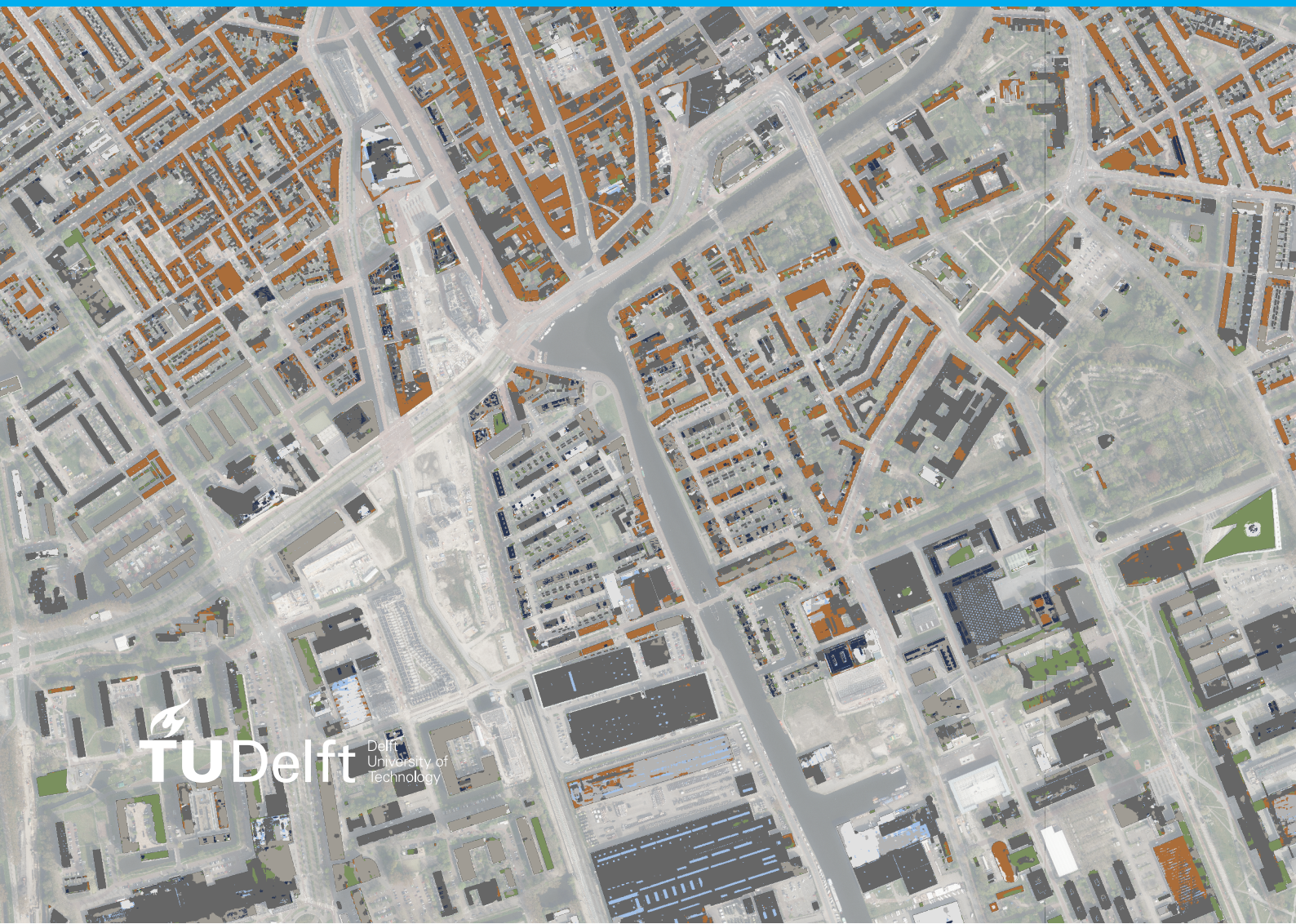
Master of Science Thesis

CNN-based Roofing Material Segmentation using Aerial Imagery and LiDAR Data Fusion

Dimitris Mantas

October 2024



Master of Science Thesis

CNN-based Roofing Material Segmentation using Aerial Imagery and LiDAR Data Fusion

Dimitris Mantas

October 2024

Submitted in partial fulfilment of the requirements for the degree of Master of Science in Geomatics at the Delft University of Technology.

Mantas, D. (2024, October). CNN-based Roofing Material Segmentation using Aerial Imagery and LiDAR Data Fusion [Master of Science thesis, Delft University of Technology]

This thesis was carried out in the:

3D Geoinformation Research Group
Department of Urbanism
Faculty of Architecture and the Built Environment
Delft University of Technology

Supervisors : **Dr. Hugo Ledoux**
Associate Professor
3D Geoinformation Research Group

Weixiao Gao
Research Fellow
3D Geoinformation Research Group

Co-reader : **Felix Dahle**
Doctoral Student
Department of Geoscience & Remote Sensing
Faculty of Civil Engineering and the Geosciences
Delft University of Technology

Front cover: Pixel-wise roofing material map of the region around Zeeheldenbuurt, Delft, overlaid onto the corresponding aerial photograph.

Abstract

Roofing material classification is becoming pivotal in urban decision-making, supporting processes like asbestos mapping, disaster preparation, and urban heat island detection. However, research has mainly focused on a restricted set of materials. Additionally, many studies rely on expensive multi- or hyper-spectral imagery and often use outdated or ineffective classification methods, overlooking the transformative capabilities of advanced deep learning and data fusion. This thesis explores using a convolutional neural network for pixel-level classification of Dutch buildings by merging standard aerial images with LiDAR data, enabling detailed mapping and addressing concerns about classification efficacy relative to image- and object-level methods.

A framework was devised to generate a new semantic segmentation dataset with over 15.5 million pixels from 200 randomly selected images nationwide, covering eight distinct materials. To facilitate material identification in unfavourable lighting conditions, true-colour aerial imagery from the BM5 dataset was combined with rasterised features extracted from the national point cloud (AHN4), specifically reflectance, slope, and planar point density. Additionally, a quasi-normalised elevation model (nDRM) was employed, based on the corresponding digital surface model and median roof elevation of buildings in each scene, as provided by the 3DBAG dataset. The research was further investigated using the DeepLabv3+ semantic segmentation architecture with a ResNet-18 backbone, and the model was trained end-to-end on the generated dataset. In this context, a novel stratified splitting algorithm and weighting scheme to combat class imbalance in the training subset were introduced.

After thorough hyperparameter tuning, we achieved a 64.68% mean intersection over union on the test subset. Membranes and gravel outperformed almost every other study. However, there were notable confusion and omission errors with light-permitting surfaces and metal. Further testing of pixel-wise material maps' generalization to different LoDs of the 3DBAG considerably decreased gross errors. However, it might overlook some minor original predictions, thus not improving overall performance notably. Generally, LoD1.1 was inadequate for modelling multi-material roofs of different heights. While LoD1.3 improved this, it still missed small roof sections, unlike LoD2.2, which also had more outliers. Additionally, an ablation study on the LiDAR-derived component of the new dataset showed that removing slope and nDRM reduced performance by 10.31% and 8.61%, respectively, while density had the least impact. All ablated features were semantically linked, suggesting they should be combined into a single dataset.

The thesis showcases the relevance of pixel-based classification with DL and data fusion, providing resources for future research and indicating areas for dataset expansion and improved annotation.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Dr. Hugo Ledoux and Weixiao Gao, for their continuous feedback, insights, support, and trust throughout the duration of my thesis. In particular, I would like to thank them for their availability and eagerness to help me through various setbacks in various stages of the project, helping me to not lose motivation and track of my end goal. In addition, I would like to thank my co-reader, Felix Dahle, for his availability and willingness to read this undoubtedly very long document in due time and provide constructive feedback for my P4 submission.

Furthermore, I would like to thank Dr. Giorgio Agugiaro of the 3D Geoinformation Group and Dr. Daniela Maiullari of the Department of Urbanism, BK, TU Delft, for each granting me a short interview in the early stages of my thesis. Although these interviews are not included in this document, they provided me with valuable insights into various aspects of roofing material classification from a practitioner's and a researcher's point of view, such as which materials are the most important in various applications, as well as how relevant intelligence is or could be used in relevant fields, guiding my early research efforts.

Moreover, I would like to thank the teaching teams of the courses "Machine Learning for the Built Environment" at BK, TU Delft, and "Data Science and Artificial Intelligence for Engineers" at the Faculty of Civil Engineering and the Geosciences, TU Delft, for providing me with the required knowledge and most importantly curiosity to successfully carry out the project. Similarly, thank you to the TorchGeo and TorchSeg maintainers and contributors for their work and prompt responses to any issues I encountered while using said software.

Finally, I would like to thank my friends and family for always encouraging and supporting me throughout my studies. Thank you for believing and having patience in me.

Dimitris Mantas
October 2024
Athens, Greece

Table of Contents

1. Introduction 1
- 1.1. Problem Statement . . . 1
- 1.2. Research Objective Overview . . . 3
- 1.3. Research Scope . . . 4
- 1.4. Research Outline . . . 4

2. Theoretical Background and Related Work 7
- 2.1. Related Work . . . 7
  - 2.1.1. Image-based Classification . . . 7
  - 2.1.2. Object-based Classification . . . 10
  - 2.1.3. Pixel-based Classification . . . 15
  - 2.1.4. Image and LiDAR Data Fusion . . . 18
  - 2.1.5. Conclusion on the State of the Art . . . 20
- 2.2. Deep Learning for Semantic Segmentation . . . 22
  - 2.2.1. Image Convolutions . . . 22
  - 2.2.2. Residual Neural Networks . . . 25
  - 2.2.3. The DeepLab Architecture . . . 28
  - 2.2.4. Splitting and Weighting . . . 29
  - 2.2.5. Data Augmentation . . . 31
  - 2.2.6. Loss Functions . . . 32

3. Methodology 35
- 3.1. Overview . . . 35
- 3.2. Overview of Source Datasets . . . 36
  - 3.2.1. The 3DBAG Dataset . . . 36
  - 3.2.2. The Dutch Aerial Imagery Programme and the BM5 Dataset . . . 38
  - 3.2.3. The Dutch Elevation Programme and the AHN4 Dataset . . . 40
- 3.3. Reference Data Generation . . . 42
  - 3.3.1. Raster Stack Generation . . . 43
    - 3.3.1.1. 3DBAG Tile Downloading and Parsing . . . 43
    - 3.3.1.2. 3DBAG Tile Asset Downloading and Parsing . . . 43
    - 3.3.1.3. Raster Concatenation . . . 49
  - 3.3.2. Raster Stack Splitting . . . 50
- 3.4. Reference Dataset Annotation . . . 52
  - 3.4.1. Material Classes . . . 52
    - 3.4.1.1. Membrane . . . 52
    - 3.4.1.2. Concrete . . . 54

3.4.1.3. Gravel . . . 54
3.4.1.4. Light-permitting Surface . . . 55
3.4.1.5. Metal . . . 55
3.4.1.6. Solar Panel . . . 56
3.4.1.7. Tile . . . 57
3.4.1.8. Vegetation . . . 58
3.4.1.9. Other . . . 59
3.4.2. Annotation Procedure . . . 59
3.4.3. Sources of Error . . . 63
3.4.3.1. Material Appearance . . . 63
3.4.3.2. Lighting Conditions & Scene Geometry . . . 64
3.4.3.3. Reflectance Noise & Relativity . . . 64
3.4.3.4. Misalignment between BM5 and AHN4 . . . 65

4. Implementation and Experimental Framework 67
4.1. Software Implementation . . . 67
4.2. Reference Dataset . . . 67
4.2.1. Modified Raster Stack Splitting Algorithm . . . 67
4.2.2. Overview . . . 68
4.3. Model Training . . . 71
4.3.1. Reference Dataset Splitting and Class Weighting . . . 71
4.3.2. Data Augmentation . . . 74
4.3.3. Loss Function . . . 75
4.3.4. Performance Metrics . . . 75
4.3.4.1. Confusion Matrix . . . 75
4.3.4.2. Accuracy . . . 78
4.3.4.3. Precision & Recall . . . 78
4.3.4.4. Jaccard Index . . . 79
4.3.5. Model Design . . . 80
4.3.6. Hyperparameter Optimisation . . . 82
4.3.6.1. Initial Configuration . . . 82
4.3.6.2. Exploration - Manual Experimentation . . . 83
4.3.6.3. Exploration - Random Search . . . 86
4.3.6.4. Exploitation . . . 88
4.4. Model Inference . . . 89
4.4.1. Chip Inference . . . 89
4.4.2. Tile Inference . . . 90
4.4.3. Map Generalisation . . . 91
4.5. Qualitative Performance Evaluation . . . 92

5. Results and Analysis 95
5.1. Hyperparameter Optimisation . . . 95
5.1.1. Initial Configuration . . . 95
5.1.2. Manual Experimentation . . . 97

5.1.3. Round 1 . . . 105
5.1.4. Round 2 . . . 107
5.1.5. Round 3 . . . 111
5.2. Pixel-based Performance Evaluation . . . 116
5.2.1. Quantitative Evaluation . . . 116
5.2.2. Qualitative Evaluation . . . 119
5.3. Generalised Performance Evaluation . . . 121
5.3.1. Quantitative Evaluation . . . 121
5.3.2. Qualitative Evaluation . . . 123
5.4. Ablation Study . . . 124
5.4.1. Reflectance . . . 125
5.4.2. Slope . . . 126
5.4.3. nDRM . . . 128
5.4.4. Density . . . 129
5.4.5. LiDAR . . . 131

6. Conclusions and Future Work **133**
6.1. Research Objective Resolution . . . 133
6.2. Discussion . . . 138
6.2.1. Contributions . . . 138
6.2.2. Limitations . . . 139
6.3. Recommendations and Future Work . . . 139

A. Hyperparameter Optimization **141**
A.1. Manual Experimentation . . . 141
A.2. Round 1 . . . 142
A.3. Round 2 . . . 146
A.4. Round 3 . . . 150

B. Failure Cases **153**

C. Qualitative Performance Evaluation **159**
C.1. Overview . . . 159
C.2. Performance Check Regions . . . 159
C.3. Pixel-level Predictions . . . 164
C.4. Generalised Performance Evaluation . . . 169

D. Reproducibility self-assessment **173**
D.1. Marks for each of the criteria . . . 173
D.2. Self-reflection . . . 173

List of Figures

1.1. True-colour aerial image of an example building and the corresponding LoD2.2 roof segment footprints, visualised as red dash-dotted polygons. Image-based classification approaches assign one or more labels to the whole scene, OBIA to each roof segment, and pixel-based methods to each pixel of the image. Hence, the solar panel array which belongs to the larger of the two roof segments may be directly delineated using only pixel-based classification since OBIA requires that appropriate subdivision of the pertinent segment is first performed. 3

2.1. Example single- and multi-label image-based roofing material classification tasks. 8
2.2. Example single- and multi-label object-based roofing material classification tasks. 11
2.3. Example single- and multi-label pixel-based roofing material classification tasks. 15
2.4. Visual representation of the convolution of a 4×4 image with a 3×3 kernel, shown in blue and dark blue, respectively. The operation is parametrised by a stride and an atrous rate of one along each dimension of the image. The corresponding output feature map is shown in green has two rows and columns. The starting position of the filter does not matter. Reproduced from Dumoulin and Visin (2016). 23
2.5. Visual representation of the convolution of a 4×4 image with a 3×3 kernel, shown in blue and dark blue, respectively. The operation is parametrised by a stride of one and an atrous rate of two along each dimension of the image. The corresponding output feature map is shown in green has two rows and columns. The starting position of the filter does not matter. Reproduced from Dumoulin and Visin (2016). 24
2.6. Example of no or valid padding (a), same padding (b), and full padding (c). Reproduced from Dumoulin and Visin (2016). 24
2.7. Example traditional (left) and residual (right) neural network blocks. The optimal training parameters of the former block are arbitrary, while those of the of the latter are clearly zero. Reproduced from https://classic.d2l.ai/chapter_convolutional-modern/resnet.html. . 26
2.8. Standard basic ResNet blocks. Identity skip connections (left) are used when x and $f(x)$ have the same dimensionality. Otherwise, projection shortcuts (right) are used. Reproduced from https://classic.d2l.ai/chapter_convolutional-modern/resnet.html. 26

2.9. Standard ResNet-18 model. The global average pooling and fully connected layers at the exit are used for image classification and are not part of the actual architecture. Reproduced from https://classic.d2l.ai/chapter_convolutional-modern/resnet.html. 27
2.10. Standard basic and bottleneck ResNet blocks. Reproduced from K. He et al. (2015). 27
2.11. ResNet-C and D modifications to the standard ResNet architecture. The former changes the stem, whereas the latter latter improves upon it by modifying the projection shortcuts. Reproduced from T. He et al. (2018). 28
2.12. Standard DeepLabv3+ architecture. Reproduced from Chen et al. (2018). 29
2.13. Binary ground truth mask and corresponding prediction loss for various objective functions and values of the conditional probability of the positive (i.e., white) class at the corresponding location. The probability of the negative (i.e., black) class everywhere else is always one. 33

3.1. Overview of the proposed methodological framework. 35
3.2. The correspondence between an arbitrary, real-life building, its BAG footprint, and its various representations in the 3DBAG. Adapted from Peters et al., 2022. 37
3.3. Tiling structure of the 3DBAG. 38
3.4. Parcel and block layout of the 2023 edition of the BM5 dataset. 39
3.5. Quick and final ortho-imagery of an example scene. Reproduced from Het Waterschapshuis, 2024. 40
3.6. Parcel layout of the AHN4 dataset. 41
3.7. Example roof surface footprints and the corresponding BM5 data. If the footprint MBR had been used to download this data, the bottom left area would have been processed although it is clearly contextually irrelevant. 44
3.8. Frequency distribution of the planar point density of the AHN4 points which are classified as building or ground and their 2D projection intersects a BAG polygon. The data is taken from the `b3_puntdichtheid_ahn4` attribute of the 3DBAG (v2024.02.28) for models which were reconstructed using said dataset, as indicated by the corresponding `b3_pw_bron` attribute. 47
3.9. Example scene and corresponding DTM and DRM. In the absence of valid data at building locations in the official product, the DTM presented here has been constructed by rasterising the relevant ground points using the procedure described in Section 3.3.1.2. 48
3.10. Example scene and corresponding nDSM and nDRM. 49
3.11. Binary building map generation process. 51
3.12. Example stack chips with varying background contents. 52
3.13. Typical examples of dark-coloured membrane roofs. 53
3.14. Typical examples of light-coloured membrane roofs. 53

3.15. Typical examples of built-up roofs with gravel ballast. The middle and right figures also show a semi-intensive green roof on the middle-left and tiles on the centre–middle-right, respectively. 55

3.16. Typical examples of light-permitting surfaces on a membrane (a), metal (b), and tile roof (c). 55

3.17. Typical examples of metal roofs. 56

3.18. Typical examples of solar panel arrays on a metal (a), tile (b), and membrane roof (c). 57

3.19. Examples of non-roof or irrelevant objects in the reference dataset. . . 60

3.20. Examples of base surfaces in chips. 61

3.21. Annotation process for an example chip. The building in the top left corner of the chip is not annotated for simplicity. 62

4.1. Pixel- and image-level class distribution across the reference dataset. . 69

4.2. Possible random and actual splits of the implementation dataset. . . . 72

4.3. Class weights of the training set using inverse frequency and the TF-IDF inspired method. 74

4.4. Jaccard index, $J(a; A, B)$ as a function of the fractional overlap, a, of two linear segments, A and B, of equal length, l, as one is "slid" across the other. Notice that $J(0.5) = \frac{1}{3}$ and that $J\left(\frac{2}{3}\right) = 0.5$. Adapted from Baharav et al., 2020. 80

5.1. Baseline performance as a function of training time. The shaded region represents the 95% confidence interval computed across three identical trials using the same random seed. 96

5.2. Training (a) and validation (b) loss as a function of training time using the baseline training protocol (Table 5.1). The shaded region represents the 95% confidence interval computed across three identical trials using the same random seed. 97

5.3. Training loss (a) and standard deviation of the validation mIoU (b) as a function of training time using the baseline training protocol (Table 5.1). The loss plateaus after the approximately 3500^{th} training step. The standard deviation was computed across three identical trials using the same random seed. The spike in validation performance is ignored as it was present in only one trial. 98

5.4. Validation loss as a function of training time for the baseline training protocol and various configurations with different amounts of stochastic depth. All regularised configurations achieve a lower loss score than the base line (sanity check) by the last training step. 101

5.5. Training (a) and validation (b) loss as a function of training time for the baseline training protocol and various configurations with different amounts of label smoothing. The baseline configuration (sanity check) fits the training set well but displays significant miscalibration. On the other hand, label smoothing results in convergence to a higher loss score, but solves the miscalibration issue. 102
5.6. Improved baseline performance as a function of training time. The baseline is marked as sanity check. 103
5.7. Training (a) and validation (b) loss as a function of training time using the improved baseline training protocol (Table 5.2). 104
5.8. Improved baseline performance as a function of training time. The baseline is marked as sanity check. 107
5.9. Training (a) and validation (b) loss as a function of training time using the best training protocol discovered in the first round of the HPO process (Table 5.3). 107
5.10. round 2 performance as a function of training time. 110
5.11. Training (a) and validation (b) loss as a function of training time using the best training protocol discovered in the second round of the HPO process (Table 5.4). 111
5.12. round 3 performance as a function of training time. The baseline is marked as sanity check 114
5.13. Training (a) and validation (b) loss as a function of training time using the best training protocol discovered in the second round of the HPO process (Table 5.6). 115
5.14. Pixel-level confusion matrix of the best model configuration discovered during the HPO process, evaluated on the test subset. Each row is normalised by the corresponding class support. The material corresponding to each class label is given in Table 4.1. 116
5.15. Generalised confusion matrices. 121
5.16. Confusion matrix before and after removing reflectance band. 125
5.17. classification report without the reflectance band. results worse than the baseline are marked with red while better with blue. 125
5.18. Confusion matrix before and after removing slope band. 126
5.19. classification report without the slope band. results worse than the baseline are marked with red while better with blue. 127
5.20. Confusion matrix before and after removing nDRM band. 128
5.21. classification report without the nDRM band. results worse than the baseline are marked with red while better with blue. 128
5.22. Confusion matrix before and after removing density band. 129
5.23. classification report without the density band. results worse than the baseline are marked with red while better with blue. 130
5.24. Confusion matrix before and after removing LiDAR-derived bands. . . 131
5.25. classification report without the LiDAR-derived bands. results worse than the baseline are marked with red while better with blue. 131

A.1. Parallel coordinate of the first automated round of the HPO process. . 142
A.2. slice plots of the continous variables of the first hpo round 143
A.3. Terminator improvement of the first automated round of the HPO process. The expected improvement potential is estimated according to Makarova et al., 2022. Each plot represents one of 10 evaluations of the underlying algorithm using different random seeds. 144
A.4. Paremeter importance of the first automated round of the HPO process. The expected improvement potential is estimated according to Hutter et al., 2014. The distribution of each parameter importance is computed across 100 evaluations of the underlying algorithm using 100 different random seeds. 145
A.5. Parallel coordinate of the second automated round of the HPO process. 146
A.6. slice plots of the continous variables of the second hpo round 147
A.7. Terminator improvement of the second automated round of the HPO process. The expected improvement potential is estimated according to Makarova et al., 2022. Each plot represents one of 10 evaluations of the underlying algorithm using different random seeds. 148
A.8. Paremeter importance of the second automated round of the HPO process. The expected improvement potential is estimated according to Hutter et al., 2014. The distribution of each parameter importance is computed across 100 evaluations of the underlying algorithm using 100 different random seeds. 149
A.9. Parallel coordinate of the third automated round of the HPO process. 150
A.10. slice plots of the continous variables of the third hpo round 151
A.11. Terminator improvement of the third automated round of the HPO process. The expected improvement potential is estimated according to Makarova et al., 2022. Each plot represents one of 10 evaluations of the underlying algorithm using different random seeds. 152

B.1. Failure case where residential solar panels installed onto a ceramic tile roof were confused for the base material (i.e., negative confidence regions in left and middle buildings). color-label associations available in Table 4.1. 153
B.2. Failure case where a dirty dark metal roof was completely confused for one with ceramic tiles (i.e., negative confidence region in the top building). The polycarbonate skylight on the roof was also not detected.color-label associations available in Table 4.1. 154
B.3. Failure case where a dirty dark metal roof was completely confused for one with ceramic tiles (i.e., negative confidence region in the top building). The polycarbonate skylight on the roof was also not detected. 155
B.4. Failure cases where gravel (a) and dark-coloured membranes/light-permitting surfaces (b) were confused with vegetation.color-label associations available in Table 4.1. 156

B.5. Failure cases where clearly visible solar panels where not fully identified.color-label associations available in Table 4.1. 157
B.6. Failure case where an atrium below a tar-and-gravel roof was mostly missed. Tile hallucinations are visible in the parts of the atrium which were actually detected.color-label associations available in Table 4.1. . 158

C.1. 3DBAG tile 9-284-556. The corresponding LoD2.2 surface footprints have been dissolved by building ID and are also shown in blue. 159
C.2. bk performance check region 160
C.3. main campus buildings performance check region 161
C.4. industrial performance check region 162
C.5. residential performance check region 163
C.6. Pixel-level predictions on the qualitative performance evaluation tile (Section 4.5). color-label associations available in Table 4.1. 164
C.7. Ground truth and corresponding predictions for the building of the Faculty of Architecture and the Built Environment.color-label associations available in Table 4.1. 165
C.8. Ground truth and corresponding predictions for the building of the main campus buildings.color-label associations available in Table 4.1. 166
C.9. Ground truth and corresponding predictions for the industrial area.color-label associations available in Table 4.1. 167
C.10. Ground truth and corresponding predictions for the residential area.color-label associations available in Table 4.1. 168
C.11. LoD1.2 predictions on the qualitative performance evaluation tile (Section 4.5). color-label associations available in Table 4.1. 169
C.12. LoD1.3 predictions on the qualitative performance evaluation tile (Section 4.5). color-label associations available in Table 4.1. 170
C.13. LoD2.2 predictions on the qualitative performance evaluation tile (Section 4.5). color-label associations available in Table 4.1. 171

D.1. Reproducibility criteria to be assessed. 173

List of Tables

3.1. Band composition of the raster stacks comprising the reference dataset. 50
3.2. Initial class name-colour-label associations. This table is used only for annotation purposes. 63

4.1. Final class name-colour-label associations. This table is used only in reference to the reference dataset. 68
4.2. Reference dataset overview. 70
4.3. Augmentations used to rescale each image band of the reference dataset and artificially increase its (i.e., the dataset's) size. 75
4.4. Typical structure of a multi-class confusion matrix. The elements of this particular matrix are arbitrary. 76
4.5. Typical structure of a binary confusion matrix. The first and second class are commonly called "negative" and "positive", respectively. . . . 77
4.6. Initial training protocol. Any parameter not mentioned assumes its default value as per PyTorch v2.2.2. 83
4.7. Search space for the first automated round of the HPO process. 87
4.8. Search space for the second automated round of the HPO process. . . 87

5.1. Baseline training protocol. Any parameter not explicitly defined assumes its default value as per PyTorch v2.2.2. 95
5.2. Improved baseline training protocol. Any parameter not explicitly defined assumes its default value as per PyTorch v2.2.2. 103
5.3. Best training protocol discovered in the first round of the HPO process. Any parameter not explicitly defined assumes its default value as per PyTorch v2.2.2. 106
5.4. Best training protocol discovered in the second round of the HPO process. Any parameter not explicitly defined assumes its default value as per PyTorch v2.2.2. 110
5.5. Search space for the third automated round of the HPO process. . . . 111
5.6. Best training protocol discovered in the third round of the HPO process. Any parameter not explicitly defined assumes its default value as per PyTorch v2.2.2. 114
5.7. Pixel-level performance scores of the best model configuration discovered during the HPO process, evaluated on the test subset. The material corresponding to each class label is given in Table 4.1. 117

5.8. Generalised classification report. The material corresponding to each class label is given in Table 4.1. results worse than the baseline are marked with red while better with blue. 122

A.1. Manual experiments results better than the baseline are marked in green. 141

List of Acronyms

2D	two-dimensional	36
3D	three-dimensional	19
ALS	airborne laser scanning	2
ASPP	à trous spatial pyramid pooling	28
BK	Faculty of Architecture and the Built Environment	92
BN	batch normalisation	75
CE	cross entropy	32
CIR	colour infrared	1
CNN	convolutional neural network	1
DL	deep learning	3
DT	decision tree	13
ECA	efficient channel attention	99
EU	European Union	1
EWI	Faculty of Electrical Engineering, Mathematics and Computer Science	93
DRM	digital roof model	35
DSM	digital surface model	12
DTM	digital terrain model	16
FN	false negative	76
FOSS	free and open-source software	31
FP	false positive	76
GMM	Gaussian mixture model	89
GSD	ground sampling distance	1
HPO	hyperparameter optimisation	5
HSI	hyperspectral imagery	1
HVAC	heating, ventilation, and air conditioning	61
ID	identification	37
IDE	Faculty of Industrial Design Engineering	93
IDW	inverse distance weighting	46

IoU intersection over union . . . 79
LiDAR light detection and ranging . . . 3
LoD level of detail . . . 2
MBR minimum bounding rectangle . . . 43
mIoU mean intersection over union . . . 77
ML machine learning . . . 1
MNF minimum noise fraction . . . 16
MSI multispectral imagery . . . 1
NaN not-a-number . . . 47
nDRM normalised digital roof model . . . 35
nDSM normalised digital surface model . . . 16
NDVI normalised difference vegetation index . . . 12
NIR near-infrared . . . 7
OA overall accuracy . . . 10
OBIA object-based image analysis . . . 2
OFA operating flying altitude . . . 19
PCA principal component analysis . . . 12
PV photovoltaic . . . 56
RC reinforced concrete . . . 54
RF random forest . . . 13
RGB true-colour . . . 1
RS remote sensing . . . 10
RQ research question . . . 3
SAM spectral angle mapper . . . 16
SE squeeze and excitation . . . 99
SOTA state of the art . . . 2
SVM support vector machine . . . 13
TGI triangular greenness index . . . 98
TN true negative . . . 76
TP true positive . . . 76
TPE Tree-structured Parzen estimator . . . 89
UHI urban heat island . . . 1
ViT vision transformer . . . 1
VLR variable length record . . . 41
WV WorldView . . . 10

1. Introduction

1.1. Problem Statement

The identification of roofing materials is fast becoming an increasingly important consideration in the context of urban planning and decision making. In particular, the long-established asbestos ban in the European Union (EU) (European Commission, 1999) as well as various other countries around the world (Gibril et al., 2017) due to its detrimental health (Burdett, 2006) and environmental impact has motivated policymakers to facilitate the mapping of pertinent materials in preparation for their eventual replacement, as evidenced by the sheer number of publications on the subject (Abbasi et al., 2022). In addition, certain materials may be of particular interest due to their thermal properties (Hamedianfar, Shafri et al., 2014). For instance, dark-coloured surfaces inherently absorb significantly more solar radiation than lighter-hued alternatives, aggravating the urban heat island (UHI) effect[1]. Similarly, cool and green roofs have been shown to be an effective means of ameliorating this phenomenon (Ilehag et al., 2018). Therefore, it is potentially in the best interest of officials to map these materials in order to improve the identification of UHI hotspots in their cities. In addition, some key roof superstructures, such as solar panels, are intertemporally highly relevant, in both solar potential estimation and large-scale urban retrofitting studies. Finally, several works have pointed out the general lack of comprehensive material inventories, which are naturally highly resource consuming to obtain at a building, block, or even neighbourhood level of detail via in situ audits (Abriha et al., 2018; Braun et al., 2019; Chisense, 2012).

Although all of these points have been individually recognised in pertinent literature, only few works have addressed the topic of general-purpose material classification, with most limiting themselves to a single use case or very particular study area. Furthermore, various authors rely on either multispectral imagery (MSI) or hyperspectral imagery (HSI) (Abbasi et al., 2022). However, these products, while offering superior spectral resolution compared to more traditional imagery (e.g., true-colour (RGB), colour infrared (CIR) etc.), are generally not as widely available. In addition, they are particularly constrained by acquisition and processing costs Krówczyńska et al. (2020) and Raczko et al. (2022), as well as sufficiently low ground sampling distance (GSD), which is clearly critical for accurate data labelling through optical photointerpretation. Moreover, modern machine learning (ML) approaches, such as convolutional neural networks (CNNs) and vision transformers (ViTs) (Dosovitskiy et al., 2020), are able to infer existing relationships amongst neighbouring pixels. As such, they have

[1] For a definition of the UHI effect, see Oke, 1982.

hence largely eliminated several classic issues with pixel-level methods, particularly mixed pixels (Feng & Fan, 2021), of course given enough training samples. However, the state of the art (SOTA) still stands largely in favour of object[2]- or image-based techniques. This often supports the nowadays simply outdated notion that pixel-level approaches cannot handle the increased intra-class variability commonly associated with so-called very high resolution (VHR) products in terms of GSD.

However, both (i.e., image- and object-based) methods may be problematic in use cases where fine material delineation is required, especially over large or diverse regions. On the one hand, image-based classification entails very small inputs for effective label localisation. This can potentially result in insufficiently utilised computational resources and slow inference speeds. Additionally, the underlying scenes are limited to only a single corresponding material. Otherwise, automated output interpretation becomes difficult without further processing. Similarly, the latter issue is also present in OBIA techniques, which additionally suffer from accuracy and automation issues due to under- or over-segmentation in the absence of building or roof boundaries to concretely define the notion of "objects'' in a given scene. In fact, various authors spend most of their time discussing their pre-processing and segmentation pipelines, because the optimality and generalisability of a given set of segmentation parameters is still debated in pertinent works. What is more, only pixel-based approaches natively support the identification of multiple materials per roof segment since the output "resolution'' is spatially bound only by the GSD of the input imagery. This observation is important because several materials of interest, such as solar panels, are commonly only supported in standardised building models of the highest level of detail (LoD), which are still extremely rare in real-world applications (Biljecki et al., 2016). Therefore, the only alternative opportunity to capture these materials is to perform OBIA, albeit with significantly increased difficulty (Figure 1.1).

Finally, although several authors adopt rudimentary data fusion methods (e.g., pan-sharpening, geometric correction, radiometric correction, etc.) as a pre-processing step, they rarely consider the potential integration of heterogeneous data sources, in particular aerial imagery and airborne laser scanning (ALS) products (Abbasi et al., 2022), which has almost always proven to be beneficial (Hamedianfar, Shafri et al., 2014; Norman et al., 2020). In fact, it is oftentimes the case that ancillary data is actually in a pre-processing step, but then apparently ignored in later ones.

[2]Object-based image classification is commonly referred to as object-based image analysis (OBIA) in pertinent literature.

Figure 1.1.: True-colour aerial image of an example building and the corresponding LoD2.2 roof segment footprints, visualised as red dash-dotted polygons. Image-based classification approaches assign one or more labels to the whole scene, OBIA to each roof segment, and pixel-based methods to each pixel of the image. Hence, the solar panel array which belongs to the larger of the two roof segments may be directly delineated using only pixel-based classification since OBIA requires that appropriate subdivision of the pertinent segment is first performed.

1.2. Research Objective Overview

In this context, this thesis aims to explore the applicability of contemporary deep learning (DL) and data fusion techniques for the general-purpose roofing material identification of the Dutch building stock. In particular, the main research question (RQ) is:

> To what extent is DL-based roofing material segmentation possible using aerial imagery and ALS data fusion?

The answer to the research question depends on the following sub-questions:

RQ1. Which imagery- and light detection and ranging (LiDAR)-derived features are the most effective considering the task at hand?

RQ2. Which classification and data fusion techniques are the most contextually relevant?

RQ3. How does the availability of LiDAR-derived products affect predictive performance?

RQ4. How does the generalization of pixel-wise roofing material maps to each of the building LoDs offered by the 3DBAG influence results?

Because they are explicitly associated with later design choices concerning the proposed methodological framework, the first two RQs will be answered through the relevant literature review which naturally precedes its derivation. In contract, the last two RQs will be resolved using the results of original research and experiments conducted in this work.

1.3. Research Scope

The scope of this thesis is the Dutch building stock and relevant datasets, in particular the 3DBAG (Section 3.2.1), AHN4 (Section 3.2.3), and BM5 (Section 3.2.2). This is due to the wide variety of openly available datasets, as the point of thesis thesis is to research pixel-based classification rather than blindly apply it. Therefore, any special characteristics of these datasets will be exploited in order to obtain better results. The aim is for the proposed methodology to apply to any part of the country, considering 3DBAG tiles as the unit of analysis. Therefore, this is an integral part of the methodology; the study area is the whole country.

In addition, due to time constraints, this project does not take into account any pertinent data quality issues unless they can be easily fixed. For instance, the spatiotemporal misalignment between BM5 and AHN4 (Section 3.4.3.4) is taken as granted.

Furthermore, a special focus of the methodology is the investigation of the effects of data fusion, as almost all pre-processing is clearly going to concern the AHN4 point cloud. Therefore, it is important to determine whether LiDAR actually helps as incorporating it into the methodology is not straightforward. In addition, it will be later discussed that a relevant ablation study, in which each LiDAR-derived feature used is incrementally removed from the pipeline to gauge its individual effect, has not yet been conducted.

Finally, the thesis will consider the effects and potential use cases of cartographic generalisation applied to pixel-wise material predictions. In particular, generalisation will be performed at the roof segment level using the building LoDs available in the 3DBAG.

1.4. Research Outline

Chapter 2 conducts a critical literature review of the field by organising relevant publications per classification method; pixel-, superpixel- or object-, and image-based. In addition, relevant works employing data image-lidar data fusion are discussed separately before a general conclusion on the SOTA is given. Finally, the basic theory regarding semantic segmentation, that is pixel-wise classification, in the context of deep learning is presented.

Chapter 3 initially presents the datasets used in this thesis and then conducts and then describes the methodological framework for the generation and annotation of the reference dataset, later divided into a training, validation, and test subset and used to train an appropriately designed model.

Chapter 4 begins with a modification to the pipeline presented in the previous chapter to facilitate small teams and then presents the reference dataset. After the dataset has been split, a loss function has been adopted and relevant performance metrics have been selected. The model used for the task at hand is designed and a methodological framework is presented for hyperparameter optimisation (HPO). Finally, large-scale inference strategies are discussed and a separate 3DBAG tile used for qualitative performance evaluation is presented.

Chapter 5 discusses the results of all experiments, in particular HPO, quantitative and qualitative performance evaluation of pixel-wise and generalised predictions, and the aforementioned ablation study.

Chapter 6 presents the answers to the above-introduces research questions, explains the contributions and limitations of this work, and finally provides recommendations for future work and research.

2. Theoretical Background and Related Work

2.1. Related Work

2.1.1. Image-based Classification

In the context of ML, image-based roofing material classification refers to the task of supervised image classification, that is the assignment of a single material class to a given image of a roof or roof segment, depending on the particular application requirements (Figure 2.1). In particular, image classification aims to answer the question of which material is the most prominent or important in the scene. At this point it should be noted that although it is technically possible, multi-label material classification, where more than one labels may correspond to the same scene, has not been explored yet in the context of roofing material classification, at least according to the best of the author's knowledge at the time of writing. In contrast to its single-label counterpart, this task aims to resolve the question of which materials are generally present in the image. However, it would not be able to delineate them, in turn limiting its added value and even potentially introducing confusion concerning the interpretation of relevant model predictions. This is perhaps the reason why it is not preferred in relevant literature. In any case, image-based classification has been shown to be an effective approach in cases where the input images are particularly small in size , so as to achieve better prediction localisation, or when this type of accuracy is not of particular importance, such as when material classification is an auxiliary task of a larger methodology (e.g., building extraction).

In terms of input data, most authors prefer aerial RGB imagery (J. Kim et al., 2021; Santos et al., 2023; Wyard et al., 2023). Similarly, Wyard et al. (2023) supposed that near-infrared (NIR) data could potentially mitigate reported misclassification issues in vegetated and shadowy scenes, but did not investigate any further. Also of note is the overwhelming preference for "general-purpose" imagery in contrast to MSI or HSI, which is very popular in object- and pixel-based classification (Sections 2.1.2 and 2.1.3) . Concerning the actual data source, even though all other authors used aerial surveying products, Santos et al. (2023) used satellite imagery from Google Earth. Still, the nature of such products did not appear to negatively affect their results. Comparably, the GSD of the input data ranged from 5 cm (Wyard et al., 2023) to 25 cm (Krówczyńska et al., 2020; Raczko et al., 2022). Again, no author reported any performance issues due to excessively low or high GSD. Finally, image dimensions varied from 27×27 pixels (Krówczyńska et al., 2020) to 224×224 pixels (J.

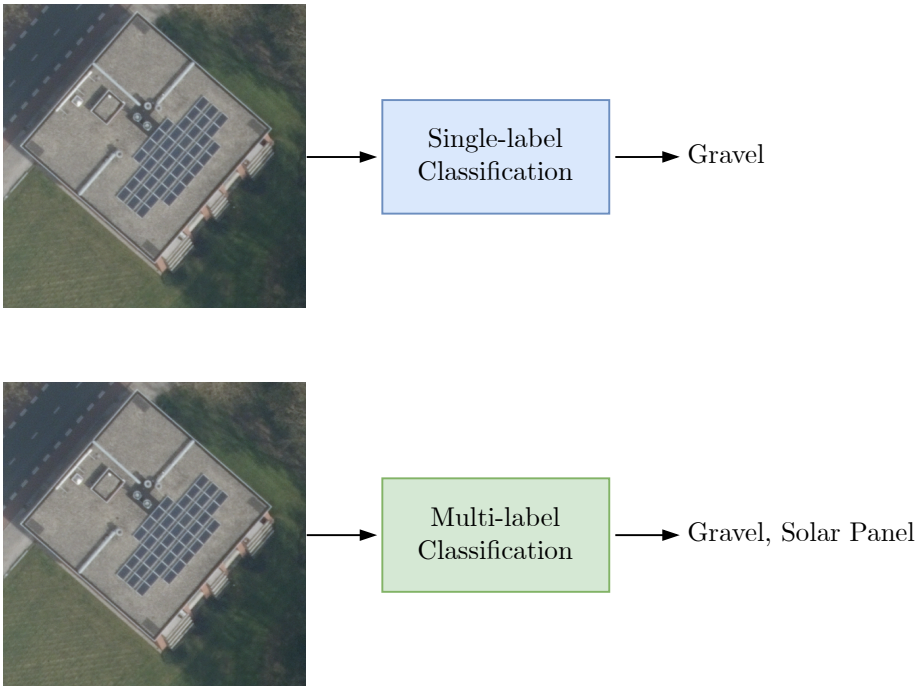


Figure 2.1.: Example single- and multi-label image-based roofing material classification tasks.

Kim et al., 2021). Although Krówczyńska et al. (2020) mention relevant computational constraints for their particular choice of input size, other authors do not touch upon the topic. However, it is hypothesised that it was at least in part selected so as to avoid scenes containing multiple buildings in order to facilitate label localisation. This assumption may be corroborated by several attempts to centre each image around a single building (Krówczyńska et al., 2020; Raczko et al., 2022; Wyard et al., 2023).

In general, most authors limit themselves to either a particular application or study area and hence have either very broad or specific classes. For instance, J. Kim et al. (2021) attempted to conduct a more general study, but two out of their four classes concerned metal roofs, with the third being "incomplete". It is not clear what this class contained, but it was assumed to mean buildings under construction. The work of Santos et al. (2023) is more complete in this regard, as it covers various types of metal, concrete, tiles, and organic materials. However, they only used three classes, with one containing both concrete and tiles, and thus potentially having a relatively lower contextual added value in terms of semantic information. This considered, the most complete study, which is also the closest to this one in terms of material classes is that of Wyard et al. (2023), which covers twelve different materials, ranging from tiles, to membranes, metal, vegetation, gravel, slates, and solar panels.

The data processing pipelines proposed by most authors vary but are generally relatively simplistic, yet arguably effective. For instance, because their input data was in

the form of chips extracted from larger mosaics, no downstream image size control was required. The only exception to this was Santos et al. (2023), who chose to resize their patches to 64×64 pixels instead of generating them at the required size in the first place. The motivation behind this decision is not clear, nor is its effect on predictive performance, although it was likely insignificant. However, it should be noted that this approach is generally unsafe because it may distort shape features in certain cases. In addition, author disagreement was also observed in terms of feature normalisation, which is a standard regularisation technique to improve convergence. In particular, only Wyard et al. (2023) scaled their data band-wise to a standard Gaussian distribution, whereas all others did not perform any sort of rescaling. Because all other authors used images of a single area, with only three, relatively equivalent bands , it is not clear whether normalisation could have improved their results. Moreover, Wyard et al. (2023) artificially increased the size of their datasets by employing slight geometric augmentations, namely reflections and rotations. In general, the most comprehensive pipeline was designed by Wyard et al. (2023) and featured the removal of under- and over-exposed images in the HSV colour space, contrast limited adaptive histogram equalisation, and rotation to a common orientation using the histogram of oriented gradients to supposedly achieve rotation invariance., with feature normalisation and data augmentation following suit. Still, the contribution of each of these steps in the final result is again not clear. In any case, the first step is particularly interesting as it was meant to mitigate potential confusion issues due to differences in scene lighting Wyard et al. (2023). Unfortunately, Wyard et al. (2023) still reported relevant problems indicating that their approach in this regard was likely insufficient. Finally, Santos et al. (2023), used synthetic minority over-sampling technique (SMOTE) (Chawla et al., 2011) to mitigate label imbalance in their dataset.

Although all authors employed CNN-based architectures, they chose to not use existing models, but instead design their own. In particular, Krówczyńska et al. (2020) used a two-block model featuring spatial dropout at the exit of the second block as well as traditional dropout after each of two densely connected layers. Although the simplicity of this model in the context of modern DL is apparent to the informed reader, the Krówczyńska et al. (2020) actually reported that a larger or less regularised model would lead to overfitting. However, it is not clear as this is something they actually experienced because their later work used a significantly more complex Inception-based model (Szegedy et al., 2014), which improved their original results on the same dataset (Raczko et al., 2022). On a related note, the latter design was very similar to that of J. Kim et al. (2021), even though Raczko et al. (2022) arrived at it independently. Finally, both Santos et al. (2023) and Wyard et al. (2023) used similar models to Krówczyńska et al. (2020), with no model performing significantly better or worse than the other. In particular, Wyard et al. (2023) employed four convolutional layers, following by max pooling, dropout and a single fully connected layers. On the other hand, Santos et al. (2023) used four convolutional blocks and a single dense layer. Nevertheless, their blocks did not follow the standard convolution-pooling-batch normalisation structure, but where instead composed of two convolutional layers.

In general, results showed significant promise regardless of the aforementioned design

differences. On the other hand, J. Kim et al. (2021) managed to obtain an average accuracy of 72.45% on the metal, 77.8% in concrete, and 72.1% on the incomplete class for an overall accuracy (OA) of 73.3%. Similarly, Wyard et al. (2023) achieved an OA of 81%, with solar panels performing the best at 96.4% and dark-coloured ceramic tiles the worst at 67.9%. Their performance on the metal class was 74.1%. Finally, Santos et al. (2023) achieved an impressive OA of 96.7%, with an F_1 score of 96% on both their metal and tile classes.

In conclusion, Krówczyńska et al. (2020) admit that while their approach was effective in their particular study area, the same cannot be automatically assumed for more densely built regions, and that further evaluation may be required. Similarly, Santos et al. (2023) reached the the same conclusion regarding generalisability. In addition, Wyard et al. (2023) noticed confusion between materials of similar colour and texture, and recommended label merging where appropriate. Finally, they noted the need for a better technique to handle abnormal scene lightning, and that auxiliary data, in particular slope information, could potentially improve predictive performance.

2.1.2. Object-based Classification

In the context of ML, object-based roofing material classification refers to the compound task of image segmentation[1] and classification (Figure 2.2). In remote sensing (RS), this task is equivalently known as OBIA. In particular, images of roofs or roof segments are first segmented using relevant clustering or segmentation algorithms. This results in the pixels of these images being grouped into contiguous clusters according to their value. These groups are called superpixels or objects, hence the name of this approach. However, it should be noted that these groups do not necessarily need to represent real-world objects or have any semantic meaning beyond the fact that they are similar to their neighbours. Finally, the pixels corresponding to each group are classified as a single entity using aggregated statistics which may be either pre-computed or learned, depending on the particular classifier.

In contrast to image-based classification (Section 2.1.1), most authors prefer aerial or satellite MSI and HSI. In particular, Gibril et al. (2017), Hamedianfar and Shafri (2015) and Shafri (2013) used WorldView (WV)-2[2] imagery, with the latter authors citing the high acquisition and processing cost, as well as potentially limited coverage of airborne HSI. Similarly, Trevisiol et al. (2022) used WV-3 products, which are similar to those from WV-2, albeit at a significantly higher GSD of 1.24 m for the MSI and 0.31 m for the panchromatic band, respectively (Maxar Technologies, 2024b). On the other hand, de Pinho et al. (2012) used MSI from IKONOS II (1 m panchromatic and 4 m MSI; European Space Agency, 2024), whereas Hamedianfar, Shafri et al. (2014) preferred airborne HSI. Finally, Wyard et al. (2021) used conventional RGB and NIR imagery with a GSD of 25 cm.

[1] Image segmentation is an image processing technique and is not to be confused with the ML task of semantic segmentation.

[2] The WV-2 sensor provides 8-band MSI and panchromatic imagery with a GSD of 1.84 m and 0.46 m, respectively (Maxar Technologies, 2024a).

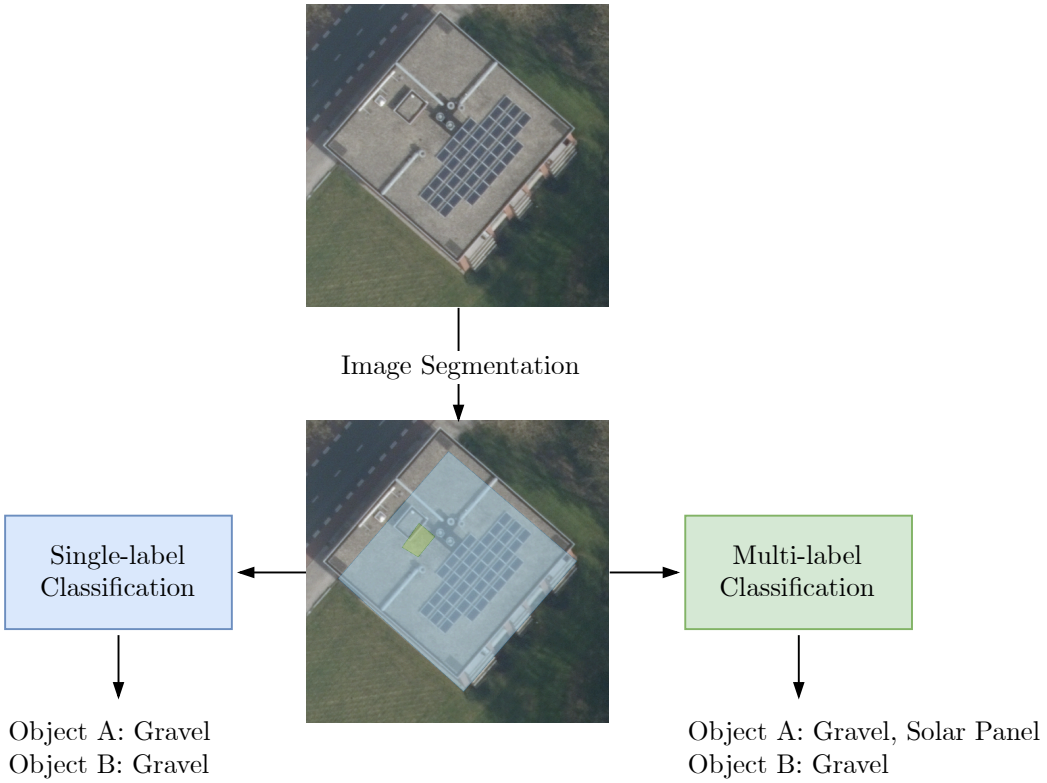


Figure 2.2.: Example single- and multi-label object-based roofing material classification tasks.

Because most works in the field actually concern themselves with urban scene classification, they generally include a large variety of classes only some of which are relevant to the context of this thesis. However, asbestos remains the most popular class (de Pinho et al., 2012; Gibril et al., 2017; Hamedianfar, Shafri et al., 2014; Hamedianfar & Shafri, 2015; Shafri, 2013), with tiles (de Pinho et al., 2012; Shafri, 2013; Trevisiol et al., 2022; Wyard et al., 2021, 2022) and metal (de Pinho et al., 2012; Hamedianfar, Shafri et al., 2014; Hamedianfar & Shafri, 2015; Shafri, 2013; Trevisiol et al., 2022; Wyard et al., 2021, 2022) following suit. Interestingly, de Pinho et al. (2012), Gibril et al. (2017), Hamedianfar, Shafri et al. (2014) and Hamedianfar and Shafri (2015) all include a "shadow" class to handle under-exposed scenes, albeit at a limited added value, since this class clearly does not correspond to a palpable object. In addition, de Pinho et al. (2012) and Wyard et al. (2021) use separate classes for the different hues of the same material. On the one hand, de Pinho et al. (2012) differentiates concrete and tiles by lightness, whereas Wyard et al. (2021) by hue. Finally, Trevisiol et al. (2022) explicitly grouped all tiles into under a single label.

Due to the nature of the corresponding input data, the processing pipelines proposed by most authors are largely similar. For example, all authors who used MSI or HSI except Hamedianfar, Shafri et al. (2014) and Trevisiol et al. (2022) applied

pan-sharpening. However, Hamedianfar, Shafri et al. (2014) did not report having access to a panchromatic band, whereas Trevisiol et al. (2022) apparently resampled all bands to a GSD of 1.2 m in order to better accommodate two short-wave infrared channels with a resolution of 3.7 m. In addition, geometric and radiometric calibration was equally popular. In particular, de Pinho et al. (2012), Hamedianfar, Shafri et al. (2014), Hamedianfar and Shafri (2015) and Trevisiol et al. (2022) all used specialised software to perform relevant corrections. Similarly, Shafri (2013) reported that their imagery had already been appropriately pre-processed by the supplier. Furthermore, Trevisiol et al. (2022) applied orthorectification using rational polynomial coefficients. Moreover, several authors integrated derivatives of their primary input data into their workflow. The most common such product are various spectral indices. For instance, Gibril et al. (2017) used normalised difference vegetation index (NDVI) and Ratio G. The former index was also used by de Pinho et al. (2012) and Hamedianfar, Shafri et al. (2014), who also used the 3:1 index and fused their panchromatic and MSI bands using principal component analysis (PCA) and converted the RGB component to the HIS colour space. What is more, Hamedianfar, Shafri et al. (2014) and Hamedianfar and Shafri (2015) experimented with various indices while the latter author introduced one for and.

As mentioned in the main issue with the segmentation step in OBIA stems from the fact that there is no general consensus at to what constitutes an semantically valid "object". Clearly, at least in the context of this thesis, valid objects would be roofs and roof segments. In other words, although the informed reader is most likely aware of various image segmentation methods, it is obvious that not all possible segmentations of a given image are equally informative, and it is hence in the best interest of the researcher to produce as few and as "clean" pixel clusters as possible. However, as raster data does not inherently include any topological information, three potential approaches arise. The first is to obtain the building or roof footprints in either raster or vector form and use these as object boundaries (Wyard et al., 2021), try to infer them directly from the image (Trevisiol et al., 2022), or simply segment the image using conventional methods (de Pinho et al., 2012; Gibril et al., 2017; Hamedianfar, Shafri et al., 2014; Hamedianfar & Shafri, 2015; Shafri, 2013).

Naturally, the second and third techniques are the most interesting as they offer the most opportunity for experimentation. For instance, Trevisiol et al. (2022) extracted the digital surface model (DSM) of their study area from the input imagery, and used as well as slope to detect building boundaries using a contrast split filter. Subsequently, they further split the resulting objects according to their average slope, and used thresholding based on NDVI, minimum height, and maximum elevation as well as area roofs to discard false positives. Similarly, de Pinho et al. (2012) also adopted a multi-scale segmentation approach. In particular, they divided their input data into five-level clusters, with each segment encompassing all previous ones. Therefore, the fifth level contained neighbourhood-sized pixel groups, whereas the first vegetation and non-vegetation elements. Each level was produced using a different approach, depending on which the authors considered to be the most contextually appropriate, from fuzzy logic, to hierarchical classification and chessboard segmentation (de Pinho

et al., 2012). On the other hand, Hamedianfar, Shafri et al. (2014), Hamedianfar and Shafri (2015) and Shafri (2013) used conventional edge-based segmentation techniques, but post-processed the results by merging similar clusters using specialised software. Finally, Gibril et al. (2017) used a region-growing algorithm and was the only author who attempted to optimise the hyperparameters of the segmentation algorithm, in particular using Taguchi methods.

In terms of classification methods, most authors prefer classic ML and statistical approaches. In particular, support vector machines (SVMs) (Gibril et al., 2017; Trevisiol et al., 2022), decision trees (DTs) (de Pinho et al., 2012; Hamedianfar, Shafri et al., 2014), and random forests (RFs) (Gibril et al., 2017) were the most popular methods. Similarly, Shafri, 2013 employed linear discriminant analysis (LDA), which is similar in concept to logistic regression, and classified image objects according to their Euclidean distance from the resulting decision boundary. In addition, Gibril et al., 2017 compared the performance of various classifiers in addition to SVM and RF, namely Bayes and nearest-neighbour classifiers, as well as a manually constructed rule-based system. Interestingly, Hamedianfar and Shafri, 2015 also used a rule-based classifier based on expert opinion rather than a statistical method, and even supported its supposed superiority in comparison.

Since each object is generally composed of multiple pixels, the input to all these algorithms are aggregate statistics or so-called features. Based on a collective conclusion drawn from relevant literature, these features may be spectral, textural, or spatial. The first type typically refers to ranges or specific values of either raw band values or various descriptive statistics (e.g., the mean Gibril et al., 2017; Hamedianfar, Shafri et al., 2014; Hamedianfar & Shafri, 2015; Shafri, 2013; Trevisiol et al., 2022, standard deviation Gibril et al., 2017; Shafri, 2013; Trevisiol et al., 2022, minimum Shafri, 2013, or maximum value Gibril et al., 2017; Shafri, 2013, etc.), which are computed directly on the input imagery or spectral indices based on the input imagery. For instance, the sign of NDVI is commonly used to identify vegetation (Gibril et al., 2017; Shafri, 2013). However,compound colour-related quantities such as brightness (Gibril et al., 2017) and hue (de Pinho et al., 2012) have also been used. Spectral features appear to be the most important feature category, as they are used by all authors and most frequently almost exclusively preferred by data mining algorithms (de Pinho et al., 2012). The second most important feature type are textural features. These features generally refer to ranges or specific values aforementioned descriptive statistics or metrics such as entropy Shafri, 2013, or homogeneity and correlation (Trevisiol et al., 2022). What clearly separates textural from spectral features is that the former are commonly computed with respect to a small (e.g., 3×3 pixel) sliding window (Hamedianfar & Shafri, 2015; Shafri, 2013; Trevisiol et al., 2022). Finally, spatial features refer to the area , roundness, compactness or convexity of each object (Hamedianfar & Shafri, 2015; Shafri, 2013).

In general, results were similar to those of image-based classification. However, there was increased interest in this case because not only did each image contain several object and in turn evaluation regions, making for a significantly more difficult task, but also since some authors chose to compare the performance of various methods.

For instance, Gibril et al., 2017 found that their relatively simple rule-based system outperformed RF in terms of OA by almost 10% (93.1% v. 82.02%). However, it should be noted that although the authors did tune the hyperparameters of SVM, which they also experimented with but found to perform even worse (81.75%), they did not appear to have done the same for RF, clearly making for an unfair comparison.. In any case, they managed to achieve an outstanding performance of 97.78% on the clay class. Similarly, Hamedianfar, Shafri et al., 2014; Hamedianfar and Shafri, 2015 achieved comparable results with an OA of 90.89% and 98.33% over two and three separate test areas, respectively. The contextually relevant class in their case was metal, which scored 92.13% and 79.83%, correspondingly. In addition, Trevisiol et al., 2022 achieved similar results both with all 16 (91.3%) and 8 WV-3 bands (89.5%). In particular, they reported 89.4/80.3% in clay, 90.5/92.2% in metal and 100/99.5% in gravel for each band combination (8/16 bands). Furthermore, Wyard et al., 2021 reported more than 80% in green roofs and orange tiles, and 70% in solar panels and white membranes, respectively.

What is more, de Pinho et al., 2012; Shafri, 2013 performed quite similarly overall with an OA of 71.91% and 84.55%, respectively. Their individual performance on ceramic was 63.21% and 67.73%, and 74.07% and 84% on metal. Interestingly, Shafri, 2013 also tested a Bayes classifier, which, despite achieving 96.93% on metal, only managed an OA of 45.64%. Finally, it is worth pointing out that all authors who used data mining techniques to build their rule-based systems found that only 13–15 features were required to achieve the aforementioned results (de Pinho et al., 2012; Hamedianfar, Shafri et al., 2014; Shafri, 2013).

In conclusion, numerous authors have documented a notable source of confusion wherein certain background classes, particularly bare soil and asphalt, exhibit similar spectral responses to corresponding roof materials, specifically clay and tar, respectively de Pinho et al., 2012; Gibril et al., 2017; Hamedianfar and Shafri, 2015. Additionally, another prevalent source of confusion was observed near or along building edges (de Pinho et al., 2012; Hamedianfar, Shafri et al., 2014), seemingly due to the mixed spectral responses from the surrounding materials. Moreover, several specific problematic instances were identified, such as small objects situated near roofs Gibril et al., 2017 and regions that are either significantly under-exposed Gibril et al., 2017 or over-exposed (de Pinho et al., 2012). It is also pertinent to note that this issue was previously identified by Wyard et al., 2023. Interestingly, de Pinho et al., 2012; Trevisiol et al., 2022 independently identified metal as one of the most frequently misclassified materials in their studies; the former author attributed this to the non-Lambertian nature of such surfaces, and the latter due to its resemblance to white membranes, which were notably not included as a class, thereby causing significant interpretability challenges. Furthermore, de Pinho et al., 2012 reported confusion stemming from both substantial intra-class variability (i.e., varying appearance of the same material) and inter-class similarity (e.g., dark tile and dark concrete). According to Trevisiol et al., 2022 the aforementioned edge problem can be mitigated by generalisation to the building-level by taking the majority label of all objects belonging to the same building, assuming that it can be identified. In fact, they mentioned that this technique can

also reduce noise due to potential clutter or foreign objects on the roof. However, the authors warn against using this approach in the case of large roofs featuring surfaces with different materials.

2.1.3. Pixel-based Classification

In the context of ML, pixel-based roofing material classification refers to the task of supervised semantic segmentation, that is the assignment of a single material class to each pixel a given image of a roof or roof segment, depending on the particular application requirements (Figure 2.3).

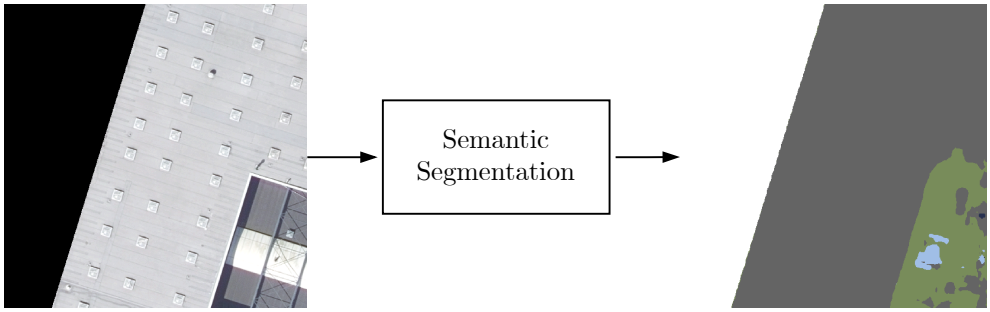


Figure 2.3.: Example single- and multi-label pixel-based roofing material classification tasks.

In terms of input data, most authors rely on airborne MSI from the MIVIS sensor (Cilia et al., 2015; Fiumi et al., 2012). The next most popular product is WV-2 (Abriha et al., 2018; Hamedianfar & Shafri, 2014; Osińska-Skotak & Ostrowski, 2015) and WV-3 imagery (Tommasini et al., 2019). On the other hand, Szabo et al., 2014 used Sentinel II and EAGLE II images, respectively..

Similarly to all other works presented so far, most authors concern themselves with asbestos detection and urban scene classification, and hence only certain classes are of interest in the context of this thesis. Tiles remain one of the most popular classes (Cilia et al., 2015; Ilehag et al., 2018; Osińska-Skotak & Ostrowski, 2015; Szabo et al., 2014). However, there still appears to be some disagreement regarding the specific terminology, and most importantly, exactly which materials are labelled as such. For instance, Hamedianfar and Shafri, 2014 modelled clay tiles regardless of appearance. Similarly, Ilehag et al., 2018 concerned themselves exclusively with cement tiles. On the other hand, Fiumi et al., 2012 explicitly modelled ceramic tiles, but under the label of "bricks". In addition, Osińska-Skotak and Ostrowski, 2015 used different labels depending on the material. Finally, Abriha et al., 2018; Szabo et al., 2014 both examined tiles of different colours and brightness due to potential shadows, but did not specify their material. In any case, metal was also a popular class Fiumi et al., 2012; Hamedianfar and Shafri, 2014; Osińska-Skotak and Ostrowski, 2015; Szabo et al., 2014. Interestingly, Ilehag et al., 2018 initially included two metal classes, "Colorbond" and "Zincalume", both of which are actually trade names, but later merged them due to

lack of reference data. Moreover, bitumen was also included in various studies, but also under several labels (Fiumi et al., 2012; Osińska-Skotak & Ostrowski, 2015; Szabo et al., 2014). Finally, Szabo et al., 2014 included two classes representing blocks of flats with and without insulation, but did not provide any reference as to what this meant. Similarly, four out of the six classes in the study conducted by Tommasini et al., 2019, are simply labelled sequentially from 2 to 5. These observations indicate both a lack of general terminology to facilitate understanding, but also the difficulty of material identification, even after relevant in situ audits.

As was the case with, most authors followed relatively standard processing techniques for airborne and satellite imagery, such as pan-sharpening (Abriha et al., 2018; Hamedianfar & Shafri, 2014; Tommasini et al., 2019), geometric correction (Cilia et al., 2015), atmospheric (Cilia et al., 2015) and general radiometric correction using either a standardised method (Fiumi et al., 2012; Tommasini et al., 2019) or by simply rescaling the input bands to a predetermined lower and upper quantile (Ilehag et al., 2018). In addition, Frassy et al., 2014 performed digital terrain model (DTM) orthorectification, whereas Osińska-Skotak and Ostrowski, 2015 also experimented with true-orthorectification, that is orthorectification using the DSM to account for the height of the various object in the given scene. Furthermore, some authors explicitly mentioned steps to discard contextually irrelevant regions from their input data. For instance Braun et al., 2019 ignored or masked out cloudy areas. Interestingly, they also discarded all thermal bands, whereas Ilehag et al., 2018 included them. On the topic of data cleaning, Cilia et al., 2015 discarded any bands with a signal-to-noise threshold beyond a predefined threshold. Moreover, they performed dimensionality reduction on their input data using the minimum noise fraction (MNF) transform, which was also used by Szabo et al., 2014 for the same purpose. On the other hand, Hamedianfar and Shafri, 2014 investigated whether the relatively high dimensionality of their dataset would affect performance by also training a pixel-wise model using three-band combination from their dataset with the highest optimum index factor (OIF), which turned out to be CIR. It is not clear why they decided to settle on only three bands. Apart from that, Frassy et al., 2014 ignored no-data pixels. Similarly, Tommasini et al., 2019 did the same for all background pixels, having had access to building footprint data. A related idea was proposed by Szabo et al., 2014, who only classified pixels with low NDVI and high normalised digital surface model (nDSM) values, which "denoted a building with high probability".

The most preferred classification method was spectral angle mapper (SAM) (Cilia et al., 2015; Fiumi et al., 2012; Szabo et al., 2014). This algorithm treats each pixel value as a vector with as many dimensions as the total number of bands of the input imagery, and computes its angle to as many reference vectors as there are classes, assigning it to the class corresponding to the lowest angle. Each reference vector is commonly computed by averaging the spectra of several training pixels, or end members. Therefore, SAM is in principle a nearest neighbour algorithm. On the other hand, statistical approaches included linear and quadratic discriminant analysis Abriha et al., 2018, whereas the only ML-based methods were once again Bayes classifiers andSVMs (Hamedianfar & Shafri, 2014; Szabo et al., 2014), as well as RFs (Abriha et

al., 2018; Ilehag et al., 2018; Tommasini et al., 2019). Although Tommasini et al., 2019 based their decision to data constraints which could potentially cause more advanced algorithms to be ineffective as they would be improperly utilised, the preference of most authors to even then relatively outdated techniques remains unclear.

In a similar fashion to all other methods, results varied significantly depending on the particular input data and methodology. In terms of input data, Abriha et al., 2018 found that coastal blue and blue bands did not provide sufficient material discrimination, with green being a little bit better, and yellow the best. In addition, Hamedianfar and Shafri, 2014 concluded that using only the CIR band combination improved performance, but not at a statistically significant level. Furthermore, Ilehag et al., 2018 conducted a PCA and determined that their thermal and texture bands were the least significant, but did not provide any reasoning as to why that was. Moreover, Braun et al., 2019; Cilia et al., 2015; Frassy et al., 2014 all saw their results influenced by the low GSD of their datasets, which lead to low per-pixel building contribution, resulting in the mixed pixels with significant background content (Braun et al., 2019), omission errors (Frassy et al., 2014) and general ineffectiveness when it came to dense predictions (Cilia et al., 2015). What is more, Szabo et al., 2014 made a general observation that increased spectral and spatial resolution typically resulting in improved performance.

In terms of preprocessing, Abriha et al., 2018 found that pan-sharpening did not only improve performance regardless of the number of classes or method but also did not statistically degrade the spectral resolution of their input data, suggesting it to be an effective approach to increasing spatial resolution where possible. On the other hand, Osińska-Skotak and Ostrowski, 2015 found that intensity correction using the cosine method only improved performance on relatively flat roofs with simple geometries, whereas the intensity of pixels on roofs with high slope was overcorrected. The latter observation was also made by Szabo et al., 2014.

In terms of classification methods, Abriha et al., 2018 found that RF outperformed both linear and quadratic discriminant analysis methods, especially as the amount of training data increased. Similarly, Szabo et al., 2014 observed that SVM outperformed both SAM, which was unable to handle increased variability in the spectral signatures of various materials in their study area, as well as the Bayes classifier. In fact, they authors found that SVM could handle both their original 126-band dataset and an MNF-reduced 15-band version equally well, with an OA of 79.9% and 79.5%, respectively. In contrast, SAM, which was only tested on the original dataset performed only managed to score an OA of 59.8%, while Bayes scored 76.3% using 9 MNF bands. These observations were also supported by Hamedianfar and Shafri, 2014, who found that Bayes could not handle the inherent intra-class diversity and inter-class similarity which are associated with many roofing materials, and that SVM was significantly more robust. However, the authors also compared pixel- and object-level classification and found the later approach to achieve better results. Nevertheless, this result is expected given that a good segmentation quality and an effective classification rule-set, simply due to the fact that OBIA works with pixel-clusters and classifies based on value ranges, and hence considers rudimentary relationships amongst neighbouring

pixels, which traditional classifiers do not when they are naively applied at the pixel level. On the other hand, Ilehag et al., 2018 found pixel-wise classification with RF to outperform OBIA both with "naive" and building-level segmentations, although the authors noted that this could have been due their labels building building-wise, and therefore ignoring potential superstructures and general material variability within the same roof.

Finally, in terms of actual results in classes of interest to this thesis, Abriha et al., 2018 managed to achieve an impressive OA of 99.5% across red and brown tiles using the pan-sharpened version of their input data. Interestingly, they found that having a separate class each class to account for under-exposed roof segments introduced confusion and degraded performance. Similarly, Fiumi et al., 2012 scored a perfect 100% on their "brick" and metal classes. However, their method did not perform as well with bituminous materials, only achieving an OA of 59.6%. On the other hand, Hamedianfar and Shafri, 2014 achieved an accuracy of 78.48% and 67.31% on their metal and clay classes, respectively with RF. In addition, Osińska-Skotak and Ostrowski, 2015 managed 43.9% and 46.2% on sheet metal using ortho and true-ortho imagery, respectively. The corresponding accuracies for ceramic tiles, where 94.5% and 39.7%, whereas in the case of roofing felt performance fell once again to 52.8% and 48.4%. In general, it was evident that true-ortho imagery did not offer significant performance improvement. Furthermore, the authors found that generalising pixel-wise predictions to the building-level as a post-processing step consistently improved performance at a significant level. On the other hand, the application of a 3×3 pixel median filter on the pixel-wise results did lead to minor performance improvements but was ultimately not worth it (Osińska-Skotak & Ostrowski, 2015). Finally, Szabo et al., 2014 achieved an average accuracy of 97.37% across brown, green, and red tiles on 15 MNF-derived bands using SVM. Although the authors did include a "red tile in shadow" class in their study, it is not considered here due to its significantly lower performance, which could skew results. In addition, they scored, 88% and 82.5% on metal and tar, respectively, using their original dataset.

At this point it should be noted that most authors agreed on a variety of issues which were also identified by works concerning themselves with image- and object-based classification. For instance, Cilia et al., 2015 observed that material condition affected its spectral signature. Furthermore, many authors noticed confusion caused by significant intra-class variance and inter-class similarity (Cilia et al., 2015; Fiumi et al., 2014; Ilehag et al., 2018). In particular, Fiumi et al., 2014 observed that non-Lambertian surfaces caused issues, while Hamedianfar and Shafri, 2014 noted the similarity of ceramic tiles to roads and other impervious surfaces and the various potential appearances of metal. Moreover, (Fiumi et al., 2014) concluded that flat roofs are the easiest to model, followed by pitched and then vaulted.

2.1.4. Image and LiDAR Data Fusion

Although various authors have incorporated various products into their datasets, as are spectral indices derived from certain original bands, RGB, CIR, and thermal imagery,

as well as ancillary data, such as the general type of roof based on pitch and year of construction (Wyard et al., 2022), only two works concern themselves with the integration of imagery and LiDAR data. Interestingly, many authors, particularly in the OBIA sector did have access to LiDAR measurements but only used for image segmentation and not classification. However, as it will be made clear later in this section, three-dimensional (3D) information provided by ALS data has been shown to compliment the planimetric nature of conventional imagery, especially regarding the delineation of ground and roof surfaces featuring materials with otherwise similar spectral signatures (e.g., asphalt and membranes, bare soil and clay, etc.), as well as the mitigation of shadow effects as the typically NIR LiDAR sensors are not affected by them and can even operate at night.

In particular, Hamedianfar, Shafri et al., 2014 compared the efficiency of WV-2 and LiDAR data fusion in the context of urban scene classification conducting using both pixel-wise and OBIA tasks. As part of their study, they fused the previously pan-sharpened imagery with the corresponding nDSM, which was produced by subtracting the DTM from the DSM of their study area. The authors mentioned that the relevant ALS survey was conducted using single-return sensors at a time close to that of the acquisition of the WV-2 data in order to avoid potential temporal misalignment issues. Specialised software was used for rasterisation and to extract the ground points (Hamedianfar, Shafri et al., 2014). The exact fusion technique is not specified, but the authors implied that the nDSM was appended to the imagery stack as a pre-processing step. This type of data fusion is called feature-level fusion and is only one of three types of fusion. Unfortunately, the effect of the additional band was studied only in the OBIA task. However, there it resulted in an OA improvement of 7.84% (27% (!) for the metal class), which was later found to be statistically significant (Hamedianfar, Shafri et al., 2014).

The other paper which explored the performance effect of LiDAR data was that of Norman et al., 2020, who also used WV-3 imagery to compare SVM and DT as well as several fusion methods in the context of an OBIA task with the aim to classify metal, concrete and asbestos into "new" and "old", according to their weathering level. Their LiDAR data was collected at an operating flying altitude (OFA) of 600 m with a planar point density of 5 pt/m^2. In contrast to Hamedianfar, Shafri et al., 2014, the authors integrated five derivate products into their original dataset, namely the DSM, DTM and nDSM, as well as the slope and intensity scalar fields. Once again, specialised software was used to generate these datasets and information on pertinent processing techniques was not provided. However, close inspection of screen capture of the resulting rasters points to a triangulated irregular network (tin)-based rasterisation method as triangular artifacts are present in areas of large water bodies, where point density is inherently low. An exception to this observation is the DTM which was produced by taking the mean elevation of the points within each grid cell (Norman et al., 2020). Interestingly, the authors mention that the GSD of the rasterised data was 0.5×0.5 cm, but clearly meant meters, as such as a resolution would clearly result in a significant number of interpolation artifacts which were not present in the screen captures. In addition, it is mentioned that the slope layer was extracted

from the DTM but this is not possible as building outlines are clearly present in the former. Most likely, it was extracted from the DSM. In any case, the fusion methods tested where feature-level raster stacking similar to Hamedianfar, Shafri et al., 2014 and pan-sharpening using Gram-Schmidt and PCA. Although it is not explicitly mentioned, it is assumed that stacking involved resampling all bands to a common GSD, whereas pan-sharpening did not involve the LiDAR-derived bands as they were not included in either of the corresponding rule-sets (Norman et al., 2020). The results revealed stacking to be significantly better than any other combination of classifier and fusion method with the exception of PCA and Gram–Schmidt (GS) in the case of SVM and DT, respectively, where the observed improvement found to be statistically insignificant.

In conclusion, it is clear that the incorporation of LiDAR data and conventional MSI and HSI is generally beneficial. Nevertheless, the field remains significantly under-explored, with the only relevant works not conducting a particularly thorough analysis. For instance, Hamedianfar, Shafri et al., 2014 conducted and ablation study only with OBIA. On the other hand, Norman et al., 2020 compared heterogeneous data fusion to pan-sharpening which is an admittedly unrelated method as it relies upon data from the same sensor. Furthermore, Ilehag et al., 2018 mentioned that point density could be beneficial to performance on membranes, but to be best of the author's knowledge this had not been tested up to the time of writing.

2.1.5. Conclusion on the State of the Art

To the best of the author's knowledge, the literature review conducted in the context of this thesis is the first in the field to be structured per unit of analysis and include general-purpose material classes. In actuality, Abbasi et al., 2022 conducted a similar analysis which served as inspiration for this work, but it was focused on asbestos-containing materials, and only included Krówczyńska et al., 2020 as an example of DL-based classification, which was it own category, along with pixel-based classification and OBIA. Hence, Abbasi et al., 2022 did not thoroughly explore image-based classification. However, many of the challenges and corresponding opportunities identified by the authors are not only relevant to the context of this thesis but also continue to apply despite recent advancements in the field.

In particular, one issue identified in image-based and OBIA approaches is that most authors clearly spent significant amounts of time and effort to design dataset-, problem, or study-area–specific pipelines, be it either classifiers (J. Kim et al., 2021; Santos et al., 2023; Wyard et al., 2023), image segmentation configurations , or classification rule-sets . However, the supposed advantages of such approaches are not clear. For instance, several OBIA researchers achieved competitive results using Taguchi methods (Unal & Dean, 1990) to optimize their segmentation and classification results at a fraction of the effort as those who did the same manually. In addition, various authors successfully employed data mining algorithms to automatically produce statistically optimal classification rule-sets using features from a predefined bank , saving the effort of having to manually design them and eliminating potential human bias from

the process. The latter issue is also supported by Abbasi et al., 2022, who actually identified data mining and DL-based feature extraction as potential solutions.

Furthermore, there is a problem with how OBIA methods are presented and compared to pixel-wise classification. Before the artificial intelligence (AI) boom in 2012, OBIA had been established as an effective way to mitigate confusion issues of traditional ML classifiers when applied pixel-wise Gibril et al., 2017; Hamedianfar, Shafri et al., 2014. The most important such issue was the so-called "salt-and-pepper" error, which basically meant that as the classifier considered each pixel individually and did not take its relationship to its neighbours into account, potentially significant differences in the spectral signature of adjacent pixels due to intra-class variability, inter-class similarity, sensor noise, inadequate preprocessing or high spectral resolution could result in unreasonable or noisy predictions called hallucinations. However, modern DL-based image processing methods, such as CNNs, are specifically designed to deal with this issue and have actually made manual OBIA rather obsolete as they also (effectively) act as feature extractors, replacing the relevant segmentation step. Moreover, they are also technically able to operate without issue at a pixel level, at least given enough training data, thereby offering the opportunity for dense predictions, in turn reducing potential omission errors. Again, this recommendation is also supported by Abbasi et al., 2022, although they mainly mention it as a means of improving label localisation in the case of low-GSD imagery. What is more, predictions may then be generalised to any desired degree as post-processing step (Osińska-Skotak & Ostrowski, 2015). On the contrary, OBIA fixes the degree of generalisation at the segmentation step. All this considered, it is not particularly fair for modern OBIA papers to compare themselves to pixel-based methods using traditional methods as their technical inferiority is not contested in the first place. Instead, it would be more interesting to compare OBIA to DL-based pixel-level methods, but to the best of the author's knowledge no such work existed at the time of writing. At this point it is also important to note that modern DL-based methods do not generally suffer from the curse of dimensionality to the degree traditional ones did, meaning that dimensionality reduction in the case of HSI may no longer be necessary. In fact, several specialised mechanisms such as depthwise separable convolutions and spatial and channel attention may be able to learn both intra- and inter-channel relationships which are otherwise not easily known or predicted by humans.

Another point of interest is input data, in particular MSI and HSI, which typically require significant acquisition and processing costs Krówczyńska et al., 2020. Although it is clear that the spectral information offered by specialised sensors is unmatched and generally allows for intricate spectral signature modelling which can in turn facilitate both data annotation and classification, all authors in the image-based classification field used conventional RGB imagery without a significant, if any, performance penalty. In fact, Krówczyńska et al., 2020 heavily supported that RGB imagery is as competitive as HSI. This claim was also backed by their later work, which achieved SOTA performance in its field(Raczko et al., 2022). Given that these products are generally captured at extremely fine spatial resolutions, potentially making up for their lack of spectral richness, the trade-off between high spatial and spectral resolution, especially

in the context of DL-based classification should be explored further.

Finally, most authors reported confusion issues regarding both over- and under-exposed regions, mixing of background and roof pixels, and significant intra-class variability and inter-class similarity in various materials due to their appearance. Although few attempted to mitigate these issues (Wyard et al., 2023), nobody was particularly successful, especially when it came to shadows, given that the background could generally be identified and ignored in case it was not of interest (Tommasini et al., 2019; Trevisiol et al., 2022). In this context, the incorporation of LiDAR data was found to be particularly effective (Hamedianfar, Shafri et al., 2014; Norman et al., 2020). However, only two authors explored this avenue and were not particularly thorough, focusing their work on the particular classification method rather than the input data. As such, the motivation behind the selection of certain LiDAR-derived attributes over others remains unclear, and so does their individual performance effect, if any.

2.2. Deep Learning for Semantic Segmentation

2.2.1. Image Convolutions

In image processing, the convolution of an image $I \in \mathbb{R}^{h_I \times w_I}$, of width w_I and height h_I, with a filter or kernel $K \in \mathbb{R}^{h_K \times w_K}$, is defined as Eq. (2.1). In practice, this is generally the case that the kernel has and the same, odd number of rows and columns and that its dimensions are smaller than those of the image.

$$\begin{aligned} C\left(p, q; s_w, s_h, d_w, d_h\right) &\equiv \left(I \star K\right)(p, q) \\ &:= \sum_{m=0}^{h_K-1} \sum_{n=0}^{w_K-1} K\left(m, n\right) \cdot I\left(ps_h + md_h, qs_w + nd_w\right) \end{aligned} \tag{2.1}$$

where C is a matrix whose rows and columns are indexed by the integer coordinates p and q, correspondingly. This matrix is called the output feature map of the convolution and highlights the patterns that K detects in I. Formally, Eq. (2.1) corresponds to the cross-correlation of the image and the kernel, defined as discrete bivariate functions. In addition, the 2-tuples $(s_h, s_w) \in \mathbb{Z}$ and $(d_h, d_w) \in \mathbb{Z}$ are termed stride and à trous or dilation rate along the vertical and horizontal directions, respectively. The function of these parameters is presented below.

To perform a convolution, the filter can be thought of as a sliding window which is passed across the image in discrete steps. However, since cross-correlation is commutative, the opposite operation would produce the same result. In any case, the value corresponding to each position of this window is equal to the grand sum of the Hadamard product of the kernel with the relevant image segment (Figure 2.4).
In this context, the stride controls the step size of the kernel along each dimension of the image. As such, standard convolutions have a stride of one because the distance between any two subsequent positions of any kernel element is equal to one pixel., while larger strides are used as a means of compression or downsampling in order to

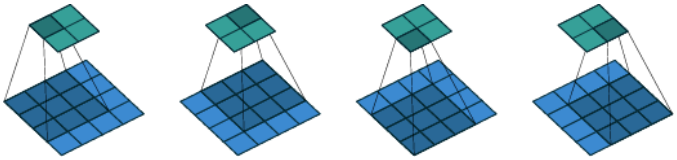


Figure 2.4.: Visual representation of the convolution of a 4×4 image with a 3×3 kernel, shown in blue and dark blue, respectively. The operation is parametrised by a stride and an atrous rate of one along each dimension of the image. The corresponding output feature map is shown in green has two rows and columns. The starting position of the filter does not matter. Reproduced from Dumoulin and Visin (2016).

preserve a minimum level of computational efficiency when multiple operations are used in series. In practice, it is generally the case that the stride along each dimension is the same, and it hence oftentimes specified using a single number.

The region of the image which contributes to the calculation of each element of the output feature map is called the receptive field. In the case of a single convolution, the dimensions of the receptive field are the same as those of the kernel. However, it is clear that the receptive field becomes wider in a superlinear fashion as more and more operations are performed in order. In practice, a large receptive field is generally beneficial as its maximises the number of contextual cues which influence each element of the corresponding output feature map. In fact, so-called global context is the main concept of ViTs (Dosovitskiy et al., 2020), the main drivers of the AI boom in the field of computer vision. Although the receptive field can be increased by simply increasing the size of the kernel, it is perhaps evident from Eq. (2.1) that this may be computationally prohibitive. Therefore, an alternative way of increasing the receptive field without incurring a significant performance penalty is through à trous convolutions.

In particular, a convolution with an atrous rate of d along a particular dimension of the image is mathematically equivalent to separating the elements of the kernel by $d - 1$ rows or columns of zeros (Figure 2.5). Thus, standard convolutions have a dilation rate of $(1, 1)$. In practice, it is generally the case that the atrous rate along each dimension is the same, and it hence oftentimes specified using a single number.

As implied by Eq. (2.1) and Figure 2.4, certain kernel positions are invalid because it partially extends "beyond" the image "bounds". For instance, the calculation of $C(h_I, w_I)$ includes image pixels which do not actually exist. Since these positions are ambiguous, they are generally not included in the output feature map (Figure 2.4). Apart from the stride and the kernel size, an alternative way of controlling the size of the feature map is through padding. In particular, a convolution with a padding of p along a particular dimension of the image is mathematically equivalent to appending p rows or columns to the image to both its bottom and top or left and right (Figure 2.6). The elements of these rows or columns may have any value, although zero is generally preferred because it essentially eliminates the contribution of no-data regions to the

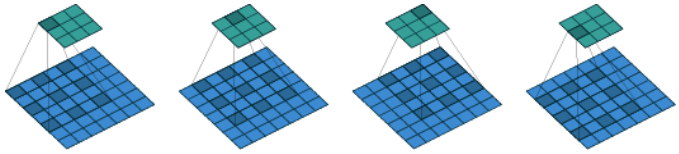
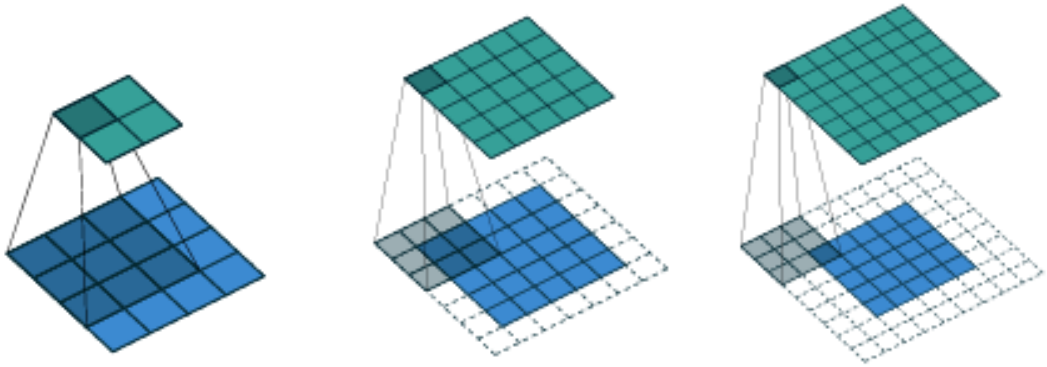


Figure 2.5.: Visual representation of the convolution of a 4×4 image with a 3×3 kernel, shown in blue and dark blue, respectively. The operation is parametrised by a stride of one and an atrous rate of two along each dimension of the image. The corresponding output feature map is shown in green has two rows and columns. The starting position of the filter does not matter. Reproduced from Dumoulin and Visin (2016).

feature map. In practice, it is generally the case that padding along each dimension is the same, and it hence oftentimes specified using a single number.



(a) a (b) b (c) c

Figure 2.6.: Example of no or valid padding (a), same padding (b), and full padding (c). Reproduced from Dumoulin and Visin (2016).

The convolution of an image featuring b_I bands is mathematically equivalent to performing b_I standard convolutions using Eq. (2.1) and subsequently computing the element-wise sum of each output feature map to produce a single-channel result. At this point it should be noted that each channel is generally convolved with a different kernel of the same size. However, in practice it is generally the case that the feature map has b_C bands, meaning that b_C convolutions must be performed as required and the individual results concatenated along their channel dimension. In this context, a special case of convolution is the 1×1 convolution, that is using an 1×1 kernel, which is mathematically equivalent to a linear transformation of the input bands. As such, 1×1 convolutions are used to control the size of the output feature map along its channel dimension (e.g., as a means of band compression before potentially resource intensive operations and at the end of semantic segmentation architectures to encode the output of the model in one-hot.)

Finally, another topic of note are separable convolutions, which exploit the mathematical properties of matrix operations to perform operations in multiple steps at an overall reduced computational cost in comparison to traditional convolutions. In particular, spatially separable convolutions apply to the case of a single input and output channel and replace the standard $k \times k$ convolution, which involves $k^2 + 1$ learnable parameters with a $k \times 1$ followed by a $1 \times k$ equivalent with a total of $2\,(k + 1)$ parameters, which is strictly less than $k^2 + 1$ of the original for all values of k greater than three.

Similarly, the depthwise separable convolution of an image featuring b_I bands when the corresponding output feature map has b_C channels is performed by convolving each input band with a single kernel and concatenating the resulting products along their channel dimensions, and subsequently applying an 1×1 convolution to the output to achieve the required band length. The first step is called depthwise convolution, while the second pointwise convolution.

2.2.2. Residual Neural Networks

In a traditional neural network, each layer learns a latent representation of the collective output from all previous layers. However, the inherent non-linearity of each layer can lead to significant variation in the input-output mapping as the network becomes deeper. Consequently, the degree of semantic coherence among the learned representations from layer to layer is often ambiguous, making it unlikely that deeper models will consistently outperform their shallower counterparts in practice.

According to K. He et al. (2015), this issue is sometimes not likely due to overfitting, particularly when specialised parameter initialisation and feature normalisation techniques are used, but rather the inability of the network to learn the required representations beyond a certain depth, either entirely or within a reasonable training duration. In order to combat this phenomenon, the authors proposed the now-pivotal concept of deep residual learning. Given an input signal x and a mapping $f\,(x)$ to be modelled, the main idea of this framework is to learn the corresponding residual mapping $f\,(x) - x$ and then simply transform the output of the network to the desired one by concatenating it with x using a residual or skip connection in an operation referred to as identity mapping. In this way, intra-layer representation coherence is now explicitly established as the initial input signal is propagated through the network, which also becomes significantly easier to train (T. He et al., 2018, Figure 2.7).

In this context, K. He et al. (2015) introduced a family of CNNs termed residual networks or ResNets using the building blocks presented in Figure 2.8.

By employing a common entry block or stem of a $7 \times 7 \times 64$ convolution with a stride of two and padding of three to reduce the spatial compression rate of the original input, along with batch normalisation, a ReLU, and a 3×3 max pooling layer with a stride of two, various models can be constructed by stacking basic blocks. For instance, the authors propose ResNet-18 (Figure 2.9), 34, 50, 101, and 152. The names of these models stem from the number of convolutional layers they contain, in addition to a fully-connected layer originally used by the authors for experimentation with image

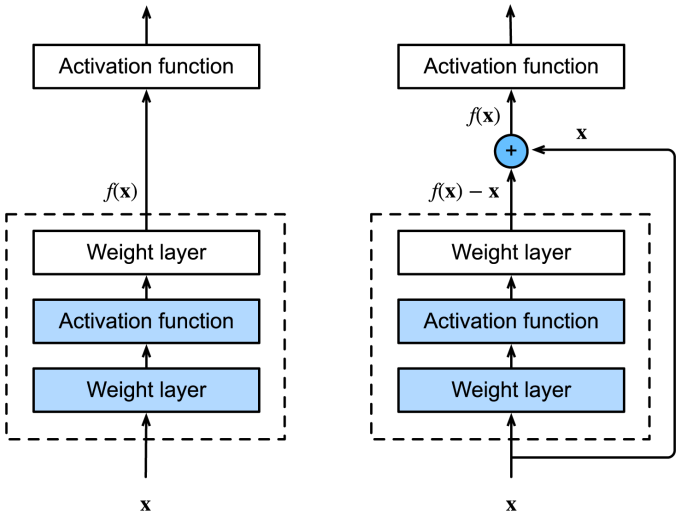


Figure 2.7.: Example traditional (left) and residual (right) neural network blocks. The optimal training parameters of the former block are arbitrary, while those of the of the latter are clearly zero. Reproduced from https://classic.d2l.ai/chapter_convolutional-modern/resnet.html.

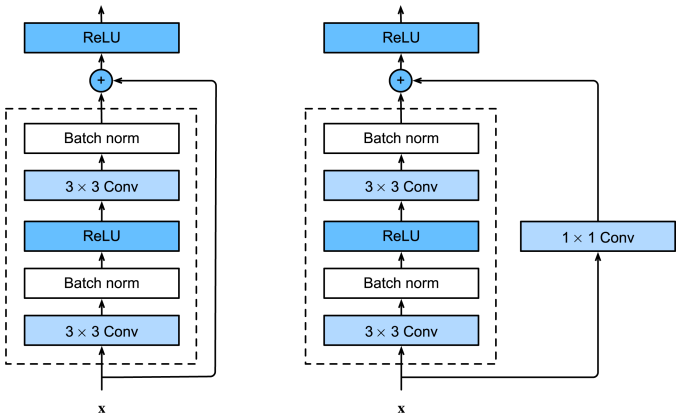


Figure 2.8.: Standard basic ResNet blocks. Identity skip connections (left) are used when x and $f(x)$ have the same dimensionality. Otherwise, projection shortcuts (right) are used. Reproduced from https://classic.d2l.ai/chapter_convolutional-modern/resnet.html.

classification. The blocks in each model are divided into four stages; with spatial downsampling being conducted in the first block of the second stage and beyond with a stride of two. Thus, concatenation is applied for identity mapping in the first stage, with projection shortcuts being used everywhere else.

At this point it should be noted that ResNet-50 and its more powerful counterparts use modified blocks called bottleneck blocks. The only difference between a basic and

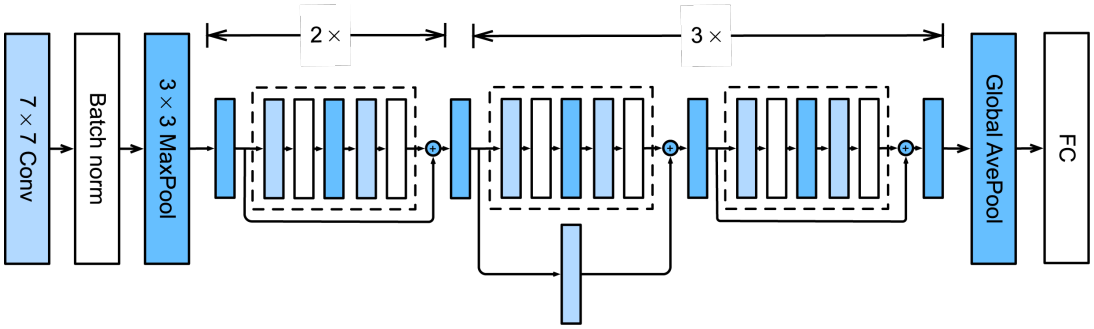


Figure 2.9.: Standard ResNet-18 model. The global average pooling and fully connected layers at the exit are used for image classification and are not part of the actual architecture. Reproduced from https://classic.d2l.ai/chapter_convolutional-modern/resnet.html.

a bottleneck block is that the two 3×3 convolutions are replaced by a single one which is both preceded and followed by a 1×1 convolution (Figure 2.10). This design aids in maintaining manageable computational complexity as the model deepens, though it is not as effective as the basic block (K. He et al., 2015).

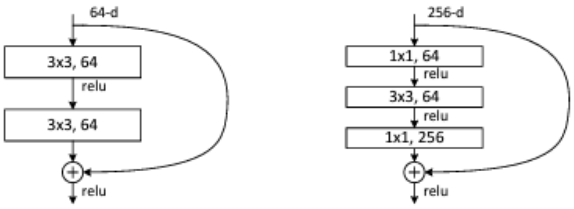


Figure 2.10.: Standard basic and bottleneck ResNet blocks. Reproduced from K. He et al. (2015).

The significant influence and popularity of ResNet[3] have led to numerous refinements of the original architecture over the years. Indeed, T. He et al. (2018) not only highlight the most substantial modifications warranting distinct names (Figure 2.11) but also present their own alongside various other notable techniques pertaining to enhancements in the overall training process for image classification. The first refinement, ResNet-B concerns improvements to the bottleneck block which is not relevant to the context of this thesis, and is thus not presented here. On the other hand, ResNet-C, which introduces a less resource intensive stem, composed of three 3×3 convolutions, is applicable. Finally, inspired by ResNet-B, ResNet-D replaces the original projection shortcut with 2×2 average pooling with a stride of two, followed by a standard 1×1 convolution, in order to preserve as much of the input feature map as possible.

[3]The original paper by K. He et al. (2015) had over 240,000 citations at the time of writing.

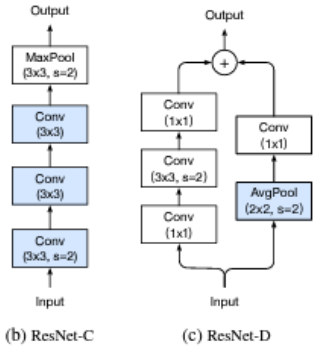


Figure 2.11.: ResNet-C and D modifications to the standard ResNet architecture. The former changes the stem, whereas the latter latter improves upon it by modifying the projection shortcuts. Reproduced from T. He et al. (2018).

2.2.3. The DeepLab Architecture

DeepLab is a family of semantic segmentation architectures which utilise a trous convolutions to achieve wide receptive fields at a minimal computational cost in comparison to larger kernels or deeper networks, while maintaining feature maps with adequate spatial resolution for performing dense predictions (Chen et al., 2016). The the main idea of DeepLab is to repurpose traditional image classification models for use in semantic segmentation by replacing their later pooling and strided convolution layers with dilated convolutions (Chen et al., 2016, 2017). This modification allows for preserving or even increasing the spatial resolution of the final feature map without requiring transpose convolutions, which are relatively resource intensive (Chen et al., 2016).

In addition, a significant contribution of these models is the so-called à trous spatial pyramid pooling (ASPP) block, which captures multi-scale contextual information by integrating feature maps generated using varying dilation rates into a single product (Chen et al., 2016, 2017, 2018). Although the first iteration of DeepLab (Chen et al., 2016) did not use ASPP, this was changed in the same publication which also introduced DeepLabv2. The block was composed of four parallel series of a single atrous 3×3 convolutions with a dilation rate of six, twelve, eighteen, and twenty four, followed by two standard 1×1 operations. The ASPP was later refined for DeepLabv3 (Chen et al., 2017) to include image-level features as a means of capturing global context cues while using relatively smaller atrous rates such that number of valid kernel parameters in the parallel atrous convolutions is maximised. These features are extracted by compressing the spatial dimensions of the input feature map to the ASPP block to $1x1$ using average pooling, convolving the output with a number of 1×1 kernels, originally 256, normalising the resulting product using batch normalisation and then bilinearly upsampling it to the required spatial dimensions to ensure compatibility with the remaining ASPP components. As such, the original atrous convolution with a dilation rate of 24 was replaced by a standard 1×1 operation.

Finally, DeepLabv3+ (Chen et al., 2018, Figure 2.12), preserves the ASPP block of its predecessor but modifies the final feature map upsampling step by adopting an encoder-decoder structure for improved object delineation.

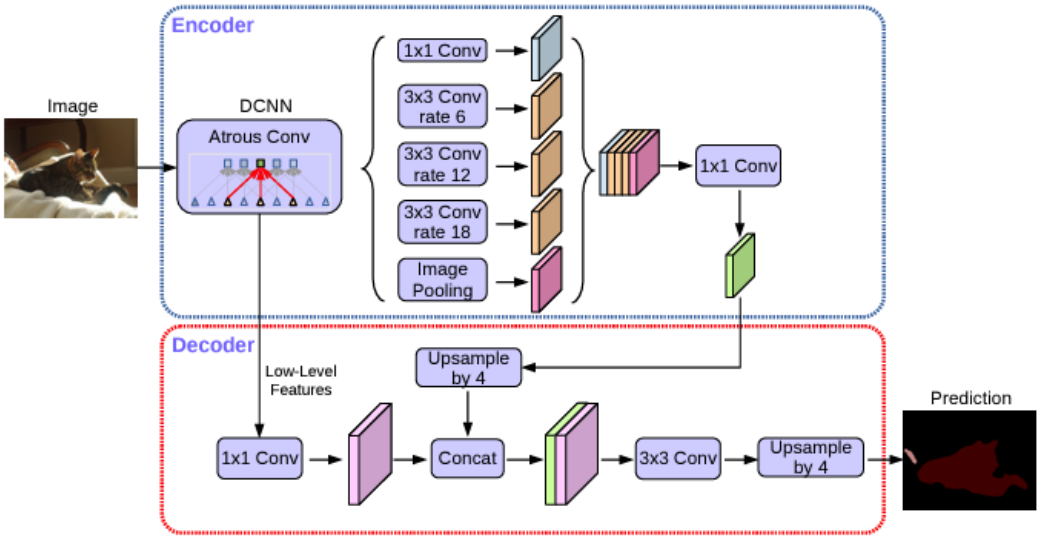


Figure 2.12.: Standard DeepLabv3+ architecture. Reproduced from Chen et al. (2018).

In particular, the output of the ASPP is once again convolved with a standard 1×1 kernel. However, the resulting product is not immediately upsampled by a factor of eight to match the spatial dimensions of the initial input image, as was the case in DeepLabv3, but instead two or four corresponding to an output stride[4] of eight or sixteen, respectively. Subsequently, the output feature map is concatenated with the input feature map to the ASPP block after it has been convolved with a standard 1×1 kernel to reduce its channel dimension, originally to 48. Finally, the resulting product is passed through a classification head composed of a standard 3×3 convolutional and a four-times bilinear upsampling layer to match the spatial dimensions of the initial input image.

At this point it should be noted that all convolutions specific to DeepLabv3+ are implemented as depthwise separable for increased computational efficiency (Chen et al., 2018). In addition, it is implied that each convolutional layer is always followed by a batch normalisation and a ReLU layer, even if this is not explicitly mentioned.

2.2.4. Splitting and Weighting

It is commonly known that supervised learning tasks require the corresponding reference dataset to be split into at least two subsets, with one being used for training

[4]The output stride is defined as the ratio of the spatial resolution of the initial input image to that of the input feature map to the ASPP block (Chen et al., 2017).

and the other reserved exclusively for testing purposes. This approach allows for unbiased predictive performance and generalisability estimations as the outcome of the learning process is decoupled and hence statistically independent from the test set. However, training is commonly influenced by various parameters which should generally be *optimized* or *tuned* with respect to a certain baseline in order for the model to approach or even achieve the true performance potential of the underlying model. These parameters are called *hyperparameters* because they typically do not influence the core structure of the model. However, they may be both intrinsic and extrinsic to it (e.g., the number of kernels in a certain layer, learning rate, optimisation algorithm, etc.). In this case the henceforth referred to as the HPO must be performed on a third *validation* subset in order to maintain the statistical integrity of the test set.

When training models with relatively low computational requirements (e.g., clustering algorithms, RFs, SVMs, etc.), as well as in situations where the reference dataset is so constrained in size such that a potential reduction of the training subset would aggravate or result in relevant statistical reliability issues (e.g., high step-to-step or trial-to-trial performance variance), the validation set may be simulated using cross-validation (CV). Nevertheless, because this approach generally requires models ab initio training for each relevant fold, implementing it in the case of deep neural networks (DNNs) is considered to be exceptionally expensive computationally to the point where it is commonly prohibitive.

Therefore, once annotated (Section 3.4.2), the reference dataset is split into a training, validation, and test subset. Although a random splitting approach is typically standard practice in most relevant tasks, it may be undesirable or even simply inadequate in situations where a particular inter- or intra-set distribution is necessary. For example, Boguszewski et al., 2021 split their country-level dataset randomly. On the other hand, Rottensteiner et al., 2012 first divided each of their city-level datasets into an orthogonal grid of approximately city-block–neighbourhood-sized cells, and consequently generated ensured that the corresponding training and test subsets such each contained all rows of this grid in a predetermined proportion. In other words, a certain number of columns of said grid were assigned to the former set and the remaining to the latter. Finally, Maggiori et al., 2017, purposefully excluded certain countries represented in their global dataset from the relevant test set in order to ensure that it contained truly unique data, in turn guaranteeing its spatial independence.

In addition, there is comparable variety in class-based splitting methods. Perhaps the most significant works in this context concern themselves with the development of various sampling techniques in the case of reference datasets with various degrees of label imbalance. This is because the are two critical issues this phenomenon may likely cause. The first is that random splitting under severe class imbalance regimes, be it naturally-occurring, due to an inadequate reference dataset size, or otherwise problematic sampling method, may lead to some minority examples being all assigned to one or two subsets instead of all. Clearly, this is a cause for concern because it means that the underlying model will either not be trained, validated or tested on these items. This type of issue is typically mitigated by *stratified sampling*, which aims to preserve the global class distribution across each split. In the case of multi-

label problems, as is semantic segmentation, stratification is commonly performed using iterative approximation algorithms, with the most prominent works being those of Sechidis et al., 2011 and Szymański and Kajdanowicz, 2017a.

Finally, the second problem potentially caused by class imbalance occurs after splitting the reference dataset and concerns the fact that stratification does not actually guarantee that the training will be balanced. Consequently, this may result in insufficient learning of the minority classes or, equivalently in terms of predictive performance, statistical bias towards the majority classes. This issue is generally resolved by assigning a greater importance to under-represented samples via appropriate modifications to the loss function (King & Zeng, 2001; Lin et al., 2020; Pazzani et al., 1994), targeted over- (Ling & Li, 1998) or under-sampling (Kubat et al., 1998) of the minority and majority labels, respectively, or a combination of the above (G. E. A. P. A. Batista et al., 2004; G. E. d. A. P. A. Batista et al., 2003). Furthermore, more advanced methods basted on boosting, bootstrap, and bootstrap aggregating or bagging have been implemented in free and open-source software (FOSS) (Lemaître et al., 2017).

2.2.5. Data Augmentation

As explained in, image convolutions are generally considered invariant to translations but are affected by other transformations, such as rotations and transpositions. In addition, there are various cases where models, especially in the realm of deep learning, would greatly benefit from larger reference datasets but the corresponding data collection or annotation process is too resource-consuming. In this context, data augmentation is a technique used to tackle both issues simultaneously.

In training, the main purpose of this method is to complement the corresponding dataset with artificial data by generating mutated views of original samples. Depending on the particular transformations used to produce this data, augmentation can introduce various types of invariance at the prediction level and simulate potentially missing or rare training cases, in turn improving model generalizability and inference performance. On the other hand, augmented data is commonly used in validation and testing to ensemble predictions for a given input in order to increase the predictive power of the model.

Data augmentation may be applied either as data preprocessing step (offline) or as per-batch after is it sampled and before it is received by the model (online). Offline augmentation is generally more resource-efficient and may help better visualize the actual transformations being applied facilitating reproducibility and data management. However, pre-augmented datasets obviously require more disk space to store. In addition, updating an augmentation parameter clearly requires partial or even complete dataset regeneration, depending on the particular change, along with everything that entails in terms of resources.

In general, there are three main types of data augmentation transforms: geometric, intensity, additive, and subtractive. Geometric transforms model different scene orientations, views, and scales by applying affine transformations or cropping to the input data. On the other hand, intensity transforms generally simulate adjustments in

the colorimetric properties of the input data (e.g., brightness, contrast, hue, etc.) to simulate various changes in colour and lighting conditions. Next, additive transforms inject foreign data into the input in order to model certain real-world cases witch may not be covered in the original dataset. For instance, out-of-focus scenes and relative camera-object movement is generally modelled using Gaussian blur. Similarly sensor dust or noise is simulated by using salt-and-pepper and Gaussian noise, respectively. In addition, there exist transforms which aim to create mosaics by combining random crops from various inputs (Yun et al., 2019) The semantic opposite of the latter augmentation, where parts of the input are replaced with invalid pixels (DeVries & Taylor, 2017) is a type of subtractive transform. Finally, there exist quasi-augmentations such as data scaling or concatenating the samples with additional data computed on demand in cases of online augmentation. At this point it should be noted that this in no case aims to be a comprehensive review of all available augmentations and their types, but a relatively simplistic overview for the uninitiated reader. After all, they are oftentimes applied sequentially for additional complexity.

The choice of appropriate augmentations for a certain task and reference dataset is not arbitrary. For instance, geometric augmentations may cause so much distortion to the input data to the point where it becomes invalid. In addition, there are certain cases where the invariance introduced by a particular augmentation is undesirable. For example, applying flip augmentations to traffic signs turns would clearly result in left and right turns being semantically merged. This is further complicated in the field or remote sensing, where imagery is not only true-colour, and hence certain intensity augmentations are undefined (Stewart et al., 2023). However, care is required even in the case of RGB imagery since each band often represents a particular spectral signature. In fact, it has been shown that certain augmentation do not corrupt this information at all, while others only up to a certain degree of intensity (Burgert & Demir, 2024). Furthermore, it has been shown that CNNs pretrained on the ImageNet dataset are biased towards texture (Geirhos et al., 2018). This appears to also stand true in the case of earth observation (EO) imagery, where models are relatively robust to colour distortions, where applicable (e.g., in the case of pan-sharpened imagery), but not so to texture (Willbo et al., 2024). Nevertheless, Willbo et al., 2024 proceed to argue that this result is class-specific.

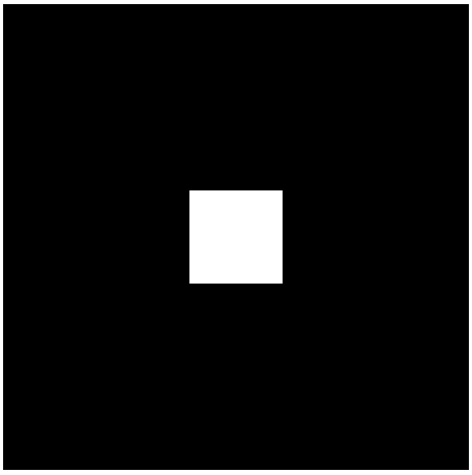
2.2.6. Loss Functions

The most common loss function in the field of semantic segmentation of RS imagery is cross entropy (CE), optionally combined with a boundary- or shape-aware component, most commonly Dice (Sudre et al., 2017) and Hausdorff distance (Karimi & Salcudean, 2020) loss, respectively.

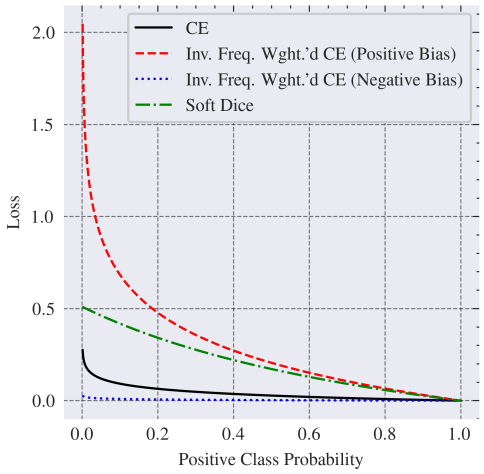
The rationale behind compound objectives is that CE basically acts as a proxy for the actual metric(s) of interest, mainly because it generally provides a smoother gradient landscape. However, it is rarely itself the actual learning goal. For example, given a binary classification problem, this may be obtaining a satisfactory accuracy score on the test subset. Clearly, this is related to maximising CE, as the model

is encouraged to be increasingly confident about its correct predictions. However, it should be noted that given a certain input, the output probability vectors $\langle 0.99, 0.01 \rangle^\intercal$ and $\langle 0.51, 0.49 \rangle^\intercal$ correspond to the same score despite the fact that the former may require significantly more extensive training in order to be achieved than the latter.

In addition, in semantic segmentation tasks where performance is typically evaluated in terms of the overall overlap between prediction and ground truth masks, CE may be at a particular disadvantage in comparison to set distance or similarity measures due to its pixel-wise nature. For instance, let the example mask and related prediction presented in Figure 2.13. This configuration models a very common scenario in the ground truth of the implementation dataset; a relatively small object (e.g., skylight, solar panel, window, etc.) surrounded by a uniform material (e.g., gravel, membrane, tile, etc.). Since each class in the reference dataset is equally important, the model should ideally learn both classes equally or at least almost equally well.



(a) Ground truth.



(b) Model Prediction.

Figure 2.13.: Binary ground truth mask and corresponding prediction loss for various objective functions and values of the conditional probability of the positive (i.e., white) class at the corresponding location. The probability of the negative (i.e., black) class everywhere else is always one.

Considering only CE and Dice loss in Figure 2.13b, it is clear that the former is overwhelmed by all the correct predictions of the majority class that it doesn't penalise the model too harshly, at least in comparison to the latter, for "minor" classification errors, even when it is extremely confident in predicting the wrong class. On the other hand, Dice loss assumes is maximum possible value in this example in said regime, naturally treating the underlying class imbalance issue Furthermore, although it is obvious that this problem can also theoretically be mitigated by appropriately weighting CE, it should be stressed that the adopted weighting scheme must happen to be biased towards the positive class. While this seems to be intuitively true, and

it most frequently is in practice, at least in the implementation dataset, one obvious counterexample is green roofs, where some membrane or gravel substrate may be visible due to weathering or other technical reasons. Then, the loss becomes negatively biased since the majority mask segment in Figure 2.13a now belongs to an otherwise minority class, and the substrate may be missed.

3. Methodology

3.1. Overview

The purpose of the proposed methodological framework (Figure 3.1) is the automatic construction of a semantic segmentation dataset which may be used to train a downstream ML model for the task roofing material classification. As is the case with all supervised learning datasets, it contains several input-output examples to be used for training and testing, as well as validation, where applicable.

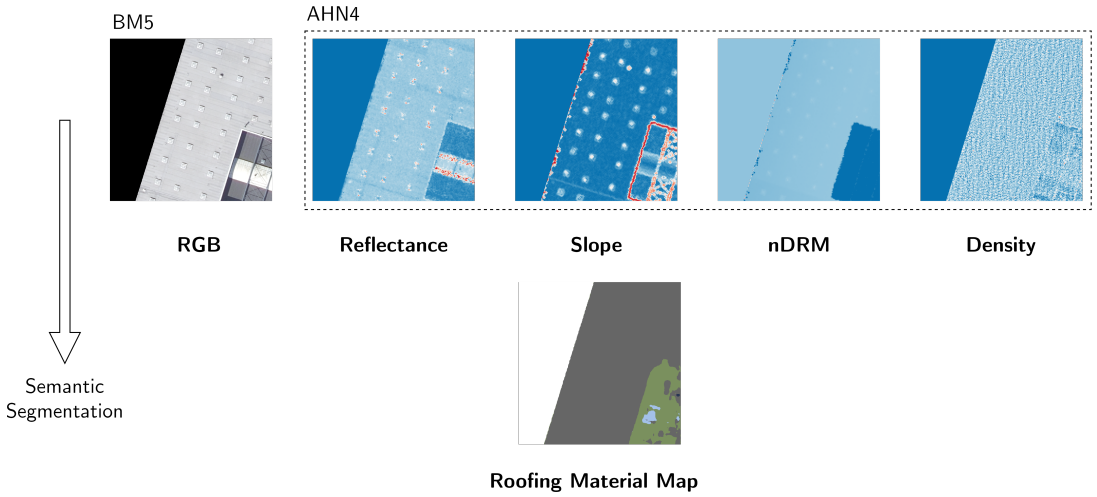


Figure 3.1.: Overview of the proposed methodological framework.

Each input example is a single 512×512 pixel image. The image contains seven bands and is extracted from a larger raster stack corresponding to a randomly sampled 3DBAG tile. Each stack may provide multiple images. The first three channels of the image comprise its RGB component, which is generated from mosaicked BM5 data. Conversely, the last four bands are derived from the AHN4 point cloud. In particular, the reflectance and planar point density of the dataset are directly rasterised. On the other hand, the slope channel is computed from the corresponding DSM. Similarly, the normalised digital roof model (nDRM) is defined as the difference of the DSM and the digital roof model (DRM), a map of the median roof elevation in areas where buildings are present and zero elsewhere. Buildings are delineated by their LoD2.2 footprints, which are extracted from the 3DBAG. In addition, this dataset serves as the source of the median roof elevation of each building. In order to facilitate downstream tasks, all bands are masked after they are concatenated using the corresponding footprints and

background or non-building pixels have been assigned a value of zero. Furthermore, a maximum limit on the number of background pixels is imposed on each image when splitting the corresponding raster stack. Images exceeding this limit are eliminated from further processing.

Each image is accompanied by the corresponding output example, which is a single 512×512 pixel image called the ground truth segmentation mask. The mask contains a single band mapping each non-background pixel to a numerical label representing a unique material and is produced manually by annotating the RGB component of the image using specialised external software. In order to facilitate the indication of ambiguous image regions which should be ignored during training, any building pixel not annotated or explicitly marked as invalid is masked in the same way as the image, along with all background pixels, when the annotations are reintroduced into the proposed workflow.

Finally, once the reference dataset has been generated it may be used to train a downstream ML model for the task roofing material classification. An implementation of this downstream task is provided in Chapter 4.

3.2. Overview of Source Datasets

3.2.1. The 3DBAG Dataset

The 3DBAG (3D geoinformation group & 3DGI, 2024; Peters et al., 2022) is an open dataset of the Dutch building stock co-developed by the 3D Geoinformation Group and its spin-off, 3DGI. It contains more than ten million automatically generated and individually validated, two-dimensional (2D) and 3D models at three LoDs, namely 1.2, 1.3, and 2.2, as specified by Biljecki et al. (2016).

Each model is constructed at the location of the footprint of the corresponding building as presented in pertinent cadastral data. The footprint is represented by a 2D polygon. In addition, height information is provided by the AHN point cloud (Section 3.2.3). The latest version of the 3DBAG at the time of writing (v2024.02.28) relies on AHN3 and AHN4 with each model being constructed using the dataset with the best point coverage. The first step in model generation process entails the splitting of the footprint into segments of equal or linearly varying elevation by applying a plane detection algorithm on the relevant subset of the point cloud. Each detected plane represents a different roof segment and must have a minimum number of points associated with it as well as minimum surface area. Subsequently, each plane is appropriately extruded from the reference ground level computed around the footprint based on the elevation of their vertices. The main difference between the various LoD models lies in the number of footprint partitions before the extrusion step. LoD1.3 models are constructed such that no neighbouring segments have a height difference of less than 3 m. The reference height of non-flat planes is defined as the 70^{th} percentile of the height of all corresponding points. Similarly, LoD1.2 models have a single roof segment which is extruded at the same percentage (Figure 3.2 and Section 3.2.3).

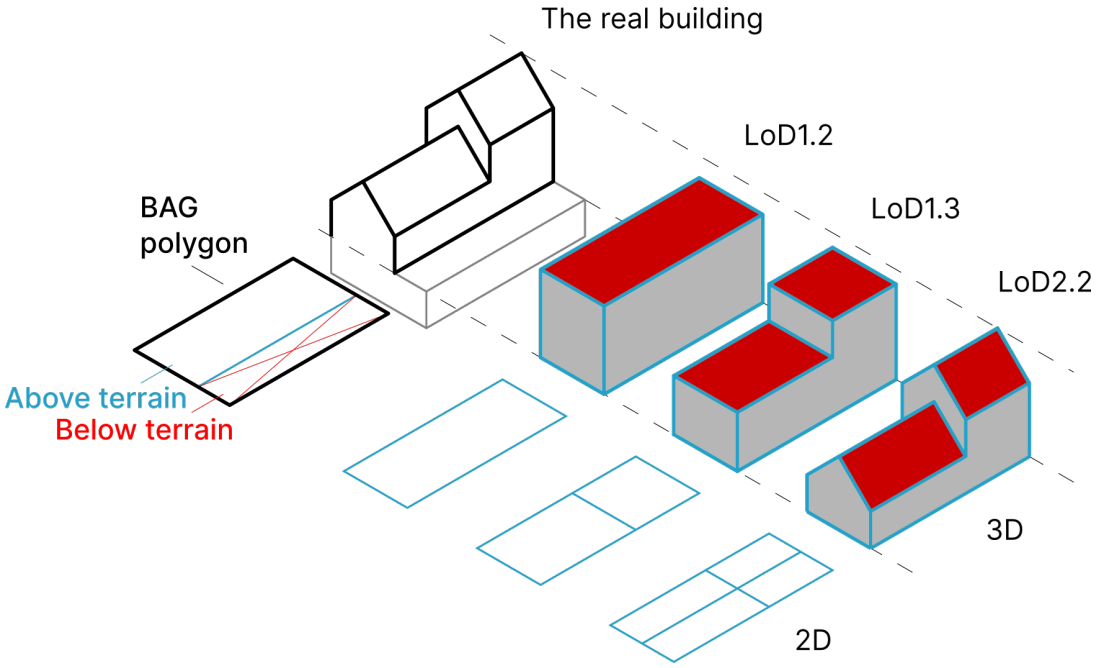


Figure 3.2.: The correspondence between an arbitrary, real-life building, its BAG footprint, and its various representations in the 3DBAG. Adapted from Peters et al., 2022.

In this context, underground and spatially overlapping building parts are not reconstructed due to the absence of pertinent elevation data. In addition, buildings which cannot by surveyed accurately using LiDAR techniques, such as greenhouses, are modelled only in LoD1.1. Interestingly, the same is true for large storage facilities, which are explicitly marked in relevant cadastral data, due to relatively uniform and flat morphology of their roofs. However, because buildings of this type are not processed normally, the corresponding models may by exception include any underground parts if they are present in the pertinent footprints. Furthermore, each model is associated with various semantic attributes, such as the corresponding cadastral information, labelled wall and roof surfaces, as well as data gathered during its construction (e.g., the median roof elevation, the total number of floors, the slope of each roof segment, etc.).

The 3DBAG is tiled using a quad-tree structure (Figure 3.3). Each tile contains a maximum of 3,500 models and is uniquely identified by a using a three-part identification (ID) defining the level of the tile in the tree and its location in a local orthogonal coordinate system. Naturally, tiles whose ID starts a with higher digit have a smaller surface area.

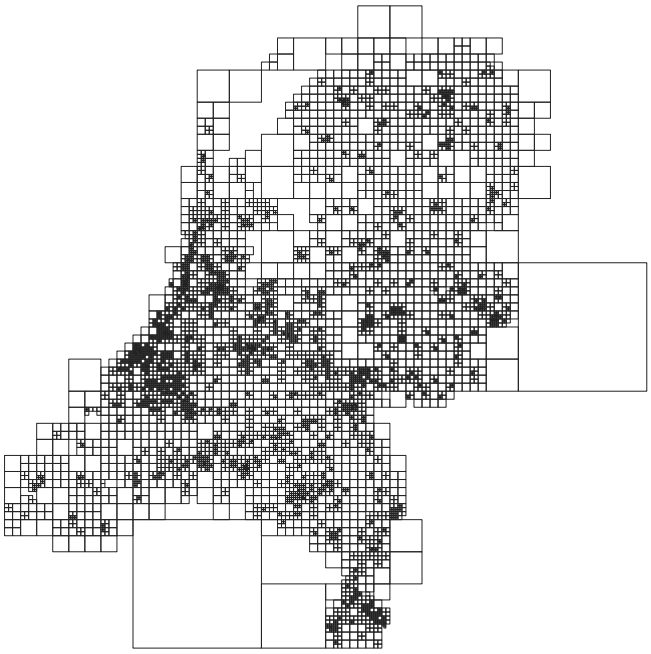


Figure 3.3.: Tiling structure of the 3DBAG.

3.2.2. The Dutch Aerial Imagery Programme and the BM5 Dataset

The Dutch Aerial Imagery (Beeldmateriaal Nederland; BM) programme is a continuous effort of several governmental organisations with the purpose of providing nationwide monocular and stereoscopic RGB and CIR aerial ortho-imagery at a yearly time scale. The program has been ongoing since 2009 and is currently in its fifth iteration (BM5), with the most recent products at the time of writing being those from 2023. Apart their actual content, images may also divided by their GSD into high- and low-resolution. The former products are captured during the leafless winter season (i.e., mid-February up to mid-April), whereas the latter in the summer. Since one of the primary purpose of CIR images is vegetation monitoring, they are only available in low resolution. In the context of this thesis, the high-resolution RGB imagery is used.

Flights are conducted incrementally by various participating parties following a pertinent tendering process. In 2023, the country was divided into five parcels which where further split into sixty six blocks (Figure 3.4).

As a result, the GSD of these products varies due to differences and limitations or restrictions in flight and instrument characteristics. In general, the high-resolution images are captured at a GSD of 4–10 cm, whereas the low-resolution products 5–10 cm. Hence, both high- and low-resolution images are later resampled to 8 cm and 25 cm, respectively. All images are captured using stereoscopic sensors with an overlap of at least 80% and 20–30%[1] along the and perpendicular to the direction of flight,

[1]The high-resolution images have a transverse overlap of at least 30%, whereas the low-resolution

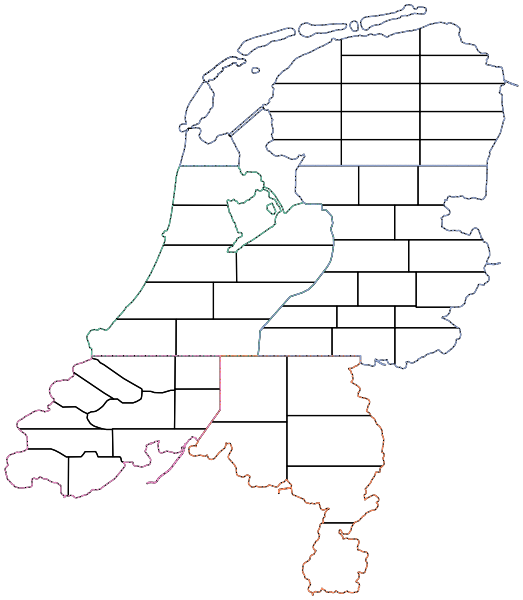


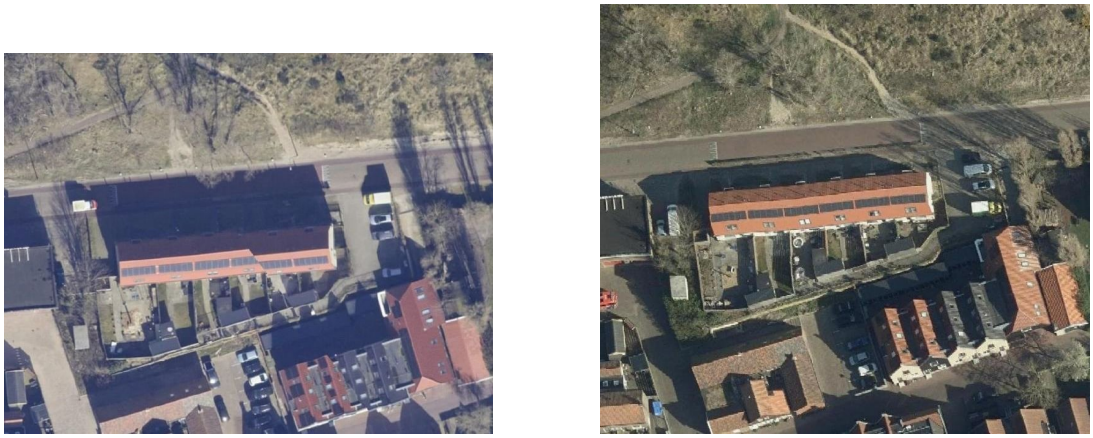
Figure 3.4.: Parcel and block layout of the 2023 edition of the BM5 dataset.

respectively. In addition, special care is taken for flights to be conducted on days with clear skies[2] and at a time frame which ensures a minimum sun elevation. Afterwards, these products are georeferenced using aerial triangulation and block adjustment and in turn appropriately collated to create a single mosaic of the entire country. Special attention is given to placing cutting lines at ground level such that they intersect as little infrastructure as possible. Furthermore, the mosaic is orthorectified using a DTM. Moreover, the mosaic is colour-corrected for increased homogeneity. Because this process is generally time-consuming and not required for certain time-sensitive applications (e.g., detecting mutations, issuing permits, etc.) a so-called "quick" mosaic (Figure 3.5) is also made available as soon as it is available. What is more, strict data quality requirements result in the final high-resolution mosaic having a minimum guaranteed planimetric error of 20 cm along either of the longitudinal and latitudinal axes, whereas the low-resolution product 37.5 cm. In addition, there are other pertinent requirements on which the mosaics are actually allowed to deviate by a maximum of 2%, albeit under the condition that these errors do not occur across contiguous areas.

Finally, both high- and low-resolution mosaics are delivered as singular ECW files. Alternatively, the high-resolution products are also made available as individual 1×1 km tiles under the GeoTIFF format with an 8-bit colour depth per band.

products 20%.

[2]The maximum cloud and shadow cover for any given image is 2%.



(a) Quick ortho-imagery.

(b) Final ortho-imagery.

Figure 3.5.: Quick and final ortho-imagery of an example scene. Reproduced from Het Waterschapshuis, 2024.

3.2.3. The Dutch Elevation Programme and the AHN4 Dataset

The Dutch Elevation (Actueel Hoogtebestand Nederland; AHN) programme is a continuous effort of several governmental organisations with the purpose of providing a nationwide LiDAR point cloud as well as the derivative DTM and DSM at a GSD of 0.5 and 5 m and a five- to six-, and later two-year time scale. The programme has been ongoing since 1996 and is currently in its fourth iteration (AHN4; 2020-2022). In the context of this thesis, the raw point cloud is used.

The point cloud is captured during the leafless season of each collection year (i.e., December up to March). Flights are conducted incrementally by various participating parties following a pertinent tendering process. For AHN4, the country was divided into five parcels (Figure 3.6).

As a result, the planar point density of these point cloud varies due to differences and limitations or restrictions in flight and instrument characteristics. In general, the point cloud is captured with a density of 10–24 points per square metre (pts./m^2). All measurements are conducted using full-waveform sensors which are mounted on special stabilisers such that they always point along the vertical direction regardless of the movement of the aircraft. The overlap between adjacent flight lines is 20–35%. In addition, special care is taken for flights to be conducted on days with clear skies and acceptable terrain visibility in terms of potential flooding, hail, sleet, or snow cover, etc.. Afterwards, the point clouds corresponding to each individual flight line are appropriately collated to create a single mosaic of the entire country and the resulting product is geometrically calibrated along both the longitudinal and latitudinal axes by aligning the ridge lines of roof gables which are visible along adjacent flight lines using inertial measurement unit (IMU) data. This process results in a maximum systematic and stochastic planimetric error of 8 cm and 5 cm, respectively. Furthermore, the elevation of the point cloud is separately calibrated using pertinent reference measure-

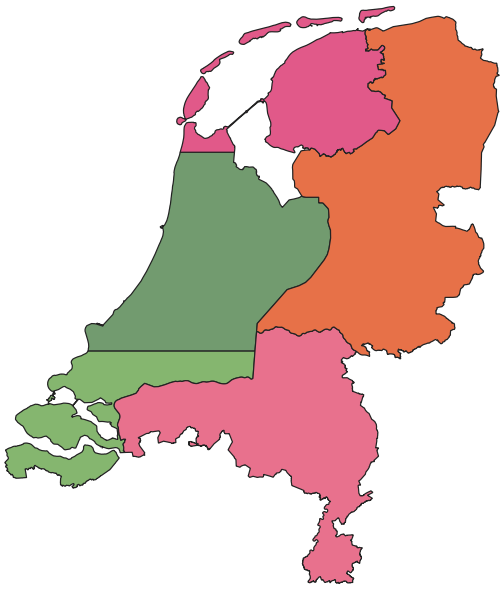


Figure 3.6.: Parcel layout of the AHN4 dataset.

ments. The corresponding maximum systematic and stochastic planimetric errors are both equal to 5 cm. What is more, the point cloud is classified into several predefined semantic categories according to the LAS 1.4 specification. In particular, AHN4 includes labels for buildings, ground, and water bodies. Points corresponding to the first two categories are labelled automatically. On the other hand, water bodies are initially classified as ground, and are hence manually corrected. In addition, an infrastructure class containing objects such as bridges, jetties, and gantry masts, is provided. These items are also manually labelled. All other points are labelled as unclassified.

Finally, the point cloud is delivered as individual 5×6.25 km tiles under the LASzip file format. In addition to the field specified by the LAS 1.4 specification, all files contain three additional attributes in the Extra Bytes variable length record (VLR), namely amplitude, reflectance, and deviation, which are specific to RIEGL sensors. In particular, according to RIEGL Laser Measurement Systems GmbH, 2019, reflectance is defined as the range-normalised difference between the amplitude of a particular target and that of a "white flat target [...], oriented orthonormal to the beam axis, and with a size in excess of the laser footprint". In turn, the amplitude, A_{dB} is defined as:

$$A_{db} = 10 \log_{10} \frac{P_{act}}{P_{min}}$$

where P_{act} and P_{min} are the power of the corresponding backscattered signal and the detection threshold of the particular instrument, respectively. Alternatively, 1×1.25 km tiles are also made available by the GeoTiles project of the Optical and Laser

Remote Sensing group of the Department of Geoscience and Remote Sensing at the Delft University of Technology. However, these tiles feature a 25 m overlap amongst them to facilitate parallel processing, which needs to be appropriately handled when merging them.

3.3. Reference Data Generation

In order to facilitate a fair representation of the various geographic regions within the study area, as well as the material classes under consideration (Section 3.4.1), a multistage approach is used to generate the reference dataset at the national level. In this way, pertinent statistical biases (e.g., systematic errors due to similar lightning and viewing conditions across spatially neighbouring scenes) are assumed to be minimised, in turn increasing model generalisability on unseen data.

The sampling heuristic is inspired by the works of Stewart et al. (2023) and Y. Wang et al. (2023), which are themselves based on that of Mañas et al. (2021), supporting the notion that urban scene variability is normally distributed around major population centres. However, this hypothesis is also taken to be true concerning material variability. Indeed, experience dictates that building use in increasingly rural areas is commonly limited to only certain types (e.g., commercial, industrial, etc.), mainly owing to relevant zoning policies and legislation which have become commonplace around the world. Hence, the fact that numerous elements of such buildings — including their roof coverings — are generally uniform and even standardised in numerous cases (e.g., owing to prefabrication, which is used as a means of meeting special use requirements at minimal cost) means that repeated sampling of said areas offers limited contextual added value beyond a particular threshold. Furthermore, it should be noted that this method caters to the tiling structure of the 3DBAG, which is proportional to building density, which is naturally positively correlated with population density. This means that population- instead of uniformly-weighted sampling is bound to achieve higher building coverage.

The main steps of the sampling process are listed below:

1. Sample one city with a population of at least 100,000 from a uniform distribution. This sample is henceforth referred to as the *seed city* and is represented by a 2D point whose geographic coordinates are determined by a geocoding service[3].

2. Sample a 3DBAG tile within 15 km from the seed city from a symmetric normal distribution. This distribution is centred around the seed. Although prior works both employ a 50 km sampling radius, this was deemed to be too large given the relatively low proximity amongst major Dutch urban centres, especially in the south.

[3] In actuality, Statistics Netherlands releases population statistics for municipalities rather than cities. Hence, high geocoding quality is crucial for appropriate point placement; especially so because naive approaches (e.g., placing points at the centroid of the corresponding municipal polygons) are not guaranteed to be valid.

3. Assuming that the tile has not been sampled before, generate the related *raster stack* (Section 3.3.1) and split it into square *chips* or *patches* of a given side length. These chips comprise the *reference dataset* to be annotated (Section 3.4) and in turn used to train the model (Section 4.3). The benefits of using patches instead of tiles are outlined in Section 3.3.2.

4. Add the patches to a common pool.

5. Repeat Steps 1–4 until the pool reaches the predefined size.

3.3.1. Raster Stack Generation

3.3.1.1. 3DBAG Tile Downloading and Parsing

The first stage in the raster stack generation process concerns the acquisition and parsing of the underlying 3DBAG tile to a format which is appropriate for further processing. Given a valid tile ID, the corresponding data is initially downloaded in the CityJSON[4] data format. Once this operation has been completed, the LoD1.1, 1.3, and 2.2 roof surface representations are extracted, projected onto a 2D plane, and ultimately saved to disk as individual GeoPackage files. In addition, these files contain the median roof elevation of the roof corresponding to each segment (`b3_h_dak_50p`). The use of this attribute is presented in Section 3.3.1.2.

Although map generalisation (Section 4.4.3) is the main function of the LoD1 footprints, it is only auxiliary in the case of the LoD2.2 surfaces which are primarily used to delineate roof boundaries in later stages of the proposed workflow. This is in part because tiles are not guaranteed to be the minimum bounding rectangles (MBRs) of the models they contain, and are hence unreliable geometry indicators.
In addition, it should be noted that roof polygons may have interior boundaries representing open-air spaces (e.g., atria, courtyards, parking spaces, etc.) which naturally do not contain contextually relevant information, and should therefore be discarded at a suitable time (Section 3.3.1.3). However, it was empirically found that the LoD1 footprints oftentimes did not include these openings. In actuality, this was the also case for all LoDs, but higher-order ones appeared to contain less errors of this type, and were thus preferred over their less detailed counterparts.

3.3.1.2. 3DBAG Tile Asset Downloading and Parsing

The second stage in the raster stack generation process concerns the acquisition and parsing of the underlying and BM5 and AHN4 tiles to a format which is appropriate for further processing.

[4] At the time of writing (3DBAG v2024.02.28), CityJSON was the only available data format which achieved complete tile coverage, and was hence preferred over GeoPackage. Wavefront OBJ was not considered because it does not support attribute and semantic information.

Tile Asset Downloading Given the LoD2.2 roof surface footprints extracted in Section 3.3.1.1, the appropriate subset of each said dataset is first identified by computing the spatial intersection of each footprint with the corresponding sheet index and discarding duplicate IDs. The surfaces rather than their MBR are used directly to avoid processing contextually irrelevant tiles (Figure 3.7).

Figure 3.7.: Example roof surface footprints and the corresponding BM5 data. If the footprint MBR had been used to download this data, the bottom left area would have been processed although it is clearly contextually irrelevant.

The resulting tiles are henceforth collectively referred to as *tile assets*, signifying their functional and semantic dependence on related 3DBAG tiles. Once this operation has been completed, the assets are downloaded in a concurrent fashion and ultimately saved to disk.

BM5 Tile Parsing Given the BM5 component of the tile assets downloaded in Section 3.3.1.2, the corresponding parsing step concerns the merging of the relevant tiles into a single image. This process is performed sequentially, by initially reading the spatial boundary of each file from its header and computing its intersection with the MBR of the LoD2.2 surface footprints extracted in Section 3.3.1.1. The output rectangle is used to create a view into the related data section and copy only the most contextually relevant segment into a common data buffer. This approach exploits the 256-bit block size of BM5 tiles and allows for resource-efficient input/output (I/O) operations. In order to resolve any potential overlap amongst the tiles, the global buffer is populated in a reverse painter's fashion (i.e., the buffer value at any given location is determined by the first valid input at that position). Once the buffer has been filled, its spatial bounds are adjusted such that the global coordinates of the related pixel

centres are integer multiples of the input GSD. This means that the resulting image is guaranteed to envelope the footprints but its bounds may be up to a pixel wider in any direction. Hence its boundary is used in later stages of the proposed workflow instead. This adjustment is performed because it was observed to result in pixel-wise alignment of the image with its constituents, presumable because they were themselves mosaicked in this way. Finally, the image is saved to disk as a GeoTIFF file.

AHN4 Tile Parsing Given the AHN4 component of the tile assets downloaded in Section 3.3.1.2, the corresponding parsing step concerns the merging of the relevant tiles into a single point cloud, the subsequent rasterisation of its planar point density, reflectance, and elevation fields, and ultimately the further processing of the latter to generate the corresponding slope field and the henceforth referred to as *nDRM*.

Tile Merging Given the MBR of the image generated in Section 3.3.1.2, the first file listed in the asset manifest is loaded into memory and cropped to the corresponding bounds. This point cloud is henceforth referred to as the *reference point cloud*[5]. Cropping is performed directly on the point records by transforming to the input MBR to the relevant coordinate system. This approach is used because world point coordinates are actually stored in LAS files and must hence be computed on demand at a potentially significant computational cost, depending on the total number of points they contain. Thereafter, each remaining file is also read and cropped in a sequential fashion, and the scale and offset vectors listed in its header are modified to match those of the reference cloud. This adjustment is performed because in order to merge any two LAS files, their point records must be concatenated under a common header. This header may be arbitrarily chosen from the inputs to this process since it only affects which one the output mosaic will be constructed around. However, its naive application to foreign datasets invalidates the original association between their point record and the related world coordinates. For instance, in the case of tiled point clouds, whose offsets are commonly incremented according to their bounds and point records represent local coordinates within each tile, this would result in the inputs being placed exactly "on top" of each other. Thus, it is necessary to embed the input scales and offsets directly into the corresponding point record fields as a means of preserving them before they are overwritten by the reference header. The relationship between the original and updated point record, $\mathcal{R}$ and $mathcal R'$, respectively, is given by Eq. (3.1):

$$\mathcal{R}' = \frac{(\mathcal{R} \times \mathbf{s} + \mathbf{o}) - \mathbf{o}'}{\mathbf{s}'} \tag{3.1}$$

where $\mathbf{s}, \mathbf{o}$ and $\mathbf{s}', \mathbf{o}'$ are the corresponding scale and offset vectors, respectively, while the bracketed quantity in the numerator represents the original world coordinates.

Once the merging process has been completed, any duplicate points are identified and removed. This step is particularly important in the case of the AHN tiles provided

[5]The reference point cloud is not to be confused with the reference dataset.

by GeoTiles due to the 20 m overlap between them. In addition, it is an essential prerequisite of accurate point density calculations. Similarly to cropping, it is performed using point records instead of the related world coordinates, mostly for the same reason. However, an additional benefit of using this approach in this case is that the arithmetic operations required to identify duplicate points are exact since they are performed on integers. Therefore, only collections of points sharing exactly the same coordinates are affected, achieving lossless point cloud decimation. First, the merged point cloud is sorted in ascending Z order, and each of its points is in turn uniquely hashed according to the corresponding X and Y fields. Subsequently, points with the same hash to be discarded in an iterative fashion, keeping the last encountered record of each duplicate point group. This has the effect of preserving the locally highest points, which are more likely to be contextually relevant. Then, the original point order is restored so as to not disturb the spatial coherence of the point cloud, which is ultimately saved to disk as a LASZip file.

Rasterisation Given the merged point cloud produced in Section 3.3.1.2, the underlying planar point density, reflectance, and elevation fields are rasterised. The MBR of the output datasets is that of the image generated in Section 3.3.1.2. The bounds of the point cloud are not used because they may be smaller due to the absence of points along at at least one edge of the corresponding boundary. In contrast the GSD of the density raster is 1 m, by convention, while that of the reflectance and elevation rasters is three times that of the related image (i.e., $3 \times 8\text{cm} = 24\text{cm}$). This is due to the technical specifications of AHN4. In particular, the minimum average planar point density of the dataset is 10 points per square meter (pts./m^2) which corresponds to a GSD of approximately $\sqrt{10}^{-1} \approx 31.6\text{cm}$ assuming a perfectly uniform point distribution. Because of the spatial interpolation algorithm used in this step (see below), this would have possibly been a more sensible resolution in order to minimise discontinuities in the surfaces modelled by the rasters. However, statistical analysis of relevant 3DBAG attributes at the time of writing (v2024.02.28) showed that the mean density of the AHN4 points classified as building or ground was approximately 25.9 pts/m^2 at the locations of BAG polygons (Figure 3.8), corresponding to a GSD of ca. 19.6 cm.

The cells of the resulting rasters are populated by interpolating the corresponding field at their centre using inverse distance weighting (IDW) with a Chebysev radius of 12 cm and a power of two. Although the choice of power is relatively standard albeit admittedly arbitrary in the context of this thesis, that of the L^∞ norm is not since it guarantees that the value of each pixel is influenced by exactly the points whose 2D projection intersects it. This means that points lying along the interface of adjacent cells contribute equally to the value of both. As is the case with duplicate point removal, the distance metric used to define the radius is a crucial aspect of obtaining an accurate point density raster. In addition, nearest-neighbour queries are backed by a 2D binary tree containing world point coordinates for simplicity. However, it should be noted that IDW is scale-invariant, and so point records could have been used instead for increased numerical precision. Once the initial rasterisation has been

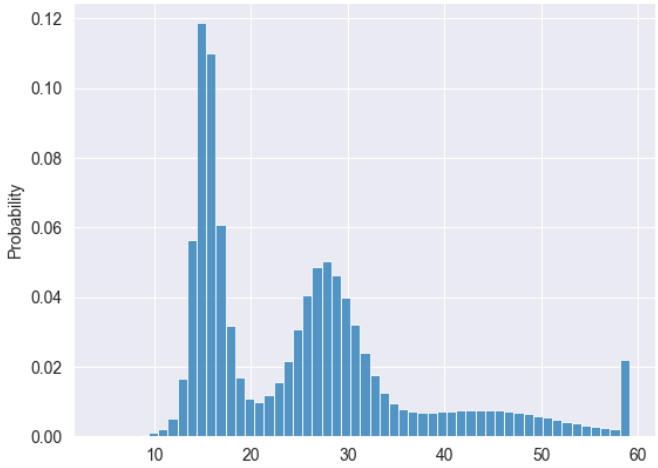


Figure 3.8.: Frequency distribution of the planar point density of the AHN4 points which are classified as building or ground and their 2D projection intersects a BAG polygon. The data is taken from the `b3_puntdichtheid_ahn4` attribute of the 3DBAG (v2024.02.28) for models which were reconstructed using said dataset, as indicated by the corresponding `b3_pw_bron` attribute.

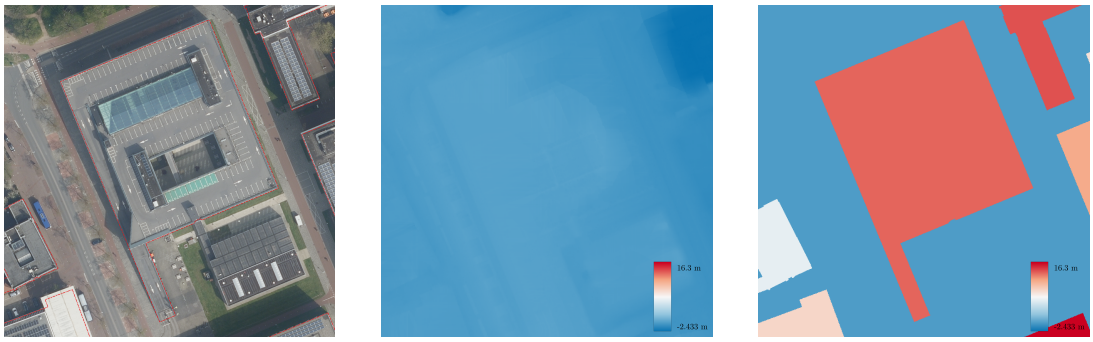
completed, any no-data cells are identified and filled in an iterative fashion using the same interpolation algorithm, but with a radius of 100 cells. Since any real number is technically a valid reflectance or elevation value, the no-data cells of these rasters are in actually populated with not-a-number (NaN). Hence, it is important to validate these cells because arithmetic operations involving NaN also return NaN, by definition, resulting in erroneous behaviour during model training (Section 4.3). Nevertheless, this step does not apply to the density raster because a density is defined as the number of points per square meter. This quantity is clearly a natural number, and hence the corresponding raster is is constructed using a unsigned 16-bit integer data buffer with a no-data value of $2^{16} = 65536$.

Once completely populated, the values of the reflectance raster are converted from decibels (i.e., a logarithmic scale) to the underlying optical power ratio (i.e., a linear scale) to enable it be correctly upsampled to a GSD of 8 cm using bilinear interpolation during the raster concatenation stage of the proposed workflow (Section 3.3.1.3). In addition, resulting values corresponding to non-Lambertian reflectors (i.e., ≥ 1) are discarded by clipping the whole raster to the interval $[0, 1]$. This correction is performed because although certain materials of interest (e.g., glass, metal, etc.) can behave as specular reflectors under certain lightning or viewing conditions, their exceedingly high signal can overpower that of neighbouring pixels in the context of convolution and pooling operations. Furthermore, interpolation to or from these values may result in erroneous intermediates being introduced. For instance, interpolating pixel values corresponding to a vegetated area situated between two bodies of water

would result in them being assigned potentially exceedingly high diffuse reflectance even though this is highly unlikely to be the case in reality.

Finally, the density and reflectance rasters are saved to disk as individual GeoTIFF files, while the elevation raster remains in memory to be used in Sections 3.3.1.2 and 3.3.1.2.

Normalised Digital Roof Model Generation The *digital roof model* (DRM) is defined as the median roof elevation in areas where buildings are present and zero elsewhere (Figure 3.9).



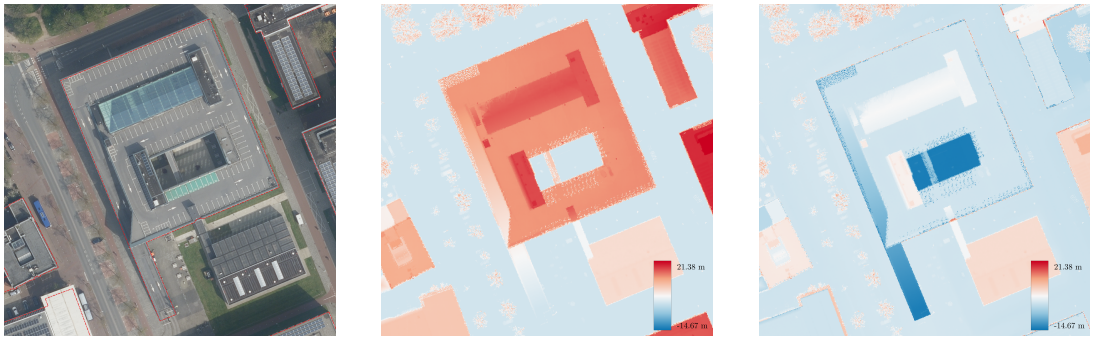
(a) DTM. (b) DRM.

Figure 3.9.: Example scene and corresponding DTM and DRM. In the absence of valid data at building locations in the official product, the DTM presented here has been constructed by rasterising the relevant ground points using the procedure described in Section 3.3.1.2.

The DRM is computed by dissolving the LoD2.2 roof surface footprints extracted in Section 3.3.1.1 by building ID in order to represent each building with a single polygon, in turn minimising the computational cost of the following operations, and in turn rasterising the relevant attribute of the resulting geometry to the MBR of the corresponding DSM. The cells whose centre is in the interior the geometry are populated with the corresponding median roof elevation, while the rest are filled with zero.

The *normalised digital roof model* (nDRM) is a quasi-normalised DSM akin to the more commonly employed nDSM. However, instead of using the DTM a reference elevation surface, the nDRM is computed with respect to the DRM. The benefit of using a nDRM instead of a nDSM is that the former is designed to represent elevation changes relative to median roof height, and therefore is signed, with negative and positive heights being below and above the reference, respectively. Therefore, the nDRM complements slope by effectively giving it a direction.

Once the DRM has been constructed, it is subtracted from an in-memory copy of the DSM and the output raster is saved to disk as a GeoTIFF file.



(a) nDSM. (b) nDRM.

Figure 3.10.: Example scene and corresponding nDSM and nDRM.

Slope Field Generation The slope, s, is defined as the first derivative of the DTM and is computed as the inverse tangent of its gradient (Eq. (3.2)).

$$s = \tan^{-1} \sqrt{\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2} \tag{3.2}$$

where $\partial z/\partial x, \partial z/\partial y$ are the partial gradients of the DTM along its longitudinal and latitudinal, correspondingly. These quantities are obtained using second-order accurate central differences, except along the raster edges, where first-order backward and forwards differences are respectively employed for the bottom and right, and top and left sides. This approach is better known in geospatial literature and software as the Zevenbergen-Thorne method (Zevenbergen & Thorne, 1987). Finally, the resulting raster values are converted from radians to degrees for easier human understanding and saved to disk along with the DTM as individual GeoTIFF files.

3.3.1.3. Raster Concatenation

Given the LoD2.2 roof surface footprints extracted in Section 3.3.1.1 and this rasters generated in Section 3.3.1.2, the third and final stage in the raster stack generation process concerns their concatenation into a single multi-band dataset. In particular, this file contains the red, green, and blue bands of the parsed BM5 tiles, followed by reflectance, slope, nDRM, and planar point density rasters extracted from the corresponding AHN4 subset (Table 3.1).

The channel order is mostly arbitrary as each one is convoluted separately from the rest when passed through the model. The only specific design decision in this context was that the BM5- and AHN4-derived bands would not interleaved to facilitate easier handling of the reference dataset. The concatenation process is performed sequentially by loading each raster into memory and copying its contents into a common data buffer of the same pixel dimensions as those of the parsed aerial imagery. This means that the rasterised datasets must first be upsampled the GSD of the orthophotos in order to correctly populate the buffer. This step is performed using bilinear interpolation

Table 3.1.: Band composition of the raster stacks comprising the reference dataset.

Name	Original Dataset	Stack Order
Red	BM5	1
Green	BM5	2
Blue	BM5	3
Reflectance	AHN4	4
Slope	AHN4	5
nDRM	AHN4 & 3DBAG	6
Density	AHN4	7

because it preserves the original value range, which is not only desirable but necessary in the case of reflectance, as explained in Section 3.3.1.2. Subsequently, the nDRM and density bands are clipped between their respective second and ninety-ninth percentiles. This post-processing step is applied specifically to these bands because it was observed that they both tend to contain erroneous values distorting the underlying distribution. In particular, the nDRM may contain extreme values due to temporal or geometric misalignment of AHN4 and BAG resulting in building segments not appearing in both datasets.
On the other hand, point density along building edges is artificially inflated and generally much greater than that of its surrounding cells due to the planar projection of points corresponding to wall surfaces.
No other bands are post-processed due to lack of insufficient theoretical indication that is it necessary or helpful.

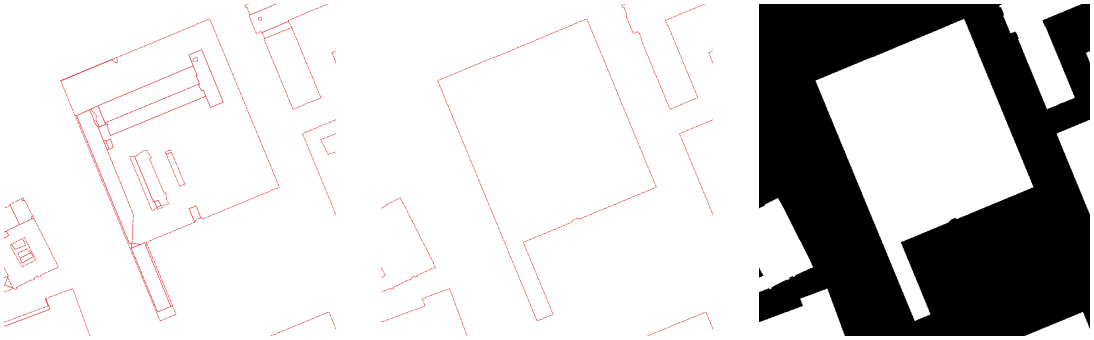
Thereafter, the footprints are used to invalidate or *mask* the henceforth referred tp as *background* cells of the stack. These are defined as those pixels whose centre are on the surface geometry. Because these regions are not only contextually irrelevant, but also similar to roofs in certain bands (), and therefore bound to cause model confusion during training and inference due to their equal contribution as roof pixels in convolution and pooling operations, they are mapped to zero.
The masked stack is computed in a similar fashion to the DRM, with the only exception that valid pixels are filled with ones to produce a binary roof map (Figure 3.11) which is in turn multiplied with the stack.

Finally, the stack is saved to disk as a GeoTIFF file.

3.3.2. Raster Stack Splitting

In order to facilitate granular handling of the reference dataset and ensure that it contains as much contextually relevant information as possible, raster stacks are split into 512×512 pixel non-overlapping chips for further processing.

Although this approach is generally beneficial in both model training (Section 4.3) and inference because it distributes the total computational resources required to completely process a stack over several batches, it is not strictly necessary at this



(a) LoD2.2 roof surface footprints.

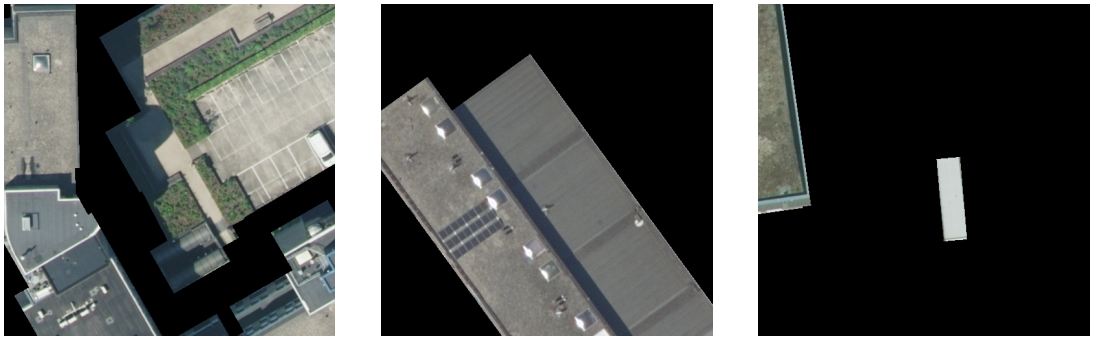
(b) Dissolved footprints by building ID.

(c) The corresponding binary roof map. Black represents 0, while white 1.

Figure 3.11.: Binary building map generation process.

stage of the proposed workflow. For instance, stacks can be tiled synchronously as a quasi-augmentation step. In fact, this is how inference is performed. However, this complicates training in cases where the background is ignored (i.e., predictions at the corresponding locations do not contribute to loss calculations and backpropagation), because invalid patches (i.e., chips which contain only background cells) represent null targets averaging over which leads to undefined behaviour due to division by zero. While, this issue can be mitigated by summing instead of averaging individual losses over samples in a given batch or not performing any gradient updates whenever such an anomaly is detected, neither approach is optimal. Since a non-negative loss component is generally computed for each item in a batch, reducing the loss by summation would effectively make the result proportional to the batch size and by extension the learning rate, resulting in a situation where a smaller batch size or learning rate would result in artificial inflated model performance in terms of validation loss by default. In addition, passing invalid samples through the model only to effectively discard them once they have been processed is clearly a waste of computational resources and time.

Furthermore, stack splitting offers two major advantages in data annotation(Section 3.4) and quality control. First, it was empirically found that patch labelling, especially when chips were shuffled, resulted in reduced cognitive fatigue due to frequently changing visual signals, in turn increasing productivity. In addition, it should be noted that certain human-in-the-loop annotation tools based on Kirillov et al., 2023 may face difficulty with capturing fine-details in images whose resolution or size deviates significantly from the one they were originally trained on (Lynn, 2023). Finally, the aforementioned issue with invalid patches can now be easily solved by simply identifying and discarding them from the reference dataset altogether. By extension, a maximum background percentage limit may be imposed on each patch in order to maximize its contextual added value (Figure 3.12).
In the context of this thesis, this limit is set to 80%.



(a) 30% ± 5% background content.

(b) 60% ± 5% background content.

(c) 90% ± 5% background content.

Figure 3.12.: Example stack chips with varying background contents.

3.4. Reference Dataset Annotation

3.4.1. Material Classes

3.4.1.1. Membrane

This class encompasses building materials which are used in the top or visible layer of membrane roofs as well as a means of damp proofing in built-up (e.g., "asphalt-and-gravel" or "tar-and-gravel"; Section 3.4.1.3 and Figure 3.15) roof systems. These roof types are commonly constructed with no to low pitch (i.e., $\leq 5 - -6°$) in commercial and industrial buildings as well as buildings with relatively large bays. In addition, membranes are generally used as vapour barriers or retardants, depending on the particular use requirements, in most — if not all — warm and cold roof systems regardless of their geometry. This also includes special types, such as green roofs (Section 3.4.1.8).

Contemporary roofing membranes typically refer to composite materials based on asphalt, various potentially vulcanised elastomers (e.g., chlorosulfonated polyethylene, ethylene propylene diene monomer, neoprene, etc.) and thermoplastics (e.g., ketone ethylene ester, polyvinyl chloride, thermoplastic polyolefins, etc.). Furthermore, tar was widely used up to the mid-20$^{\text{th}}$ century, but was eventually replaced with bitumen as crude oil production increased. Depending on the material, roof system and use requirements, they are applied in multiple layers in either liquid or semi-liquid form, or as individual sheets which are applied and secured using nails or similar fasteners, glue or heat. The surface of membrane roofs constructed using the former type appears relatively uniform, as if painted. On the other hand, parallel seams due to the necessary overlap between adjacent elements is a characteristic indication of membrane sheets.

Dark-coloured Membrane This subclass includes dark-coloured (e.g., black, off-black, dark-grey, etc.) coatings (Figure 3.13).
These materials are especially common due to the relatively lower cost of their primary

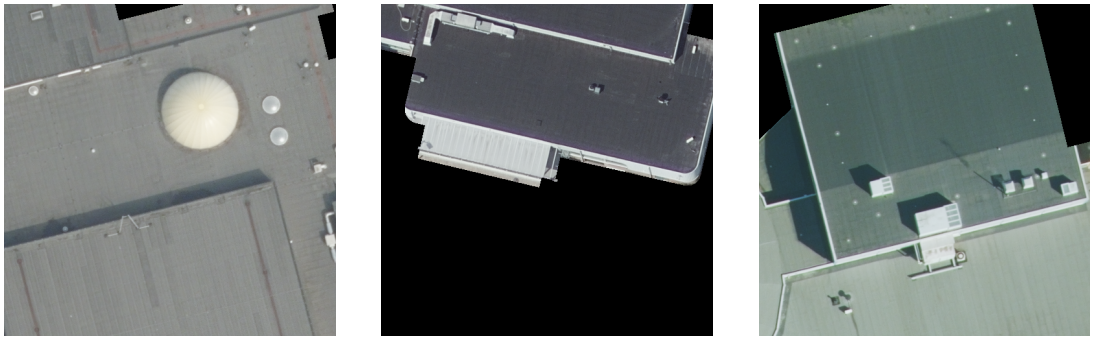


Figure 3.13.: Typical examples of dark-coloured membrane roofs.

components — which are naturally dark-coloured — in comparison to their light-coloured counterparts (Section 3.4.1.1). They are generally encountered in the base and intermediate layers of most roofing systems with the exception of membrane roofs, in cases the inherently increased thermal mass of these coatings is desirable or otherwise unimportant, where they form the top construction layer.

Light-coloured Membrane This subclass includes light-coloured (e.g., white, off-white, light-grey, etc.) coatings (Figure 3.14).

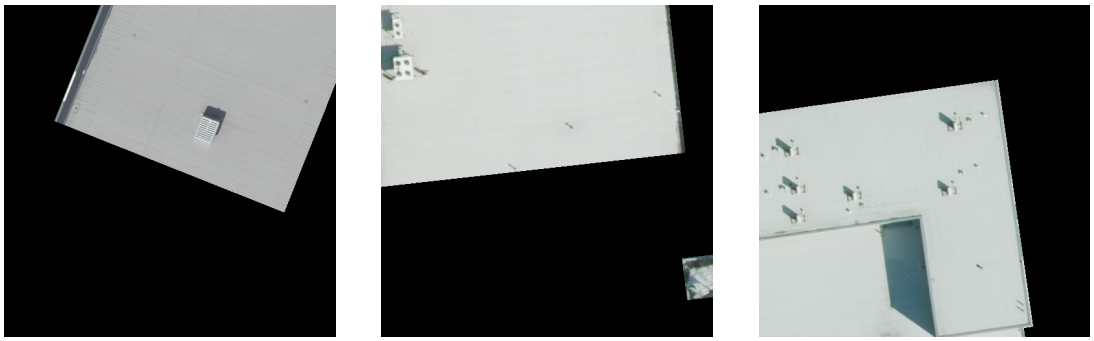


Figure 3.14.: Typical examples of light-coloured membrane roofs.

In contrast to their dark-coloured counterparts (Section 3.4.1.1), these materials are commonly based on acrylics or silicones, hence their brighter appearance. They are generally less common due to their relatively increased manufacturing cost, but are gradually growing in popularity in the construction of cool roofs due to their inherently decreased thermal mass. However, this is only relevant in membrane roofs where coatings are the top construction layer.

At this point it should be noted that the distinction between a so-called "dark-" and a "light'-coloured" material ought in fact to be based on its brightness and not hue, saturation, or any other colorimetric property. However, because this quantity is not explicitly modelled in true-colour images and may also be heavily influenced by the particular lighting conditions and scene geometry, material classification in the case of coatings is performed according to the expert opinion of the annotator. Therefore, there is a certain degree of semantic ambiguity — and consequently potential error — when dealing with under- and over-exposed surfaces.

3.4.1.2. Concrete

This class encompasses cement mortar, fibre cement, and reinforced concrete (RC).

The first material is commonly used instead of membrane roofs in the case of accessibility requirements due the sensitivity of the former system to live loads. The mortar is applied in a tiled fashion with expansion joints around individual elements.

On the other hand, fibre cement is used to tile pitched roofs with at least 5–6°pitch, especially those under 15°. Hence, it is generally available as flat tiles or corrugated sheets. In particular, the latter type is typically used in industrial buildings, due to its relatively low weight and high durability with minimal maintenance requirements. It is commonly encountered in its natural colour, often with visible moss and fungi on its porous surface, although several pigment options are possible. At this point, it should be noted that fibre cement used to be widely reinforced with chrysotile or white asbestos, which has been banned in the EU since 2005 (European Commission, 1999), due to the its detrimental health effects, including extensive lung scarring and cancer. However, its fibres have been shown to continue to be released as building materials containing it deteriorate with no significant chemical or structural changes in their composition (Burdett, 2006). Hence, it could be of particular interest to policy- and decision-makers to map the spatial distribution of asbestos-containing fibre cement roofing in order to facilitate the large-scale replacement.

Finally, RC is generally used as covering in flat and low-slope roofs in garage buildings as well as buildings with relatively large bays. In addition, it may be used to tile pitched roofs with at least 15°pitch, particularly those between 60–90°. In the first case, it is commonly applied as a slab with intermittent expansion joints, depending on the particular use requirements, while in the latter, individual, typically prefabricated panels are the norm.

3.4.1.3. Gravel

This class encompasses loose gravel which is commonly used as to ballast built-up roof systems and green roofs, as well as a means of improved percolation in the latter roofing type.

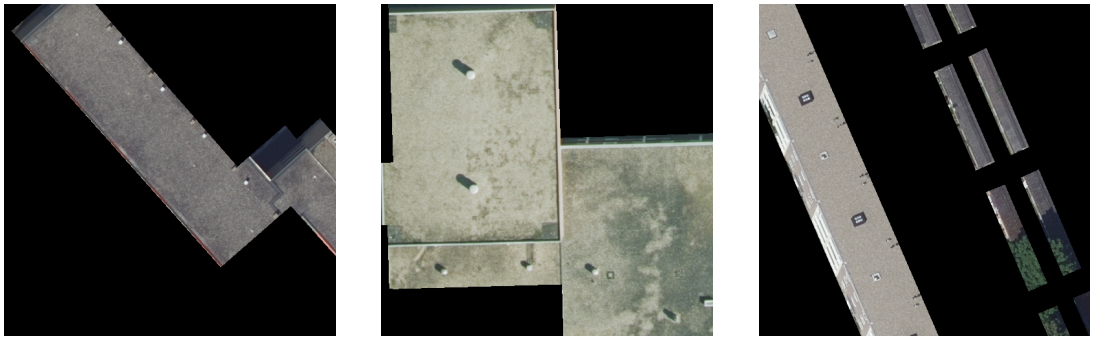


Figure 3.15.: Typical examples of built-up roofs with gravel ballast. The middle and right figures also show a semi-intensive green roof on the middle-left and tiles on the centre–middle-right, respectively.

3.4.1.4. Light-permitting Surface

This class encompasses translucent and transparent surfaces, such as skylights and greenhouse roofs, which are commonly constructed using glass or plastic panels (Figure 3.16). Nevertheless, surrounding elements, such as window heads, rails, sashes, and grilles, are also semantically assigned to this category.

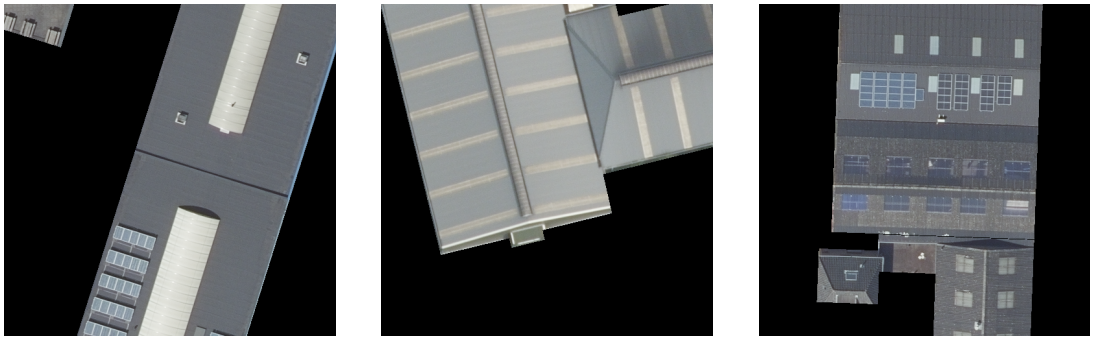


Figure 3.16.: Typical examples of light-permitting surfaces on a membrane (a), metal (b), and tile roof (c).

3.4.1.5. Metal

This class encompasses metal tiles and roof systems (Figure 3.17). These roof types are commonly constructed with low to medium pitch (i.e., ca. 2-5–30-32.5°) in commercial and industrial buildings, especially in cases when air- or water-tightness is not a use requirement. On the other hand, metal tiles are used in pitched roofs of all slopes, particularly those under 15°, and they are the only relevant non-membranous alternative in the case of low-slope roofs. Contemporary roofing metal generally refers

to aluminium, copper, lead, steel or zinc panels, sheets, and tiles which may be galvanised, where applicable, or painted. The typical appearance of metal panels and sheets includes parallel striations perpendicular to roof ridges to facilitate run-off as well as similarly oriented seams or uniformly scattered bolts or similar fasteners. In addition, the typical patina of copper is a characteristic indication of metal.

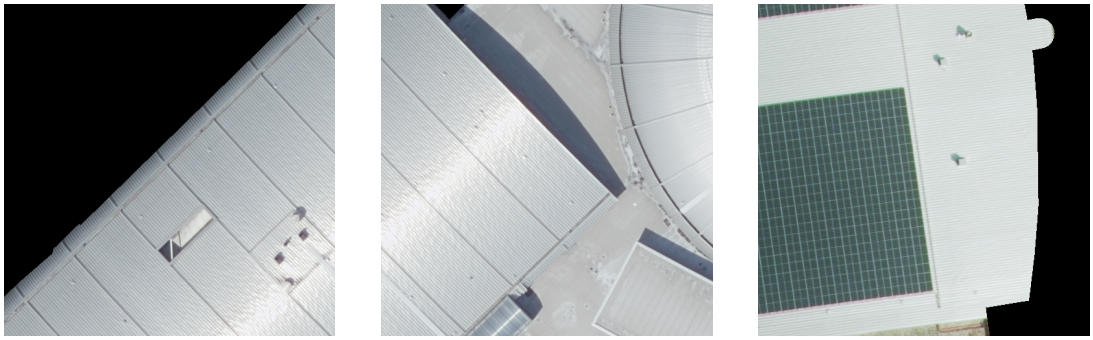


Figure 3.17.: Typical examples of metal roofs.

3.4.1.6. Solar Panel

This class encompasses photovoltaic (PV) and solar panels and their frames (Figure 3.18).

PV panels are used to convert solar radiation directly into electricity to be either consumed immediately or stored for later use. They are generally composed of individual modules of flat, dark-coloured (e.g., black, blue, navy, etc.) cells containing parallel, grey bus lines lines which may or may not appear in groups along at least one of the horizontal and vertical directions. The cells can be either rigid or flexible, in which case they are called thin-film. Depending on the cell type and use case, the panels may be installed directly on rooftops or first mounted on metal frames. The frames may be in their natural colour or painted to resemble that of the corresponding panels. In addition, frames enable various array formations, from one where the panels lay flat, to one where they are all tilted towards a particular direction, to one where their tilt alternates in a zigzag pattern (Section 3.4.1.6).

On the other hand, solar panels convert radiation to thermal energy as means of heating a medium, typically water. They mostly share the same characteristics with PV panels with the exception that they are often in close proximity to or even paired with an external storage tank holding said medium. In addition, arrays of this type may have a corrugated appearance due to the use of evacuated tube instead of flat panels.

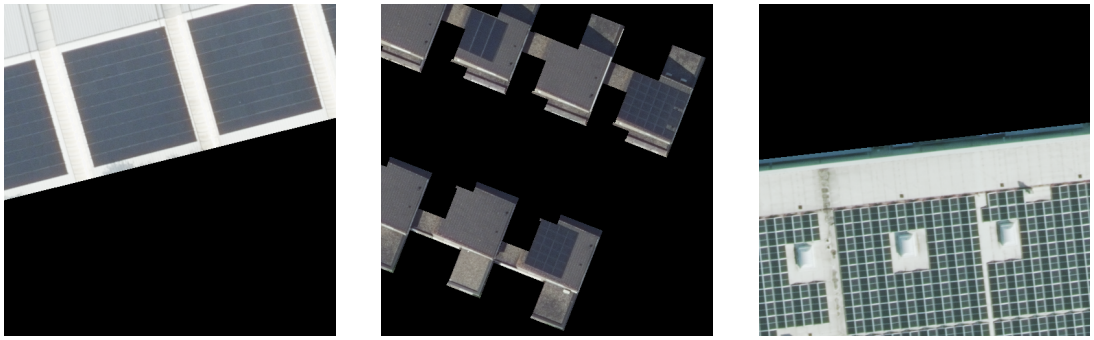


Figure 3.18.: Typical examples of solar panel arrays on a metal (a), tile (b), and membrane roof (c).

3.4.1.7. Tile

This class encompasses tile roof systems. These roof types are commonly constructed with at least medium pitch (15-16+) to facilitate run-off, which is otherwise hindered due to the gap between adjacent tiles.

Contemporary tiles generally refer to fired clay, cinder block or fibre cement, both of which are oftentimes mislabelled as made of "concrete", metal, plastic, rubber, stone, various synthetic materials and wood. In addition, various special types, such as solar shingles, are commercially available. Furthermore, tiles are typically divided into three main categories according to their shape: flat, curved and interlocking. However, due to their inter-temporal worldwide popularity since at least age of Ancient Egypt, multiple combinations and local variations of these types exist. Moreover, it is common for certain styles to be preferred in certain regions, depending on their architecture.

This, in combination with contemporary fabrication and manufacturing methods, minimise and in certain cases even eliminate previously telltale differences between different tile materials, such as their colour and shape, in turn making it neigh impossible to cover all of them accurately and reliably. Hence, in the context of this thesis, only asphalt and stone shingles, as well as ceramic tiles considered, as they were deemed to be both the most typical examples of this class and the easiest to identify in the reference dataset. At this point it should be noted that, "concrete" and metal tiles continue to be semantically assigned to the titular classes, and solar tiles to solar panels, although their identification is significantly harder, as will be later explained.

Asphalt Shingle This subclass encompasses flat tiles or shingles which commonly refer to tiles based on various organic materials (e.g., compacted cellulose, wood, paper, etc.) or fibreglass. The base material is generally infused with asphalt or similar material as a means of waterproofing, and ballasted with ceramic, stone, or synthetic granules, which impart the final product its colour and characteristic appearance.

Stone Shingle This class encompasses natural (e.g., limestone, sandstone, slate, etc.) and synthetic stone shingles. This tile type commonly used in roofs with at least 25°pitch, especially those under 30°and between 40—50°in the case of limestone and slate, respectively. Their appearance depends on the type, origin, and fabrication method of the particular stone or product meant to resemble it. However, the naturally occurring irregularities in their shape, especially along their edges, and colour are generally sufficient for their identification.

Ceramic Tile This class encompasses fired clay tiles. As explained in Section 3.4.1.7, modern tiles are commercially available in a multitude of variations which are not specific to any particular colour or shape. This is especially true for this tile type due to the high workability of damp clay, which allows it to be sculpted or molded, as is the case with "concrete'' and metal tiles. Hnece, it is impossible to truly circumscribe this tile type by shape, as was the case with asphalt and stone shingles. This means that apart from certain characteristic differences in surface texture (e.g., the existence of moss or fungi, roughness, shine, thickness etc.), the impact of most of which is minimal and even absent depending on post-fabrication treatment (e.g., cleaning, painting etc.), the most reliable indicator of clay tiles is their color. This is generally a shade of brown-orange-red, altough it should be noted that other color option are commercially available (e.g., anthracite, granite, etc.).

3.4.1.8. Vegetation

This class encompasses green roofs. These roofing types are commonly encountered in flat roofs and low-slope residential and commercial roofs but can also be constructed with medium-low to medium-high pitch (i.e., ca. 18–34°), as is traditionally the case in Scandinavia. In either case, their construction most relevantly includes damp- or water-proofing membranes (Section 3.4.1.1), depending on their slope, followed by a root barrier to prevent damage to the membranes, a drainage layer which may contain loose rubble or gravel (Section 3.4.1.3) for improved percolation, and ultimately the growing media and vegetation.

Depending on the substrate depth and plant cover, green roofs are distinguished in three main categories: extensive, semi-intensive, and intensive.

Extensive roofs are typically not meant for public access and are almost completely covered with grass, moss and various low vegetation such as herbs as a means of storm-water mitigation. On the other hand, intensive green roofs are commonly known as "roof gardens" and are meant to replicate a natural landscape shrubs, native forbs and grasses, larger perennials, tropical, non-native vegetation. Finally, semi-intensive green roofs form a middle ground between extensive and intensive offering both rainwater mitigation and biodiversity. They are usually semi-public with small shrubs, forbs, and grasses.

3.4.1.9. Other

This class encompasses any material type not explicitly represented by a corresponding class (e.g., plastic, rubber, thatch, wood, etc.).

3.4.2. Annotation Procedure

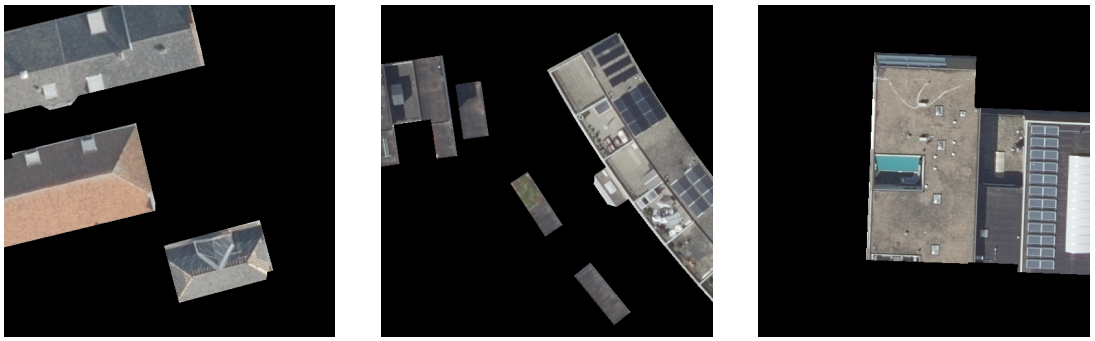
The purpose of annotating the reference dataset is to assign a so-called ground truth to each chip the sampled raster stacks have been split into (Section 3.3.2). This means that ideally each pixel of each chip should be associated with integer labels the background or a particular class, as presented in Section 3.4.1. These associations are then compiled into a patch-specific single-band image of the same spatial dimensions and GSD as the corresponding chip. These rasters are henceforth referred to as the ground truth *segmentation masks* and generally represent the expected output of a perfectly performing model. Hence, these masks are used in the training, validation and test phases to quantify the correctness of model predictions and in turn guide, where applicable, and monitor its performance during training, validation and testing.

As previously explained, image annotation for semantic segmentation typically involves assigning a single numerical label to each pixel. This is commonly performed by digitally overlaying a blank canvas over the image and colouring it using a predefined colour-label association (e.g., blue corresponds to one which corresponds to asphalt shingles, etc.). The canvas may have variable opacity such that the image underneath is visible, and therefore it can be thought of as an equivalent of tracing paper. This process is performed in this way such that images and the corresponding ground masks are aligned at a pixel level.

The canvas may be coloured using various approaches. For instance, various image annotation tools offer manual brush tools, while others allow the user to draw polygons representing the loci of pixels with the same colour, and thus label. In addition, there exist various interactive methods such as the generalisation of automatically generated image over-segmentations and prompt-based segmentation using only positive, positive and negative point samples, as well as polygons. Furthermore, there exist completely automatic segmentation tools which generally use pretrained foundation models which can optionally be fine-tuned on the reference dataset and continuously updated using active learning.

In the context of this thesis, a model-assisted polygon annotation method is employed to label the reference dataset. This facilitates the exploitation of the inherently polygonal nature of most rooftops and their elements because instead of needing to completely label areas of uniform material, their boundaries could simply be "traced'' by marking their vertices to define the endpoints of relevant linear segments to be joined. In particular, the true-colour component of each chip is labelled in this fashion using a layered canvas (Cordts et al., 2016; Yang et al., 2012). This means that the canvas has a so-called z-index such that each polygon is drawn at a separate height, with older polygons being placed lower than newer ones by default. The primary benefit of this technique is that polygons are completely independent from each other,

and so may overlap until flattened and rasterised. Consequently, polygons may be edited individually along their corresponding plane and even moved along the z-index without immediately affecting their neighbours. This is contrast to drawing at a constant height where any overlap must be resolved at once, meaning that changes to a particular polygon may interfere and even corrupt others, making annotation errors more complicated and time-consuming to mitigate. This observation is especially relevant in this case, because annotation masks may include a special *invalid* class which is used to mark non-roof objects or objects of unknown material in the interior of otherwise inside valid polygons (Figure 3.19).



(a) Partially constructed asphalt shingle or membrane roof. Depending on the granularity of the annotation, this roof may also be left completely unlabelled.

(b) Paved balconies with visible clutter.

(c) Open-air atrium partially obscured by awning.

Figure 3.19.: Examples of non-roof or irrelevant objects in the reference dataset.

Because invalid polygons are essentially used to discard information from the reference dataset (see below), it is crucial that they do not interfere with their neighbours until the mask is finalised for their overall impact to be minimal. All this greatly simplifies the annotation of relatively complex scenes, such as solar panel arrays on membrane-roof dormers extruding from a ceramic tile roof, which only requires 3–4 rectangles to be completely annotated using a layered canvas.

Given a chip in the reference dataset, each visible rooftop or rooftop segment is annotated incrementally using the following approach. First, the henceforth referred to as the *base* surface of the rooftop is found and labelled. This surface does not have a strict definition nor description and may not even be visible in significantly cluttered scenes. Hence it requires experience to identify and is subject to the expert opinion of the annotator. However, observing that building construction generally progresses vertically, with newer components and elements being raised upon older ones, as well as the fact that roofing installation clearly belongs to the final stages of erection, the base surface may be loosely described as the lowest visible rooftop segment such that any surface at a higher elevation (e.g., the top surface of elevator or

stair bulkheads, chimneys, various enclosures, electromechanical equipment, skylights, windows, etc.) represents a later point in construction. In addition, the base surface generally corresponds to a single roofing material, most commonly concrete, gravel, membranes, metal, or tiles. This is because it is clearly illogical to build on top of light-permitting surfaces, solar panels or vegetation. Some examples of base surfaces in chips from the reference dataset are given in Figure 3.20.

(a) Dark-coloured membrane. (b) Metal. (c) Light-coloured membrane.

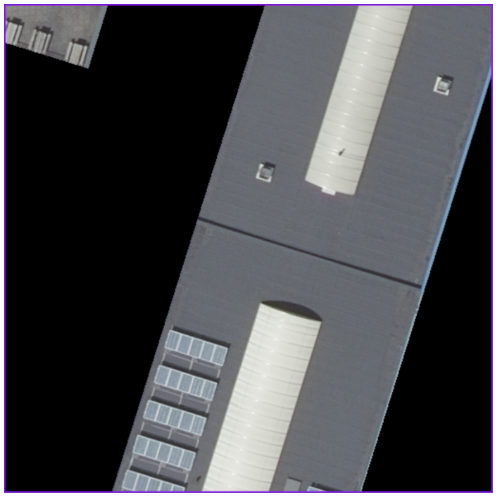
Figure 3.20.: Examples of base surfaces in chips.

Once identified, the base surface is annotated with a single polygon representing said material, as though the rooftop does not contain any other surfaces. Thereafter, the next highest "base'' surface is identified and labelled in the same manner, with this process continuing iteratively until the whole rooftop has been completely annotated. Finally, the invalid class is used to mark any ambiguous, unknown, or irrelevant regions of significant size[6] (e.g., clearly movable objects, chimneys, heating, ventilation, and air conditioning (HVAC) systems, ridge caps, gutters, ridge caps, flashing, vents, parapets, etc.). The annotation process for an example chip is presented in Figure 3.21.

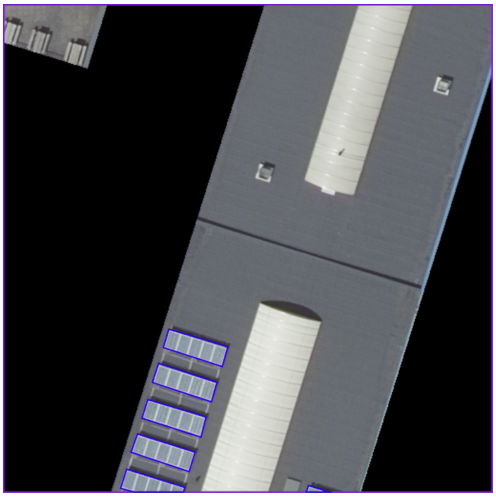
Special cases in this process include built-up roofs where both the vapor barrier and gravel are visible to a significant degree, in which case they are both labelled. The same is true in the case of green roofs regarding waterproofing, gravel and vegetation. In fact, gravel-paved intensive green roofs with scattered vegetation are actually labelled using the gravel class, with individual plants being labelled as vegetation. The labelling order in these cases, as well as any other case where the current base surface is composed of multiple materials is subject to the expert opinion of the annotator.

Furthermore, non-roof regions (e.g., façade segments due to the non true-ortho nature of BM5 (Section 3.4.3.4)) are not labelled at all, and neither are roof segments which are clearly designed for recreational purposes (e.g., balconies, patios, decks, etc.) due to the large amount of clutter, much of which is movable objects, and variation in paving materials. Moreover, in cases where the material of a certain region is

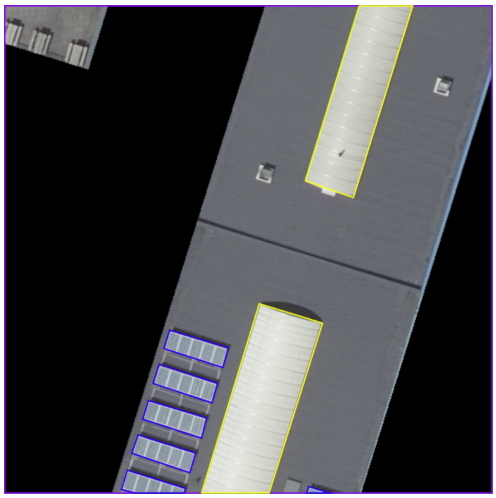
[6]The critical area of an area to be considered significant is subject to the expert opinion of the annotator, but is generally in the 1—10 square meter range, depending on the size of the pictured object.



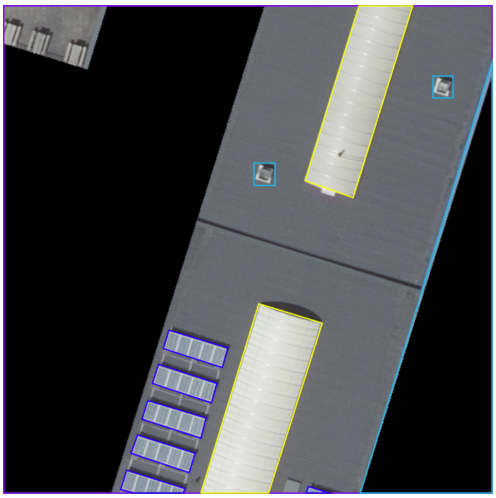
(a) Step 1. The whole image is labelled using a single polygon



(b) Step 2. The solar panels are labelled.



(c) Step 3. The skylights are labelled.



(d) Step 4. Invalid areas are marked.

Figure 3.21.: Annotation process for an example chip. The building in the top left corner of the chip is not annotated for simplicity.

ambiguous in the aerial view, the corresponding street panorama is consulted. If the material is still ambiguous, it is not labelled or marked as invalid, as previously mentioned, whichever is easier.

Once the reference dataset has been annotated, the polygons corresponding to each chip are projected to the same plane and rasterized with any potential overlap is resolved using a painter's algorithm. Thereafter, the background in each chip is remasked using the LoD2.2 roof surface footprints extracted in Section 3.3.1.1 to ensure that no polygon covers includes background cells. In addition, this allows the annotation to be "slopy" along building edges, further accelerating the process. Subsequently,

all invalid pixels are mapped to background and all pixel values greater than the label of the now discarded invalid class (Table 3.2) are decremented by one to preserve continuity.

Table 3.2.: Initial class name-colour-label associations. This table is used only for annotation purposes.

Name	Colour	Label	Name	Colour	Label
Background		0	Light-coloured Membrane		7
Asphalt Shingle		1	Light-permitting Surface		8
Ceramic Tile		2	Metal		9
Concrete		3	Other		10
Dark-coloured Membrane		4	Solar Panel		11
Gravel		5	Stone Shingle		12
Invalid		6	Vegetation		13

Finally, a relevant continuity check is performed across the masks to check for missing classes. In that case, the present classes are all remapped to a continuous range which is saved to disk alongside the masks as a JSON file.

3.4.3. Sources of Error

3.4.3.1. Material Appearance

Although the material classes presented in Section 3.4.1 have been specifically defined to be as visually distinct as possible, the advent of contemporary manufacturing techniques and various ongoing changes in design and construction norms have significantly complicated the delineation of certain classes solely based on appearance. Many materials which were commonly known as having a certain colour, shape, or texture (e.g., ceramic tiles, solar panels, membrane) are now commercially available in a multitude of variations, some of which have been specifically designed to resemble different materials (e.g., synthetic stones). In addition, one must consider the material age and condition; exposure to the elements (e.g., ultraviolet (UV) radiation, rain, snow, etc.) can cause the colour of certain materials to not only fade, but change completely (e.g., copper).

Clearly, there is significant overlap between most semantically similar classes, which is also reflected in the reference data.

3.4.3.2. Lighting Conditions & Scene Geometry

Similarly to Section 3.4.3.1, one additional factor which can affect material appearance are the lighting conditions at the time the corresponding image was captured in combination with the scene geometry. As mentioned in , the BM5 dataset is updated yearly over the course of the autumn period. This means that it is highly unlikely for every flight to be undertaken at exactly the appropriate time and under the right weather and visibility conditions such that the corresponding images are captured with perfect illumination under directly overhead sunlight. Clearly, the reference dataset is bound to contain regions with variable amounts of blur due to suboptimal visibility or shadow due to taller structures casting a shadow over neighbouring surfaces at a lower elevation, both of which naturally hinder material identification.

3.4.3.3. Reflectance Noise & Relativity

As mentioned in Section 3.2.3, the AHN4 point cloud contains a reflectance attribute in the Extra Bytes VLR of the corresponding LASzip files. According to RIEGL Laser Measurement Systems GmbH, 2019, reflectance is defined as the range-normalised difference between the amplitude of a particular target and that of a "white flat target [...], oriented orthonormal to the beam axis, and with a size in excess of the laser footprint". In turn, the amplitude, A_{dB} is defined as:

$$A_{db} = 10 \log_{10} \frac{P_{act}}{P_{min}}$$

where P_{act} and P_{min} are the power of the corresponding backscattered signal and the detection threshold of the particular instrument, respectively.

However, laser pulses are generally partially absorbed or scattered as they propagate through the atmosphere due to the presence of airborne particles. This phenomenon is called atmospheric attenuation and its impact on P_{act} depends on the output power and wavelength of each pulse, the target range, and particle density, which is itself a function of various weather parameters whose combined effect is modelled by the attenuation or extinction coefficient, μ, measured in reciprocal meters and negatively correlated with visibility. In addition, even under perfect atmospheric conditions, it is not safe to assume that all backscattered pulses actually reach the instrument receiver, which is commonly assumed to be collocated with its transmitter. This is because the angle of incidence is not guaranteed to be perfectly right, especially in the case of systems with rotating or oscillating transmitters, as well as due the fact that most real-life targets do not behave as perfectly specular reflectors. Finally, it is obvious that any incident energy absorbed by the target cannot contribute to the return signal. This means that the effective range of the instrument is correlated with both μ and target reflectance, resulting in certain targets being impossible to correctly capture

above a certain operating flying altitude or below a particular visibility threshold at a given laser pulse rate and output power level. Nevertheless, it is not safe to assume that any of these parameters was calibrated with respect to a certain minimum reflectance threshold, as this is not an official technical requirement of the AHN program, nor that they or the corresponding weather conditions remained constant for every flight. For instance, this is clearly stated to not be the case for OFA in the AHN informational documentation.

Since the reflectance field is only calibrated for range, it is clear that measurements may exhibit significant systematic errors. Nevertheless, because intensity data is not range-corrected and otherwise suffers from the same issues, the preference to reflectance instead of intensity, which was used instead by both Hamedianfar, Shafri et al. (2014) and Norman et al. (2020), appears logical.

3.4.3.4. Misalignment between BM5 and AHN4

Geometric Misalignment As explained in Section 3.2.2, BM5 imagery is only corrected for ground deformities using relevant DTM information. This means that although ground-level features are accurately georeferenced and remain fixed throughout different views of the same scene, lens distortion and camera tilt may still cause increasingly higher or taller objects (e.g., buildings, lighting poles, trees, etc.) to appear significantly tilted away from the focal point of the camera. In fact, this effect increases in intensity with smaller focal lengths and towards image edges due to lens curvature.

However, since ALS measurements do not suffer from the same issue, it follows that there exist numerous cases in the reference dataset where its true-colour and LiDAR components are severely misaligned.

Although, this issue can be mitigated somewhat during training with dilated convolutions, the fact that ground truth masks are based on the RGB bands of the corresponding chips clearly introduces a statistical bias towards them and may even cause significant confusion in cases where objects are completely misaligned. In fact, this is actually quite common with smaller features (i.e., light-permitting surfaces and solar panels).

Temporal Misalignment As explained in Sections 3.2.2 and 3.2.3, the BM5 8 cm imagery is updated yearly during the autumn months. On the other hand, each update of the AHN is a multi-year venture. This means that it is highly unlikely that the true-colour and LiDAR components of the reference dataset are temporally aligned, in turn leading to mask ambiguity and model confusion in cases where objects visible in one subset are absent from the other. This is oftentimes the case with solar panels, which can be installed or removed at any point, as well as buildings whose roofing has been replaced. As is the case with Section 3.4.3.4, this issue can be mitigated by using the global positioning system (GPS) timestamps of the AHN4 tiles to download and parse images from the appropriate year. However, misalignment may still exist at the seasonal level, and it is not clear whether all AHN4 timestamps represent the time of capture or end-product creation.

4. Implementation and Experimental Framework

4.1. Software Implementation

In the context of this thesis, the methodological framework presented in Chapter 3 was implemented as a Python library (Mantas, 2024a). Data collection and pre-processing operations are conducted mainly using GeoPandas, SciPy, NumPy, rasterio, and Pandas, whereas the learning pipeline is written in PyTorch using the TorchGeo (Stewart et al., 2022) and Lightning frameworks. Models are provided by timm Wightman, 2019 and TorchSeg. Performance metrics are implemented in TorchMetrics, and HPO is conducted with Optuna.

4.2. Reference Dataset

4.2.1. Modified Raster Stack Splitting Algorithm

Although the appropriateness of the reference dataset generation algorithm has been logically established (Section 3.3), it should be noted that its practical effectiveness in achieving satisfactory building coverage is highly sensitive to the size of the reference data pool. This is due to the relatively large tile size in comparison to that of the chips it is split into for annotation (Section 3.4.2) and model training purposes (Section 4.3). In particular, the mean surface area of the 3DBAG tiles under consideration was approximately 878,743.36 m^2 at the time of writing. Assuming a spatial resolution of 8 cm and ignoring any potential background filtering (Section 3.3.2), this means that, on average, a single tile is comprised of ca. two hundred and fifteen 512×512 pixel chips. However, it should be noted that data annotation is oftentimes extremely laborious and time-consuming — especially in the case of relatively small teams. After all, this is the main motivational driver behind the recently ongoing advent of foundation models for image segmentation (e.g., Kirillov et al., 2023 for general-purpose and Dionelis et al., 2024 for aerial imagery, respectively), some of which have already been integrated into commercial labelling software (Lynn, 2023).

Consequently, the tile splitting step described in Section 3.3.2 was adjusted to accommodate reduced annotation capacity, as was encountered in this thesis. Instead of merely accepting all chips with background content below or equal to the specified maximum threshold, a decision is made on whether to process a tile initially based on a simulated coin toss, even if the tile might ultimately be rejected due to its background content. Specifically, a random bit is sampled from a Bernoulli distribution,

and further processing is undertaken only if the bit is positive. This approach encourages the sampling of more tiles, given that statistically, half of them will not undergo processing regardless of their background content. Even so, a tile can occasionally be so large that only a portion of it exhausts the entire sample order. To ensure sampling of the entire tile, only three *accepted* tiles in a row are permitted, with the fourth being automatically rejected, regardless of the coin toss result or its background content.

4.2.2. Overview

The reference dataset (Mantas, 2024b) contains 200 chips representing four 3DBAG tiles (400 ha); one in Delft, two in Dordrecht and one in Enschede. The total area sampled is 335544.32 m^2 out of which approximately 45.4% (152274.42 m^2) represents contextually relevant areas. This equates to 1,170 buildings, which makes the dataset not only the *first* for semantic segmentation but also the *largest* only *free and open-source* and for roofing material classification, in general, at least to the author's best knowledge at the time of writing. Out of this, 15685909 pixels or 100389.82 m^2 were annotated over the course of three months with eight different classes, namely: dark- and light- coloured membranes, ceramic tiles, gravel, light-permitting surfaces, metal, solar panel arrays, and vegetation (Table 4.1).

Table 4.1.: Final class name-colour-label associations. This table is used only in reference to the reference dataset.

Name	Colour	Label	Name	Colour	Label
Background		0	Solar Panel		7
Dark-coloured Membrane		1	Vegetation		8
Ceramic Tile		2			
Gravel		3			
Light-permitting Surface		4			
Metal		5			
Light-coloured Membrane		6			

The other supported classes were not encountered; either at all or not to a significant degree (i.e., in more than one image), indicating either a flaw in the sampling process or that they are not actually required to effectively model Dutch roofs. It is important to note that there were considerable challenges in accurately classifying residential

roofs under the ceramic tile category. While light-coloured roofs are assumed to be generally correctly identified, dark-coloured roofs exhibited much greater uncertainty. Often, even with close examination of street view imagery, it was unclear whether these roofs were made of painted concrete, matt-finished metal, or indeed tiles. For simplicity and completeness, such instances were always categorized under ceramic tiles rather than being left unlabelled. However, the uncertainty of these labels is admittedly very high, and any related predictions should be approached with extra caution. Currently, in the absence of supplementary data, the ceramic tile class functions more as a broad label for generic "tile-like" materials.

In any case, the pixel- and image-level class proportions are presented in Figure 4.1.

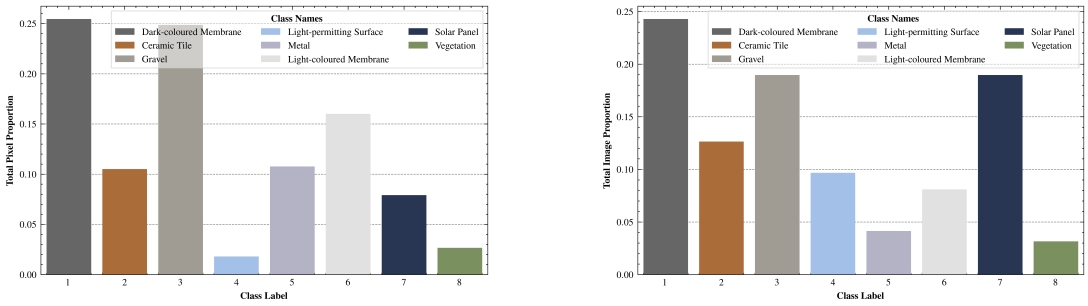


Figure 4.1.: Pixel- and image-level class distribution across the reference dataset.

The majority of samples represent dark-coloured membranes (25.5%; 3994104 pixels), and gravel (24.9%; 3901064 pixels), with light-coloured membranes (16%; 2511431 pixels) trailing a bit behind the two former classes but still achieving a significantly larger support than the remaining ones, each of which represents less than 11% of the total. This majority grouping is expected and may be explained by the fact that most of these materials are encountered in buildings of relatively large size which are naturally more likely to be sampled than smaller ones, especially given the imposed maximum limit on the background content of sampled patches. In addition, specifically regarding dark-coloured membranes, it was found that they are especially common as roofing materials for flat dormers, which are a defining characteristic of typical Dutch homes. Hence, this material was essentially found in almost all buildings, with more than 60% of all chips (123) containing it. In essence, a recurring trend seen in all sampled tiles is that the class distribution typically consists of two or three distinct groups. These groups are primarily determined by the types of buildings present and the relative size of each building compared to its surroundings and the imposed sampling limitations.

This considered, the second most populous material group consists of metal (10.8%; 1689750 pixels), ceramic tiles (10.5%; 1647803 pixels), and solar panels (7.9%; 1241648 pixels). Similarly to the first group, this one may also be explained when considering the characteristics of each material (Section 3.4.1). Specifically, metal is frequently used as a roofing material for large commercial and industrial buildings, which have been previously noted to have a higher odds of being included in the sample. However,

it does not offer as much versatility as membranes, resulting in a significantly lower presence in the reference dataset, except for a few specific instances of large commercial and industrial buildings with sloping roofs, where it is the preferred option. In addition, it is currently a widespread practice to utilize the unoccupied roof space of most suitable buildings by mounting solar panel arrays. Therefore, this material is almost as common as dark-coloured membranes, being present in 48.5% (97) of all sampled chips, and thus has a fairly sizeable support despite its relatively small size. The only apparently paradoxical point about this group is the limited representation of ceramic tiles. This is particularly surprising, given that most sampled structures seem to be residential, where tiles are predominantly utilized for pitched roofs, which are very common because of the high likelihood of heavy rain and snowfall in the Netherlands. It is assumed that the most likely reason behind this apparent undersampling due to a combination of the relatively small size and general design of the typical Dutch house in combination with the imposed background content limit. In particular, even though residences are commonly constructed in close proximity or right next to each other, forming dense parcels, they usually include gardens and patios, or are bordered by canals, roads, etc., all of which are not relevant for the context and contribute to the limit. This is actually why the threshold was established at a relatively high value of 80%, as lower values led to almost no residential buildings being sampled.

Finally, the minority group contains vegetation (2.7%; 417762 pixels) and light-permitting surfaces (1.8% 282347). This observation is not surprising, as green roofs remain relatively rare because they typically have higher construction and maintenance costs compared to other alternatives. In contrast, while light-permitting surfaces are present in a substantial portion of the reference dataset (25%; 50 chips), they are generally smaller than the objects represented by other categories. This is because they primarily refer to residential skylights and opaque polycarbonate openings on metal industrial roofs, thus making it challenging to gather a large and diverse sample in terms of the total number of pixels sampled.

A summary of the sample statistics for each tile is given in Table 4.2.

Table 4.2.: Reference dataset overview.

Tile ID	Region	Sampled Chips	Sampled Buildings	Sampled Pixels
9-272-552	Delft	15	25	1334889
9-368-464	Dordrecht	60	392	4605217
9-380-480	Dordrecht	76	637	5166454
9-976-648	Enschede	49	116	4579349
Total		200	1170	15685909

4.3. Model Training

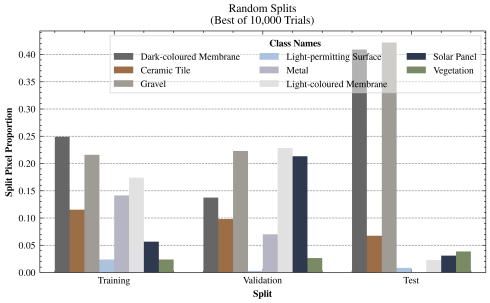
4.3.1. Reference Dataset Splitting and Class Weighting

The proposed workflow recognises the fact that the implementation dataset (Section 4.2) is likely too small, at least according to Mather and Koch, 2022, who recommend a minimum training subset size of at least thirty times its spectral dimensionality (i.e., 30×7 bands per chip $= 210$ examples). Moreover, it is obvious that roofing material datasets are most probably inherently imbalanced due to oftentimes major differences in the applicability, nature, and popularity of various materials in the given study area (Wyard et al., 2023) (Wyard et al., 2023). For instance, one likely cannot expect to produce a dataset representing an industrial region which contains a significant number of green roofs.

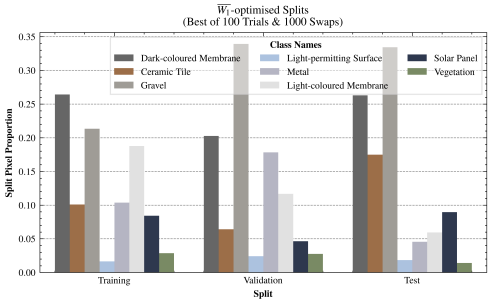
This considered, an iterative stratification algorithm inspired by Xiao et al., 2018 was developed as an alternative to existing implementations of the aforementioned relevant methods (Szymański & Kajdanowicz, 2017b), which were found to not be able to split reference datasets into three separate subsets. At this point it should be noted that a potential adaptation of said methods to the requirements of this thesis was not considered due to pertinent time constraints.

First, the reference dataset, $\mathcal{D}$ is split randomly into a training, $\mathcal{S}_{\text{train}}$, validation, $\mathcal{S}_{\text{val}}$, and test, $\mathcal{S}_{\text{test}}$, set of size, $q_{\mathcal{S}} \forall \mathcal{S} \subset \mathcal{D}$, equal to *70%*, *15%*, and *15%* of the original, respectively. Because $q_{\mathcal{S}}$ may not be a factor of $|\mathcal{D}|$, each subset is initially populated with $\lfloor q_{\mathcal{S}} \times |\mathcal{D}| \rfloor$ elements from $\mathcal{D}$ and any remainders are in turn distributed in a round robin-fashion starting from $\mathcal{S}_{\text{train}}$. After the split has been performed, each subset is checked for complete class coverage (i.e., at least one pixel from a given class is present in at least one corresponding ground truth mask, for each class under consideration). If a set is found to not contain one or more labels, the whole split is discarded and the splitting process is repeated until said condition is satisfied. Once an appropriate split has been found, it is greedily optimised using the following heuristic. First, the mean Wasserstein 1-distance, $\overline{W_1}$ between each subset pair is computed as an initial approximation of the overall disparity between the underlying class distributions. This quantity is normalised by the relevant set size such that it may be compared across subsets. Then, for each pair, two items are randomly sampled; one related to the first set and one to the other. These items are swapped if and only if $\overline{W_1}$ would be improved while still maintaining complete class coverage. This process is repeated for a predefined number of successful swaps or until $\overline{W_1}$ stops improving significantly, or becomes equal to zero. Any difference between two consecutive changes in $\overline{W_1}$ which is greater than 10^{-6} is considered an significant improvement. Subsequently, the optimized split and pertinent $\overline{W_1}$ are saved in-memory, and the whole process is repeated for a predefined number of steps in order to potentially discover better initial splits. Finally, the split with the lowest overall $\overline{W_1}$ is returned. At this point it should be noted that this algorithm has not been tested against similar works, but was experimentally shown to significantly better than random splitting (Figure 4.2).

Finally, label imbalance in the training subset is mitigated using the following



(a) Random splitting; best in terms of mean normalised wassserstein distance out of 10000 trials.



(b) Implemented splitting.

Figure 4.2.: Possible random and actual splits of the implementation dataset.

weighting scheme. This approach is again adopted mainly on the basis of pertinent time constraints prohibiting the appropriate adaptation of sampling algorithms as most already existing implementations (Lemaître et al., 2017) do not consider multi-label problems. Given n_c classes present across $n_m \equiv |\mathcal{D}|$ ground truth masks, $M \in \mathbb{R}^{h \times w}$ of width, w, and height, h, in $\mathcal{D}$, the inverse overall frequency f_k^{-1} of a class $k \forall k \in \{0, 1, \ldots, n_c - 1\}$ is given by Eq. (4.1):

$$f_k^{-1} \coloneqq \frac{\sum_{M \in \mathcal{D}} h \times w}{\sum_{M \in \mathcal{D}} \text{count}_k\left(M\right)} \tag{4.1}$$

where $\text{count}_k\left(M\right)$ denotes the pixel count of k across M and is defined as:

$$\text{count}_k\left(M\right) \coloneqq \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \mathbf{1}_k\left(M\left(i, j\right)\right)$$

where $\mathbf{1}_k\left(M\left(i, j\right)\right)$ is an auxiliary indicator function which is equal to one if k is equal to $M\left(i, j\right)$, otherwise zero. The numerator of Eq. (4.1) is simply to the total number of pixels across all masks, whereas the denominator the total pixel count of k. This is the primary weighting scheme in the case of class imbalance. However, one issue with this approach is that it does not account for class distribution throughout the dataset because it is a microscopic heuristic, i.e., it ignores the existence of multiple masks and instead treats the dataset as a single mosaic. For example, it is trivial to prove that the (inverse) frequency of a class with a certain number of samples in one mask is the same as that of one with equal presence across all masks. However, one could argue that the former class should be assigned a greater weight because it is less likely to be learned as well as the latter.

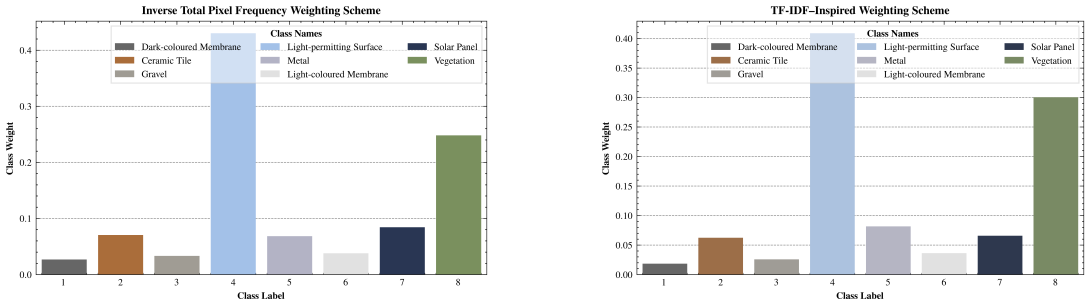
In order to account for its rate of occurrence at the image level, each class is weighted by the product of its *inverse mean pixel-level* and *inverse image-level* frequencies. This measure is henceforth referred to as the inverse mean pixel frequency–inverse image

frequency (impf-iif) and is inspired by term frequency–inverse document frequency (tf-idf), which is used in information retrieval systems to represent the relative importance of a given word in regard to both the document it appears in and the collection it itself potentially belongs to (Rajaraman & Ullman, 2011).

$$\begin{aligned}(\text{impf-iif})_k &:= \text{impf}_k \times \text{iif}_k \\ &= \left(\frac{1}{n_m}\sum_{M\in\mathcal{D}} \text{count}_k\left(M\right)\right)^{-1} \\ &\times \left[\log_b\left(\frac{n_m+1}{|\{M\in\mathcal{D}|\text{count}_k\left(M\right)>0\}|+1}\right)+1\right]\end{aligned} \tag{4.2}$$

In contrast to Eq. (4.1), the first bracketed term of Eq. (4.2), i.e, the inverse mean pixel frequency, is defined macroscopically as the mean pixel count of k across all masks. On the other hand, the second term, i.e., the inverse image frequency, represents class frequency at the image instead of the pixel scale according to whether the class under consideration exists in each image or not. The unit constant in the numerator and denominator is mathematically equivalent to assuming an additional mask was seen containing exactly one pixel of each class, which prevents zero divisions. This quantity is then scaled logarithmically to compute the information content of the class, which is related to its entropy, i.e., the expected "surprise" upon its occurrence. The base of the logarithm determines the unit of information, with common ones being the *shannon* ($b = 2$), the *natural unit of information* ($b = e$), the *trit* ($b = 3$), and the *hartley* ($b = 10$). In the context of this thesis, b is set to the maximum information capacity of the dataset, i.e., the total number of unique classes it contains. This is equivalent to assuming an equal prior probability of occurrence for each class throughout the dataset, further augmenting the weights of particularly rare classes. Finally, a smoothness term of 1 is added to the information content to enforce strictly positive weights for classes present in all images.

Similarly to, it should be noted that Eq. (4.2) not been mathematically tested for correctness, but was experimentally shown to significantly better than Eq. (4.1) (Figure 4.3).



(a) Inverse frequency.

(b) Proposed.

Figure 4.3.: Class weights of the training set using inverse frequency and the TF-IDF inspired method.

4.3.2. Data Augmentation

In the context of roofing material classification, only select works make any mention to the concept of data augmentation. In particular, only Krówczyńska et al., 2020; Wyard et al., 2023 mentioned basic geometric augmentations, such as reflections, rotations, and transpositions. Similarly, this thesis adopts the augmentations presented in Table 4.3, which are applied sequentially in an online fashion.

The geometric transforms during training correspond to the D_4 dihedral group (i.e., the symmetries of the square), since the reference dataset contains square chips, with the exception of the two diagonal reflections which do not appear to be favoured in practice. Intensity augmentations are not applied because it is assumed that the reference dataset has enough inherent variation in this regard, which is always preferable relative to synthetic data.

Table 4.3.: Augmentations used to rescale each image band of the reference dataset and artificially increase its (i.e., the dataset's) size.

Stage	Augmentation	Element Probability	Batch Probability	Same across Batch
Training	Scaling to [0, 1]	100%	100%	✓
	Reflection about the y-axis	50%	100%	
	Reflection about the x-axis	50%	100%	
	Anti-clockwise rotation by 90°	50%	100%	
	Anti-clockwise rotation by 90°	50%	100%	
	Anti-clockwise rotation by 90°	50%	100%	
Inference*	Scaling to [0, 1]	100%	100%	✓

* In this context, "inference" refers to model states where the underlying back-propagation graph has been disabled,batch normalisation (BN) layers are not updated, and dropout layers are inactive (i.e., validation, test, and traditional inference).

4.3.3. Loss Function

This considered, the loss function employed in the context of this thesis is a linear combination of CE and Dice loss, with the former being weighted according to Eq. (4.2).

4.3.4. Performance Metrics

4.3.4.1. Confusion Matrix

It is commonly known that the primary tool for predictive performance evaluation in the domain of supervised learning is the confusion matrix because most relevant predictive performance metrics may be derived from it.

The *confusion matrix*, $C \in \mathbb{Z}^{n \times n}$, is defined as a square matrix of order equal to the overall number of classes involved in a given classification problem, n, such that each of its elements $c_{ij} \coloneqq C(i,j)\, \forall i,j \in \{0,1,\dots n-1\}$, that is the entry at the intersection of row i and column j or position (i,j), represents the total number of observations in a particular sample known to belong to class i and classified into j (Table 4.4).

Table 4.4.: Typical structure of a multi-class confusion matrix. The elements of this particular matrix are arbitrary.

	1	1	5	1
Actual Class	2	8	7	7
	3	3	10	1
		1	2	3
		Predicted Class		

This considered, the following relevant nomenclature and related observations are presented below:

1. The element at position $(k, k), k \in \{0, 1, \dots n-1\}$ corresponds to the *true positive* (TP) of class k. Hence, the sum of all entries along the main diagonal of C is equal to the overall number of correct predictions made by the classifier regardless of class.

2. The sum of all off-diagonal items in column k (i.e., $\sum_{i \neq k} c_{ik}$) is called the *false positive* (FP) of class k. This count is equivalent to the total type I error of the model.

3. The sum of all elements excluding those along row and column k (i.e., $\sum_{i \neq k} \sum_{j \neq k} c_{ij}$) represents the *true negative* (TN) of class k.

4. The sum of all off-diagonal entries along row k (i.e., $\sum_{j \neq k} c_{kj}$) is corresponds to the *false negative* (FN) of class k. This is equivalent to the total type II error of the model.

5. The sum of all TPs and FNs of a given class is called its *support*.

In addition, it follows from the primary definition of the confusion matrix that the sum of all elements along a given row and the corresponding column is equal to the overall number of actual and predicted instances of the related class, respectively. By extension, the sum of all such instances is equivalent to that that of all entries in the confusion matrix (i.e., $\mathrm{Tr}\,(C^{\intercal}C) \equiv \sum_i \sum_j c_{ij}$), which represents the sample size. Furthermore, it stems from Definitions 2 and 4 that the sum of all off-diagonal elements is equal to the overall number of incorrect predictions made by the classifier regardless of class.

Finally, it should be noted that a special case of the confusion matrix is the binary case. This is because, as there are only two possible classes, the TPs of one are by definition the TNs of the other and vice versa. Hence, in this case and this case **only** the first class is be commonly chosen arbitrarily to be the "negative" and the second the "positive", and therefore the matrix may transformed into Table 4.5.
Although the structure of this matrix abides by all previously mentioned definitions and observations, the multi-class notation will be used **exclusively** in the context of this thesis due to potential confusion with the otherwise ambiguous meaning of TNs.

Table 4.5.: Typical structure of a binary confusion matrix. The first and second class are commonly called "negative" and "positive", respectively.

Actual Class	0	TN	FP (Type I Error)
	1	FN (Type II Error)	TP
		0	1
		Predicted Class	

One potential source of confusion with multi-class confusion matrices and classification problems in general concerns the computation of a global indicator for the overall performance of a given classifier. Although this is relatively straightforward in the case of binary classification as metrics may simply be calculated with respect to either the positive or negative class, depending on the particular metric and use-case, this concept does not generalise because in a n-class problem, where n is greater than two, it is not clear which class should be considered positive and which negative.

Macroscopic & Weighted Averaging The first possible solution to this issue is *macroscopic* or *macro*, for short, averaging where the metric of interest is computed separately for each class and then averaged over them. The reduction step is commonly performed by taking the arithmetic mean of the corresponding class-wise metrics, either ordinary or weighted by their support. This the reason why several traditionally macro averaged metrics, such as *mean intersection over union* (mIoU) (Section 4.3.4.4), are called as such. The former case constitutes traditional macro averaging while in the later, the method is known as *weighted* averaging. In particular, let an n-class classification problem and a relevant metric of interest m^k corresponding to a given class $k \forall k \in \{0, 1, \ldots, n-1\}$. Then, the pertinent macro averaged metric, $\bar{m}_{macro}$ is given by:

$$\bar{m}_{macro} = \frac{1}{n} \sum_k m^k \tag{4.3}$$

and the corresponding weighted metric, $\bar{m}_{weighted}$, by:

$$\bar{m}_{weighted} = \frac{\sum_k (TP^k + FN^k) m^k}{\sum_k (TP^k + FN^k)} \tag{4.4}$$

where TP^k and FN^k are the TPs and FNs of class k, respectively.

Microscopic Averaging On the other hand, *microscopic* or *micro*, for short, averaging directly incorporates the TPs, FPs, TNs, and FNs of each class, into a single calculation to produce an *overall* result. This the reason why several traditionally micro averaged metrics, such as accuracy (Section 4.3.4.2), are called as such. In particular, let an n-class classification problem and a relevant metric of interest $m\,(TP, FP, TN, FN)$ be a function of the corresponding elements of the confusion

matrix. For instance, m can represent accuracy (Section 4.3.4.2). Then, the pertinent micro averaged metric, $\bar{m}_{micro}$ is given by:

$$\bar{m}_{micro} = m\left(\sum_k TP^k, \sum_k FP^k, \sum_k TN^k, \sum_k FN^k\right) \tag{4.5}$$

where TP^k, FP^k, TN^k, and FN^k are the TPs and FNs of class $k \forall k \in \{0, 1, \ldots, n-1\}$, respectively.

4.3.4.2. Accuracy

Accuracy is generally defined as the *ratio* of *correct* predictions of a given class to its *support*, that is the overall empirical probability of a randomly selected prediction being correct (Eq. (4.6)).

$$Accuracy \equiv \tilde{P}\left(PositivePrediction \mid Positive\right) \coloneqq \frac{TP}{TP + FN} \tag{4.6}$$

However, this definition is actually the same as that of recall (Eq. (4.9)) at the class level, and hence accuracy is commonly reported directly at the global scale. The most common averaging scheme for this metric is micro-averaging which results in the so-called *overall accuracy* (OA) (Eq. (4.7).

$$OverallAccuracy \coloneqq \frac{\mathrm{Tr}\left(C\right)}{\mathrm{Tr}\left(C^\intercal C\right)} \tag{4.7}$$

Although this metric provides an admittedly intuitive performance estimate, it is prone to bias towards the majority class since each class is assigned equal importance in the relevant calculations. For instance, let a sample of 10 positive and 90 negative observations, correspondingly. Then, a clearly inappropriately trained model which makes only negative predictions would be able to achieve an otherwise seemingly high accuracy score of 90%. On the other hand, a perhaps more balanced performance of 8 and 80 correct positive and negative predictions, respectively, would actually result in a lower score of 88%. This phenomenon is obviously distracting and undesirable in cases where all classes are equally important regardless of their support. Hence, accuracy is rarely the primary metric of choice in segmentation tasks which are inherently imbalanced.

4.3.4.3. Precision & Recall

Precision or positive predictive power, rate, or value, or *user's accuracy*, depending on the context, is defined as the *ratio* of *correct* to all *positive* predictions of a given class, that is the empirical conditional probability of a randomly selected positive prediction being actually correct (Eq. (4.8)).

$$Presision \equiv \tilde{P}\left(PositiveObservation \mid PositivePrediction\right) \coloneqq \frac{TP}{TP + FP} \tag{4.8}$$

Similarly, *recall*, detection probability, power, *sensitivity*, *producer's accuracy*, or true positive power, rate, or value, depending on the context, is defined as the ratio of correct to all positive observations, that is the empirical conditional probability of a random selected positive item being correctly classified (Eq. (4.9)).

$$Recall \equiv \tilde{P}\left(PositivePrediction|Positive\right) \coloneqq \frac{TP}{FN+TP} \tag{4.9}$$

Although both micro-averaged precision and recall also suffer from the same potential bias issue as accuracy Section 4.3.4.2[1], their advantage in comparison is that they are generally able to better evaluate performance on FPs and FNs, respectively, and are hence more appropriate in cases where either misclassification is considered to be important.

This considered, precision and recall may be combined to form the F_β measure, which, when micro-averaged, serves as a typically more balanced global performance indicator than accuracy, as well as a comparator between models with different overall precision and recall scores.

$$F_\beta \coloneqq \left(1+\beta^2\right)\frac{Precision \times Recall}{\beta^2 Precision + Recall} = \left(1+\beta^2\right)\frac{TP}{\left(1+\beta^2\right)TP + FP + \beta^2 FN} \tag{4.10}$$

The most commonly used F-measure is F_1, which is equivalent to the harmonic mean of precision and recall.

4.3.4.4. Jaccard Index

The Jaccard or Tanimoto index or similarity coefficient or intersection over union (IoU), depending on the context, $J\colon A, B \rightarrow [0,1]$, is defined as the ratio of the cardinality of the set produced by computing the intersection of two finite sets, A and B, to that of their union (Eq. (4.11)).

$$J\left(A,B\right) \coloneqq \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| + |A \cap B|} \tag{4.11}$$

In classification problems, A and B are assumed to be the predicted and actual positive observations, correspondingly. Hence, Eq. (4.11) becomes:

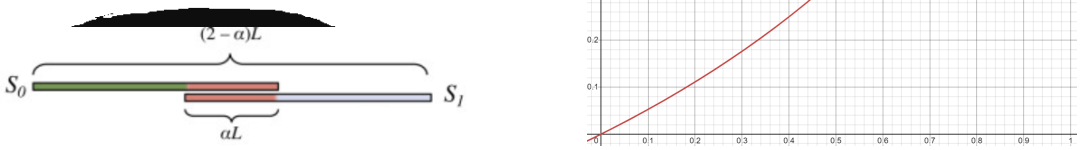
$$J = \frac{(TP+FP) \cap (TP+FN)}{(TP+FP) \cup (TP+FN)} = \frac{TP}{TP+FP+FN} \tag{4.12}$$

Notice that the structure Eq. (4.10) and Eq. (4.12) is very similar. In actuality, it may be proven through basic algebraic manipulations that:

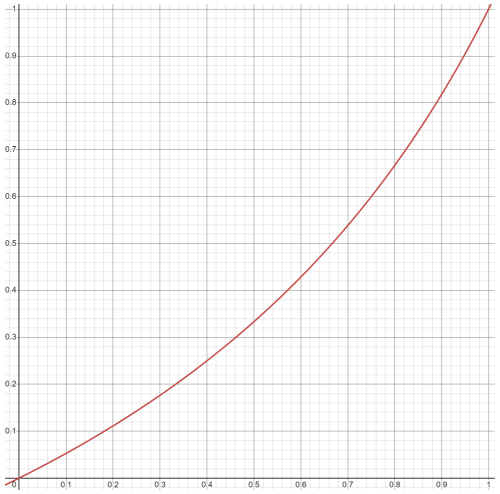
$$J = \frac{F_\beta}{\beta^2\left(1-F_\beta\right)+1} \tag{4.13}$$

[1] Let a sample of 90 positive and 10 negative observations, correspondingly, and an inappropriately trained model which only makes positive predictions. Then, the respective overall precision and recall scores would be equal to 90% and 100%, respectively.

The Jaccard index may be interpreted as a measure of the overlap between A and B, which in the case of semantic segmentation can be considered to represent a prediction and ground truth mask, respectively. Therefore, low IoU scores generally signify inferior segmentation quality than higher ones. However, it should be noted that these values don't exactly correspond to the actual overlap between the masks because any differences are exponentially penalised (Figure 4.4).



(a) Geometric configuration.



(b) $J(a; A, B)$.

Figure 4.4.: Jaccard index, $J(a; A, B)$ as a function of the fractional overlap, a, of two linear segments, A and B, of equal length, l, as one is "slid" across the other. Notice that $J(0.5) = \frac{1}{3}$ and that $J\left(\frac{2}{3}\right) = 0.5$. Adapted from Baharav et al., 2020.

However, it is clear that this property also makes IoU highly resilient against potential class imbalance issues.

4.3.5. Model Design

As implied in Section 2.2, the proposed methodology adopts an encoder-decoder model composed of ResNet-18 (Howard et al., 2019) and DeepLabv3+ (Chen et al., 2018), respectively. The main motivation drivers behind this design were the intertemporal popularity of the DeepLab architecture in both general computer vision tasks and RS problems, the scarcity of sufficient computational resources to facilitate extensive HPO (Section 5.1) and maintain acceptable latency of the overall development cycle had a more complex backbone been employed, as well as the likely limited size of the implementation dataset (Section 4.2), which was assumed would cause more powerful models to severely overfit the corresponding training subset without significant regularisation efforts[2]. However, due to the above-mentioned resource deficiency

[2]In fact, this was later observed to be true even in the case of this model, albeit to a controllable degree (Section 5.1.2)

and relevant time constraints, it would perhaps be impossible to properly guide these efforts to fruition.

In actuality, the decoder was chosen first. This was because `torchseg`, which served as the decoder provider in the implementation source code () only offered eight other models at the time of writing, and hence it was straightforward to consider each one in sequence. On the other hand, `timm` (Wightman, 2019), which was adopted as the respective encoder provider, served 858 different models which were compatible with `torchseg`. Therefore, the decided upon course of action was that once a decoder was selected, the backbone employed in the related publication or similar would be used for brevity. Indeed, time constraints were also the reason why the experimentation with and adaptation of potentially more appropriate architectures not available in `torchseg` or `timm` was not pursued.

Taking this into consideration, DeepLabv3+ was the obvious decoder choice since it combines the advantages of the U-Net (Ronneberger et al., 2015) family of models, namely skip connections between the encoder and decoder, with the concept of feature pyramids, which is also adopted by (Li et al., 2018; Zhao et al., 2016) (). This property generally allows the model to better encapsulate both low- and high-level semantic information at multiple scales, in turn enhancing its contextual scene understanding as coarse and fine features are independently captured by larger and smaller receptive fields, correspondingly (Section 2.2.3). This is particularly important in the context of this thesis, as the objects represented by the various annotation classes (Section 3.4.1) typically vary significantly in size. In addition, relatively broad receptive fields facilitate the exploitation of pertinent cues from neighbouring buildings, as each chip in the reference dataset does not necessarily depict only one. Specifically, roofing materials of nearby buildings may be similar due to architectural styles, local regulations, or the use of the same materials in a neighbourhood, and thus the human intuition that if several buildings in close proximity all have ceramic tile roofs, then a similar looking one nearby also has a tile roof is passed on to the model. Furthermore, it is worth mentioning that this idea has also been directly proposed in related previous work (Solovyev, 2020). Moreover, the notion of global context (i.e., the capturing of long-range feature dependencies) is the main driver behind the advent of the self-attention mechanism, which is integral to the transformer family, revolutionising the field of deep learning in the 2020s. With the benefits of expandable receptive fields having now been established, dilated convolutions, the unique situational advantage of DeepLabv3+ over Li et al., 2018, the only other similar architecture available in `torchseg` becomes clear: they can be widened without incurring any additional computational cost. Moreover, this property was considered to be more desirable than the attention mechanisms employed by other relevant options (Fan et al., 2020; Li et al., 2018), especially since the backbone itself contains squeeze-and-excitation blocks. On a related note, DeepLabv3+ also uses separable convolutions for substantially reduced computational complexity (Section 2.2.3),

As previously mentioned, the encoder choice was initially inspired by Chen et al. (2018) who adopted a more powerful variant (Chollet, 2016) of the Xception family (Dai et al., 2017). Nevertheless, it was found that no pertinent version offered in

`timm` was able to achieve and maintain acceptable data throughput when trained end-to-end using the most up-to-date respective protocol (Table 4.6) and a batch size of at least eight. At this point, it should be mentioned that it had already been decided that ensuring at least this batch size at least during training was more important than accommodating a potentially more powerful encoder into the final model, because all backbones tested in the context of this thesis contained multiple BN layers which require relatively large batch sizes to train effectively. In particular, Chen et al., 2017 conducted a relevant experiment which showed that this baseline in fact constituted the bare minimum to achieve competitive predictive performance in comparison to their best performing model in this regard. In addition, simply decoupling batch size and feature normalisation by replacing BN with layer, instance or group normalisation was not considered as a viable alternative, because it has been shown to hurt performance (Wu & He, 2018). Therefore, attention inevitably turned to Chen et al., 2016, 2017 both of which employed ResNet-101. However, computational resource constraints again significantly limited the de facto backbone options, and it was found that only ResNet-18 and 34 managed to satisfy the corresponding efficiency requirements imposed earlier. Finally, out of these two models, ResNet-18 was selected despite the fact that it may not able to fully utilize the advantages of its architecture in comparison with traditional deep convolutional neural networks (DCNNs) (K. He et al., 2015) because ResNet-34 failed to maintain a relatively consistent throughput throughout a series of sequential training trials.

4.3.6. Hyperparameter Optimisation

4.3.6.1. Initial Configuration

Once the model design (Section 4.3.5) had been finalized, a baseline which would serve as the starting point of the HPO process (Section 4.3.6.4) was established. Hence, the main purpose of this configuration was not so to be or even approach the performance of the final model (), although this would clearly be welcome if eventually proven to be the case, but represent a relatively simple and computationally efficient solution to the problem at hand while still maintaining acceptable predictive performance (Godbole et al., 2023). This realization stemmed from the understanding that introducing unnecessary complexity could impede future model development if it were too computationally intensive, hard to comprehend, reason through, and optimize. In the worst-case scenario, it might even degrade performance, necessitating its removal and thereby invalidating many completed experiments and their results, which could has served as a foundation for further work.

This considered, the training protocol used in the model design phase (Table 4.6) was initially adopted as it was observed during relevant experiments that it fulfilled the aforementioned performance and usability criteria.()

This was somewhat expected due to the underlying assumption that this protocol had been tested with various model configurations, tasks, and problems and was hence generally designed with flexibility in mind. However, as the so-called “no-free-lunch”

Table 4.6.: Initial training protocol. Any parameter not mentioned assumes its default value as per PyTorch v2.2.2.

Parameter	Value
Optimisation Algorithm	Adam
β_1	0.9
β_2	0.999
ϵ	10^{-7}
Learning Rate	0.001
Learning Rate Annealing	Polynomial
Polynomial Decay Exponent	0.9
Encoder	ResNet-18
Epochs	300

theorem states, there was no evidence that the performance this protocol offered was indeed representative of the problem or the true potential of the model. Hence, given that this potential issue needed to be investigated before greedily tuning the protocol, a two-phase, HPO process was designed in accordance to Godbole et al., 2023.

The primary purpose of the first phase, which is henceforth referred to as the *exploration* phase, was the experimentation with various configurations in order to gain an understanding of the interaction between various hyperparameters as well as their individual and combined effect on performance in comparison to pertinent theoretical knowledge (e.g., Goodfellow et al., 2016). In turn, this would naturally facilitate the identification of the most contextually helpful hyperparameters as well as the specification of an appropriate search space for their optimal values to be used later in the HPO process (Section 4.3.6.3). This considered, the model dynamics were explored and the original training protocol (Table 4.6) was appropriately adapted according to results of initially manual (Section 4.3.6.2) and later two automated quasi-random (Section 4.3.6.3) relevant studies. These experiments where in part designed based on the opinion of the author as formed through numerous insights gathered from conducting several unofficial trials conducted during the data collection process, comprising tens of studies on various parameters (e.g., anti-aliasing, attention mechanisms, data augmentations, etc.).

Finally, the *exploitation* phase, which assumed that any knowledge on the problem and model had already been gained, consisted of a single automated search round which used Bayesian optimization to optimize any parameters with a continuous search space as it was discretized in the exploration phase due to computational constraints, as well as those not fixed in the exploration phase.

4.3.6.2. Exploration - Manual Experimentation

As previously explained, an equally important goal to achieving a relatively performant model configuration at that juncture was obtaining as deep an understanding of the underlying optimisation problem as possible, of course within the scope of this thesis

as well as the relevant computational and time constraints. Hence, the first stage of the exploration phase entailed the undertaking of numerous manual experiments with the objective of studying the individual performance effect of various hyperparameters. In actuality, this step was considered to be of particular importance to the design of the following stages of this phase (Section 4.3.6.3) as it would hopefully provide valuable insights into exactly which parameters were the most impactful to performance, in which way and to what extent, none of which would be possible had the exploration phase consisted exclusively of random parameter searches.

However, the aforementioned project constraints did not allow for an in-depth analysis, and therefore the *tunable* or *variable* parameters needed to first be carefully determined. In this context, a set of thirteen different hyperparameters was compiled according to the recommendations of Godbole et al., 2023; Goodfellow et al., 2016, as well as relevant information gathered from studying several pertinent publications which disclosed their training protocols (T. He et al., 2018; Wightman et al., 2021).. The ultimately selected parameters and motivation behind them are presented in Table A.1, but they were generally selected so as to cover various end-to-end predictive performance aspects, from the shape of the input data and modifications to the encoder and decoder, to the optimisation algorithm and learning rate schedule. Thus, this list was considered to generally represent what was possible given the thesis constraints, in turn maximizing the relative reliability of the overall HPO process, but also not allowing for much more. Thus, it was decided all further pertinent experiments were limited to only these hyperparameters for brevity. In addition, any conditional parameters introduced by certain values of others (e.g., the size of the anti-aliasing kernel and stride when the corresponding layer is added to the encoder blocks) were not experimented with and left to their predefined defaults () according to their respective provider.

Once the tunable hyperparameters had been determined, at least two or three experiments were performed for continuously increasing values of each one in order to sample a point in the corresponding underfitting and overfitting regimes, respectively, as well as one intermediate third as a means of validation of the former two in case it could not be inferred from the original configuration. The limiting values of each parameter were later used to define the search space for the following randomised experiments (Section 4.3.6.3). In case a particular experiment did not result in one of these points being successfully identified, it was repeated as necessary with an appropriately adjusted hyperparameter value (i.e., lower or higher to sample towards the underfitting and overfitting regimes, respectively). In general, the two first points represent suboptimal performance due to the model not being able to fit the training data either at all or within the provided epoch budget; the former due to potentially excessively high bias whereas the latter variance. Hence, the intermediate point was assumed to represent an acceptable bias-variance trade-off. Nevertheless, it should be noted that instead of sampling the validation loss curve which is commonly used in conjunction with the training loss to determine under- or over-fitting (), these experiments instead considered changes in the validation mIoU, which was the principal performance metric used in the context of this thesis. This was because loss func-

tions are typically designed to only be used as efficient supervisory agents, that is a proxy for actual performance metrics and hence have little real-world significance as model comparators in this regard.. The reason that the metrics of interest are not directly used as loos functions is that they are commonly non-differentiable or have inherently numerically unstable first derivatives (). For instance, it can happen that the training and validation sets do not have similar class distributions and therefore gradient updates do not necessarily always correspond to a reduction of both relevant losses, resulting in apparent overfitting. Similarly, the fact that loss is generally computed directly on class probabilities instead of the corresponding predictions may lead to the model learning to be so certain about its predictions regardless of whether or not they are actually current that by late training wrong predictions with an initially relatively low probability (e.g., 51%) are now "overwhelmingly'' wrong (e.g., 99%). This translates to increased loss, but because both probabilities correspond to the same prediction, performance metrics are not affected. In other words, overfitting does occur in this case but only at the probability, not the classification level. This phenomenon is called model miscalibration (Guo et al., 2017). Furthermore, it is clear from Table A.1 than not all variable parameters are continuous or that they could even lead to both under- and over-fitting. For instance, the parameters related to the input data and the model, with the exception of the base dilation rate of the ASPP kernels, can only have two or at most three states and are all, the base rate included, designed to require additional predictive capacity or effectively increase it. Therefore, no configuration which included any of these hyperparameters could actually underfit the training subset assuming that the original configuration did not. Similarly, the purpose of any regularisation hyperparameter is to decrease capacity, and thus actively regularised models can never overfit unless the original is also overfitting and they are set to a relatively low value rendering them situationally inefficient. This considered the scope of the corresponding experiments in the case of these parameters was limited to simply identifying an appropriate value range for each one without the additional requirement that its minimum and maximum value would necessarily need to sample the underfitting and overfitting regime, respectively.

Finally, the training budget allocated to the experiments was calculated as the average number of epochs required for the original configuration to fit the training set across three identical trials. In this context, fitness was abstractly defined as the beginning of a significant plateau of low training loss and high predictive performance in terms of the corresponding mIoU, and was determined empirically. At this point it should be noted that although a clearly logical approach overall, a fixed epoch budget at this phase of the HPO process could perhaps lead to the misidentification of otherwise beneficial configurations which may inherently require more training steps to achieve their true performance potential, as is commonly the case when relatively low learning rates or high regularization are used (). However, project constraints again did not allow for more extensive experimentation.

At this point it should be noted that the parameter values used for these experiments were determined according to relevant publications (T. He et al., 2018; Wightman et al., 2021)..

4.3.6.3. Exploration - Random Search

Once the manual experimentation stage (Section 4.3.6.2) had been completed, the two limiting values of each variable hyperparameter were used to define the search space for a series of automated studies where model configurations where randomly selected from a uniform compound probability distribution (i.e., there was no assumed or inferred relationship between individual parameters or earlier configurations) and trained end-to-end given the previously adopted epoch budget (i.e., 200 epochs). Although Godbole et al., 2023 recommend that at least some of the experiments in this stage be performed with a fraction of this budget, as a trade-off between thoroughness and transferability () and exploration, the severe step-to-step and overall trial-to-trial variance discovered when designing the manually configured experiments () prohibited such a discount on the basis of maintaining an acceptable degree of statistical reliability throughout the study. In any case, the goal was still to study the combined performance effect of various hyperparameters and refine or validate the results of the previous exploration stage, albeit at a slower-than-recommended pace due to the unchanged training budget. For instance, discovering the optimal point at the edge of the quasi-random search space indicates that the boundaries might need adjustment (Godbole et al., 2023). However, the manual three-point sampling approach reduces the likelihood of this happening.

On a similar note, the primary reasoning for the preference for non-adaptive optimisation algorithms at this phase () is that the middle of the search space where the optimal hyperparameters are assumed to lie, is guaranteed to be considered given enough trials. Conversely, gray- and black-box methods might overlook the search space's middle due to some early trial bias, even if it harbors equally good points. Such non-uniformities are precisely what an effective optimization algorithm uses to expedite the search (Godbole et al., 2023). This is particularly important in this thesis due to the likely small dataset size, which leads to considerable variation in training and validation loss and metrics from step to step and trial to trial (). This variance occurs even in identical trials, complicating comparison of different scientific hyperparameter values because trials may randomly end on "lucky" or "unlucky" steps and hindering the reproduction of the best trial results (Godbole et al., 2023). In addition, this issue is worsened by the fact that computational and time constraints do not allow for its appropriate handling (e.g., by increasing the batch size, obtaining more data, using a particularly low learning rate Goodfellow et al., 2016, or stochastic weight averaging[3]). Therefore, it is actually critical that black-box approaches are not adopted at this point as it was very easy to be "unlucky'' and miss or even diverge from their target. Finally, it should be noted that modern approaches based on Gaussian processes and mixtures () are basically built on top of random search in the sense that they require several warm-up runs whose number scales superlinearly with respect to the dimensionality of the search space () in order to fully initialize. This means that a potentially significant portion of the overall study budget ends up being "wasted''

[3]At the time of writing the SWA implementation in PyTorch heavily interfered with the learning rate schedule and thus significantly limited its tuning potential.

on normal random searches and one can only hope that the remaining is used optimally. However, not only is this clearly potentially suboptimal given the computational and time constraints of this projects but it has been shown to not necessarily be the case. For instance, show that a random study with 2x the budget of SMAC or TPE outperforms it on certain datasets, with a search with the same budget not being far behind ().

All this considered, the tunable hyperparameters (Table A.1) were split into two groups, those affecting the input data and the model (Table 4.7), and those affecting the optimisation algorithm and learning rate schedule (Table 4.8).

Table 4.7.: Search space for the first automated round of the HPO process.

Category	Parameter	Value
Input Data	Append HSV	[Yes,No]
	Append TGI	[Yes,No]
Encoder	Attention Block	[ECA,None]
Decoder	Base Dilation Rate	[1,21] step 5
Regularisation	Stochastic Depth	[0,0.1] step 0.01
	Weight Decay	[0,0.01] step 0.001
Optimisation	Learning Rate	[5e-4,5e-3] step 5e-4

Table 4.8.: Search space for the second automated round of the HPO process.

Category	Parameter	Value
Optimisation	Algorithm	{Adam, AdamW}
	Learning Rate	[5e-4, 5e-3] step 5e-4
	Learning Rate Annealing	{Cosine, Polynomial(0.9)}
	Warmup Length	[50,150]step 25
Regularisation	Stochastic Depth	[0,0.1] step 0.01
	Weight Decay	[0,0.01]step0.001

The reason for this division was that it allowed to split the overall study budget into two equal parts and perform two separate studies. Although this came at the potential expense of precision, since it basically assumed that the parameters in each group basically did not have significant interaction with each other, which was not actually tested, it allowed to reduce the dimensionality of the per-study search space exponentially, while only reducing its budget linearly, better catering to the explorative goal of this phase and the constraints of this project.

The parameters associated with the goal of each round, for instance those belonging to the input data, encoder, and decoder categories in Table 4.7, are called architectural, with all others named nuisance (Godbole et al., 2023). The only reason nuisance parameters are included in the study is so that configuration comparisons are fair. Hence, whichever values of the architectural parameters are found to be quasi-optimal

will be fixed for the remained of the HPO process with the exception of the base atrous rate which will be re-optimised in the final round. The reason for this is that in contrast to the recommendations of Goodfellow et al., 2016, the search subspace is heavily discretised due to computational constraints, and thus multiple potential values of the base dilation rate will not be immediately explored. Of course the same is true for continuous hyperparameters, namely the stochastic depth, weight decay, and learning rate. In addition, whichever values of these parameters end up being quasi-optimal, they will be discarded and the parameters will be "optimized" from the beginning in the following round. Only after the final round will they be finalised.

Furthermore, each study had a fixed budget of 50 trials (), with the best configuration from the first being checked for potential issues (e.g., miscalibration, overfitting, loss spikes, etc.) and used in the latter. In case any issues were discovered, the next best trial was considered for adoption. Otherwise, the trial was repeated two times to measure the mean and standard deviation of the corresponding validation mIoU. Then, the potential improvement was adopted if its mean score was greater than the mean of the other. Potentially powerful statistical tests regarding the significance of the observed improvement were not performed due to the small sample size and great trial-to-trial variance, rendering basically any but the most extraordinary results insignificant. This process continued until all trials which displayed a potential improvement were exhausted, at which point it was assumed that the study failed to produce any improvement over the previous, and was thus enriched with an additional 50 trials until a statistically significant improvement was found. At this point it should be noted that contrary to the recommendations of Goodfellow et al., 2016 early stopping was not employed on neither the validation loss nor mIoU due to said variance. Similarly, the search space was discretised due to the relatively low training budget.

4.3.6.4. Exploitation

As mentioned in , the HPO process has so far been conducted using discretised search spaces due to computational constraints. However, this is not normally standard practice (Goodfellow et al., 2016) because it does not promote fine tuning towards the optimal hyperparameter settings. For instance, assume that the search subspace for a certain hyperparameter is defined by an integer interval. Then, if the optimal value of is not an integer, then it will never be sampled. In the worst case, if the interval is $[a..b]$ and the optimal value is equal to $\frac{a+b}{2}$, then the best possible configuration can only be within a distance of $\frac{b-a}{2}$ from the optimum. In addition, although the uniform mixture model used to sample configurations so far has been useful in the exploration phase of HPO (Godbole et al., 2023), and has actually resulted in significant performance improvements in comparison to the improved baseline , it is not suitable for target optimisation.

Hence, since all the performance effect of all architectural hyperparameters involved in the manual experimentation phase has now been thoroughly explored, the exploration phase can conclude and give way to the exploitation phase (Godbole et al., 2023). In the context of this thesis, this phase entails the greedy optimisation of the

non-categorical hyperparameters involved in the previous HPO rounds. Al tough all parameters could technically be re-optimised for potentially optimal results, it should be noted that all categorical hyperparameters involved in the study only had two potential values to choose from, and so it is assumed that their search subspace is so small that random search probably already optimized them anyway, and therefore they remain fixed.

In this iteration, the meta-optimizer— the algorithm responsible for selecting trials—has been substituted with one based on Tree-structured Parzen estimator (TPE) Watanabe, 2023, in alignment with the current SOTA in the HPO domain (Bergstra et al., 2013; Bergstra et al., 2011). In brief, TPE is a Bayesian optimisation algorithm which samples trials by fitting anGaussian mixture model (GMM) to a given percentile of the best configurations found so far, as well as another to the remaining ones, and then choosing the configuration which maximizes the ratio of the first to the second GMM, that is the configuration which has the highest probability of minimizing the objective Finally, the two GMMs are updated after each trial.

Pruning (automatic detection and cancellation of unpromising trials) was deliberately avoided due to significant step-to-step variance during validation, which could prematurely terminate otherwise promising trials, mistakenly marking their configurations as suboptimal due to an "unlucky" step.

4.4. Model Inference

4.4.1. Chip Inference

Chip inference refers to the task of predicting the roofing material map corresponding to *individual* inputs with the same or smaller spatial dimensions than those the model can process at once, in this case 512×512 pixels. These chips may or may not belong to a larger tile, but are regardless small enough in size for the model to fully process at once, either sequentially or in batches, and semantically independent (i.e., the corresponding maps may be examined individually). For instance, validation and testing epochs are fundamentally special cases of chip inference because they fulfil all the above criteria but occur only during training. In this context, this type of inference does not require any special handling as the model has presumably already had to perform it successfully multiple times by this phase of its development, as is inherently designed to do. In contrast to the recommendations of Chen et al., 2018, inference is performed using an output stride of 16. The motivation behind this decision is that the output stride effectively has a smoothing effect on the feature map at the exit of the decoder due to the way it is defined, which allows it to implicitly control the degree of upsampling of the feature map as it passes through the model. In particular, the spatial dimensions of the feature map at this location are equal to 32×32 pixels, and so the concatenated output of the ASPP must be upsampled by four times in order for the input and the output of the model to share the same spatial dimensions, assuming that the upsampling operation at the segmentation head has a constant factor of four. On the other hand, the first interpolation factor is reduced to

two when the output stride is equal to 8. Hence, it is expected that a larger output stride will reduce confusion due to contextually irrelevant objects in the input scenes, as well as reduce any artifacts potentially owing to the arguably severely limited size of the reference dataset. Therefore, performance in classes corresponding to relatively large surfaces (i.e., membranes, tiles, metal, solar panels, and vegetation) is expected to improve with an increase in the output stride. However, the opposite effect is expected in the case of light-permitting surfaces, which inherently represent physically smaller objects, as they may be effectively "smoothed out" of the feature map, that is they could become practically undetectable. Nevertheless, as most predictions should improve by an increased output stride, no relevant inference-time change was made. Test-time augmentation, which is known to improve results by averaging the output of several mutated views of the same scene under certain circumstances (Shanmugam et al., 2020) was not experimented with for brevity.

4.4.2. Tile Inference

In the context of this thesis, the most important inference task is *tile inference* where the model is tasked with generating roofing material maps (i.e., segmentation masks) for whole 3DBAG tiles as opposed to single chips. Fundamentally, tile inference is identical to chip inference (Section 4.4.1), which the model has been called to perform multiple times as part of validation and testing. This is because the model is fully convolutional (i.e., it does not feature any dense connections), and therefore does not require an input of a particular size as all weight connections are local to a specific part of it. However, it should be noted that that the computational requirements of the model generally increase superlinearly with the input size, and it is thus impractical or oftentimes simply impossible for it to fully process excessively "large" images at once. This issue is clearly critical in the case of 3DBAG tiles which are commonly more than 10000×10000 pixels in size at a GSD of 8 cm, meaning that inference must again be performed on individual patches and the corresponding masks stitched together to produce the complete overall map. Therefore the only difference between chip and tile inference is that the former is a sub-operation of the latter, which also entails said post-processing step. However, this step is not arbitrary because translational variance (e.g., due to zero-padding when performing convolutions) may result in artifacts along the interface of adjacent chips. These artifacts may manifest as misaligned but correct predictions or worse completely erroneous labels.

In general, this problem may be mitigated in one of the following two ways (Huang et al., 2018). The first entails splitting the tile into overlapping chips and then averaging the output *probabilities* (i.e., the model output after the softmax function has been applied to it) corresponding to the overlapping regions. According to Huang et al., 2018, the amount of overlap varies from application to application, with a general recommendation being that it should be approximately equal to the receptive field of the model . Similarly, the averaging scheme generally varies from a simple unweighted mean to more sophisticated methods such as Gaussian weighting to reduce the impact of the aforementioned edge effects. Finally, the second method identified by Huang

et al., 2018 involves simply clipping the output patches by a small amount (e.g., 20–40 pixels along each edge) before concatenating them. Although this may fundamentally be considered to be a more aggressive version of label averaging, and Huang et al., 2018 actually found it to perform better in comparison, it is likely relatively harder to implement correctly so as to respect the output map size and not introduce gaps between adjacent patches due to insufficient overlap.

In the context of this thesis, tile inference is performed by splitting the input tile into 512×512 pixel chips with a stride of 256 pixels, running chip inference on each one, and finally concatenating the output maps by averaging the corresponding class probabilities. Patch segments which extend beyond the boundary of the tile are filled with zeros. Larger patch sizes which are actually recommended by Huang et al., 2018 were tested but inference was found to be too slow with the decreased output stride (Section 4.4.1). Similarly, the technically superior method of label clipping was not experimented with for brevity.

4.4.3. Map Generalisation

As is the case with all semantic segmentation models, the one employed in the context of this thesis is designed to pixel-wise roofing material maps. Although these maps may be useful for various purposes which require this level of semantic level, such as energy consumption or solar potential estimation studies where information on individual solar panels and light-permitting surfaces may be useful, they can be excessive or even distracting for others. For instance, urban wind flow simulations which model the surface roughness of individual buildings would probably produce reliable results with only an appropriate approximation computed by only considering the "mean" or majority material of each surface. In addition, pixel-wise maps are not semantically aligned with their parent tiles which at the time of writing were available in only three LoDs. Furthermore, these maps are likely to contain gross errors caused by incorrect predictions due to insufficient training, and, unavoidably, unknown materials, which degrade their overall usability.

A very attractive solution which has been successfully applied (Osińska-Skotak & Ostrowski, 2015) to mitigate all these issues is the generalisation, in particular merging, of neighbouring pixels into a common cluster with a single label, similarly the titular cartographic operation. This cluster can be refer to the size of a relevant kernel, a roof segment, or even a whole building, depending on data availability and the particular use case. In fact, this process is functionally equivalent to the segmentation step in OBIA, with the exception and advantage that it is now performed following the classification step, in turn allowing for a variety of map scales to be produced from the same product on demand, without requiring additional training or inference. This also means that the main map can now be as detailed as possible, to the point where it becomes agnostic of study area and use case, both of which essentially limited OBIA, with the whole process needing to be performed from the beginning if any of them changed.

Although a "mean" or "$n - th$ quantile" material which would represent a form of

amalgamation of the physical properties of its constituents is probably useful in a variety of applications, the required aggregation is very difficult to implement correctly because different properties are averaged and ordered in different ways, depending on their definition. Thus, a more attractive option is to report the minority or majority material of each cluster. This process is performed at the roof segment level by rasterising each individual polygon comprising the roof surface footprints corresponding to the original map under consideration at a particular LoD to the GSD of the map such that all pixels contained within the polygon have the value of 1 and the rest 0, and then replacing all original pixels at the ones' positions by their majority label. Given the absence of a semantic class hierarchy, which is probably application-specific anyway, any potential ties are resolved by taking the minimum majority label.

4.5. Qualitative Performance Evaluation

Apart from the obvious quantitative performance evaluation, which entails a performance assessment and analysis on the test subset as well as relevant comparisons to the SOTA where applicable, the novelty and small size of the reference dataset also necessitate a qualitative performance evaluation in order to confirm the relevant accuracy of the ground truth masks, estimate the generalisability to of the model in various regions, and mitigate potential issues arising from the admittedly minute test set.

In this context, the qualitative evaluation of the model was conducted on tile 9-284-556 (Figure C.1), which is located in the municipality of Delft, approximately from the corresponding seed city. The tile has a total surface area of circa and contains buildings (ca.). This tile was selected as it was not sampled for the generation of the reference dataset, and is hence completely unknown to the model. In addition, a significant proportion of the tile contains the main segment of the TU Delft campus, with which the author is deeply familiar. Therefore, given the absence of relevant ground truth masks, optical material identification from the corresponding aerial imagery was expected to be easier than what it would have been had a different tile been selected. Although conducting the evaluation on a whole tile instead of individual chips is highly beneficial because it simultaneously allows for the evaluation of the corresponding inference pipeline, particularly the employed patch stitching method, it can be inefficient due to the large physical size of the underlying scene. Thus, the tile was divided into four separate segments which are henceforth referred to as performance evaluation regions, each of which has been specifically designed to assess performance in all classes and major building use types, as well as various special cases, such as buildings appearing significantly slanted and roofs with peculiar geometry, featuring materials originally not taken into consideration.

In particular, the first region (Figure C.2) contains the building of the Faculty of Architecture and the Built Environment (BK). This building is particularly interesting because the sloped segments of its roof are lined with asphalt shingles, which the model has not been explicitly learned. Hence, assuming that the model is well-trained, the expectation is that shingle pixels will be assigned to the closest valid class in terms

of appearance and spectral signature, that is either dark-coloured membranes due to their general appearance or ceramic tiles due to their overlapping texture. In addition, the roofs of both its extensions and especially the restaurant feature a number of what appears to be HVAC equipment, which is again not explicitly modelled. Therefore, it remains to be seen whether the model will simply ignore them due to their relatively small size in comparison to the roofs they are installed on or classify them as metal or light-permitting surfaces due to their bright appearance.

The second region (Figure C.3) contains the main buildings of the overall visible segment of TU Delft campus. Although most buildings share a similar roof morphology, that is large, flat sections featuring dark-coloured membranes or gravel, solar panel arrays, and HVAC equipment. Similarly to the first region, this one also features an open-air parking space in the north-west corner which is paved with concrete or asphalt to allow for vehicle traffic, with the model being unaware of either material. In addition, this building features a large glass atrium, which is one of the very few instances of light-permitting surfaces which are not small skylights and windows in the reference dataset. Considering light-permitting surfaces, it may also be interesting to see how the model handles the Faculty of Industrial Design Engineering (IDE) building due to the fact that most of the tar-and-gravel segment of its roof is taken up by what appears to be a grid of skylights, making it the only building in the reference dataset where the corresponding class is the majority. Another building of interest is the conference centre, which has a highly irregular asphalt shingle or dark-membrane roof. Furthermore, the building whose footprints are marked in cyan features a domed metal roof, making it the only building in the reference dataset where the metal class does not refer to corrugated flat sheets. Moreover, both the short segment of Faculty of Electrical Engineering, Mathematics and Computer Science (EWI)1 and EWI2, whose footprints are marked in magenta and yellow, respectively, feature unconventional roof shapes. What is more, the tall segment of EWI1 appears significantly tilted in the RGB component of the corresponding stack, more so than the BK building. Finally, whole region, especially the library, feature a significant number of vegetation instances, which is a minority class in the reference dataset.

The third region (Figure C.4) represents a typical industrial zone, containing mostly buildings with large, flat roofs dark- and light- coloured membranes, metal, and extensive solar panel arrays.

Finally, the fourth performance check region (Figure C.5) exemplifies a standard residential district. It is composed of multiple relatively separate blocks of densely built or even adjoining houses with tile roofs. These roofs display a range of colours and textures that change swiftly due to the close proximity of the structures. Additionally, flat, tar-and-gravel sections and dark-coloured membrane dormers occasionally appear.

5. Results and Analysis

5.1. Hyperparameter Optimisation

5.1.1. Initial Configuration

As mentioned in, the HPO process requires the establishment of an acceptable baseline, or initial model configuration which is able to obtain generally reasonable predictive performance on the validation subset. At this point it should be noted that as the underlying learning task at hand is that of semantic segmentation, performance is always expressed in terms of mIoU unless explicitly stated otherwise. In the context of this thesis, the baseline was chosen as the default training protocol proposed by https://github.com/google-research/deeplab2 (Table 5.1).

Table 5.1.: Baseline training protocol. Any parameter not explicitly defined assumes its default value as per PyTorch v2.2.2.

Category	Parameter	Value
Input Data	Append HSV	-
	Append TGI	-
Encoder	Variant	ResNet-18
	Attention Block	-
	Anti-aliasing Block	-
Decoder	Base Dilation Rate	6
	Stochastic Depth	-
	Label Smoothing	-
	Weight Decay	-
Optimisation	Optimizer	Adam
	Learning Rate Annealing	Polynomial
	Learning Rate	0.001
	Warmup Length	-

The motivation behind this selection was that this protocol was assumed to have been tested under various configurations, tasks, and problems and was hence generally designed with flexibility in mind.

Once the baseline training protocol had been specified, the model was trained end-to-end using a batch size of 8 for a total of 300 epochs, that is 5400 steps. The validation results are presented in Figure 5.1

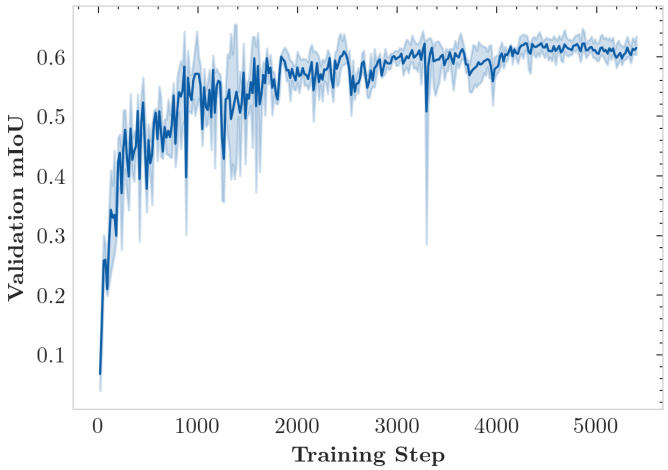
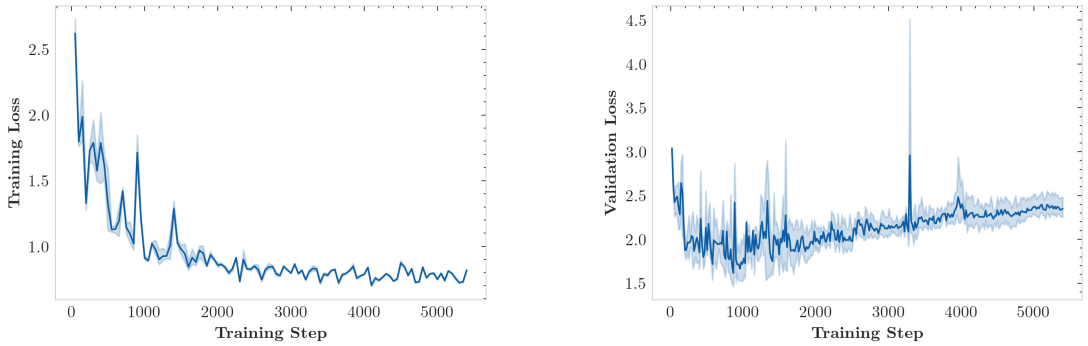


Figure 5.1.: Baseline performance as a function of training time. The shaded region represents the 95% confidence interval computed across three identical trials using the same random seed.

As shown in Figure 5.1 the model achieved an average validation mIoU of approximately 61.46% with a sample standard deviation of 1.698% by the last training step. However, there was significant step-to-step variance until the circa 2000^{th} step. In addition, a very big performance drop can be observed around the 3250^{th} step, from which the model fortunately recovered within 1–2 epochs. This is typical behaviour of a so-called "eureka" moment, which eventually turned out to be incorrect. Although relative metric instability on the validation set is generally to be expected in the early training regime, as the model is still making significant errors and corresponding weight adjustments, it should be noted that the spike actually appeared when the model had began to converge, as is visible by the training loss curve. This was relatively alarming behaviour and most likely pointed to a potentially too small validation set which, given an "unlucky" batch configuration caused certain difficult or otherwise "bad" samples to be grouped together. Alternatively, the issue could have lied with the last validation batch, which was not "complete" as the batch size was not a factor of the size of the validation set. However, further inspection of the individual validation mIoU curves revealed that the spike was actually present in only one trial. Therefore, given that the underlying matrix multiplication algorithms are not deterministic, the issue was instead primarily attributed to a "bad" path across the loss landscape and was in turn not heavily prioritized.

In addition, even though this performance was generally acceptable as a baseline, and the corresponding training protocol was in turn ultimately adopted as the baseline for the HPO process, a review of the training and validation loss curves (Figure 5.2) revealed significant miscalibration Guo et al., 2017.

This issue was deemed to be of particular importance because, although did not appear to affect performance directly, it meant that the model was becoming increasingly



(a) (b)

Figure 5.2.: Training (a) and validation (b) loss as a function of training time using the baseline training protocol (Table 5.1). The shaded region represents the 95% confidence interval computed across three identical trials using the same random seed.

overconfident in its predictions. However, given that both the literature review and reference dataset annotation processes revealed significant intra-class variability and inter-class similarity, it would perhaps be more appropriate for the model to maintain a relative degree of "hesitation" until convergence, or, in simple terms, be "careful" and not "jump into early conclusions". Hence, various regularisation techniques were included into the HPO process on an ad hoc basis, beginning from the initial, manual experimentation step (Section 5.1.2).

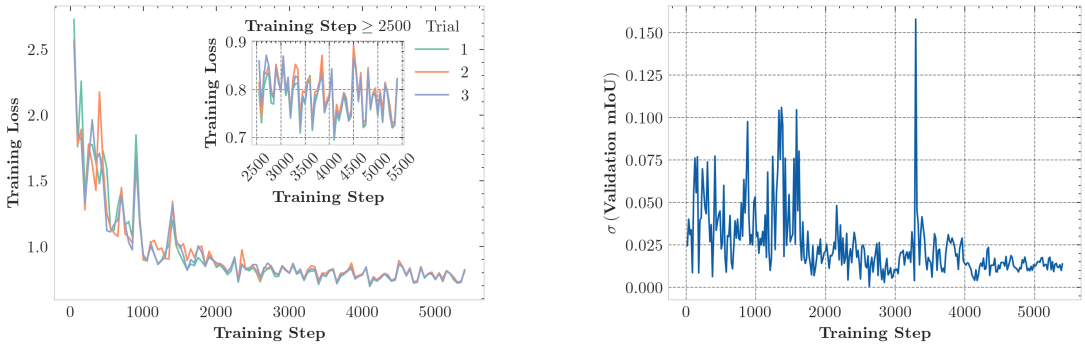
5.1.2. Manual Experimentation

As mentioned in , the initial step of the HPO process entailed manual experimentation with various hyperparameter values with the goal of investigating their individual effect on predictive performance. In addition, it was supposed that these experiments would allow the definition of an appropriate, that is reasonably small, search space in order to maximize the efficiency of later rounds given the significant time and computational constraints of this thesis.

According to, the training budget allocated to all future experiments was set to 200 epochs as that was the point were the baseline configuration achieved convergence (Figure 5.3).

The average validation mIoU at the selected step (i.e., 3600^{th}) was approximately 60.1% with a sample standard deviation of 1.227%.

In terms of potential amendments to the input data, it appears that the inclusion of the HSV and TGI bands, as computed from the corresponding RGB channels of the original input data, improve predictive performance. In general, both observations are to be expected. On the one hand, the HSV colour space represents colorimetric properties which are not explicitly modelled by RGB. For example, HSV offers a relatively ambiguous representation of chroma. In other words, the adopted transformation from



(a) (b)

Figure 5.3.: Training loss (a) and standard deviation of the validation mIoU (b) as a function of training time using the baseline training protocol (Table 5.1). The loss plateaus after the approximately 3500$^{\text{th}}$ training step. The standard deviation was computed across three identical trials using the same random seed. The spike in validation performance is ignored as it was present in only one trial.

RGB to HSV results in objects of the same perceived colour to have approximately the same value in the hue band. In addition, brightness, which the last component of HSV has been successfully used as a feature in multiple OBIA rule-sets, both man- and machine-designed. On the other hand, triangular greenness index (TGI), which has been shown to improve the identification of vegetated regions, may be of particular importance in the context of this thesis given the inherent rarity of the corresponding class both in general and in the training subset, as it could directly provide information on whether plant life is present in a given scene or not. However, it should be noted that, given the sheer number of filters in the model, it is not unreasonable for the model to learn these channels by itself supposing they are indeed beneficial to performance. For instance, TGI is defined as a linear combination of the RGB bands, which can be easily modelled using a single convolutional layer. Furthermore, although the transformation from RGB to HSV is non-linear, it may be able to be approximated using multiple filters. Perhaps, this is the reason why the observed performance improvements, especially in the case of TGI, are not particularly significant given the observed standard deviation of validation mIoU. Since it is not clear whether adjustments to the input data actually helped, further testing was deferred to the first round of the HPO process (Section 5.1.3).

In terms of potential changes to the encoder of the model, it appears that the ResNet-D variant significantly improves predictive performance. This observation is supported by T. He et al., 2018, who specifically designed this variant to ignore as little information from the input feature maps as possible. In the context of this thesis, this tweak proved to be significant, since the corresponding performance improvement was almost six times the larger than the observed standard deviation of

validation mIoU. Assuming performance is normally distributed around the observed mean, this result is highly unlikely to be due to random chance, and so the baseline encoder was swapped for a ResNet-D variant for all three of the following HPO rounds (Sections 5.1.3 to 5.1.5). Other potential improvements to the encoder included the incorporation of either efficient channel attention (ECA) Q. Wang et al., 2019 or squeeze and excitation (SE) (Hu et al., 2017) units. Perhaps unsurprisingly, ECA, a direct contemporary of SE achieved significantly better performance than SE, confirming the experimental findings of Q. Wang et al., 2019. Moreover, the observed performance degradation when using SE was approximately in the bottom 0.5^{th} percentile of validation mIoU under the aforementioned normality assumption. The reason for this deterioration in performance is not immediately clear, but is primarily attributed to the fact that an SE unit generally contains more trainable parameters than its ECA counterpart (Q. Wang et al., 2019), in turn requiring a longer training period to achieve similar results. On the other hand, ECA did not appear to impact performance in a significant way. Hence, SE was eliminated from further consideration and thus the first HPO round included only ECA as a choice of attention mechanism. Finally, anti-aliasing (Zhang, 2019) also resulted in significant performance loss. Once again, the reason for this is not immediately clear .

In terms of potential adjustments to the decoder of the model, the only parameter experimented with was the "base" dilation rate, that is the dilation rate of the first atrous convolution in the ASPP, with the rate of the two other being two and three times that, respectively. This is in line with Chen et al., 2018, who used the $(6, 12, 18)$ as dilation rates. In actuality, none of the DeepLab works explicitly mentions that the dilation rates in the ASPP are correlated. However, to the best of the author's knowledge this implicitly proposed linear relationship had not been contested in relevant literature at the time of writing. At this point it should be noted that changes to other parameters, particularly the output stride, the number of filters after the 1×1 convolution, and the number of feature map reduction channels were not investigated due to the significant computational constraints of this thesis, as it was assumed that each would have to be increased in order to potentially obtain any performance improvement. Similarly to B. J. Kim et al., 2023, who developed an equation for the base dilation rate as function of the width or height of the input image, assuming square images, and observed noticeable but admittedly relatively minor (i.e., in the order of less than 1%) performance improvement after optimizing it (i.e., the base dilation rate) accordingly, it appears that an appropriate base dilation rate may offer minor performance improvements. In particular, performance appeared to initially remain relatively constant as the base dilation rate increased, only showing minor albeit sometimes statistically significant fluctuations, such as in the case of 10. This behaviour continued up to a rate of 15, which resulted in a performance improvement of approximately 3% in comparison to the baseline, with the immediately following rate tested, namely 20, dropping performance once again to 55.66%. For completeness, it is noted that the optimal dilation rate for this particular work according to B. J. Kim et al., 2023 would be 10. All in all, as was the case with potential changes to the input data and the incorporation of ECA units into the encoder, it could not be determined with

relative certainty whether changing the base dilation rate at this point would actually help. However, given the geometric misalignment between the RGB and LiDAR-derived bands of the input data as well as the assumption that neighbouring buildings generally feature similar roofing materials, and could thus provide contextual cues, there was a strong suspicion that the dilation rate should be optimized, particularly enlarged, in order to maximize the receptive field of the model. Therefore, further testing was deferred to the first round of the HPO process (Section 5.1.3).

In terms of potential introduction of regularisation, three different methods were tested, namely: stochastic depth in the encoder of the model, label smoothing in the CE component of the loss function and weight decay in the optimization algorithm. The motivation behind the experimentation with three instead of a single method lies in the fact that each selected technique aims to regularise the model in different ways. In brief, stochastic depth controls the titular property of the encoder, randomly deactivating entire blocks. Amongst other benefits, this most importantly means mitigating potential vanishing gradient issues and motivating each the encoder to optimize each block independently and not potentially use one to correct the mistakes of the other (Hayou & Ayed, 2021). In addition, as stochastic depth is implemented in `timm` using a linear decay schedule, meaning that later blocks increased deactivation probability in comparison to earlier ones. This means that the first and empirically most important blocks for feature extraction are potentially better trained. In contrast, conventional dropout operates on individual neurons after the pooling step, thereby affecting the width of the model. The preference of stochastic depth over dropout can be attributed to the following two reasons. Despite its theoretical benefits preference for stochastic depth over dropout is mainly attributed to Wightman et al., 2021 who successfully used it to improve significantly improve SOTA ResNet performance on the ImageNet dataset, as well as the ConvNeXt (Liu et al., 2022; Woo et al., 2023) architecture, the de facto SOTA ResNet-like model at the time of writing. As expected, the incorporation of increasing amounts of stochastic depth into the decoder generally resulted in apparent performance degradation. However, performance metrics are not appropriate indicators of the effect of this regularisation technique because it directly affects the backpropagation process. A more appropriate metric would perhaps be the validation loss, where all configurations with stochastic depth achieved significantly better performance (Figure 5.4).

However, as no specific value of stochastic depth significantly dominated the others, further testing was deferred to the first round of the HPO process (Section 5.1.3).

Another regularisation technique tested was label smoothing, which is also used by Wightman et al., 2021 and specifically designed to mitigate miscalibration issues (Szegedy et al., 2015). In brief, label smoothing evaluates CE loss using a mixture of the original ground truth labels and a uniform distribution whose intensity is controlled by a single weight. Small values of this weight lead to the resulting distribution approaching the original, whereas a weight of 1 results in all the ground truth labels becoming effectively the same. In addition to its regularising properties, label smoothing is particularly important in the context of this thesis because it implicitly corrects potential annotations issues due to similar material appearance. For instance, there were

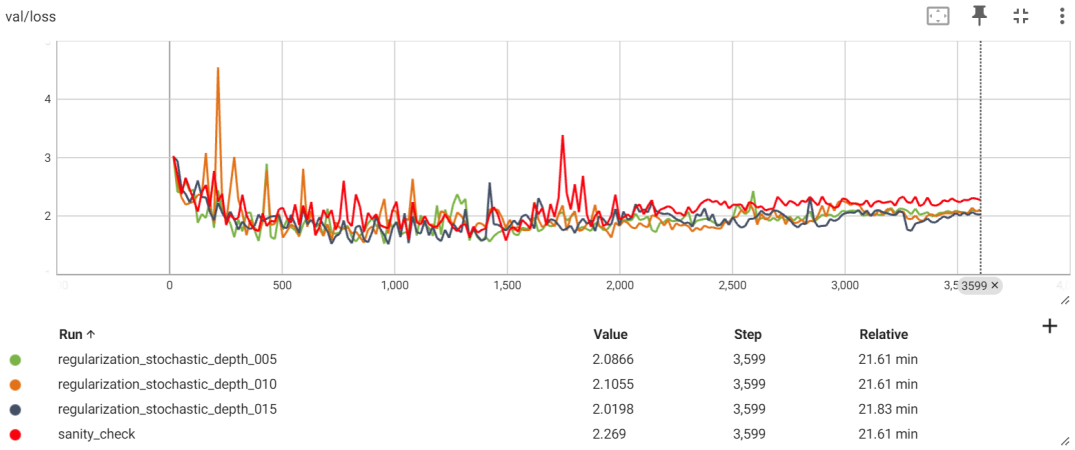
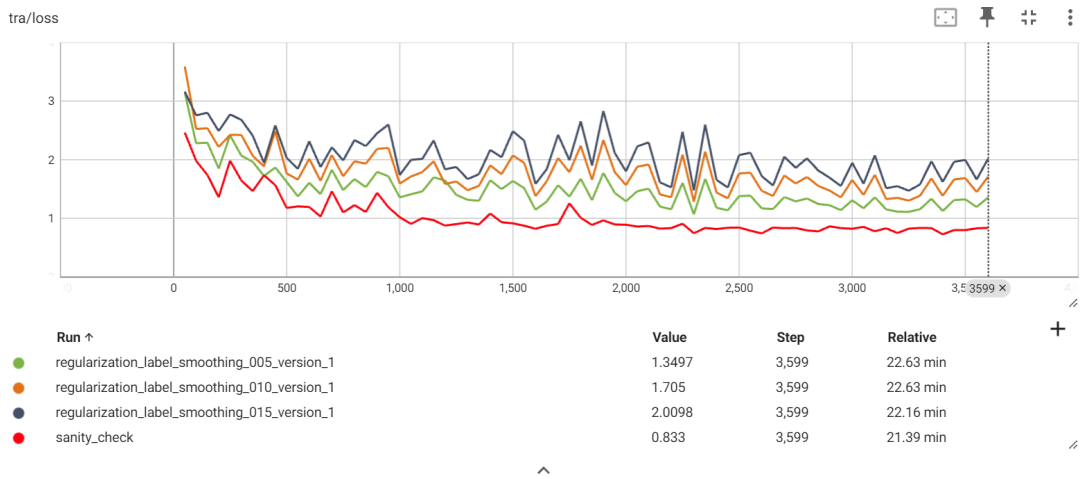


Figure 5.4.: Validation loss as a function of training time for the baseline training protocol and various configurations with different amounts of stochastic depth. All regularised configurations achieve a lower loss score than the base line (sanity check) by the last training step.

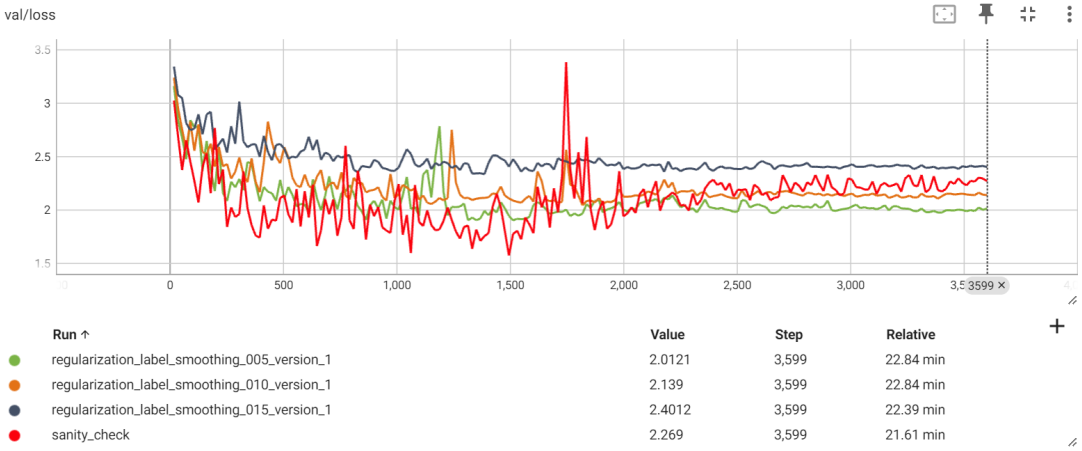
many cases in the reference dataset where light-coloured membranes resembled metal . In fact, the reader is reminded of the fact that this particular issue has been reported by multiple other authors . Supposing that such mistakes exist in the training subset, which is very likely as thorough label review was not conducted due to the significant time constraints of this thesis, it may very well be the case that since the human annotator mistook one material for the other, the model is bound to do the same, especially since there is now no way to tell that it is wrong. However, the assumption is that this confusion will be relatively minor and that the difference between the log-probabilities of the wrong and the correct material will be "smoothed out" or even overturned with little amounts of label smoothing. However, as previously mentioned, the smoothing weight would need to be tuned carefully, because many materials are generally alike, and thus the differences between them are oftentimes minute, and hence very easy to disturb. Indeed, this hypothesis appears to be true, as label smoothing with a weight of 0.1 resulted a statistically significant performance boost, especially in comparison to other weight values tested, all of which degraded performance. Nevertheless, as was the case with stochastic depth, the main effect of label smoothing should be evident in the validation loss curve, were miscalibration was expected to have decreased significantly in comparison to the baseline, as was actually the case (Figure 5.5).

This considered a label smoothing weight of 0.1 was adopted for all three of the following HPO rounds (Sections 5.1.3 to 5.1.5).

This considered, an improved baseline (mIoU: 64.41% with std. 3.525%) is introduced in Table 5.2. All future experiments are conducted with respect to it.



(a) Training loss.



(b) Validation loss.

Figure 5.5.: Training (a) and validation (b) loss as a function of training time for the baseline training protocol and various configurations with different amounts of label smoothing. The baseline configuration (sanity check) fits the training set well but displays significant miscalibration. On the other hand, label smoothing results in convergence to a higher loss score, but solves the miscalibration issue.

Table 5.2.: Improved baseline training protocol. Any parameter not explicitly defined assumes its default value as per PyTorch v2.2.2.

Category	Parameter	Value
Input Data	Append HSV	-
	Append TGI	-
Encoder	Variant	ResNet-18-D
	Attention Block	-
	Anti-aliasing Block	-
Decoder	Base Dilation Rate	6
	Stochastic Depth	-
	Label Smoothing	0.1
	Weight Decay	-
Optimisation	Optimizer	Adam
	Learning Rate Annealing	Polynomial
	Learning Rate	0.001
	Warmup Length	-

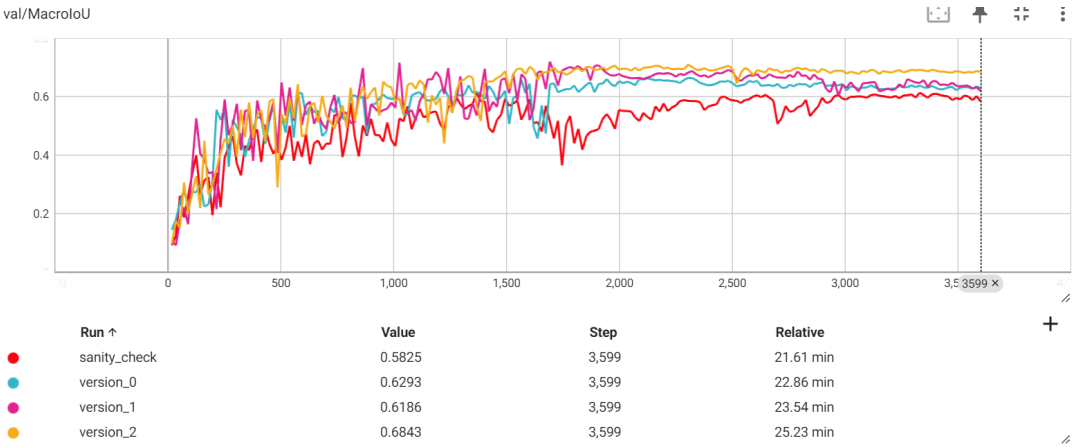
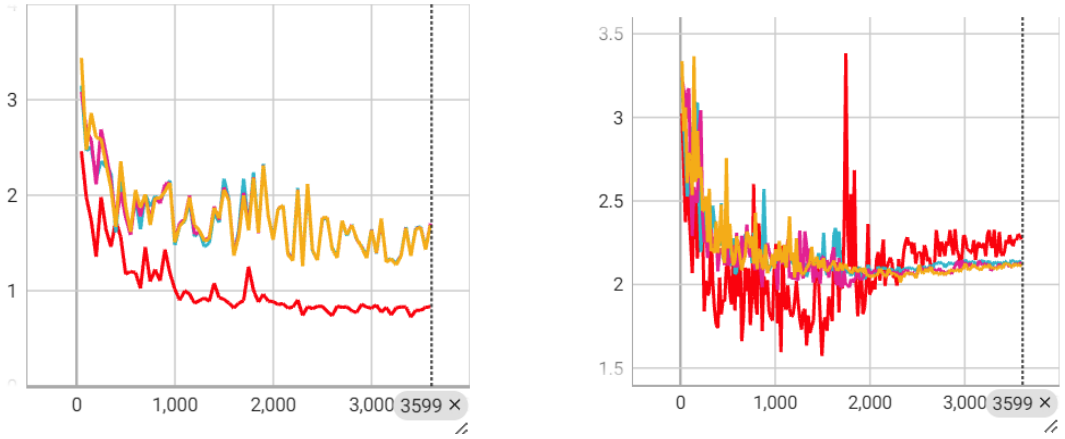


Figure 5.6.: Improved baseline performance as a function of training time. The baseline is marked as sanity check.



(a) (b)

Figure 5.7.: Training (a) and validation (b) loss as a function of training time using the improved baseline training protocol (Table 5.2).

5.1.3. Round 1

According the findings from the manual experimentation phase, the HPO process continued with an automated search for good configurations of the now-improved baseline model. As previously mentioned, this process was divided into three rounds with the hope of better allocating the limited computational resources to the overall search space. In particular, each round examined a subspace of the original search space, with the variable parameters in each case being closely related.

This considered the search space for the first automated round is presented in Table 4.7.

The architectural hyperparameters are related to the input data, encoder, and decoder, while regularisation parameters and the learning rate are considered nuisance parameters according to Godbole et al., 2023, and are optimised simply to make model comparisons fair.

As mentioned in , the first round consisted of 50 trials where each configuration was sampled from a uniform mixture model.

As illustrated in Figure A.1, the majority of the sampled configurations attained a validation mIoU between 47% and 57%, with five notable exceptions falling below 45%. Each of these trials applied a weight decay of at least 0.005, which will later be demonstrated as excessive, and used a learning rate ranging from 0.0035 to 0.005. The only other relatively consistent aspect was that all but one of these trials did not incorporate attention mechanisms; however, based on the estimated parameter importances (Figure A.4), this likely did not significantly impact their performance. Conversely, the top two trials, which nearly matched the performance of the enhanced baseline, utilized weight decays of 0 and 0.004, respectively, along with learning rates of 0.001 (i.e., the baseline) and 0.0005. The second-best trial's combination of a learning rate and weight decay is particularly noteworthy since its learning rate was an order of magnitude smaller than that in the poorest-performing trials, which nonetheless had similar weight decay values. In addition, none of the two top trials used any additional input bands, whereas both featured ECA and a base dilation rate of 1.

In general, Figure A.1 shows that all possible values of each variable parameter were sampled, while the slice plots of the continuous variables (Figure A.2) feature a relatively even spread, with the optimum value in the case of non-categorical parameters lying away from the edges of the corresponding search space, as recommended by Godbole et al., 2023.

Therefore, it is assumed that the search space was adequately sampled, both in terms of quantity of trials and degree of exploration. This is further corroborated by the expected improvement plot (Figure A.3), which showed an expected improvement of less than 5% at the end of the 50^{th} trial, meaning that more than 95% of potential improvements had probably already been observed, at least based on the observed trials.

Another interesting topic which arises from experimenting with multiple configurations is that of parameter importance estimations, that is the estimation of the degree to which each tunable parameter affects performance. In the context of this thesis,

parameter importances are computed according to Hutter et al., 2014 In the case of this round, the estimated parameter importances are presented in Figure A.4.

It is not surprising that the learning rate is the most crucial parameter, contributing around 80% to overall performance. Indeed, both Goodfellow et al., 2016 and Godbole et al., 2023 emphasize the significance of a well-calibrated learning rate. The significance of this parameter becomes even more pronounced when examining the third crucial hyperparameter, weight decay, which almost matches the base dilation rate in importance, each holding a value of approximately 8%. As discussed earlier (Section 5.1.2), the learning rate and weight decay are interdependent when using the Adam optimizer, meaning adjustments to one influence the other. Similarly, the significance of the dilation rate is also to be expected for the reasons mentioned in Section 5.1.2, particularly considering the lack of solid evidence and theoretical intuition to validate the claimed advantages of three of the remaining parameters, namely the two modifications to the input data and ECA, within the scope of this thesis. Consequently, it stands to reason that stochastic depth follows, assigned a mean importance level of 2.393%, with HSV, TGI, and ECA trailing behind, holding a combined importance of less than 5%. Notably, HSV might be considered more significant than TGI due to its complexity in modelling with convolutions, as mentioned in Section 5.1.2.

Finally, the loss and performance curves of the best configuration (Table 5.3) from this round are presented in Figures 5.8 and 5.9. After conducting two additional tests on the optimal configuration using the same seed, resulting in a total of three trials, the mean validation mIoU was recorded at 64.8%, with a standard deviation of 1.497%.

Table 5.3.: Best training protocol discovered in the first round of the HPO process. Any parameter not explicitly defined assumes its default value as per PyTorch v2.2.2.

Category	Parameter	Value
Input Data	Append HSV	-
	Append TGI	-
Encoder	Variant	ResNet-18-D
	Attention Block	ECA
	Anti-aliasing Block	-
Decoder	Base Dilation Rate	1
	Stochastic Depth	0.03
	Label Smoothing	0.1
	Weight Decay	-
Optimisation	Optimizer	Adam
	Learning Rate Annealing	Polynomial
	Learning Rate	0.001
	Warmup Length	-

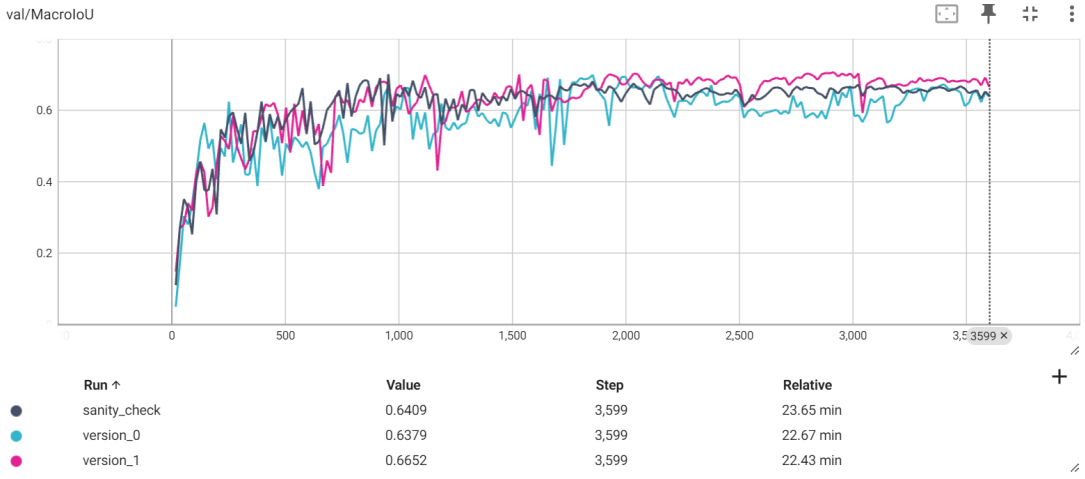
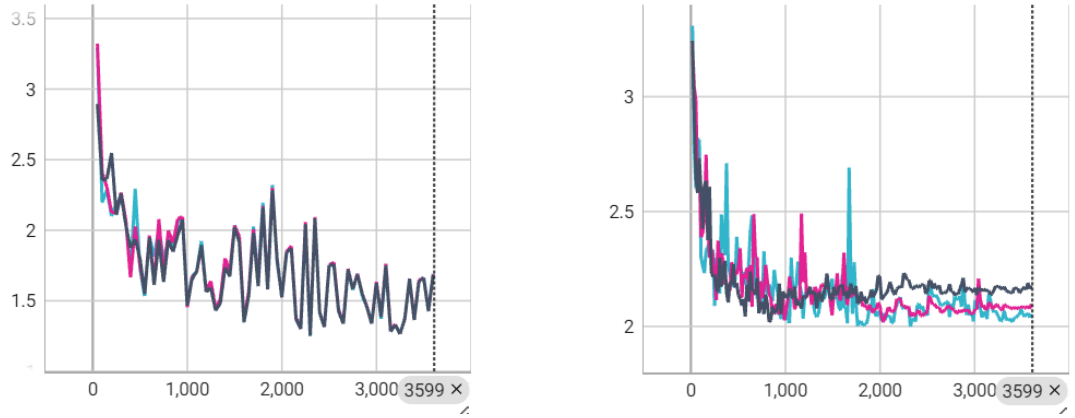


Figure 5.8.: Improved baseline performance as a function of training time. The baseline is marked as sanity check.



(a) (b)

Figure 5.9.: Training (a) and validation (b) loss as a function of training time using the best training protocol discovered in the first round of the HPO process (Table 5.3).

5.1.4. Round 2

As the first round of the HPO process (Section 5.1.3) was designed to optimize the input data and model hyperparameters, it is only logical that the second round covered the remaining parameters investigated in the manual experimentation phase (Section 5.1.2), that is those which were related to the optimization of the loss function, namely: the pertinent algorithm as well as the learning rate annealing strategy and warmup length. Maintaining the terminology used by Godbole et al., 2023, these parameters were architectural, with the nuisance parameters being stochastic depth,

weight decay, and learning rate, in a similar fashion to the first round. This considered, the search space for the second round is presented in Table 4.8.

The search space of the nuisance parameters was not adjusted from the first to the second round due to the fact that one was effectively a direct continuation of the other, that is they belonged to the same overall study, and so changing the search space of revisited parameters would invalidate any comparisons between rounds. In addition, search space for weight decay had to explicitly remain the same because any value determined to be quasi-optimal would most likely not apply to this round as the optimizer is now a tunable hyperparameter and, as previously mentioned , Adam and AdamW handle weight decay differently.

Similarly to the first round, this one consisted of 50 random trials featuring configurations sampled from a uniform mixture model. The architectural parameters optimized in the first round assumed their quasi-optimal values.

As illustrated in Figure A.1, the majority of the sampled configuration achieved a validation mIoU between 50% and 70%. The mean of this range is higher than it was during the first round, meaning that this round revealed better configurations than the first one on average. However, the range is also significantly wider than what it was in the first round. This implies that configuration changes were overall more impactful in this round than they were in the first one. This phenomenon is primarily attributed to the concurrent tuning of both the optimisation algorithm and weight decay, as mentioned in the beginning of this section. This assumption is corroborated by the fact that the trials in this performance region are clearly separated into two subgroups, those with a performance between 50% and 60%, and those with a performance between 60% and 70%. In fact, 13 out of the 14 configurations in this region featured Adam with a non-zero weight decay, with the 14^{th}, which was also the locally best-performing one employing AdamW with a weight decay of 0.007, the fourth highest possible in the whole round. Hence, it is clear that Adam does not work well with weight decay, as shown also in the first round, whereas AdamW benefits from it. Indeed, all but three out of the total 17 configurations with performance above 65% in the second group used AdamW, with two out of the three with Adam having a weight decay of zero. The only trial with Adam and non-zero weight decay (0.002) scored 65.72%, which was the second worst performance in the group. Furthermore, the round featured two trials which fell below the 50% threshold, with subsequent inspection revealing that they both had the same learning rate of 0.0045 paired with Adam and a weight decay of 0.006 and 0.007, respectively. Finally, the two best-performing trials, which were only approximately 0.8% apart in terms of performance, both used cosine learning rate annealing, which is actually proposed by Loshchilov and Hutter, 2017; Wightman et al., 2021, as well as a very similar learning rate of 0.0045 and 0.005, with the smaller of the two values corresponding to the best trial. Moreover, both trials had similar stochastic depth of 0.08 (best) and 0.1, as well as warmup length of 125 (best) and 100 epochs. However, the optimizer and weight decay choices interestingly differed, with the best-performing trial using AdamW and a weight decay of 0.007, and the second-best performing trial Adam and 0.004, respectively. With the second-best performing trial using Adam and a non-zero weight decay having almost

5% worse, it is not immediately clear why this configuration was so good. The only assumption is that this is either a statistical outlier or that the relatively low learning rate in comparison with the long warmup period reduced the effects of weight decay.

In general, Figure A.5 shows that all possible values of each variable parameter were sampled, while the slice plots of the continuous variables (Figure A.6) feature a relatively even spread, with the optimal value in the case of non-categorical hyperparameters lying away from the edges of the corresponding subspace, as recommended by Godbole et al., 2023.

Therefore, it is assumed that the search space was adequately sampled, both in terms of quantity of trials and degree of exploration. This is further corroborated by the expected improvement plot (Figure A.7), which showed an expected improvement of less than 5% at the end of the 50^{th} trial, meaning that more than 95% of potential improvements had probably already been observed, at least based on the observed trials.

Another interesting topic which arises from experimenting with multiple configurations is that of parameter importance estimations, that is the estimation of the degree to which each tunable parameter affects performance. In the context of this thesis, parameter importances are computed according to Hutter et al., 2014 In the case of this round, the estimated parameter importances are presented in Figure A.8.

Given the previous analysis regarding the explored optimizers and weight decay, it is not surprising that they are by far the most important parameters, so much so that the importance level of the learning rate, the traditionally most important hyperparameter in DL (Godbole et al., 2023; Goodfellow et al., 2016) was assessed at less than 5%. Interestingly, the optimizer was found to be significantly more important than the weight decay, and not the other way around, despite the fact that the update rules of Adam and AdamW are otherwise identical. The reason for this is most likely the fact that the same internal variable was used for weight decay regardless of optimizer, and so from that point of view, that is when keeping the weight decay effectively "constant" the parameter that makes or breaks performance is indeed the optimizer. Similarly, to the first round, stochastic depth was found to not be particularly important, most likely for the same reasons. Finally, the warmup length and annealing were found to be the two least important parameters, although the confidence bounds regarding annealing are relatively wide. Once again, this is expected, as both parameters are designed to provide stability during training, not explicitly improve performance; a seemingly stable model is not necessarily a good performer as shown in the ablation study (Section 5.4).

Finally, the loss and performance curves of the best configuration (Table 5.4) from this round are presented in Figures 5.10 and 5.11. At this point it should be noted that an unrelated data management issue resulted in the complete and irrecoverable loss of the loss logs of the original trial. Therefore, three instead of the initially planned two additional testes were conducted on the optimal configuration in order to fairly determine its confidence bounds. All trials were conducted using the same configuration and random seed. Therefore, Figures 5.10 and 5.11 do not actually include the best trial, only its three recreations. This considered, the mean validation

mIoU was recorded at 68.76%, with a standard deviation of 1.098%.

Table 5.4.: Best training protocol discovered in the second round of the HPO process. Any parameter not explicitly defined assumes its default value as per PyTorch v2.2.2.

Category	Parameter	Value
Input Data	Append HSV	-
	Append TGI	-
Encoder	Variant	ResNet-18-D
	Attention Block	ECA
	Anti-aliasing Block	-
Decoder	Base Dilation Rate	1
	Stochastic Depth	0.08
	Label Smoothing	0.1
	Weight Decay	0.007
Optimisation	Optimizer	AdamW
	Learning Rate Annealing	Cosine
	Learning Rate	0.0045
	Warmup Length	125

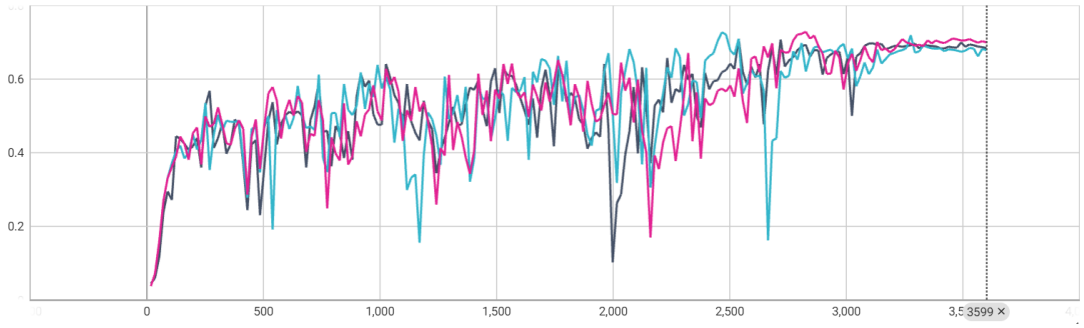
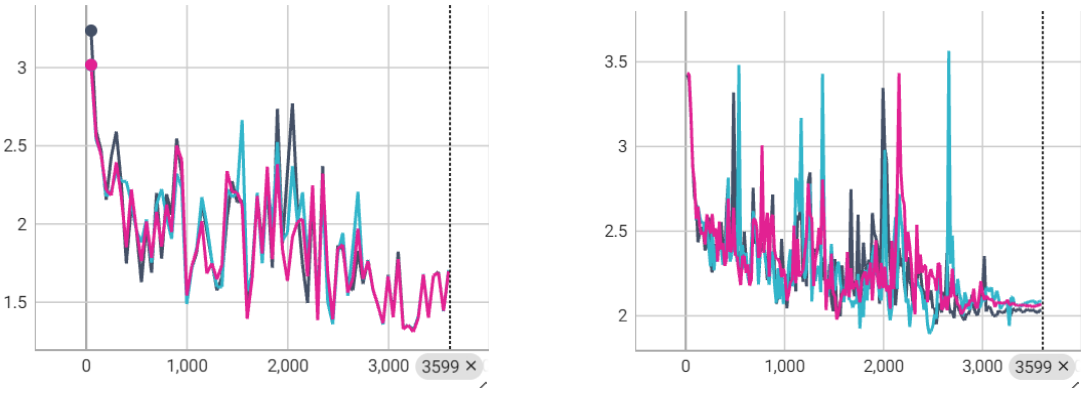


Figure 5.10.: round 2 performance as a function of training time.



(a) (b)

Figure 5.11.: Training (a) and validation (b) loss as a function of training time using the best training protocol discovered in the second round of the HPO process (Table 5.4).

5.1.5. Round 3

In the context of this thesis, this phase entails the greedy optimisation of the non-categorical hyperparameters involved in the previous HPO rounds. Although all parameters could technically be re-optimised for potentially optimal results, it should be noted that all categorical hyperparameters involved in the study only had two potential values to choose from, and so it is assumed that their search subspace is so small that random search probably already optimized them anyway, and therefore they remain fixed. In addition, with the exception of the optimizer, none of these parameters were found to be particularly important (Figures A.4 and A.8), so including them in the optimisation again when the search space is no longer discretised and thus immensely larger could potentially defeat the purpose of the third round, which is strictly fine tuning the existing quasi-optimal configuration. Furthermore, regarding the optimizer, it remains fixed as it was found to better handle weight decay, which was determined to actually be very important in the second round. After all, Wightman et al., 2021 recommends AdamW instead of Adam anyway.

This considered, the search space for the third round is presented in Table 5.5.

Table 5.5.: Search space for the third automated round of the HPO process.

Category	Parameter	Value
Decoder	Base Atrous Rate	[6..20]
Regularisation	Stochastic Depth	[0,0.1]
	Weight Decay	[0,0.01]
Optimisation	Learning Rate	[5e-4,5e-3]
	Warmup Length	[50..150]

Given that TPE necessitates a certain number of preliminary trials, in this case ten, in order to function effectively and ensure the statistical reliability of the underlying GMMs, it was determined that the search space would remain unchanged for this round. This decision was made to facilitate sampling across the entire range of possible parameter values, thereby addressing and reducing any potential biases introduced in earlier rounds. After all, since TPE is an adaptive algorithm, the search space it evaluated would get progressively narrower as the round progressed, because performance variations are usually predictable; therefore, it being initially broad was not considered to be a significant issue.

However, it should be noted that the minimum bound of the subspace corresponding to the base dilation rate was actually adjusted from one to six, despite the fact that the best performing configuration in the first round featured the former value. In actuality, this change was an oversight and was made by mistake. However, it fortunately did not appear to affect the final results as clearly shows that the optimum dilation rate was actually 19 and no configuration with a rate below 12 managed to score a validation mIoU of more than 70%. In addition, this result is actually in line with the hypothesis that a larger base dilation rate, and hence a larger receptive field is beneficial for the model because input scenes may contain multiple neighbouring buildings which are likely to share the same or similar roofing materials. Therefore, a large receptive field would likely provide additional context. Nevertheless, supposing that this is in fact true, then the question arises of why a unit base dilation rate performed so well in the first round. The reason for this is not entirely clear, but clearly shows that the dilations rates of 6 and 16 resulted in equivalent performance, which is of course somewhat contradictory. Therefore, the author's assumption on the matter is that the neighbouring building hypothesis does not always hold; sometimes neighbouring buildings do provide correct context, hence why a large base dilation rate would help, but other times (e.g., when buildings of different primary use are located relatively close to each other), it causes confusion, making a smaller dilation rate seem more attractive.

In any case, as was the case with the previous rounds, this his one consisted of 50 trials.

As shown in Figure A.9, most sampled configurations achieved a validation mIoU between 65% and 70%. The mean value in this range is higher compared to the previous round, indicating an overall improvement. Additionally, this range was the smallest among all rounds, implying that the HPO process may be nearing convergence or has already converged. Although this outcome aligns with the purpose and design of this round, it is important to note that a significant number of configurations performed noticeably worse than the mean performance of the best configuration in the second round (i.e., 68.76%). Specifically, excluding the initial 10 trials that were randomly sampled using a uniform mixture model, 18 out of the remaining 40 trials did not surpass this threshold, with 7 performing worse than the first round (i.e., 64.8%), and failing to even outperform the improved baseline (). While this is not particularly concerning as only 8 and 4 of these trials occurred after the 30^{th} and 40^{th} overall trials, respectively, suggesting that the underlying sampling algorithm was gaining

confidence, it is worth mentioning that the overall process might have benefited from a larger trial budget, especially in the preliminary stage, as many subsequent trials were arguably "wasted" on poor configurations. Alternatively, the search space could have been narrowed, or pruning could have been employed. However, pruning was deliberately avoided due to significant step-to-step variance during validation, which could prematurely terminate otherwise promising trials, mistakenly marking their configurations as suboptimal due to an "unlucky" step. Finally it is worth mentioning that the only significant difference between the two best-performing configurations, which were otherwise more than 2% apart, was the value of stochastic depth and weight decay. In particular, the best trial featured a stochastic depth and weight decay of approximately 0.05190 and 0.00134, respectively, whereas the corresponding values in the second best trial were 0.04125 and 0.003472, correspondingly. This indicates that performance is highly dependent on seemingly minute parameter changes, which most likely require continuous, targeted sampling to effectively capture, in turn justifying the third round.

In general, both Figure A.9 and Figure A.10 show that the region around the best configuration was densely sampled, of course with the exception of certain outliers belonging to either the preliminary phase or bad configurations, as previously mentioned.

The hypothesis that the optimal configuration was adequately sampled is further corroborated by the expected improvement plot (Figure A.11), which showed an expected improvement of less than 5% at the end of the 50^{th} trial, meaning that more than 95% of potential improvements had probably already been observed, at least based on the observed trials.

At this point it should be noted that in contrast to the previous two rounds, it is not relevant to discuss parameter importances because the search space was not uniformly sampled, and thus any relevant analysis would concern only the optimal region. In any case, the curious reader is informed that stochastic depth was the most important parameter with and importance level of 34.55%, followed by the learning rate (24.57%), weight decay (15.10%), warmup length (15%), and the base dilation rate (10.78%).

Finally, the , the loss and performance curves of the best configuration Table 5.6 from this round are presented in Figures 5.12 and 5.13. After conducting two additional tests on the optimal configuration using the same seed, resulting in a total of three trials, the mean validation mIoU was recorded at 70.46%, with a standard deviation of 4.054%. Overall, the HPO process resulting in a performance improvement of approximately 10% and 6% over the initial and improved baseline, respectively, while completely solving the observed miscalibration issues. The only slightly worrying thing about this configuration is that the loss/performance spikes reported in are still present and that the observed performance standard deviation is almost 3 times larger than what it was in the beginning of HPO, signifying that the model may have lost trial-to-trial stability.

Table 5.6.: Best training protocol discovered in the third round of the HPO process. Any parameter not explicitly defined assumes its default value as per PyTorch v2.2.2.

Category	Parameter	Value
Input Data	Append HSV	-
	Append TGI	-
Encoder	Variant	ResNet-18-D
	Attention Block	ECA
	Anti-aliasing Block	-
Decoder	Base Dilation Rate	19
	Stochastic Depth	0.0519
	Label Smoothing	0.1
	Weight Decay	0.00131376
Optimisation	Optimizer	AdamW
	Learning Rate Annealing	Cosine
	Learning Rate	0.00390561
	Warmup Length	135

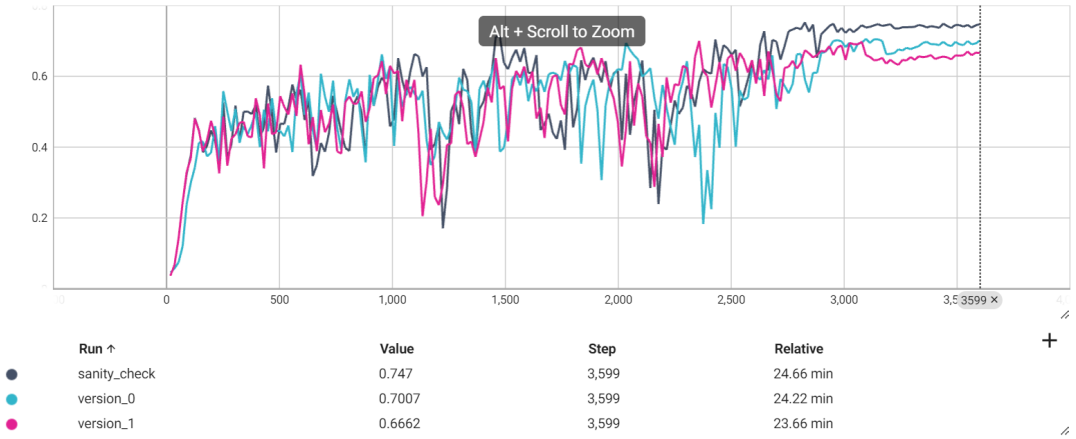
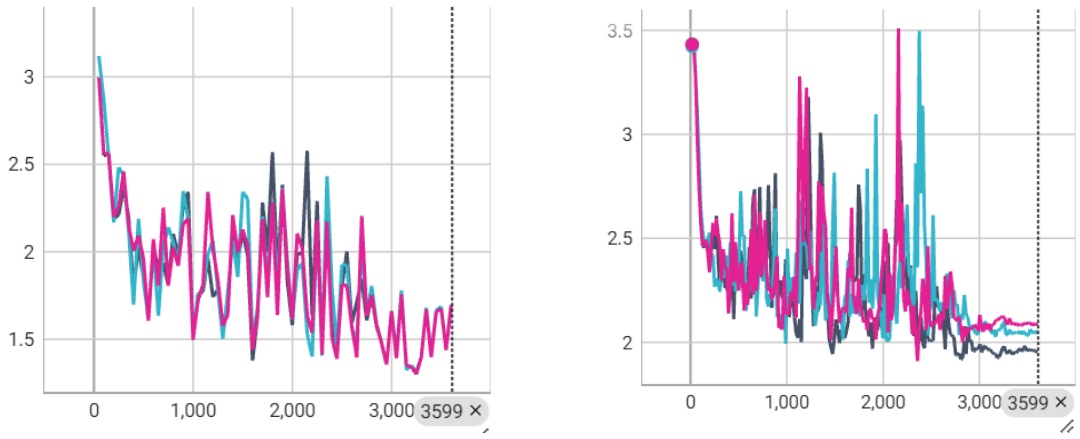


Figure 5.12.: round 3 performance as a function of training time. The baseline is marked as sanity check



(a) (b)

Figure 5.13.: Training (a) and validation (b) loss as a function of training time using the best training protocol discovered in the second round of the HPO process (Table 5.6).

5.2. Pixel-based Performance Evaluation

5.2.1. Quantitative Evaluation

The test performance of the best configuration is presented in Figure 5.14 and Table 5.7.

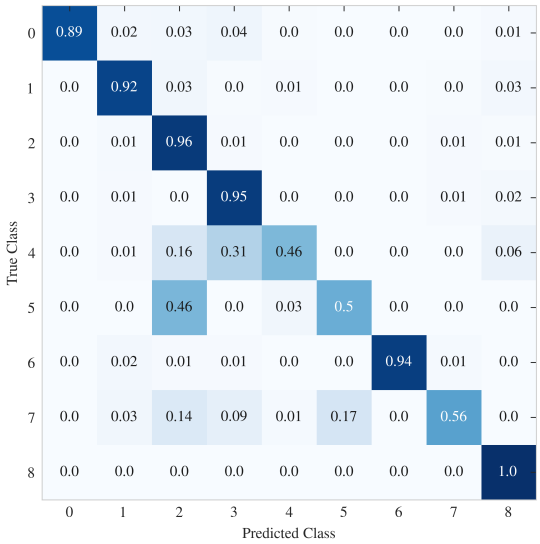


Figure 5.14.: Pixel-level confusion matrix of the best model configuration discovered during the HPO process, evaluated on the test subset. Each row is normalised by the corresponding class support. The material corresponding to each class label is given in Table 4.1.

In general, performance is more or less acceptable given the aforementioned data constraints. In particular, when the model is viewed strictly as a classifier, the OA of 85.15% competes with most works covered in (), even surpassing many of them, and most importantly Wyard et al., 2023 (81%), whose study featured many of the same classes as this thesis. However, given that the test subset is heavily imbalanced (), OA, as well as any other microscopically averaged performance metric may not be a reliable performance indicator for the reasons mentioned in . According to the confusion matrix (Figure 5.14) the average recall ignoring the background is 78.63%, whereas the average precision is 77.55% (Table 5.7). In any case, it can overall be said that the model makes an approximately equal number of Type I and II errors on average, with approximately 24% of predictions of any given class being incorrect and 21% of the ground truth labels having been positively identified. Finally, when viewed from a segmentation point of view, the mIoU of 64.68% is definitely decent for a proof of concept, but admittedly leaves a lot to be desired.

In terms of class-wise performance, Figure 5.14 and Table 5.7 reveals that the classes under consideration may effectively be divided into three distinct groups based on the corresponding classification and segmentation performance. In particular, first group, which refers to the classes where the model achieved outstanding performance in both

Table 5.7.: Pixel-level performance scores of the best model configuration discovered during the HPO process, evaluated on the test subset. The material corresponding to each class label is given in Table 4.1.

Class	Test Performance		
	Precision	F_1 Score	IoU
1	0.9570	0.9372	0.8819
2	0.7829	0.8618	0.7571
3342123 3	0.9487	0.9505	0.9056
4	0.5908	0.5141	0.3460
5	0.6020	0.5480	0.3774
6	0.9967	0.9684	0.9387
7	0.8745	0.6811	0.5164
8	0.4511	0.6218	0.4511
OA	0.8793		
mIoU	0.6468		

classification and segmentation, are dark- and light-coloured membranes and gravel. In actuality, with the exception of IoU in dark-coloured membranes, all other relevant performance metrics are above the 90% threshold. This result is not only in line with Wyard et al., 2023 but also SOTA to the best of the author's knowledge. The reason for this phenomenon is attributed to the generally characteristic appearance of dark-coloured membranes and gravel, in combination with the geometric properties of the roof systems they are typically used in, in particular their slope, which is assumed to have facilitated the discrimination of light-coloured membranes and metal, a common source of confusion in both the context of this thesis and the relevant literature. Furthermore, it should be noted that flat roofs, which are almost exclusively where these materials are encountered, were reported by Fiumi et al., 2014 to be easier to model than pitched ones.

The second class group refers to classes where the model performed acceptably overall, but had significant room for improvement in either classification or segmentation. These classes are ceramic tiles, solar panels, and vegetation. In particular, both tiles and vegetation achieved impressive recall rates (Figure 5.14), with the latter class even achieving an initially seemingly perfect 100% score. This means that the model was able to detect them very reliably in the test set, that is most relevant ground truth labels were correctly identified. However, it is evident that the model, albeit very accurate, was not very precise in its predictions (Table 5.7). For instance, almost half of all metal pixels in the test set were incorrectly identified as belonging to the tile class (Figure 5.14). In addition, the model significantly confused light-permitting surfaces and solar panels for this class. However, the precision score of 78.29% (Table 5.7) is still reasonable. Therefore, it is likely that the model encountered difficulties in distinguishing some instances of metal and tiles, which might have caused it to adopt a

more inclusive approach in its predictions to enhance overall performance. Moreover, the observed confusion can be partly explained by the fact that residential solar panels and skylights are frequently much smaller than tiles, which serve as a primary roofing material and thus most likely tend to dominate relevant feature maps, in turn making the model either eager to ignore them or in the worst case simply oblivious to them. Of course, making things even worse, as mentioned in the beginning of this section, is the fact that these feature maps are not particularly well-suited for reliable detection of small objects due to the selected output stride. In any case, a characteristic failure case concerning this issue is presented in Figure B.1.

Finally, in the case of metal, a particularly interesting failure case is that of dirty dark metal, which admittedly looks like dark ceramic tile (Figure B.2). This scene actually constituted a significant proportion of all metal instances in the test set, being only one of two images where it was present, hence the significant number of false positive tile predictions in this regard.

Perhaps more worryingly than tiles, the model is apparently too eager to classify pixels as vegetation, as revealed by the disappointingly low precision score of the corresponding class (Table 5.7). At this point it should be noted that, although the model did have 100% accuracy in this class, later inspection revealed that only one test image contained it (Figure B.3), and therefore this metric is not a reliable performance indicator in this case.

The model often seemed perplexed when distinguishing real vegetation from background elements like trees or related materials, such as gravel (Figure B.4a). Indeed, certain failure cases (Figure B.4b) made it unclear whether the model was genuinely confused or, at its worst, merely hallucinating due to incomplete training or a loss of context.

Finally, the model had issues detecting solar panels (Figure B.5) due to their small size and similar appearance to their neighbouring materials.

Finally, the last class group refers to the classes where the model did not perform well in any regard, namely light-permitting surfaces and metal. For the former category, previously noted issues of potential confusion and omissions, mainly due to the typically small physical dimensions of the corresponding objects relative to those of their surroundings, as well as the chosen output stride, are now verified. In particular, the observed confusion with tiles is explained above, whereas the significant confusion with gravel is due to the failure case presented in Figure B.6.

This case also contributes substantially to the number of false positive tile predictions. This is particularly noteworthy because there are no tiles present in the entire scene, suggesting that the model has strongly linked light-permitting surfaces with tiles due to the fact that many examples of the former class in the reference dataset were actually residential skylights.

Finally, the only two scenes where metal was present are shown in Figures B.2 and B.5a.

5.2.2. Qualitative Evaluation

The predictions of the best model configuration discovered during the HPO process (Section 5.1) on the qualitative performance evaluation tile are presented in Figure C.6.

The results for the first performance check region are presented in Figure C.7.

As mentioned in , this building is of particular interest because the sloped segments of its roof feature asphalt shingles, which the model has not been explicitly learned. Hence, assuming that the model is well-trained, the expectation is that shingle pixels will be assigned to the closest valid class in terms of appearance and spectral signature, that is either dark-coloured membranes due to their general appearance or ceramic tiles due to their overlapping texture. Indeed, Figure C.7b shows that most shingles were classified as tiles, with the exception of mostly north-west- and south-west-facing segments, which were "mistaken" for vegetation. The reason for this "error" is not clear. Although, further inspection revealed that almost no LiDAR data was available directly for these regions (i.e., the corresponding pixels in the density band were or near zero), most likely due to obstruction issues caused by the particular flight characteristics, the introduced interpolation noise did not cause major artifacts except from the slope channel. Hence, it could simply be the case that the model hallucinated, as previously mentioned in Section 5.2.1. Apart from that, the model exhibited relatively good performance, recognising almost all instances of dark-coloured membranes and even correctly ignoring most contextually irrelevant objects. However, it failed to recognize the zinc tower tops.

The next check region contains the main campus buildings (Figure C.8).

The performance of the model is once again generally acceptable, with no significant gross errors apart from certain cases with particularly confusing roof shapes. Looking at Figure C.8b from top to bottom, it is evident that the model has practically no issues modelling large flat roofs, with most materials up to the level of the conference centre being correctly classified. The open parking space in the north-west corner is of particular importance because it has a concrete or asphalt-paved surface to allow for traffic, but the model is unaware of either material. However, as was the case with BK it appropriately handled the case, modelling the roof with dark-membrane. Nevertheless, like Figure B.6 it once again failed to detect the atrium. In addition, the model appears to have issues correctly modelling small roofs, showing significant hallucinations, perhaps due to the high dilation rate and output stride, which resulted in it requiring a large context window and material uniformity in order to make a reliable prediction. Furthermore, in contrast to BK, the model did not manage to ignore clutter and oftentimes incorrectly classified HVAC systems and similar equipment as various materials, particularly light-permitting surfaces, perhaps due to their increased brightness from reflecting sunlight, and rarely gravel, most likely due to the dull gray colour of those which are not particularly reflective. Although these errors are visible in the TPE building as well as the one to its right, the most important such error is in the IDE building, where the model was confused by almost all exhausts while also failing to detect the base material (i.e., gravel) of the majority of the roof surface. However, the model performed almost perfectly in the conference

centre, which has a very peculiar shape with alternating valleys and ridges, as well as the library, where it even managed to detect the cone almost perfectly, although it confused its actual material (i.e., metal) with light-coloured membrane and solar panels.. Most other buildings are either correctly modelled or feature a combination of the aforementioned errors.

The only remaining buildings of interest are those marked in cyan, magenta, yellow, and black in the bottom of Figure C.8b. The first building features a domed metal roof on the main building and an adjoining extension on the right with a flat, gravel-covered roof. Upon further analysis, it was found that the reflectance of the metal portion was comparable to that of tiled roofs, likely due to peculiarities in its shape combined with specific flight characteristics in the observed scene. Consequently, because the roof has an apparent texture more characteristic of tiles rather than typical corrugated metal, it is understandable that the model could have misidentified it as a tiled roof. Similarly, the gravel roof displays green areas along its edges and patches of high reflectance, indicative of low vegetation such as grass. Thus, the predictions of vegetation are attributed to plant growth on the roof, fostered by pollination and frequent rain.

What is more, the predictions for the tall segment of EWI1 feature considerable ambiguity; nonetheless, the structure is visibly tilted in the RGB bands. Therefore, most predictions pertain to its facade, which, naturally, falls outside the model's intended scope. Moreover, the model appears significantly susceptible to significant shadows. Although this issue was noticed in the test subset and several authors) in the literature review have also highlighted it, it is should be noted that four out of the total seven bands of the reference dataset (i.e., the LiDAR component) are usually not affected by this issue. This observation may suggest an excessive dependence on the RGB component of the input data.

Finally, the lower segment of EWI1 as well as EWI2 are both heavily misclassified. On the one hand, EWI1, which actually has a membrane roof featuring several pyramidal, metal elements with south-facing solar panels, was predicted by the model to have a tiled roof. In addition, most solar panels were misidentified as light-permitting surfaces. However, further analysis of the corresponding input data did not reveal obvious problems and therefore the cause of this confusion remains uncertain.

The third performance check region is the industrial area to the west of the campus (Figure C.9).

Given that this region contains mostly buildings with flat roofs featuring materials where the model performed the best in the test set, it is not surprising that performance is also very promising here. However, some of the various confusion and omission issues previously mentioned are still present. For instance, in the top north-west building the model confused metal for light-coloured membrane and light-permitting surfaces. Similarly, the buildings in the south-east corner of the region which are assigned light-coloured membrane actually all have metal roofs. In addition, the model continues to occasionally hallucinate vegetation and classify HVAC systems and other bright objects as light-permitting surfaces. Finally, the model displayed significant confusion in the bottom building of the two in the north-east corner of the region, particularly regarding its skylights, which were misclassified as solar panels. In addition, the

majority of the right segment of the building was plagued by hallucinations. Although an exact reason for this phenomenon is not clear, it should be noted that the solar panels are not present in the LiDAR bands of the corresponding input data, which could explain the skylight issue.

The final performance check region is the residential area in the northeast corner of the tile (Figure C.10).

This region consistently showed similar positive and negative results as previously mentioned, making a detailed analysis of little added value. Nonetheless, it's important to note that in this region, particularly in cases with adjacent houses having different roofing materials or irrelevant contextual objects, the corresponding prediction boundaries are less distinct than usual. This indicates confusion and highlights the model's difficulty in effectively modelling densely built areas.

5.3. Generalised Performance Evaluation

5.3.1. Quantitative Evaluation

The performance of the model on the generalised test subset is presented in Figure 5.15 and Table 5.8. Both ground truth masks and the corresponding pixel-wise predictions have been generalised to the three LoDs available in 3DBAG using the method presented in . Any background pixels are preserved in both cases.

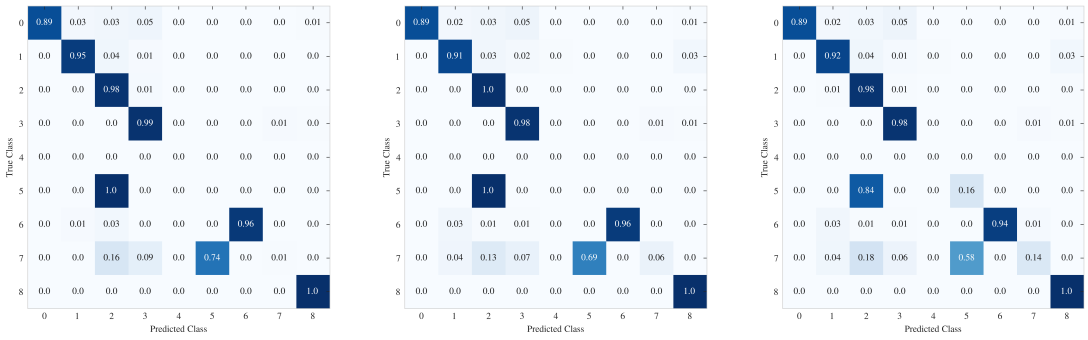


Figure 5.15.: Generalised confusion matrices.

As shown in Table 5.8, the light-permitting surface class is absent from the generalised ground truth masks. This is expected as all but one instances of this class in the test set were skylights, with the only exception being the atrium shown in Figure B.6. However, objects of this size are not included in the LoDs served by the 3DBAG. Hence, unless they happen to represent the majority of the surface area of another roof segment, which they clearly do not in this case, they cannot appear in the generalised ground truth. In actuality, the same is true for solar panels, since it is the only other class under consideration which does not refer to a true roofing material but a superstructure. However, given the fact that solar panels usually take up a significant percentage of the roof they are installed on in order to maximize they combined

Table 5.8.: Generalised classification report. The material corresponding to each class label is given in Table 4.1. results worse than the baseline are marked with red while better with blue.

Class	LoD1.2			LoD1.3			LoD2.2		
	Prec.	F1	IoU	Prec.	F1	IoU	Prec.	F1	IoU
1	0.9981	0.9741	0.9494	0.9777	0.9424	0.8911	0.9731	0.9435	0.8931
2	0.8175	0.8928	0.8063	0.8168	0.8974	0.8139	0.7735	0.8655	0.7628
3	0.9632	0.9741	0.9496	0.9656	0.9748	0.9508	0.9712	0.9773	0.9556
5	0	0	0	0.0004	0.0005	0.0003	0.0789	0.1062	0.0561
6	1	0.9805	0.9618	1	0.9778	0.9565	1	0.9715	0.9445
7	0.1408	0.0128	0.0064	0.623	0.1142	0.0605	0.7691	0.2397	0.1361
8	0.8932	0.9436	0.8932	0.5681	0.7246	0.5681	0.565	0.722	0.655
Avg. Acc.		0.6113			0.6138			0.6400	
mIoU		0.6524			0.6059			0.6290	

efficiency, they are more likely to appear even in the lowest LoD generalisations.

In contrast to the expected outcome, overall performance dropped significantly in comparison to the corresponding pixel-wise results, regardless of LoD. In particular, the average F_1 score ignoring the background and light-permitting surfaces dropped from 79.55% to as low as 68.75% in the case of LoD1.2, with the remaining two LoDs achieving somewhat improved but ultimately similar scores. Similarly, mIoU dropped from 68.97%, to 60.59% in the case of LoD1.3. However, closer inspection of Figure 5.15 reveals this result to be relatively misleading as the class-wise recall of all but the metal and solar panels classes is either in line or significantly improved in relation to the corresponding pixel-wise results. In fact, none of the two classes had a recall of more than 20% through the whole study. A potential explanation of this phenomenon is provided later in this section.

At this point it should be noted that microscopically averaged metric scores are not reported in this study because the underlying calculations assume that the ground truth masks contain nine classes, that is the background and eight material classes, which are sequentially labelled from zero to eight. However, because, the light-permitting surface class is absent from the ground truth, this assumption is invalidated. As a result, all averages are effectively computed as if the absent class actually existed, but the model did not output it, resulting in all relevant scores being zero. Although this issue can be avoided by manually computing macroscopically averaged metrics from the relevant class-wise results, microscopic averaging was unfortunately not possible without significant changes to the generalisation pipeline, which in turn not feasible due to time constraints.

In terms of class-wise performance, both classification and segmentation performance has consistently improved in all classes belonging to the first performance group (Section 5.2.1), as previously mentioned. In particular, both the F_1 and IoU scores of

membranes and gravel have increased by approximately 1–6% in the case of LoD1.2, depending on the particular class and metric, with segmentation performance having generally improved more than classification. The largest average improvement, that is the mean improvement in both F_1 and mIoU, was observed in dark-coloured metrics (5.22%), followed by gravel (3.38%), and finally light-coloured membranes (1.76%). On the other hand, gravel was the most improved class in the first group in the case of LoD1.3, achieving an improvement of 3.56% and 5.68% in F_1 and mIoU, respectively. However, the performance of the remaining two classes only slightly improved in comparison the corresponding pixel-wise results. This was also the case for LoD2.2.

With the exception of solar panels, the observed performance improvements continue to hold for the second performance group, namely ceramic tiles and vegetation. In fact, performance improved more in this group than in the first, with both F_1 and mIoU of the former class having improved by approximately 3–3.5% and 5–5.5%, respectively in both LoD1.2 and LoD1.3, before basically resuming their pixel-wise values in LoD2.2. In the case of LoD1.3, this improvement was unmatched by any class in the first group. In addition, vegetation was associated with a massive performance improvement across all LoDs clearly due to the removal of relevant false positive predictions (Figures B.4a and B.4b). In fact, the average improvement in LoD1.2 surpassed 38%, and never fell bellow 10%. This signifies the auxiliary purpose of generalisation which to reduce relatively small prediction noise.

Finally, performance in both the metal and solar panel classes surprisingly dropped to effectively zero in both LoD1.2 and LoD1.3, only for it to improve slightly in LoD2.2. The average performance degradation in each case was 46.27%, 46.23%, and 38.16% in the case of metal, and 58.92%, 51.14%, and 41.09% in the case of solar panels.

In the case of metal, the issue was apparent since instances of the titular class were present in only two images in the test set, and one of them (Figure B.2) would clearly correspond to completely incorrect predictions in any LoD. Furthermore, the second image where metal was present (Figure B.5a) clearly generalises to a solar panel roof regardless of LoD, as solar panels are not explicitly modelled as roof segments but represent the majority of valid pixels in the scene. Unfortunately, it is also clear that the corresponding model predictions, perhaps correctly depending on the particular use case, were generalised to metal, resulting in no correct metal predictions overall. The reason that performance in the case of LoD2.2, is not exactly zero is that the descending roof segment on the top left of the image is modelled separately. There, generalised predictions are obviously infallible.

Finally, in the case of solar panels, the issue becomes obvious given the fact that the model has issues fully identifying them (Figures B.1 and B.5).

5.3.2. Qualitative Evaluation

In the context of this study, the purpose of qualitative performance evaluation is not to highlight various real-world failure cases, but rather to identify which LoD provides the most consistent map experience to the user by "correcting" as many such errors as possible. After all, as generalised maps are produced by same product, which was

analysed in great detail in Section 5.2.2, no additional inaccuracies are possible. This considered, the generalized versions of both the complete tile, but also each of the four performance check regions is presented in .

In the case of BK, the LoD1.2 and LoD1.3 maps are effectively the same, with the building being modelled with a dark-coloured membrane roof and only a small number of minor segments along its edges being labelled as ceramic tiles in the latter case. In addition, the number of these segments is increased in the LoD2.2 map, which is also the only one to feature gross errors, in particular segments marked as vegetation, making it rather unsuitable for use as-is.

Similarly, the LoD1.2 map of the main campus buildings is mostly void of noise. However, various building extensions which have roofs featuring different materials than those used in their parents, are oftentimes inaccurately modelled. This is due to the fact that buildings may only have a single roof segment at a uniform elevation in LoD1.1, in turn allowing only one material type to be assigned to it. This issue is resolved in the LoD1.3 map, at the expense of some obvious errors in minor roof segments, which multiply and become clearly present LoD2.2.

This situation is largely the same in the industrial area. However, interestingly, the LoD2.2 map might have a distinct advantage over the LoD1.2 and LoD1.3 maps here due to its inclusion of some skylights, which are accurately modelled. Additionally, this map has fewer gross errors than usual as it depicts large, flat roofs which are uniform in shape and material, mostly belonging to the first performance group, thus optimizing all aspects of the model's performance.

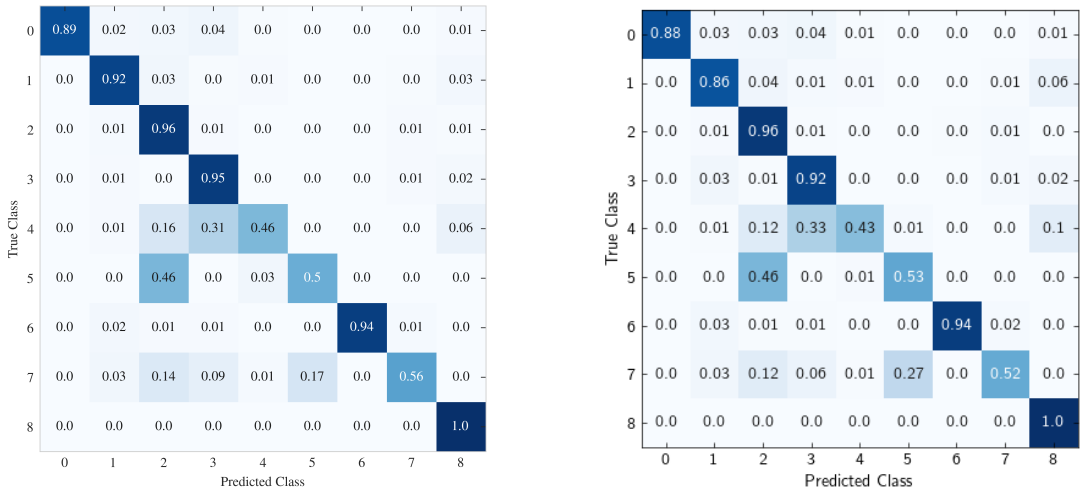
Finally, the LoD1.2 map of the residential area is likely ineffective because it omits dormers and only represents the simplest building shapes, as each roof is constrained to a single segment at a uniform elevation. However, since the dormer problem also occurs in the LoD1.3 map, users are compelled to resort to its otherwise relatively inconsistent LoD2.2 counterpart.

5.4. Ablation Study

This section presents the ablation sutdy regarding the LiDAR-derived bands of the input data in order to gauge their individual effect on performance. For each experiment, one band was removed and the model was retrained end-to-end three times using the best configuration discovered during the HPO process. The same random seed was used for each trial. Although this configuration has been optimized to operate with all bands present, any additional hyperparameter tuning would result in unfair comparisons. However, the presented results are not reliable indicators of the true performance potential of the model with each band removed. For each band, the presented results refer to the test subset using the best out of three corresponding trials according to validation mIoU.

5.4.1. Reflectance

The test performance of the model with the reflectance band removed is presented in Figures 5.16 and 5.17.



(a) With reflectance. (b) Without reflectance.

Figure 5.16.: Confusion matrix before and after removing reflectance band.

	Precision	F1	IoU
1	0.9249	0.8929	0.8065
2	0.7811	0.8612	0.7562
3	0.9490	0.9350	0.8780
4	0.6671	0.5222	0.3534
5	0.4970	0.5134	0.3453
6	0.9661	0.9542	0.9124
7	0.8378	0.6383	0.4688
8	0.3470	0.5148	0.3466
OA	0.8517		
mIoU	0.6084		

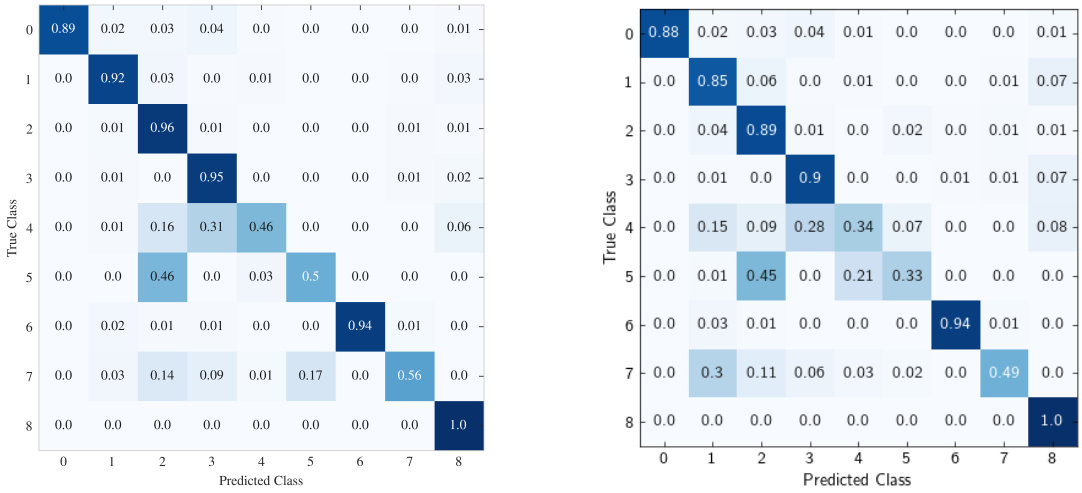
Figure 5.17.: classification report without the reflectance band. results worse than the baseline are marked with red while better with blue.

In general, a significant overall performance degradation can be observed both in terms of classification and segmentation. In particular, OA has been reduced from 87.93% to 85.17%, representing a deterioration of almost 3%. Similarly, mIoU has deteriorated by more than 4%, representing a combined average performance deterioration of 3.3%. Although accuracy in the metal class has improved by 3% due to a reduction in relevant false positives pertaining to light-permitting surfaces, significant performance degradation can be observed dark-coloured membranes (6%), solar panels (4%), gravel (3%), and light-permitting surfaces (3%). In the case of dark-coloured

membranes, this is primarily attributed to a sharp increase in vegetation false positives, implying that reflectance-derived features could have been heavily used by the model to differentiate these classes. Similarly, reflectance perhaps contributed significantly to the discrimination of solar panels and metal, as its ablation led to an increase of 10% in the solar panel instances misclassified as metal. This may be attributed to the fact that metal is generally characterised by high reflectance values, which may have been exploited by the model to facilitate its identification. However, relevant ceramic tile and gravel false positives have been reduced by 2% and 3%, respectively. In addition the absence of reflectance increased confusion between gravel and dark-coloured membranes[1] Finally, despite the fact that confusion between light-permitting surfaces and tiles dropped by 4%, the instances of the former class which were misidentified as gravel and vegetation increased by 2% and 4%, respectively.

5.4.2. Slope

The test performance of the model with the slope band removed is presented in Figures 5.18 and 5.19.



(a) With slope.

(b) Without slope.

Figure 5.18.: Confusion matrix before and after removing slope band.

In general, the overall performance degradation caused by the ablation of slope was significantly larger than what it was in the case of reflectance. In particular, OA has been reduced from 87.93% to 81.61%, representing a deterioration of more than 6%. Similarly, mIoU has deteriorated by almost 10%, representing a combined average performance deterioration of 8.32%. In actuality, this drop is the largest observed in the whole study. However, this was expected given the fact that various works

[1] in the context of this section, confusion between class a and b refers to the false positives of class b referring to class a.

	Precision	F1	IoU
1	0.8402	0.8429	0.7285
2	0.7575	0.8200	0.6948
3	0.9573	0.9290	0.8675
4	0.2947	0.3167	0.1881
5	0.6792	0.4408	0.2827
6	0.9496	0.9459	0.8973
7	0.8399	0.6189	0.4481
8	0.2425	0.3904	0.2425
OA	0.8161		
mIoU	0.5437		

Figure 5.19.: classification report without the slope band. results worse than the baseline are marked with red while better with blue.

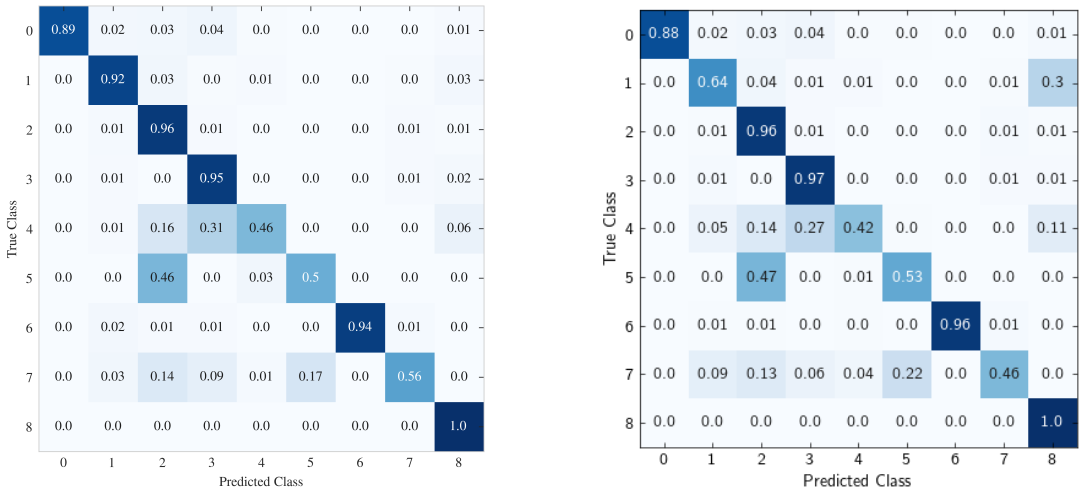
presented in the literature review not only recognised its contextual added value as it is a deciding factor for appropriate material selection when constructing any roof system, but also successfully employed it. This considered, with the exception of light-coloured membranes and vegetation, accuracy in all other classes dropped by as much as 17% and 12% in the case of metal and light-permitting surfaces respectively, with dark-coloured membranes, ceramic tiles, and solar panels following suit with (7%), and ultimately gravel with an observed degradation of 5%. In the case of metal, this is primarily attributed to a sharp increase in light-permitting surface false positives, implying that slope-derived features could have been heavily used by the model to differentiate these classes. Indeed, the absence of slope also led to an increase of 7%, from originally 0%, in the light-permitting surfaces misidentified as metal. In addition, increased confusion was observed between the former class and dark-coloured membranes (15% from 1%) as well as vegetation (8% from 6%). Nevertheless, both tile and gravel false positives pertaining to light-permitting surfaces were reduced by 7% and 3%, respectively.

In the case of dark-coloured membranes, the previously noted confusion with vegetation was still present, in addition to an increase of 3% in pertinent tile false positives. Furthermore, because dark membranes and tiles may appear similar in the absence of slope, a similar phenomenon was observed from the point of view of tiles, where some confusion with metal was also observed, most likely for the same reason. Similarly, the ablation of slope also resulted in a vast increase of 27%, from originally 3%, in the confusion between solar panels and dark-coloured membranes, as well as 2% in the confusion with light-permitting surfaces. Nevertheless, the instances of solar panels misclassified as tile, gravel, or metal all dropped, by as much as 15% to only 2% in the case of the latter class, signifying that solar panels and metal were potentially semantically coupled in the slope band.

Finally, most of the observed performance degradation in gravel is attributed to increased confusion with vegetation, with generally increased false positives pertaining to vegetation becoming a recurring theme in the study.

5.4.3. nDRM

The test performance of the model with the nDRM band removed is presented in Figures 5.20 and 5.21.



(a) With nDRM. (b) Without nDRM.

Figure 5.20.: Confusion matrix before and after removing nDRM band.

	Precision	F1	IoU
1	0.9193	0.7549	0.6064
2	0.7827	0.8642	0.7609
3	0.9571	0.9612	0.9253
4	0.5337	0.4703	0.3074
5	0.5445	0.5353	0.3655
6	0.9950	0.9783	0.9575
7	0.8416	0.5911	0.4195
8	0.1433	0.2508	0.1433
OA	0.8042		
mIoU	0.5607		

Figure 5.21.: classification report without the nDRM band. results worse than the baseline are marked with red while better with blue.

In general, the overall performance degradation caused by the ablation of the nDRM was almost as large as the one cause by that of slope. This is because as mentioned in, the nDRM conveys information on local height differences between various segments of a given roof due to the way it is defined. In particular, OA has been reduced from 87.93% to 80.42%, representing a deterioration of more than 7%. Similarly, mIoU has deteriorated by almost 9%, representing a combined average performance deterioration of 8.06%, the second largest observed in the study. This considered, with the exception of ceramic tiles and vegetation, where accuracy did not change, metal,

where it improved by 3% for the same reason as in the case of reflectance ablation, as well as gravel and light-coloured membranes where it also improved 2%, performance in all other classes decreased by as much as 28% in the case of dark-coloured membranes, followed by metal (3%) and light-coloured membranes (2%).

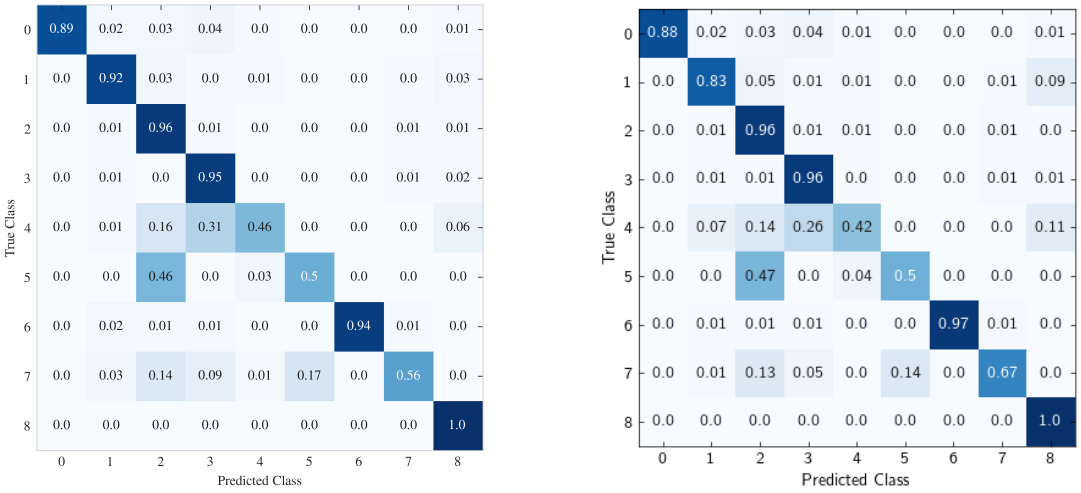
In the case of dark-coloured membranes this is once again attributed to a sharp increase in vegetation false positives, in this case the largest in study, from 3% to 30%.

This was also the case with light-permitting surface, although the increase was not nearly as drastic at 5%. In addition, although confusion with gravel decreased, similarly to all other studies, the number of light-permitting surface instances which were misidentified as dark-coloured membranes increased by 4%. This change is consistent with the one observed when slope was ablated, albeit much less pronounced.

However, unlike the prior study, the drop in the performance in solar panels was not primarily attributed to increased confusion with dark-coloured membranes. Specifically, although this problem persisted with a 4% rise, accompanied by a 3% increase in confusion with solar panels—nearly matching the slope ablation results—a significant 20% rise in instances misclassified as metal was noted. Finally, as was the case with in the previous studies, confusion with gravel dropped by 3%.

5.4.4. Density

The test performance of the model with the nDRM band removed is presented in Figures 5.22 and 5.23.



(a) With density.

(b) Without density.

Figure 5.22.: Confusion matrix before and after removing density band.

In general, the ablation of density resulted in the least significant performance degradation in comparison to all previous studies, implying that it is perhaps not a reliable material differentiator. This is corroborated by the fact that, although certain

	Precision	F1	IoU
1	0.9744	0.8965	0.8124
2	0.7680	0.8547	0.7463
3	0.9628	0.9618	0.9264
4	0.5427	0.4712	0.3082
5	0.6414	0.5591	0.3880
6	0.9887	0.9772	0.9554
7	0.8723	0.7599	0.6128
8	0.3116	0.4752	0.3116
OA	0.8707		
mIoU	0.6327		

Figure 5.23.: classification report without the density band. results worse than the baseline are marked with red while better with blue.

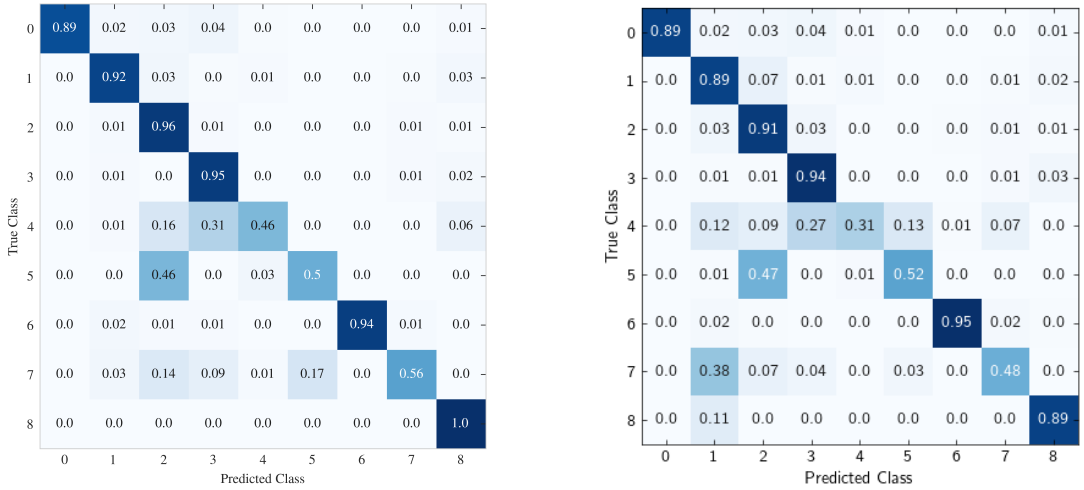
classes are indeed associated with higher and lower density than others (e.g., light-permitting surfaces), external factors also significantly influence this variation. Key among these are flight altitude and the presence of surfaces that the laser cannot reach due to their orientation in relation to the sensor-airplane system. In any case, OA still dropped from 87.93% to 87.07%, representing a perhaps statistically insignificant degradation of almost 0.9%. Similarly, mIoU has deteriorated by a little more than 1%, representing an average performance deterioration of 1.14%, which is the lowest in the study. This considered, with the exception of ceramic tiles, metal, and vegetation, where accuracy did not change, gravel where it improved by 1%, presumably due to an equivalent reduction in the relevant vegetation false positives, as well as light-coloured membranes where it also improved by 3%, similarly to the previous study, performance in almost all other classes decreased by as much as 10% in the case of dark-coloured membranes, followed by light-permitting surfaces (4%).

In the case of dark-coloured membranes, this is once again attributed to a sharp increase in vegetation false positives from 3% to 9%, the second largest in the whole study. In addition, as in the previous study, this was also the case with light-permitting surfaces, where it increased by 5% Furthermore, once again confusion with ceramic tiles and particularly gravel dropped, while the number of instances misidentified as dark-coloured membranes increased significantly from 1% to 7%.

Finally, an exception to this analysis concerns solar panels in contrast with all other studies, performance increased by 10%, making it the only class-study combination with a surely statistically significant performance improvement. In particular, this is attributed to decreased confusion with gravel and metal. Although these changes where observed in previous studies, the relationship between solar panels and metal sometimes worsened instead of improving, and confusion with other materials, especial tiles and light-permitting surfaces was introduced, which is of course not present in this case.

5.4.5. LiDAR

The test performance of the model with all the LiDAR-derived bands removed is presented in Figures 5.24 and 5.25.



(a) With LiDAR. (b) Without LiDAR.

Figure 5.24.: Confusion matrix before and after removing LiDAR-derived bands.

	Precision	F1	IoU
1	0.8321	0.858	0.7513
2	0.7608	0.8287	0.7075
3	0.9518	0.9482	0.9015
4	0.5586	0.4017	0.2513
5	0.8087	0.6343	0.4645
6	0.9709	0.9618	0.9264
7	0.8252	0.6032	0.4319
8	0.4436	0.5919	0.4204
OA	0.8496		
mIoU	0.6068		

Figure 5.25.: classification report without the LiDAR-derived bands. results worse than the baseline are marked with red while better with blue.

In general, the ablation of the whole LiDAR-derived component of the reference dataset resulted in significantly different behaviour than previous studies, particularly regarding the common issue of increased false positives pertaining to vegetation. In particular, OA dropped from 87.93% to 84.96%, representing a degradation of almost 3%. In addition, mIoU deteriorated by 4%, representing an average performance deterioration of 3.49%, which is interestingly less than half of what it was when slope or nDRM were individually ablated despite the fact that they are also missing here. This implies that the RGB bands of the dataset are perhaps independent of the rest

and convey enough information for the model to stabilize to an acceptable performance level. On the other hand, the LiDAR channels are almost surely interconnected, as the previous studies resulted in largely the same observations, this means that it is perhaps better to not use LiDAR data at all if not all relevant products are available.

In any case, with the exception of metal and light-coloured membranes where accuracy improved by 2% and 1%, respectively, individual class performance dropped by as much as 15% in the case of light-permitting surfaces, followed by solar panels (12%), vegetation (11%), which was did not have perfect recall for the first time in the whole study, ceramic tiles (5%), dark-coloured membranes (3%), and gravel (1%).

In the case of light-permitting surfaces, this drop is attributes to sharply increased confusion with dark-coloured membranes (11%), metal (13%), and solar panels (7%), none of which where present when all bands were used. Hence, the significant reduction in the number of light-permitting surface instances misclassified as tiles (7%) or gravel (4%), was not enough to bring performance back up. Similarly, increased confusion with dark-coloured membranes was once again one of the primary sources of performance degradation in the case of solar panels, where it rose from 3% to 38%, solely overshadowing big improvements in the confusion with tile (7%), gravel (5%), and metal (14%). Furthermore, this was the case with vegetation, where the number of pertinent instances misidentified as dark-coloured membranes increased from 0% to 11%. Finally, in the case of dark-coloured membranes there was increased confusion with tiles (6%), which was also observed albeit not to the same degree from the point of view of tiles. Moreover, the number of tile instances misclassified as gravel increased by 2%. The added value of the missing bands is now apparent because if not all of these issues can be attributed to the oftentimes similar appearance of the classes under consideration in the RGB spectrum under various lighting conditions.

6. Conclusions and Future Work

6.1. Research Objective Resolution

In the age of contemporary DL techniques, the unique potential advantages of pixel-based urban scene classification, particularly in comparison to OBIA, which has effectively replaced it in recent years, make it an attractive approach. In this context, this thesis investigated the applicability of general-purpose CNN-based roofing material type classification using aerial imagery and LiDAR data fusion, which has also not been thoroughly explored in the field. This considered, the aforementioned objective was accompanied by four research questions (Section 1.2) the answers to which would lead to its fulfilment. These questions are addressed below:

RQ1. Which imagery- and LiDAR-derived products are the most effective considering the task at hand?

As mentioned in Section 1.2, this question will be answered through the relevant literature review. In the case of optical imagery, the fact that most authors employed MSI and HSI, regardless of classification method, due to its inherently superior spectral resolution in comparison to conventional products, others have achieved equally competitive results with RGB imagery (Krówczyńska et al., 2020; Raczko et al., 2022; Wyard et al., 2023). However, it should be noted that all of them employed CNN-based classifiers, which were significantly more powerful in comparison to other models used in the field due to the fact that they are able to automatically learn intricate feature interrelationships. Hence, any comparison between these works and those using statistical or traditional ML techniques would most likely not be fair. In general, there is a trade-off between spectral and spatial resolution, and therefore the question effectively becomes a matter of which is more important. Interestingly, several authors who employed MSI and HSI argued against a high GSD, while others were in its favour. In any case, the arguments of both sides revolved primarily around the fact that certain roofing materials are characterised by high intra-class variance and inter-class similarity, as are metal and light-coloured membranes. Of course, this issue is naturally exacerbated in imagery with high spatial resolution. The truth of the matter is that if the downstream classifier is not able to handle this issue, then the typically lower GSD of MSI and HSI may reduce the overall error rate. On the other hand, models which can are bound to be more generalisable. However, a similar argument could be made in relation to spectral resolution. In fact, many authors used only a fraction of the original bands of their datasets, which could be said to defeat the purpose of using these products in the first place, especially given their increased acquisition and processing

costs. Therefore, an one-size-fits-all answer regarding optical imagery cannot be given as it clearly depends on other unrelated factors. Nevertheless, in the context of this thesis, where an CNN-based classifier was employed, both the conducted literature review and relevant experimental results point to the conclusion that RGB imagery is at the very least equally effective as MSI and HSI.

In terms of imagery-derived data, Ilehag et al., 2018 experimented with entropy, but eventually found it to not offer much contextual added value. On the other hand, several authors successfully employed various spectral indices, particularly NDVI, in order to facilitate material discrimination.

Finally, the situation in the case of LiDAR-derived products is much simpler as at the time of writing there were only two relevant works in the field , both of which featured and nDSM, with Norman et al., 2020 also including the DSM and DTM, as well as the intensity and slope fields. Ultimately, both authors concluded that utilizing LiDAR-derived data led to a statistically significant performance improvement in comparison to optical imagery. On a related note, Ilehag et al., 2018 hypothesised that planar point density could be effective in membrane identification, but did not investigate the matter any further.

RQ2. Which classification and data fusion techniques are the most contextually relevant?

As mentioned in Section 1.2, this question will be answered through the relevant literature review. In the case of classification approaches, there exists image-, object-, and pixel-based classification. The main difference amongst these methods is the analysis unit, that is the object[1] to which predictions refer to. In the case of image-based classification, a single label is used to characterise a given scene as a whole, meaning that the underlying model aims to answer the question of which material(s) is/are the most prevalent or otherwise important in the scene, but falls short of actually delineating them. This means that, in order to facilitate prediction localisation, the scene must generally contain a single building or building segment, or even object (e.g., solar panel), depending on the particular use case and required level of detail. However, this in turn raises the question of how to guarantee this, especially given the fact that buildings are encountered in a wide variety of physical sizes. Hence, a common approach to this issue is to simply choose the spatial dimensions of the input to be sufficiently small, so as for the corresponding predictions to be as unambiguous as possible. Nevertheless, the restriction of a single label per scene continues to apply, and therefore material hierarchy rules must be decided at the data annotation stage in order to be able to inevitably label scene containing multiple materials.

This issue may be somewhat alleviated by object-based classification or OBIA, which, instead of the whole scene, assigns a single label to each of a number of pre-defined pixel clusters it has supposedly been previously divided into, called “objects”. In this way, not only is explicit label localisation feasible, as each label is now associated with a clearly defined object, but also multi-label scene classification becomes

[1] not to be confused with “object” in object-based classification

possible. However, the preliminary task of image segmentation, that is defining the objects in a given scene, has been historically considered to be the hardest and simultaneously most important part of OBIA, with most authors in the field agreeing that a well-performing automated solution simply does not exist. In fact, there is apparent disagreement even as to what exactly constitutes an "object", with some authors clearly going for a "building-based" approach, using building outlines to delineate them from background (Tommasini et al., 2019) or otherwise attempting to identify in their in datasets (Trevisiol et al., 2022), whereas others resort to simply grouping together pixels of similar appearance or spectral signature using standard image segmentation techniques (Hamedianfar, Shafri et al., 2014). In addition, because OBIA models operate on objects, not images, this step must be performed for each image they are given, with the segmentation parameters changing every time unless relevant building or roof outlines are available. Furthermore, it should be noted that, similarly to image-based classification, each object is assigned a single label, with everything this entails.

The only method which does not involve this limitation is pixel-based classification, which assigns a single label to each pixel in a given scene. Since pixel-based classifiers operate on images and the pixel is the fundamental unit of an image, this approach can identify materials at the sub–roof-segment level without needing to be segmented beforehand. This means that, in contrast to both image- and object-based classification, predictions are given directly at the highest possible level of detail, and may thus be generalised to cater to a variety of use cases after the fact (Osińska-Skotak & Ostrowski, 2015), meaning that unless the particular application requires it, no relevant constraint or simplification (e.g., only one material per building, roof segment, or object, etc.) needs to be considered. Given the fact that the only argument against pixel-based classification in relevant literature, that is the fact that conventional classifiers which cannot infer spatial coherence cues amongst neighbouring pixels ended up making noisy predictions, is no longer valid in the age of contemporary CNN- and ViT-based models, the obvious choice of classification approach is this.

Finally, in terms of data fusion approaches, the only relevant technique discussed in relevant literature is layer stacking, that is the integration of all individual datasets into a single, multi-band raster product by concatenating them along their channel dimension. This type of fusion is called feature-level fusion, and was adopted in this work as the only relevant one which was supported by the literature review.

RQ3. How does the availability of LiDAR-derived products affect predictive performance?

As mentioned in Section 1.2, this question will be answered using the results of original research and experiments conducted in this work. As observed by Hamedianfar, Shafri et al.; Norman et al., the integration of optical imagery and LiDAR-derived products and optical imagery appears to significantly improve predictive performance. In particular, when only the RGB component of the reference dataset was used to train and test the model, both classification and segmentation performance, quantified by OA and mIoU, respectively, dropped by approximately 3% and 4%, from

87.93% and 64.68%, to 84.96% and 60.68%. Although there was a significant decrease in the light-permitting surface instances incorrectly identified as tile or gravel, as well as an equivalent improvement in the solar panel pixels which were misidentified as tile, gravel, or metal, the latter of which actually led to a remarkable precision improvement in the metal class, confusion in the minority or otherwise "difficult" classes (e.g., due to the physically small size of the corresponding objects), namely light-permitting surfaces, solar panels, and vegetation, was increased. This issue can be attributed to the similar appearance of these materials to others in the RGB spectrum under certain lighting conditions, particularly dark-coloured membranes, ceramic tiles, as well as residential light-permitting surfaces and solar panels. As such, the general conclusion regarding the LiDAR-derived products used in this work is that they are highly useful for effective material disambiguation under suboptimal viewing conditions, albeit not otherwise necessarily critical to overall performance.

Concerning the performance effect of each individual product, the absence of slope resulted in the highest average degradation (8.32%), closely followed by the nDRM (8.06%). The observed importance of slope was expected not only as an observation from the literature review (Wyard et al., 2022), but also due to the fact that different roofing materials are inherently associated with different ranges of roof pitch. By extension, the definition of the nDRM allows it to implicitly convey slope information, and it hence carries a similar predictive power to slope. However, as this information is based on the median roof elevation of each building, any relevant information extracted from it is not absolute. Perhaps somewhat surprisingly, due to its relatively insignificant physical meaning, the third most important product was reflectance, with an average performance deterioration of 3.3%. In particular, due to its definition and range invariance, as well as the typical wavelength of airborne LiDAR sensor, which is in the NIR range, at was initially assumed that reflectance would effectively behave similarly to a typical infrared band, as if it was optical imagery. However, this was not in fact exactly the case. Finally, planar point density did not appear to impact overall performance significantly (1.14%).

In the context of this RQ, perhaps the most interesting conclusion is that the LiDAR-derived bands are most likely semantically interconnected, and hence it is perhaps better to either use them all together as as single product or not at all. This assumption stems from the fact that the RGB-only configuration performed significantly better than the ones where slope or the nDRM was ablated, and almost the same as when reflectance was absent. In addition, the ablation of each LiDAR-derived band revealed certain patterns of increased confusion which were almost universally present regardless of which band was absent. In particular, all studies showed a significantly increased confusion between dark-coloured membranes and vegetation, as well as light-permitting surfaces and vegetation, both of which did not exist initially. Furthermore, increased confusion was observed between light-permitting surfaces and dark-coloured membranes in all studies but the ablation of reflectance. However, the total number of light-permitting instances misidentified as gravel dropped. Similarly, all experiments resulted in decreased confusion between light-permitting surfaces and tiles as well as solar panels and gravel. Moreover, confusion between solar panels and metal

dropped in all cases except that of the absence of the nDRM. Finally, performance in light-coloured membranes remained relatively unaffected throughout the study.

RQ4. How does the generalization of pixel-wise roofing material maps to each of the building LoDs offered by the 3DBAG influence results?

As mentioned in Section 1.2, this question will be answered using the results of original research and experiments conducted in this work. In terms of quantitative performance, the map generalisation did not explicitly improve performance. In fact, OA in the metal and solar panel classes consistently dropped, mainly due to the inability of the model to fully detect relatively small objects, in this case panels. However, observed artifacts manifesting themselves as false positives pertaining to vegetation practically disappeared, resulting in an equivalent increase in map reliability, especially from the perspective of the user. In addition, another interesting result is that generalisation resulted in the absence of light-permitting surfaces from the resulting maps due to their small size. Hence, despite its aforementioned advantage, generalisation is generally not suitable in case information on these objects is required.

Furthermore, qualitative evaluation found that not all LoDs are suitable for all use cases. In particular, LoD1.2 is unable to correctly model buildings with roof segments at various elevations, featuring different materials from each other. On the other hand, LoD1.3 solves this issue all the while having effectively no practical disadvantages compared to LoD1.2 with the exception of occasional artifacts in small roof segments. Finally, LoD2.2 maps typically include significant noise, but have applications in industrial regions because large protruding skylights are supported by the LoD2.2 specification but not LoD1. For the same reason, these maps are the only choice in residential regions where dormer information is required.

In conclusion, given the answers to the now-resolved research questions, it is clear that pixel-based roofing material classification is no longer inherently inferior to its image- or object-based counterparts. In fact, this thesis achieved SOTA classification and semantic segmentation performance in both dark- and light-coloured membranes, as well as gravel. In addition, competitive results were obtained in the case of ceramic tiles. However, only marginally acceptable results in solar panels and vegetation indicated the need for further research. Similarly, disappointing performance was observed in metal and light-permitting surfaces, signifying that a more comprehensive reference dataset is almost surely required, as the model clearly had significant issues differentiating said materials from ones of similar appearance. Finally, in terms of data fusion, this work showed LiDAR-derived data to offer significant performance improvements in scenes with suboptimal lighting conditions as well as in cases where local elevation changes are correlated with specific materials.

6.2. Discussion

6.2.1. Contributions

This thesis draws inspiration from relevant works and combines their individual strengths to modernise pixel-based roofing material classification, introducing an open end-to-end methodological framework for large-scale, general-purpose material mapping based on data fusion and DL, which were combined for the first time in the field, partially SOTA results in dark- and light- coloured membranes as well as gravel. In this context, the scientific contribution of this work may be summarised into three areas, namely the reference dataset, the corresponding splitting and weighting schemes, and the final model.

In particular, this thesis is only the second after Wyard et al., 2023 and the first in the field of pixel-based roofing material classification to publish its reference dataset (Mantas, 2024b) under a liberal copyright licences. Furthermore, the software developed in the context of this work (Mantas, 2024a) is similarly released as FOSS, the first in the field regardless of classification method. This approach ensures the transparency and reproducibility of this thesis and encourages further research and development, as interested parties are not compelled to reimplement the proposed methodology.

The second contribution of this work is the proposed reference dataset splitting and class weighting schemes. As mentioned in Section 2.2.4, none of the several existing multi-label stratified splitting algorithms have been implemented with DL in mind as the relevant implementations are explicitly designed to produce two instead of three splits. Because re-splitting one of these subsets to end up with three overall sets does not generally produce correct results as the underlying algorithm does not take the ignored set into account, the only option is to use cross-validation. However, this solution is not attractive as the model must be trained ab initio for each fold. Hence, this thesis introduces an alternative splitting scheme which is inspired by the work of Xiao et al. and operates by greedily minimising the mean Wasserstein-1 distance amongst the pixel-level class distributions corresponding to the provided subsets. Although, this algorithm has not been rigorously tested for correctness, it is, to the best of the author's knowledge, the only functional alternative to the aforementioned techniques at the time of writing. Similarly, the proposed class weighting scheme applied to the training subset, which is inspired from information retrieval systems is designed to explicitly take image-level class frequencies into account, another first in the field. Of course, the implementations of both the splitting and weighting schemes are available as part of the software developed in the context of this work.

Finally, the optimised parameters of the final model are provided alongside the implementation software, allowing interested parties to directly use it as a ready-made product in their own work. Apart from the obvious facilitation of applications which could potentially benefit from intelligence related to roofing materials, this is also the first public model in the field.

6.2.2. Limitations

The main limitation of this thesis is the reference dataset size, which is arguably insufficiently small, as evidenced by the high step-to-step and trial-to-trial variance in both training and validation. For instance, only one image in the test subset contained the vegetation class. This issue significantly undermines the reliability of the results presented in this work as it remains unclear which, if any, are actually statistically significant, particularly concerning the HPO process. Although qualitative performance evaluation did in fact aid in general performance assessment, e.g., by introducing several vegetation samples, it is of utmost importance that the dataset be expanded as soon as possible. Furthermore, it should be noted that several errors were discovered in the ground truth segmentation masks during the quantitative performance evaluation, which undoubtedly caused additional model confusion and degraded the added value of the reference dataset. Although plans exist to amend these errors in a later dataset revision, a more comprehensive annotation process is essential, potentially including multiple annotators and incorporating a feedback loop for improved quality control.

6.3. Recommendations and Future Work

To advance current research, several key recommendations are proposed and prioritised by their potential impact. In particular, high-priority items are likely to result in significant improvements to the proposed workflow and overall performance, thus warranting prompt attention in future work. In contrast, medium and low-priority tasks provide opportunities for further refinements in specific areas.

As mentioned in Section 6.2.2 immediate focus should be placed on increasing the size of the reference dataset. Subsequently, the annotation process could be enhanced by adopting more efficient, (semi-)automated annotation methods.

Moreover, a thorough review of the proposed annotation guidelines, combined with the implementation of an comprehensive quality control stage, is expected to significantly reduce the likelihood of irreparable gross errors, ensuring consistency and accuracy in annotations.

Similarly, self-supervised pretraining on a superset of the reference dataset warrants exploration. For instance, contrastive learning techniques are gaining popularity in relevant literature (Stewart et al., 2023) as effective methods for learnable parameter initialisation, enabling the model to develop comprehensive latent representations of scenes it will subsequently encounter, thereby reducing training times and enhancing performance.

In addition, future research should explore alternative data sources to complement or replace the datasets utilised in this work, e.g., replacing BM5 with a true-ortho alternative. Furthermore, some of the proposed data pre-processing techniques, such as point cloud rasterisation and feature normalisation, could be refined, and new methods like radiometric intensity correction could be introduced. Finally, additional handcrafted features to enhance the identification of small objects, such as light-permitting surfaces and solar panels, could be investigated, along with shape-aware training, e.g.,

by incorporating automatically generated kinetic partitions of the input scenes (Gao et al., 2024) into the corresponding raster stacks or the loss function.

A. Hyperparameter Optimization

This appendix contains all non-essential data used in the analysis of the HPO process.

A.1. Manual Experimentation

Table A.1.: Manual experiments results better than the baseline are marked in green.

Category	Parameter	Value	Validation mIoU (Last Step)
Input Data	Append HSV	Yes	0.6283
	Append TGI	Yes	0.6107
Encoder	Variant	ResNet-18-D	0.674
	Attention Block	ECA	0.6001
		SE	0.5695
	Anti-aliasing Block	Yes	0.5382
Decoder	Base Dilation Rate	1	0.5924
		10	0.5873
		15	0.6332
		20	0.5566
Regularisation	Stochastic Depth	0.05	0.6039
		0.1	0.5415
		0.15	0.5836
	Label Smoothing	0.05	0.5483
		0.1	0.6303
		0.15	0.5782
Optimisation	Weight Decay	0.001	0.5944
		0.01	0.5642
		0.1	0.4361
	Optimizer	AdamW	0.6189
	Learning Rate Annealing	Cosine	0.5948
	Learning Rate	1e-4	0.5191
		5e-4	0.568
		5e-3	0.5702
		1e-2	0.5014
	Warmup Length	50	0.5948
		100	0.6269
		150	0.575

A.2. Round 1

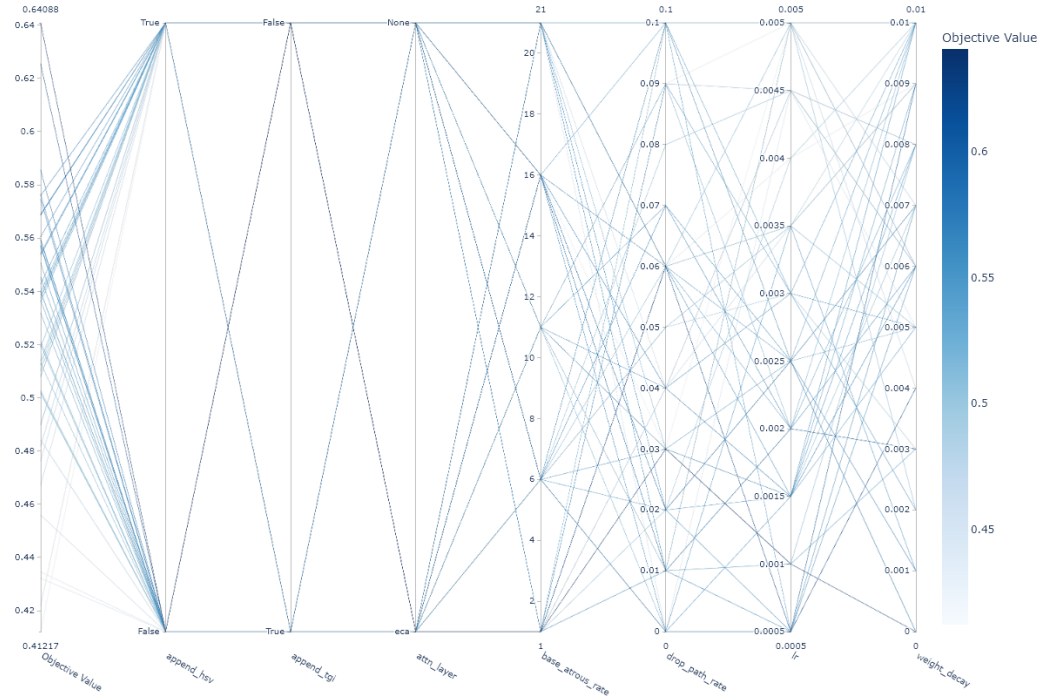
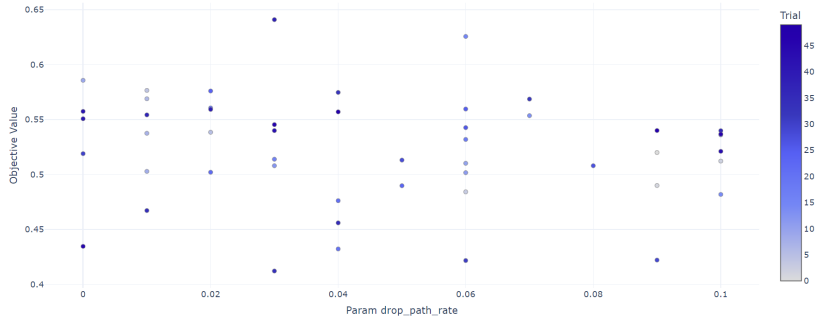
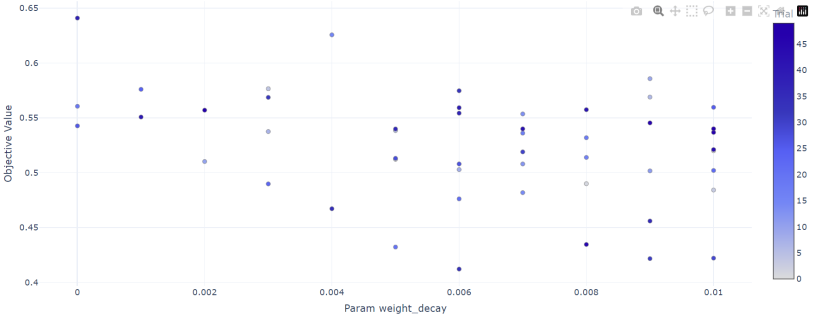


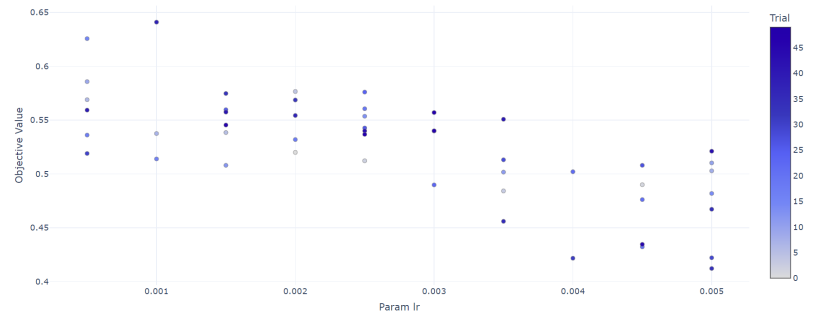
Figure A.1.: Parallel coordinate of the first automated round of the HPO process.



(a)



(b)



(c)

Figure A.2.: slice plots of the continous variables of the first hpo round

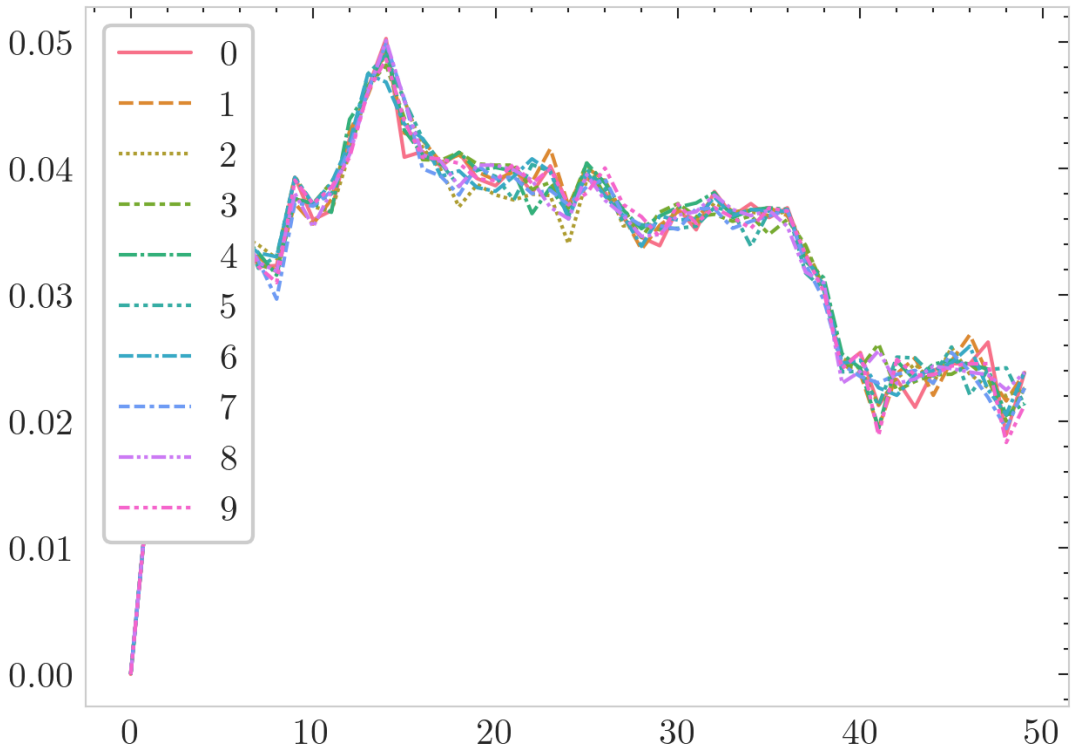


Figure A.3.: Terminator improvement of the first automated round of the HPO process. The expected improvement potential is estimated according to Makarova et al., 2022. Each plot represents one of 10 evaluations of the underlying algorithm using different random seeds.

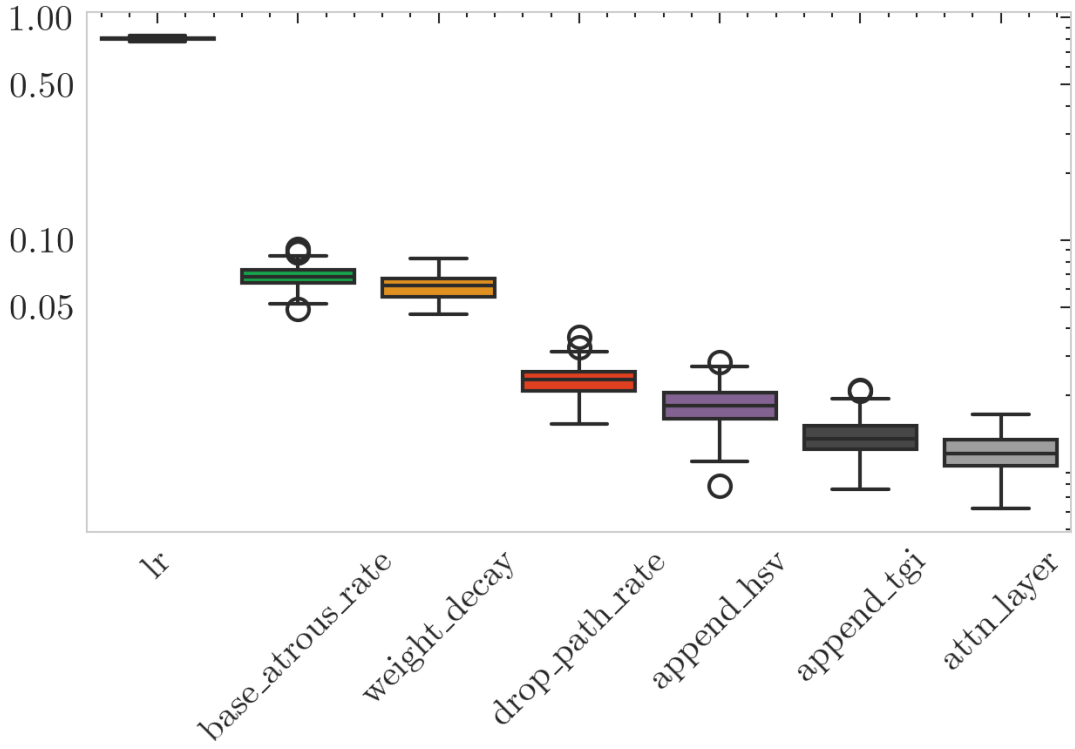


Figure A.4.: Paremeter importance of the first automated round of the HPO process. The expected improvement potential is estimated according to Hutter et al., 2014. The distribution of each parameter importance is computed across 100 evaluations of the underlying algorithm using 100 different random seeds.

A.3. Round 2

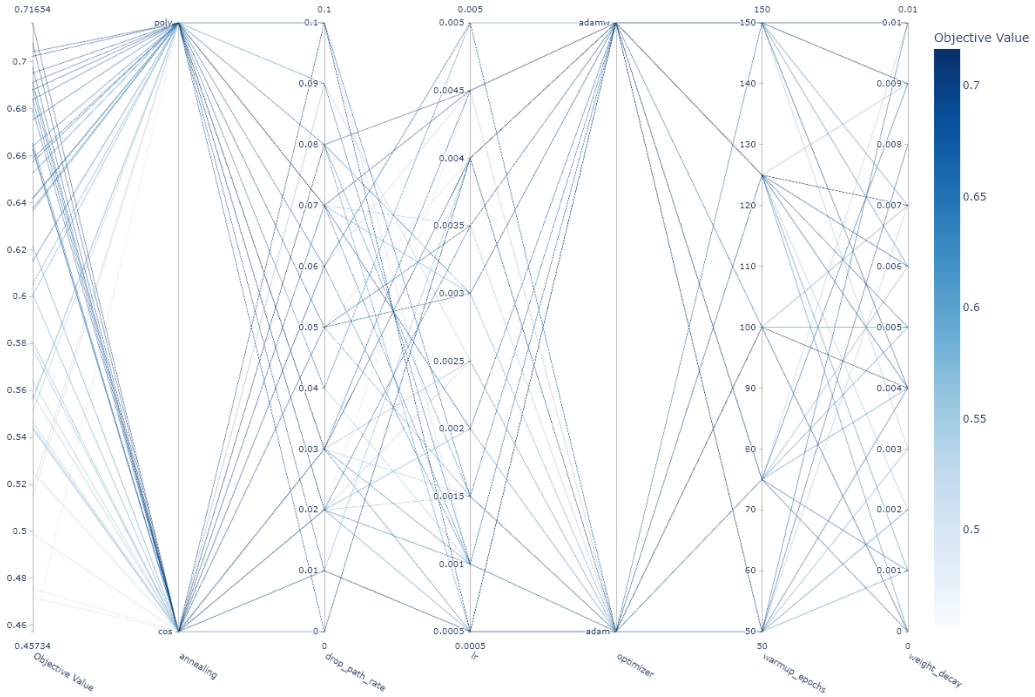
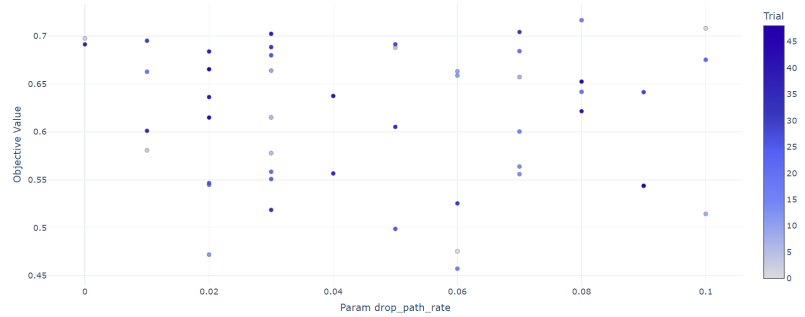
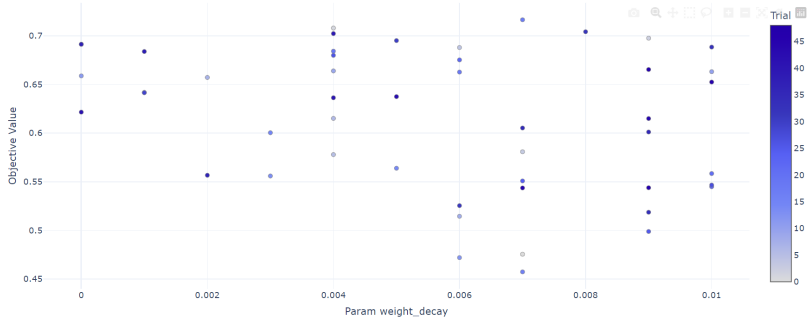


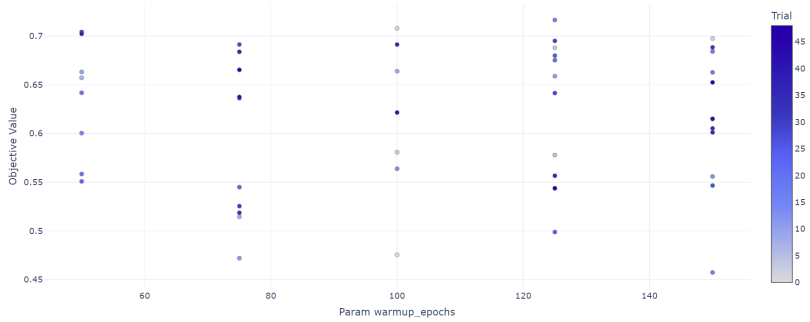
Figure A.5.: Parallel coordinate of the second automated round of the HPO process.



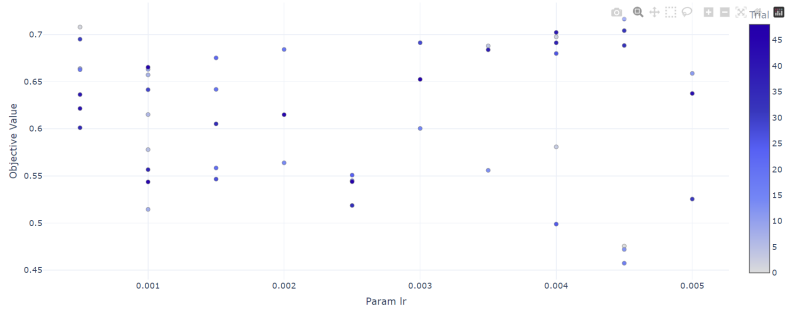
(a)



(b)



(c)



(d)

Figure A.6.: slice plots of the continous variables of the second hpo round

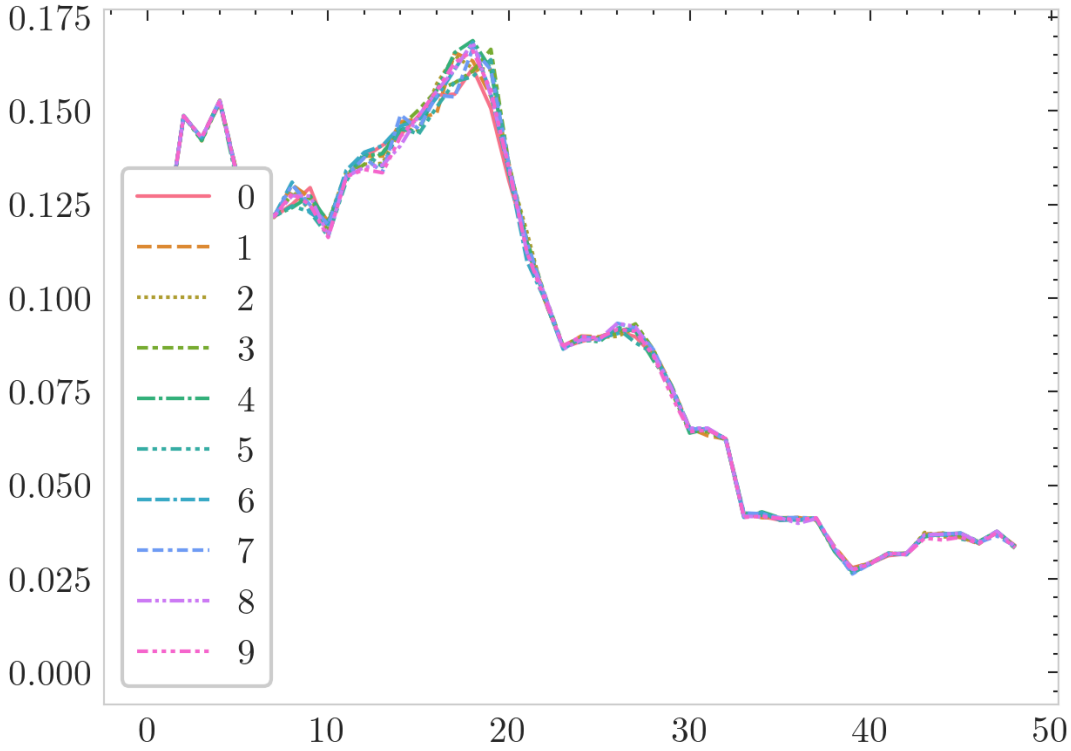


Figure A.7.: Terminator improvement of the second automated round of the HPO process. The expected improvement potential is estimated according to Makarova et al., 2022. Each plot represents one of 10 evaluations of the underlying algorithm using different random seeds.

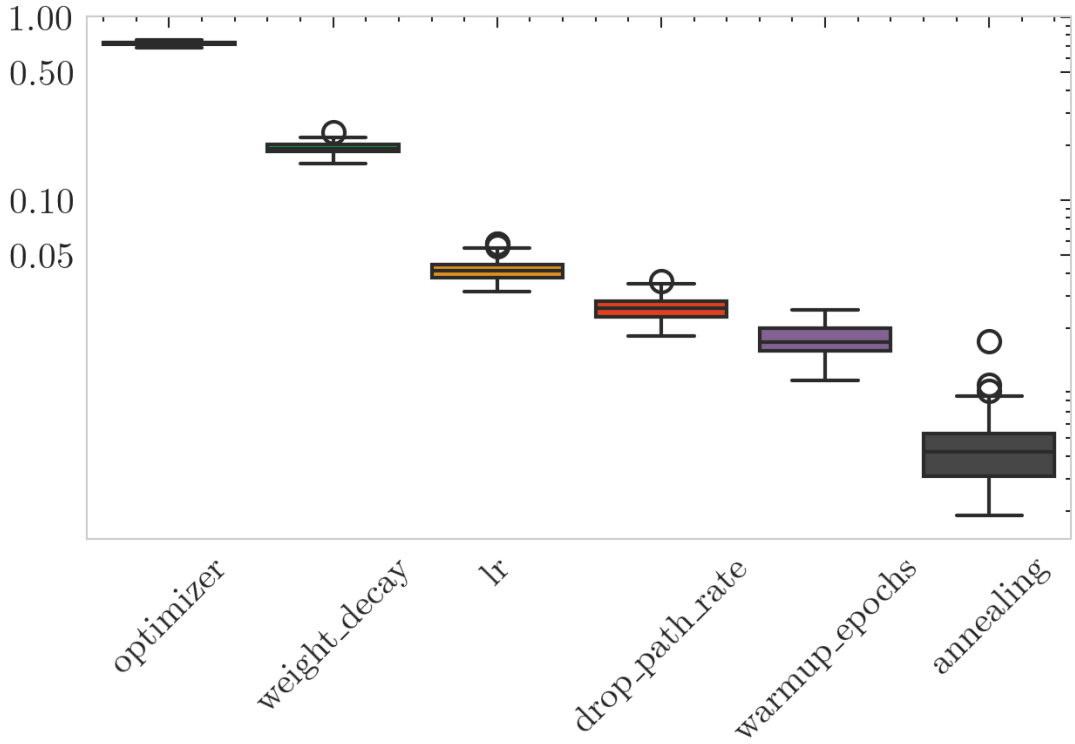


Figure A.8.: Paremeter importance of the second automated round of the HPO process. The expected improvement potential is estimated according to Hutter et al., 2014. The distribution of each parameter importance is computed across 100 evaluations of the underlying algorithm using 100 different random seeds.

A.4. Round 3

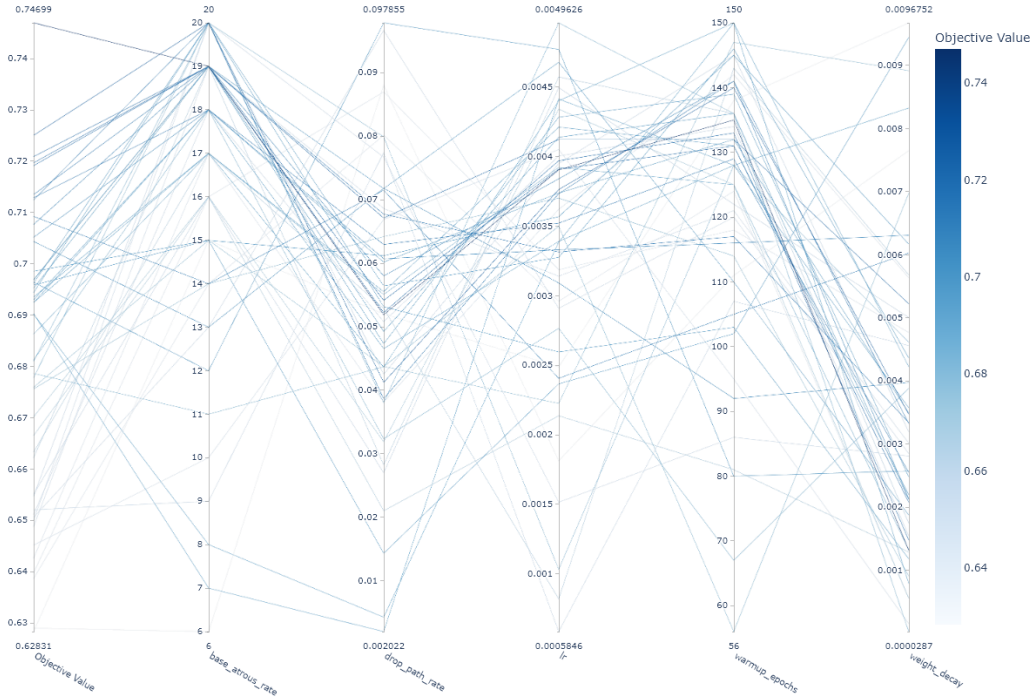
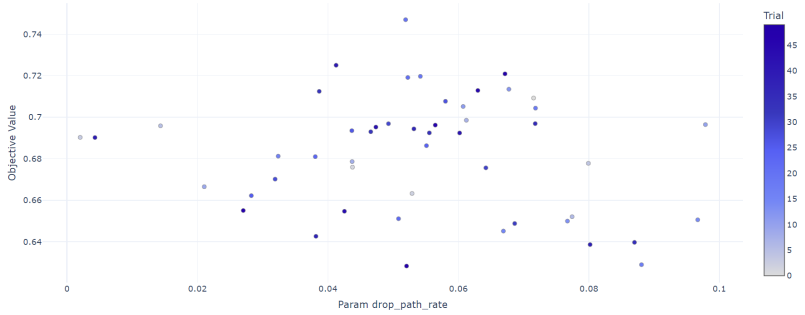
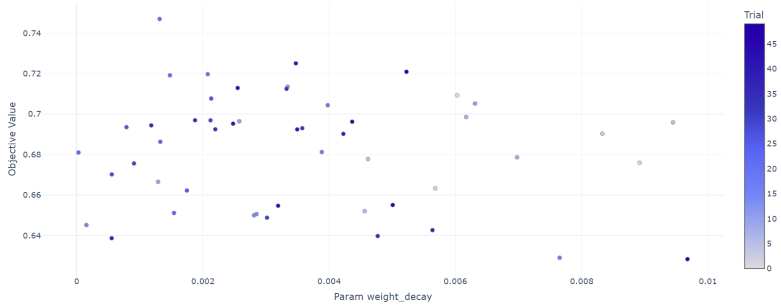


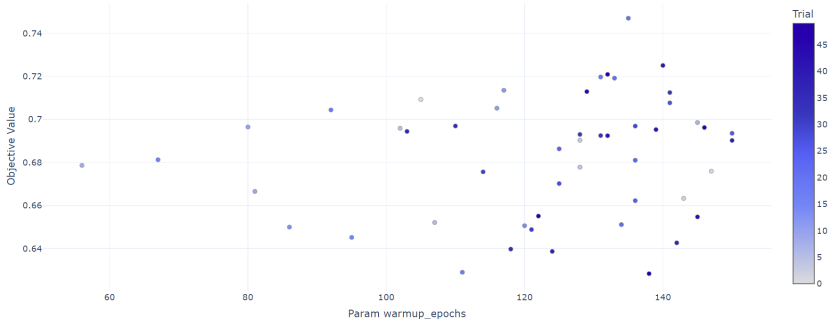
Figure A.9.: Parallel coordinate of the third automated round of the HPO process.



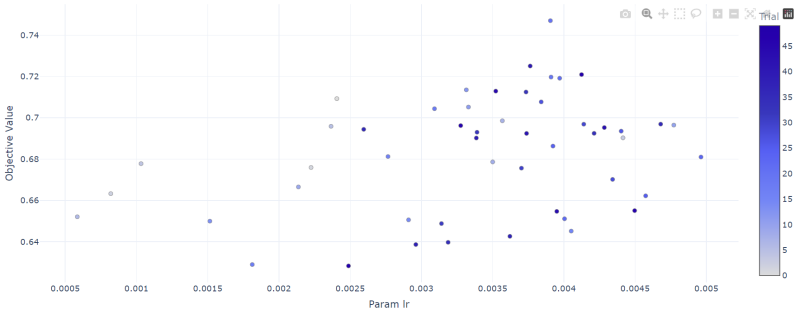
(a)



(b)



(c)



(d)

Figure A.10.: slice plots of the continous variables of the third hpo round

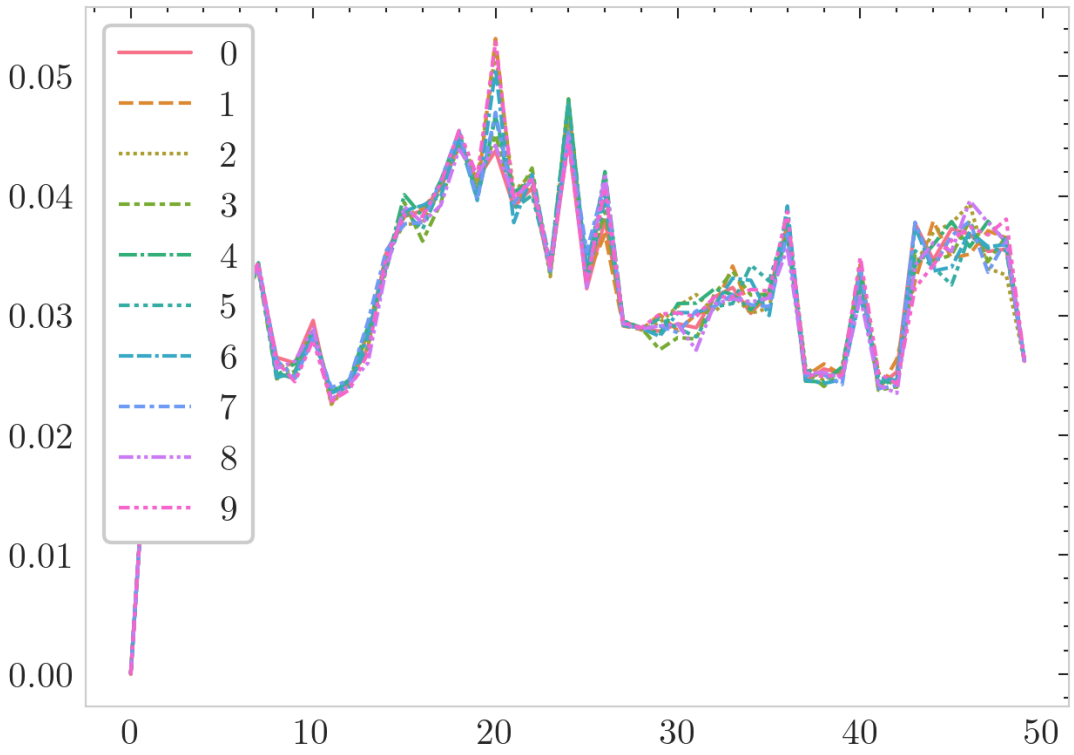


Figure A.11.: Terminator improvement of the third automated round of the HPO process. The expected improvement potential is estimated according to Makarova et al., 2022. Each plot represents one of 10 evaluations of the underlying algorithm using different random seeds.

B. Failure Cases

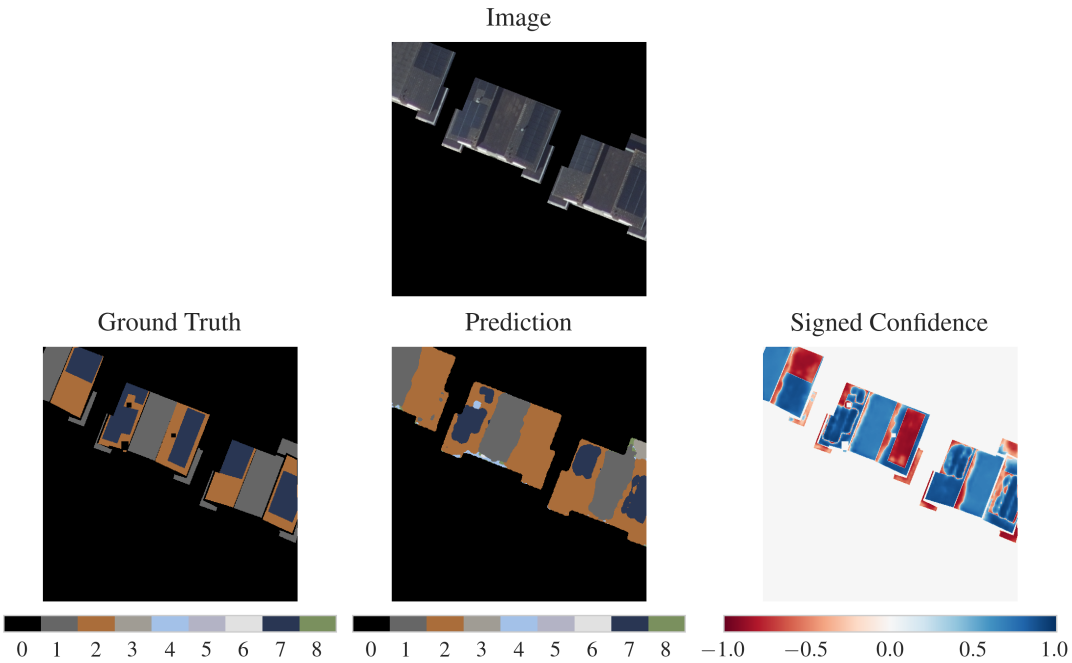


Figure B.1.: Failure case where residential solar panels installed onto a ceramic tile roof were confused for the base material (i.e., negative confidence regions in left and middle buildings). color-label associations available in Table 4.1.

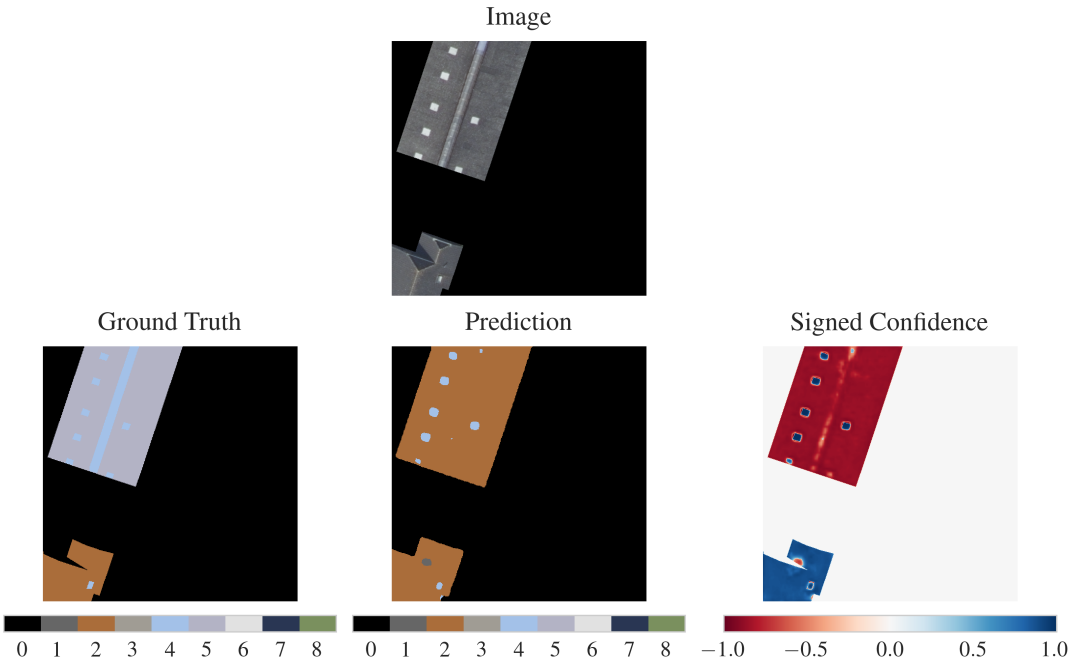


Figure B.2.: Failure case where a dirty dark metal roof was completely confused for one with ceramic tiles (i.e., negative confidence region in the top building). The polycarbonate skylight on the roof was also not detected.color-label associations available in Table 4.1.

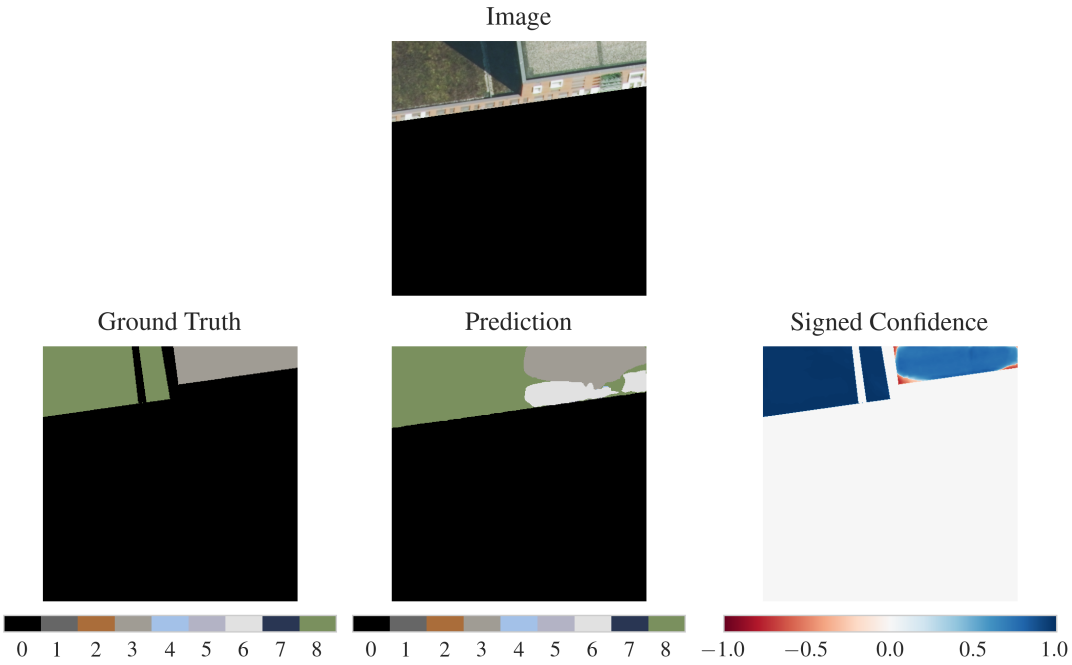
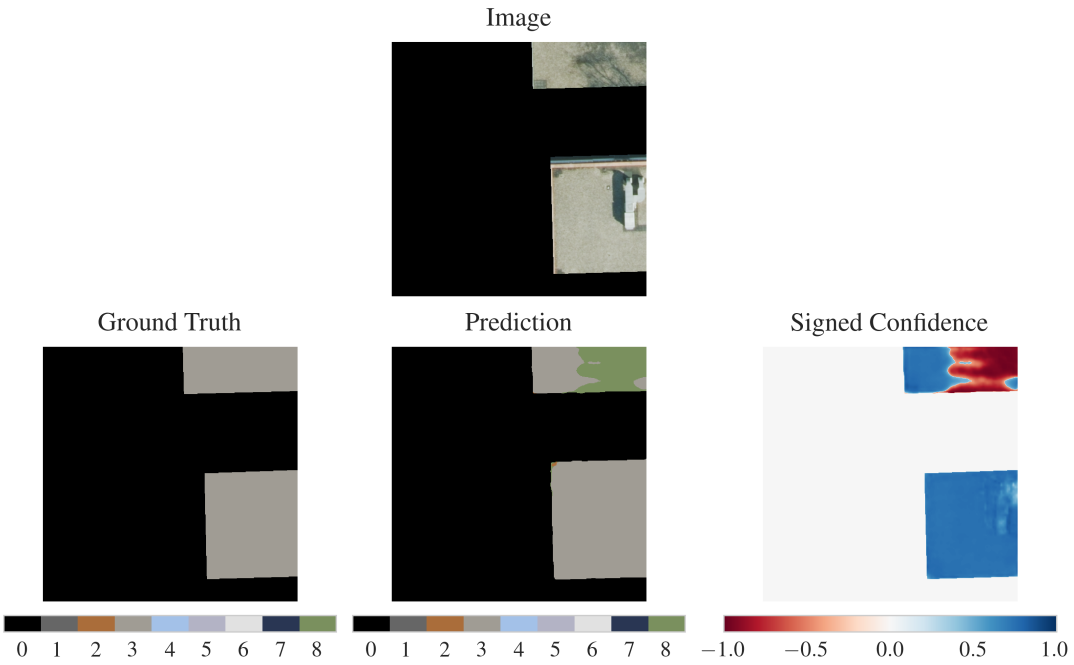
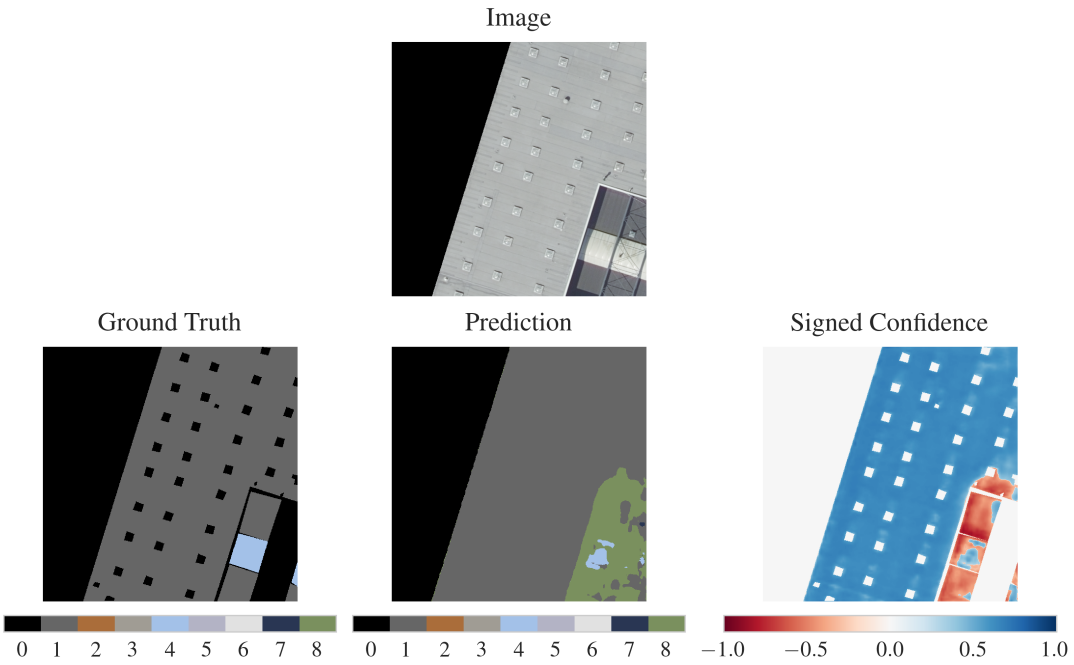


Figure B.3.: Failure case where a dirty dark metal roof was completely confused for one with ceramic tiles (i.e., negative confidence region in the top building). The polycarbonate skylight on the roof was also not detected.

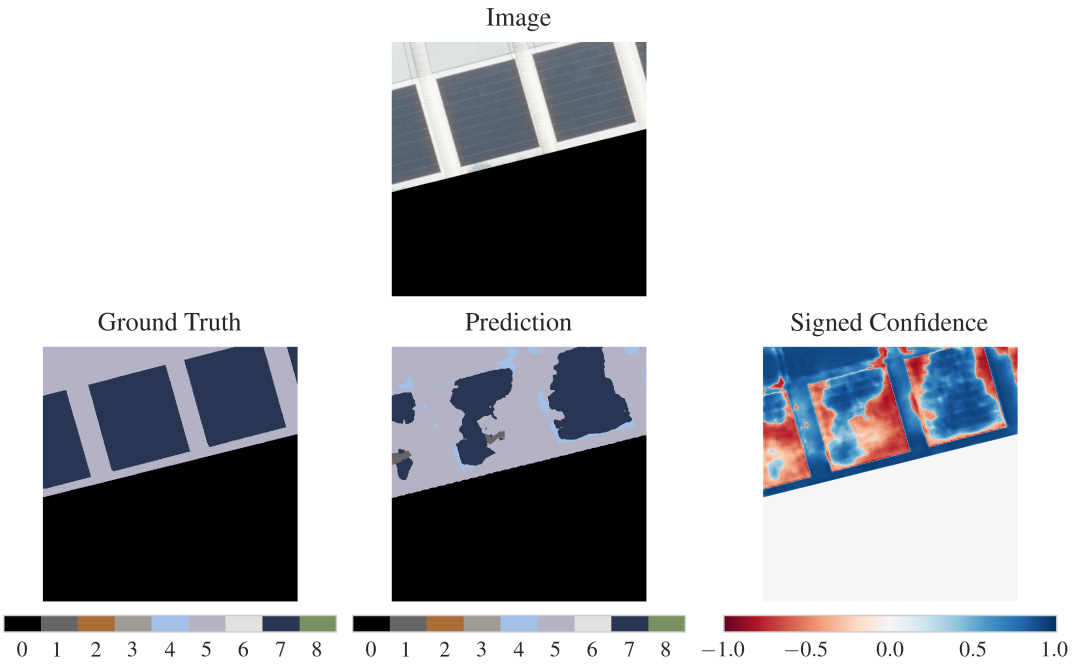


(a) Confusion with gravel.

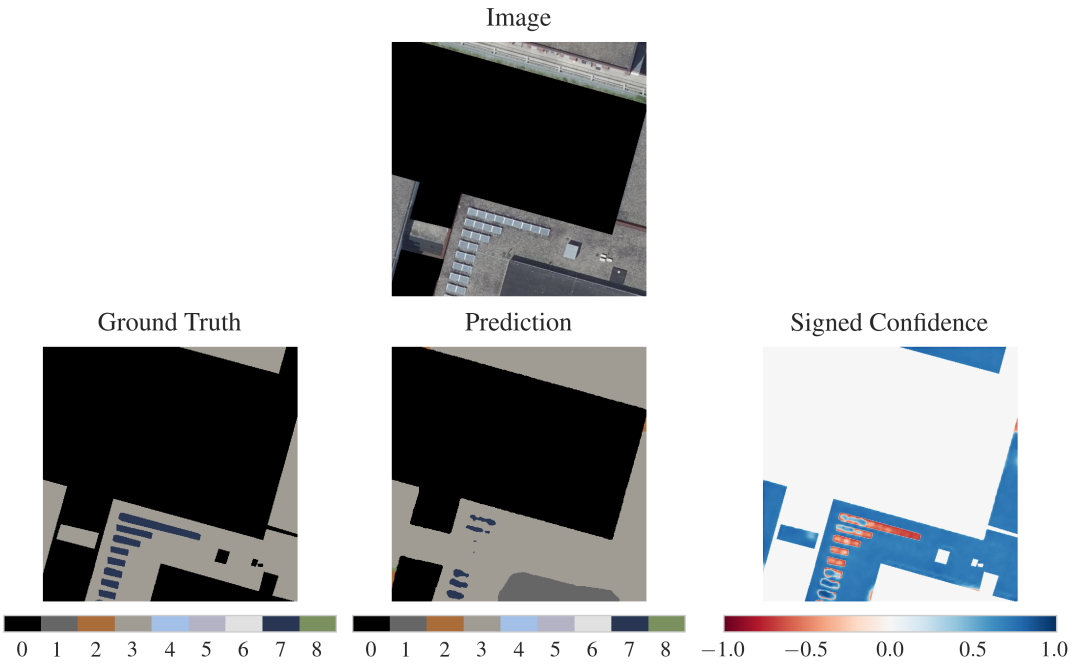


(b) Confusion with dark-coloured membrane and light-permitting surface.

Figure B.4.: Failure cases where gravel (a) and dark-coloured membranes/light-permitting surfaces (b) were confused with vegetation.color-label associations available in Table 4.1.



(a)



(b)

Figure B.5.: Failure cases where clearly visible solar panels where not fully identified.color-label associations available in Table 4.1.

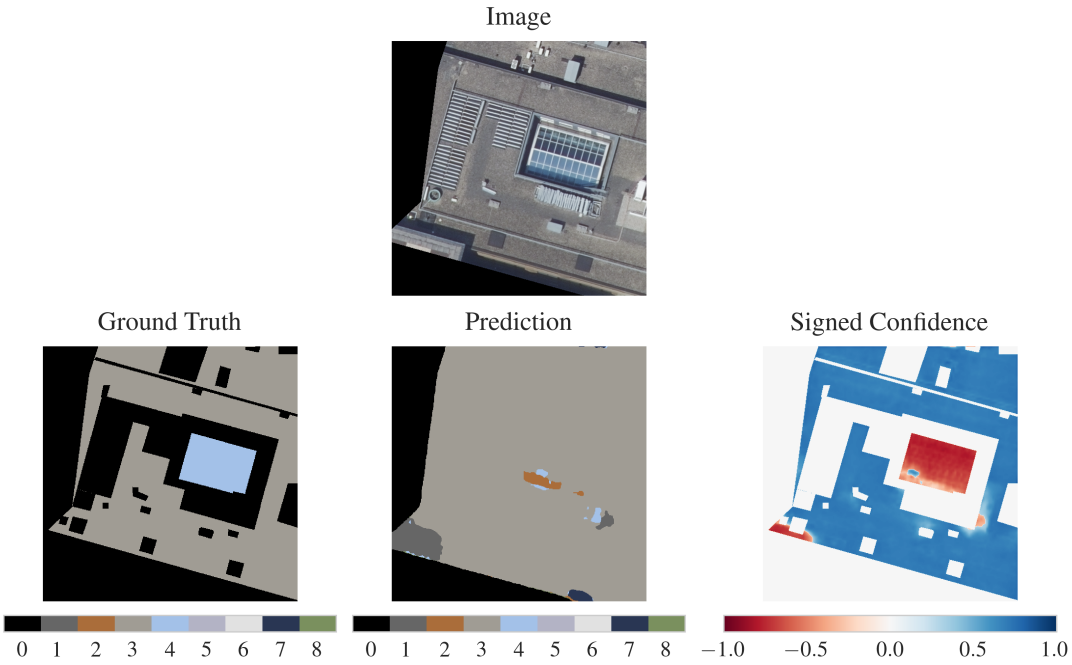


Figure B.6.: Failure case where an atrium below a tar-and-gravel roof was mostly missed. Tile hallucinations are visible in the parts of the atrium which were actually detected.color-label associations available in Table 4.1.

C. Qualitative Performance Evaluation

C.1. Overview

Figure C.1.: 3DBAG tile 9-284-556. The corresponding LoD2.2 surface footprints have been dissolved by building ID and are also shown in blue.

C.2. Performance Check Regions

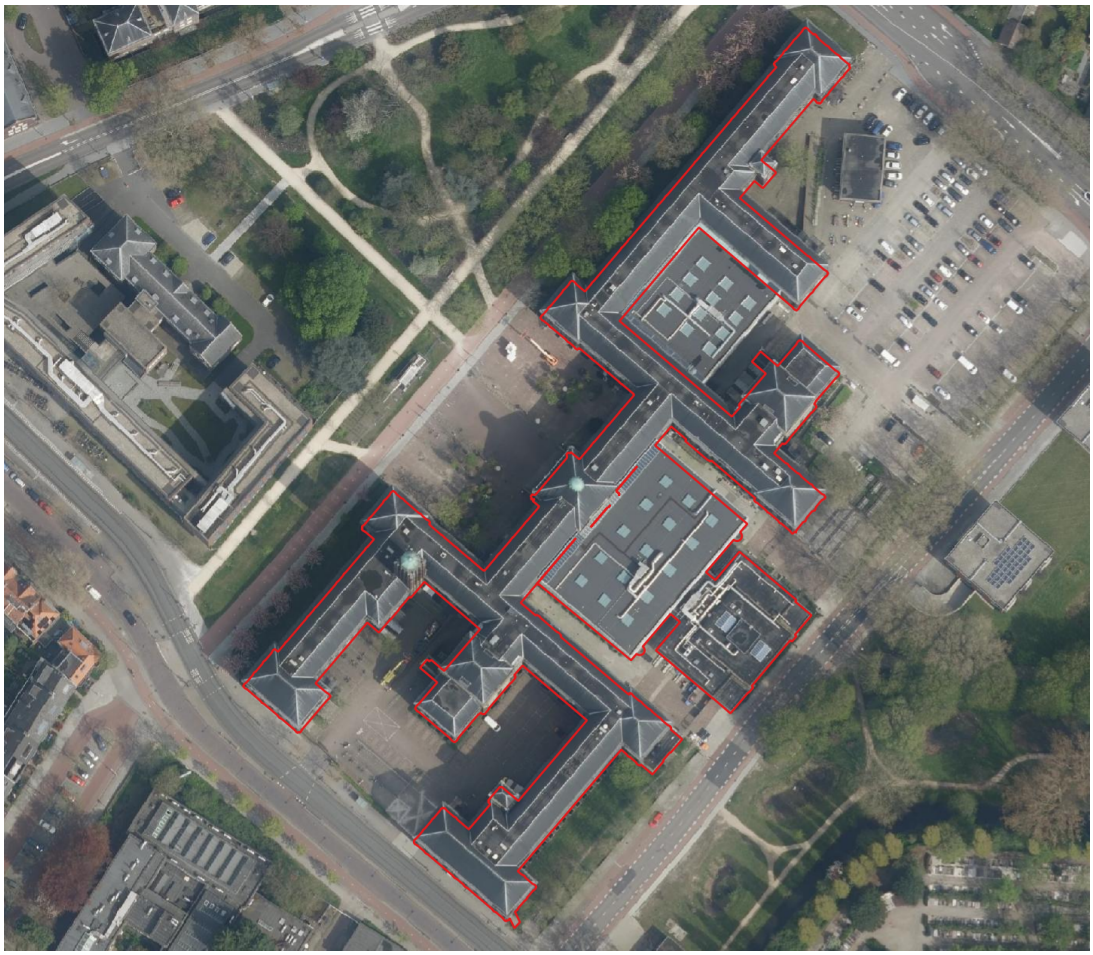


Figure C.2.: bk performance check region

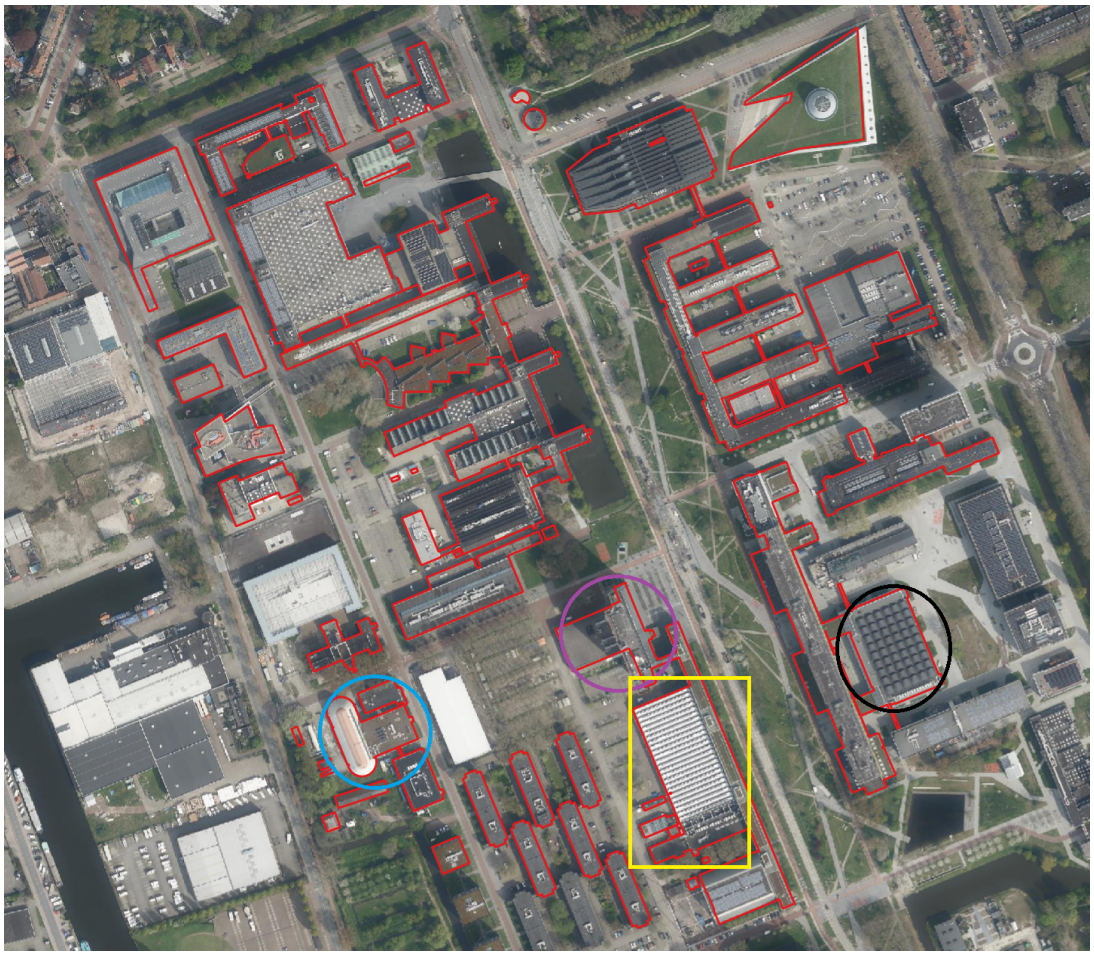


Figure C.3.: main campus buildings performance check region

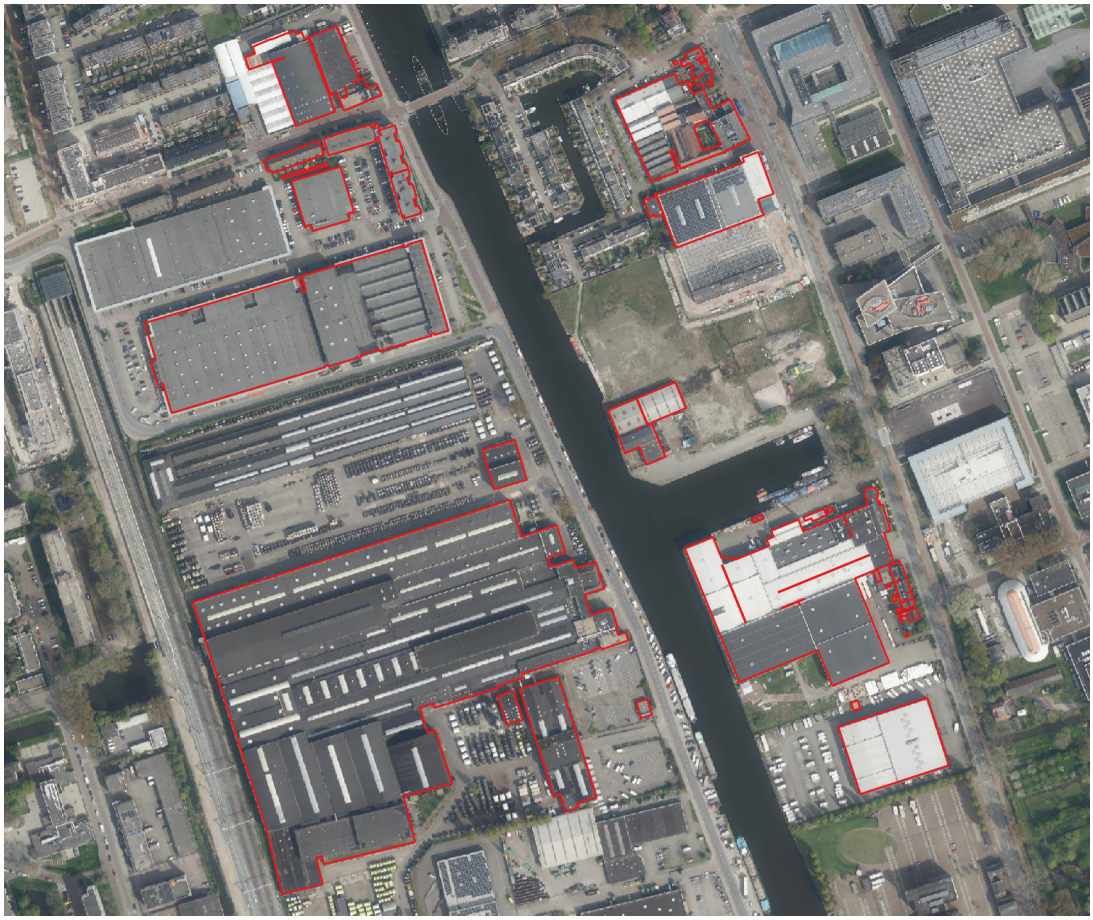


Figure C.4.: industrial performance check region

Figure C.5.: residential performance check region

C.3. Pixel-level Predictions

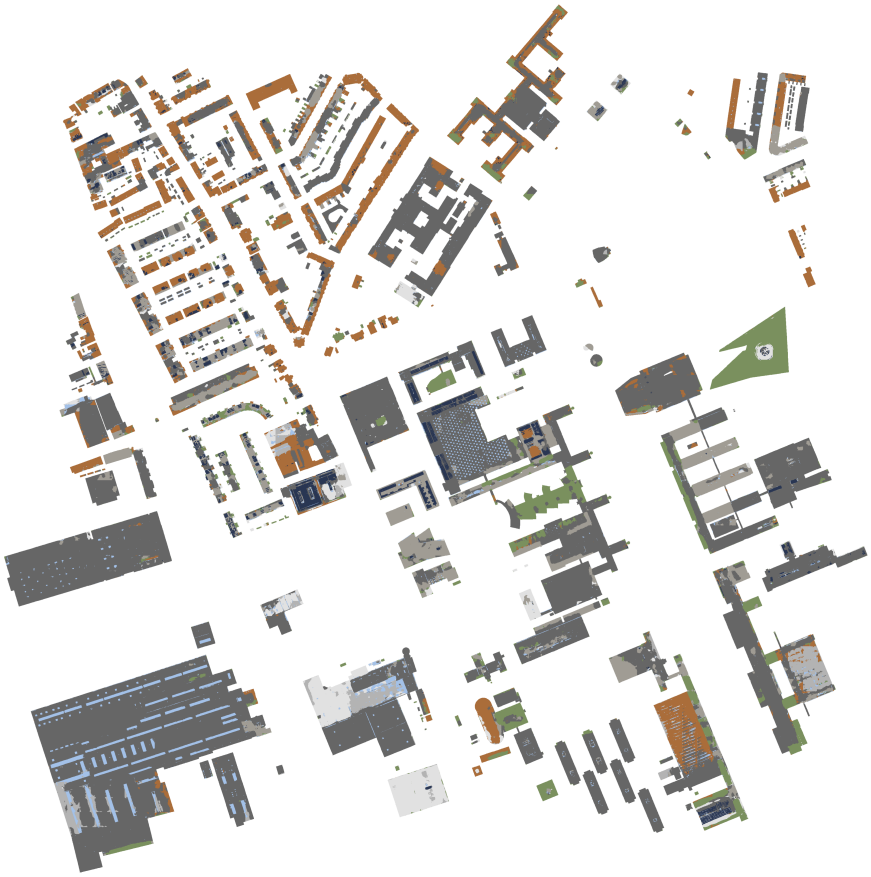
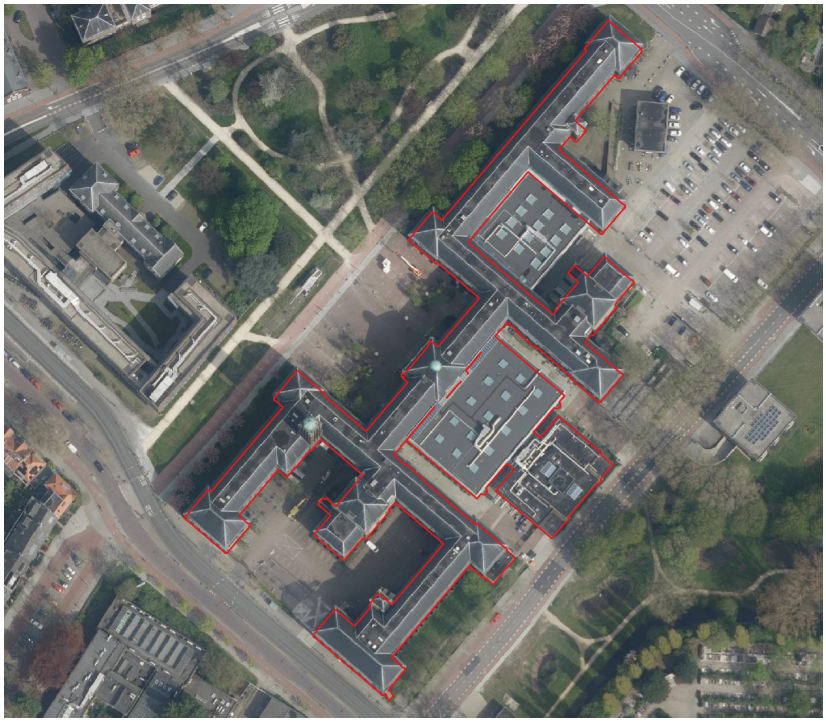


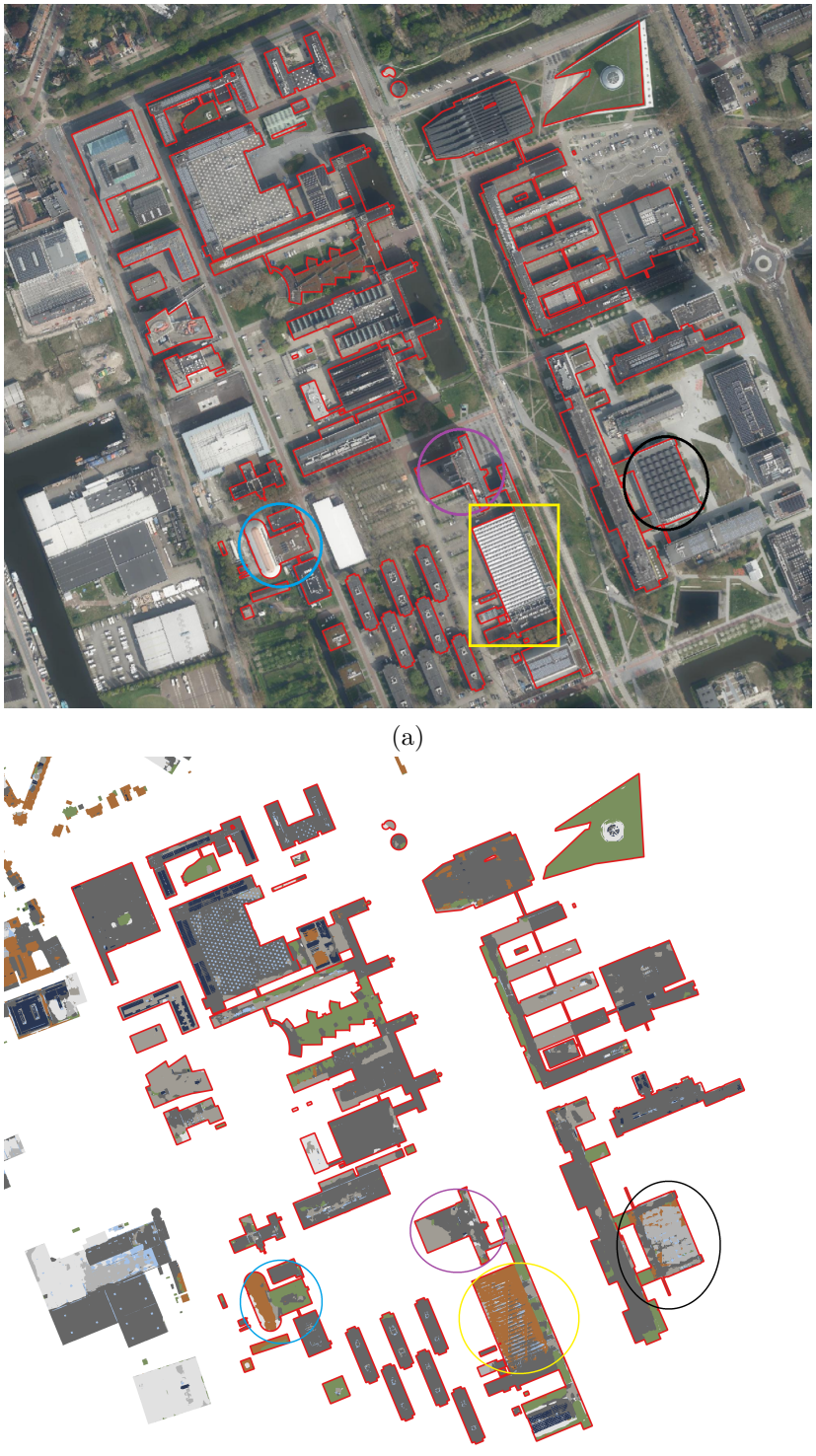
Figure C.6.: Pixel-level predictions on the qualitative performance evaluation tile (Section 4.5). color-label associations available in Table 4.1.



(a)

(b)

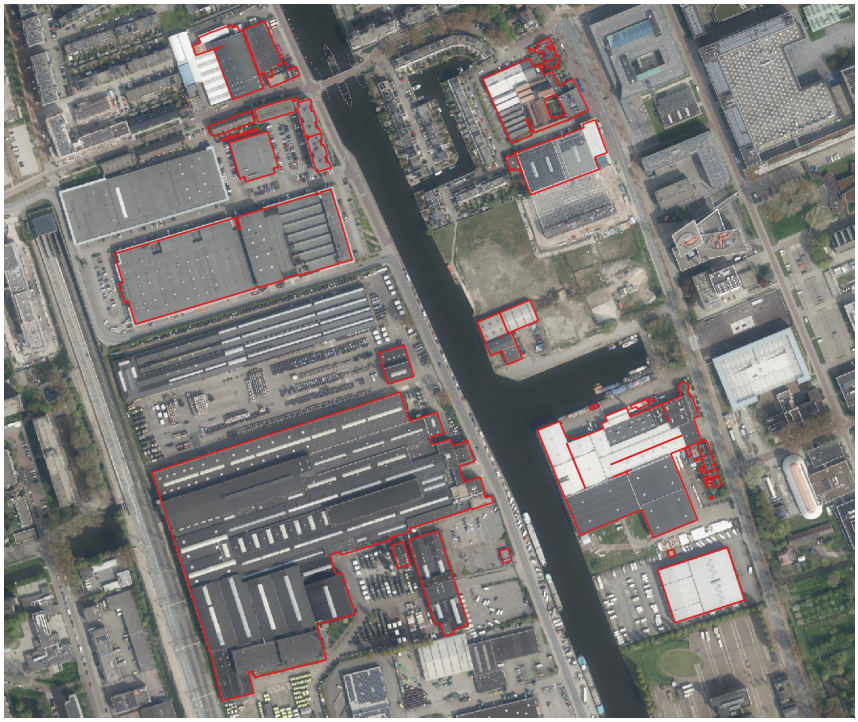
Figure C.7.: Ground truth and corresponding predictions for the building of the Faculty of Architecture and the Built Environment.color-label associations available in Table 4.1.



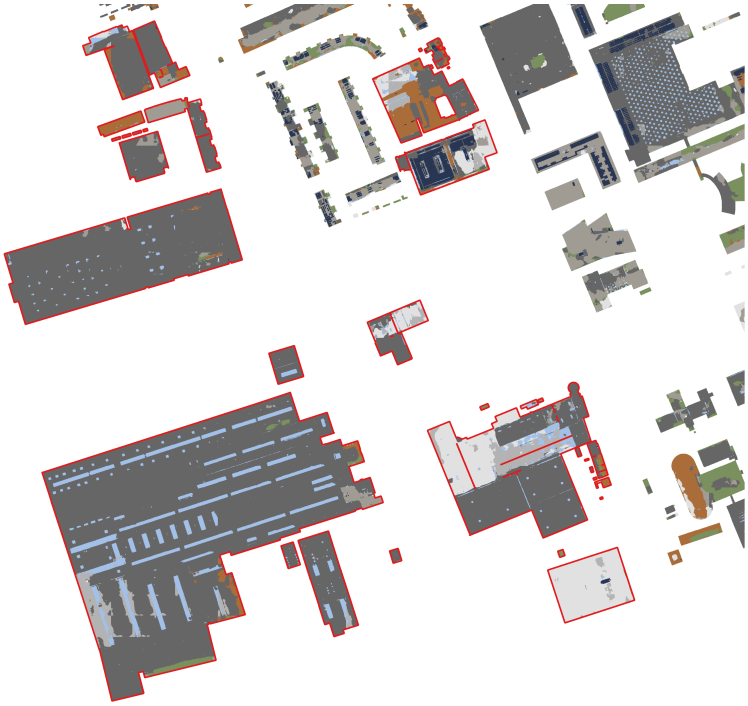
(a)

(b)

Figure C.8.: Ground truth and corresponding predictions for the building of the main campus buildings.color-label associations available in Table 4.1.



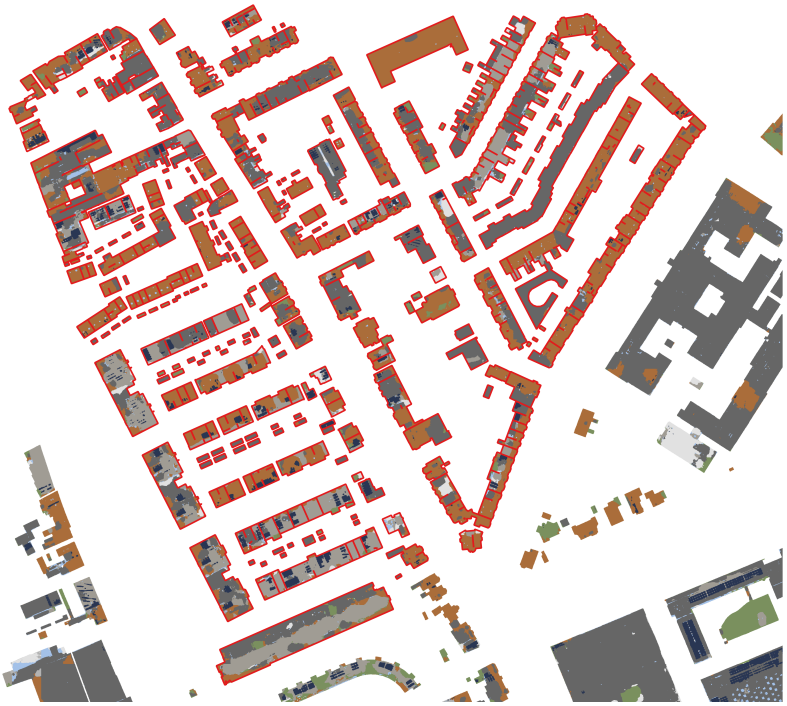
(a)



(b)

Figure C.9.: Ground truth and corresponding predictions for the industrial area.color-label associations available in Table 4.1.

(a)



(b)

Figure C.10.: Ground truth and corresponding predictions for the residential area.color-label associations available in Table 4.1.

C.4. Generalised Performance Evaluation

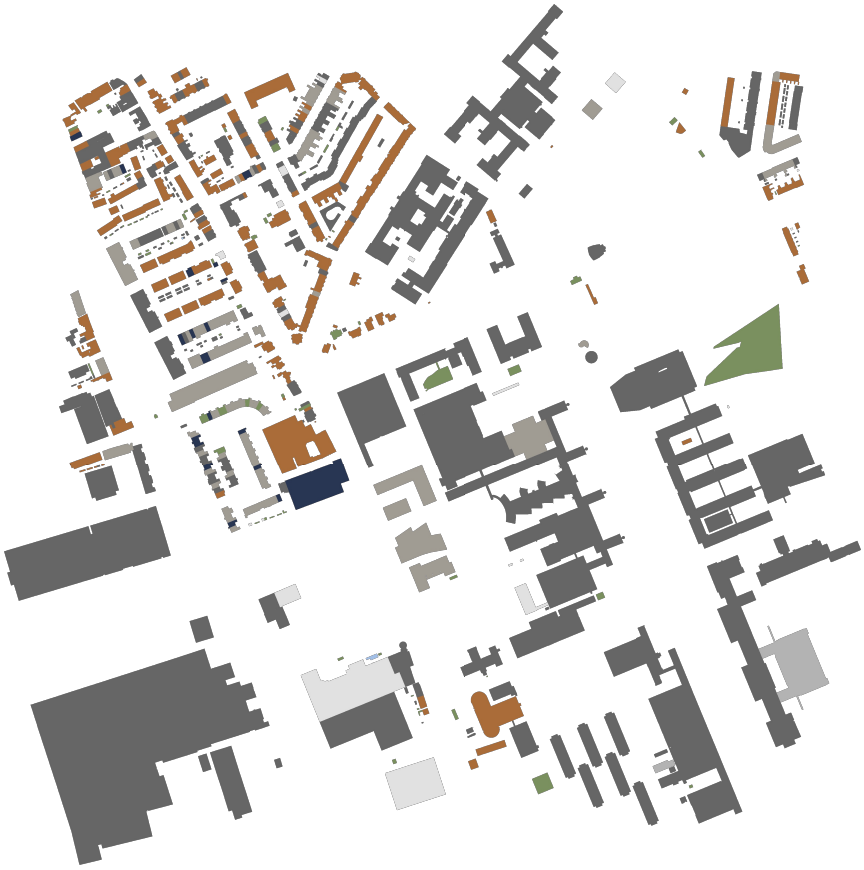


Figure C.11.: LoD1.2 predictions on the qualitative performance evaluation tile (Section 4.5). color-label associations available in Table 4.1.

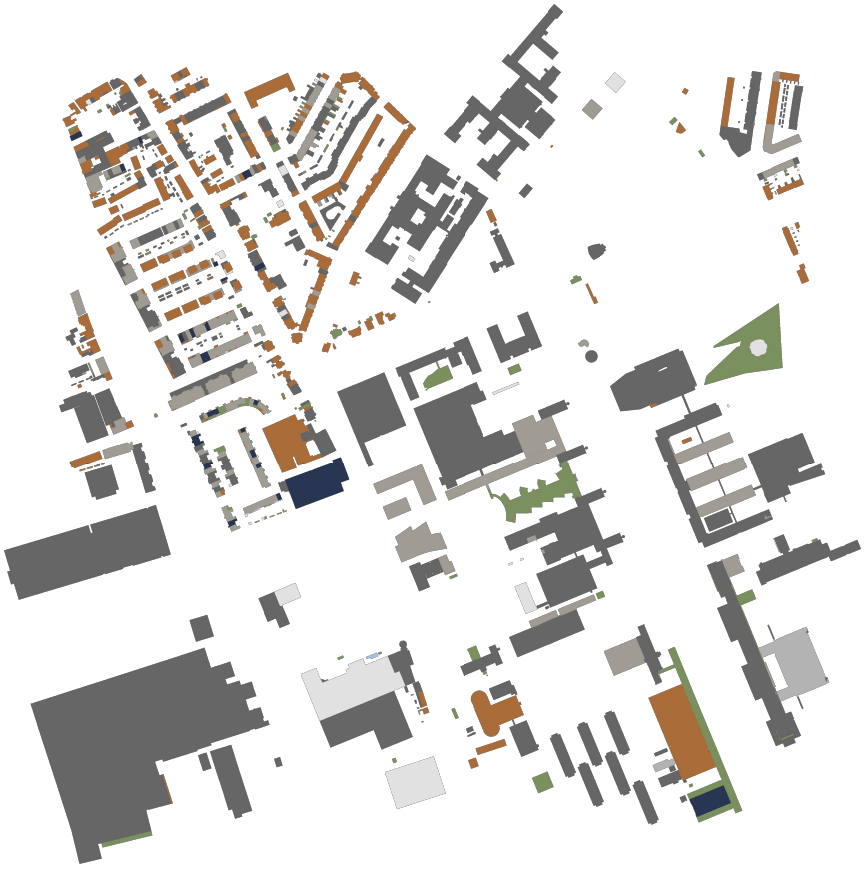


Figure C.12.: LoD1.3 predictions on the qualitative performance evaluation tile (Section 4.5). color-label associations available in Table 4.1.

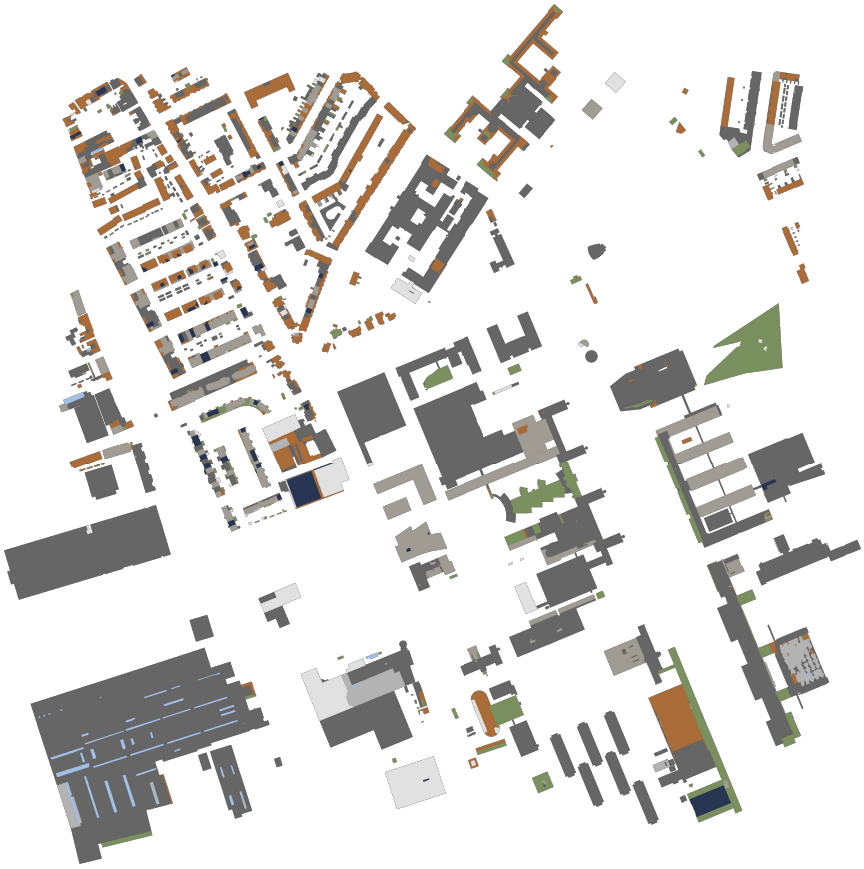


Figure C.13.: LoD2.2 predictions on the qualitative performance evaluation tile (Section 4.5). color-label associations available in Table 4.1.

D. Reproducibility self-assessment

D.1. Marks for each of the criteria

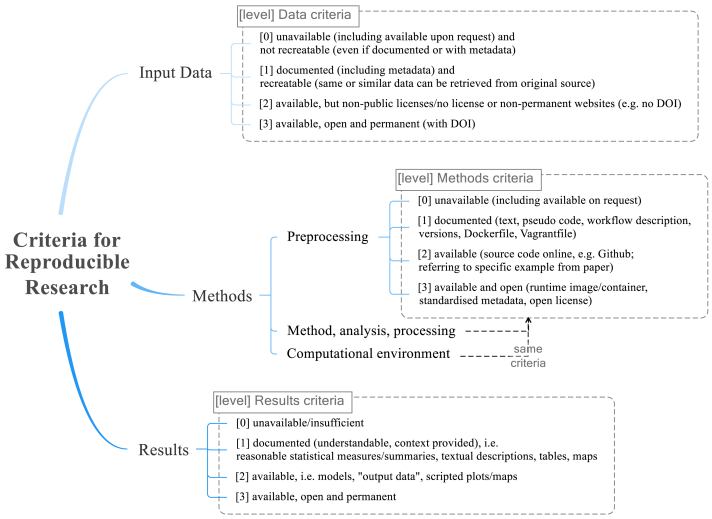


Figure D.1.: Reproducibility criteria to be assessed.

1. input data: 3
2. preprocessing: 2
3. methods: 2
4. computational environment: 2
5. results: 3

D.2. Self-reflection

The dataset is permanently stored online on Roboflow Universe under CC BY 4.0 Anyone can clone and extend it. Documentation for producing and annotating the dataset is provided in full in this document. In terms of methods, all aspects are scored with a 2 because the source code is currently not organised or documented properly. Otherwise, the criteria for a score of 3 are fulfilled. Finally, the results aspect is scored with a 3 because the model parameters and scripts to run relevant experiments are provided along with the source code.

Bibliography

3D geoinformation group & 3DGI. (2024, October). 3DBAG Documentation.

Abbasi, M., Mostafa, S., Vieira, A. S., Patorniti, N., & Stewart, R. A. (2022). Mapping Roofing with Asbestos-Containing Material by Using Remote Sensing Imagery and Machine Learning-Based Image Classification: A State-of-the-Art Review. *Sustainability*, *14*(13). https://doi.org/10.3390/su14138068

Abriha, D., Kovács, Z., Ninsawat, S., Bertalan, L., Bertalan-Balazs, B., & Szabo, S. (2018). Identification of roofing materials with Discriminant Function Analysis and Random Forest classifiers on pan-sharpened WorldView-2 imagery – a comparison. *Hungarian Geographical Bulletin*, *67*, 375–392. https://doi.org/10.15201/hungeobull.67.4.6

Baharav, T. Z., Kamath, G. M., Tse, D. N., & Shomorony, I. (2020). Spectral Jaccard Similarity: A New Approach to Estimating Pairwise Sequence Alignments. *Patterns*, *1*(6), 100081. https://doi.org/https://doi.org/10.1016/j.patter.2020.100081

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, *6*(1), 20–29. https://doi.org/10.1145/1007730.1007735

Batista, G. E. d. A. P. A., Bazzan, A. L. C., & Monard, M. C. (2003). Balancing training data for automated annotation of keywords:: a case study. *Revista Tecnologia da Informação*.

Bergstra, J., Yamins, D., & Cox, D. D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *30th International Conference on Machine Learning, ICML 2013*, (PART 1).

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 24). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf

Biljecki, F., Ledoux, H., & Stoter, J. (2016). An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*, 25–37. https://doi.org/10.1016/j.compenvurbsys.2016.04.005

Boguszewski, A., Batorski, D., Ziemba-Jankowska, N., Dziedzic, T., & Zambrzycka, A. (2021). LandCover. ai: Dataset for automatic mapping of buildings, woodlands, water and roads from aerial imagery. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1102–1110.

Braun, A., Warth, G., Bachofer, F., & Hochschild, V. (2019). Identification of roof materials in high-resolution multispectral images for urban planning and mon-

itoring. *2019 Joint Urban Remote Sensing Event (JURSE)*, 1–4. https://doi.org/10.1109/JURSE.2019.8809026

Burdett, G. (2006, November). *Investigation of the chrysotile fibres in an asbestos cement sample* (tech. rep.). Health and Safety Laboratory. Derbyshire, United Kingdom.

Burgert, T., & Demir, B. (2024). Estimating Physical Information Consistency of Channel Data Augmentation for Remote Sensing Images. *arXiv e-prints*, arXiv:2403.14547. https://doi.org/10.48550/arXiv.2403.14547

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2011). SMOTE: Synthetic Minority Over-sampling Technique. *arXiv e-prints*, arXiv:1106.1813. https://doi.org/10.48550/arXiv.1106.1813

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2016). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *arXiv e-prints*, arXiv:1606.00915. https://doi.org/10.48550/arXiv.1606.00915

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv e-prints*, arXiv:1706.05587. https://doi.org/10.48550/arXiv.1706.05587

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *arXiv e-prints*, arXiv:1802.02611. https://doi.org/10.48550/arXiv.1802.02611

Chisense, C. (2012). CLASSIFICATION OF ROOF MATERIALS USING HYPERSPECTRAL DATA. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XXXIX-B7*, 103–107. https://doi.org/10.5194/isprsarchives-XXXIX-B7-103-2012

Chollet, F. (2016). Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv e-prints*, arXiv:1610.02357. https://doi.org/10.48550/arXiv.1610.02357

Cilia, C., Panigada, C., Rossini, M., Candiani, G., Pepe, M., & Colombo, R. (2015). Mapping of Asbestos Cement Roofs and Their Weathering Status Using Hyperspectral Aerial Images. *ISPRS International Journal of Geo-Information*, *4*(2), 928–941. https://doi.org/10.3390/ijgi4020928

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. *arXiv e-prints*, arXiv:1604.01685. https://doi.org/10.48550/arXiv.1604.01685

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable Convolutional Networks. *arXiv e-prints*, arXiv:1703.06211. https://doi.org/10.48550/arXiv.1703.06211

de Pinho, C. M. D., Fonseca, L. M. G., Korting, T. S., de Almeida, C. M., & Kux, H. J. H. (2012). Land-cover classification of an intra-urban environment using high-resolution images and object-based image analysis. *International Journal*

of Remote Sensing, *33*(19), 5973–5995. https://doi.org/10.1080/01431161.2012.675451

DeVries, T., & Taylor, G. W. (2017). Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv e-prints*, arXiv:1708.04552. https://doi.org/10.48550/arXiv.1708.04552

Dionelis, N., Pro, F., Maiano, L., Amerini, I., & Saux, B. L. (2024). Learning from Unlabelled Data with Transformers: Domain Adaptation for Semantic Segmentation of High Resolution Aerial Images. *arXiv preprint arXiv:2404.11299*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv e-prints*, arXiv:2010.11929. https://doi.org/10.48550/arXiv.2010.11929

Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv e-prints*, arXiv:1603.07285. https://doi.org/10.48550/arXiv.1603.07285

European Commission. (1999, July). The European Commission bans White Asbestos.

European Space Agency. (2024). IKONOS-2 - Earth Online. https://earth.esa.int/eogateway/missions/ikonos-2

Fan, T., Wang, G., Li, Y., & Wang, H. (2020). MA-Net: A Multi-Scale Attention Network for Liver and Tumor Segmentation. *IEEE Access*, *8*, 179656–179665. https://doi.org/10.1109/ACCESS.2020.3025372

Feng, S., & Fan, F. (2021). Analyzing the Effect of the Spectral Interference of Mixed Pixels Using Hyperspectral Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 1434–1446. https://doi.org/10.1109/JSTARS.2020.3045712

Fiumi, L., Campopiano, A., Casciardi, S., & Ramires, D. (2012). Method validation for the identification of asbestos–cement roofing. *Applied Geomatics*, *4*(1), 55–64. https://doi.org/10.1007/s12518-012-0078-0

Fiumi, L., Congedo, L., & Meoni, C. (2014). Developing expeditious methodology for mapping asbestos-cement roof coverings over the territory of Lazio Region. *Applied Geomatics*, *6*(1), 37–48. https://doi.org/10.1007/s12518-014-0124-1

Frassy, F., Candiani, G., Rusmini, M., Maianti, P., Marchesi, A., Nodari, F. R., Via, G. D., Albonico, C., & Gianinetto, M. (2014). Mapping Asbestos-Cement Roofing with Hyperspectral Remote Sensing over a Large Mountain Region of the Italian Western Alps. *Sensors*, *14*(9), 15900–15913. https://doi.org/10.3390/s140915900

Gao, W., Peters, R., & Stoter, J. (2024). Unsupervised Roofline Extraction from True Orthophotos for LoD2 Building Model Reconstruction. *Lecture Notes in Geoinformation and Cartography*. https://doi.org/10.1007/978-3-031-43699-4{_}27

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2018). ImageNet-trained CNNs are biased towards texture; increasing shape

bias improves accuracy and robustness. *arXiv e-prints*, arXiv:1811.12231. https://doi.org/10.48550/arXiv.1811.12231

Gibril, M. B. A., Shafri, H. Z. M., & Hamedianfar, A. (2017). New semi-automated mapping of asbestos cement roofs using rule-based object-based image analysis and Taguchi optimization technique from WorldView-2 images. *International Journal of Remote Sensing*, *38*(2), 467–491. https://doi.org/10.1080/01431161.2016.1266109

Godbole, V., Dahl, G. E., Gilmer, J., Shallue, C. J., & Nado, Z. (2023). Deep Learning Tuning Playbook. http://github.com/google-research/tuning_playbook

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. *arXiv e-prints*, arXiv:1706.04599. https://doi.org/10.48550/arXiv.1706.04599

Hamedianfar, A., Shafri, H., Mansor, S., & Ahmad, N. (2014). Improving detailed rule-based feature extraction of urban areas from WorldView-2 image and lidar data. *International Journal of Remote Sensing*, *35*. https://doi.org/10.1080/01431161.2013.879350

Hamedianfar, A., & Shafri, H. Z. M. (2014). Development of fuzzy rule-based parameters for urban object-oriented classification using very high resolution imagery. *Geocarto International*, *29*(3), 268–292. https://doi.org/10.1080/10106049.2012.760006

Hamedianfar, A., Shafri, H. Z. M., Mansor, S., & Ahmad, N. (2014). Combining data mining algorithm and object-based image analysis for detailed urban mapping of hyperspectral images. *Journal of Applied Remote Sensing*, *8*(1), 085091. https://doi.org/10.1117/1.JRS.8.085091

Hamedianfar, A., & Shafri, H. Z. M. (2015). Detailed intra-urban mapping through transferable OBIA rule sets using WorldView-2 very-high-resolution satellite images. *International Journal of Remote Sensing*, *36*(13), 3380–3396. https://doi.org/10.1080/01431161.2015.1060645

Hayou, S., & Ayed, F. (2021). Regularization in ResNet with Stochastic Depth. *arXiv e-prints*, arXiv:2106.03091. https://doi.org/10.48550/arXiv.2106.03091

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv e-prints*, arXiv:1512.03385. https://doi.org/10.48550/arXiv.1512.03385

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., & Li, M. (2018). Bag of Tricks for Image Classification with Convolutional Neural Networks. *arXiv e-prints*, arXiv:1812.01187. https://doi.org/10.48550/arXiv.1812.01187

Het Waterschapshuis. (2024). BM5. https://www.beeldmateriaal.nl/

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. *arXiv e-prints*, arXiv:1905.02244. https://doi.org/10.48550/arXiv.1905.02244

Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2017). Squeeze-and-Excitation Networks. *arXiv e-prints*, arXiv:1709.01507. https://doi.org/10.48550/arXiv.1709.01507

Huang, B., Reichman, D., Collins, L. M., Bradbury, K., & Malof, J. M. (2018). Tiling and Stitching Segmentation Output for Remote Sensing: Basic Challenges and Recommendations. *arXiv e-prints*, arXiv:1805.12219. https://doi.org/10.48550/arXiv.1805.12219

Hutter, F., Hoos, H., & Leyton-Brown, K. (2014). An efficient approach for assessing hyperparameter importance. *31st International Conference on Machine Learning, ICML 2014*, *2*.

Ilehag, R., Bulatov, D., Helmholz, P., & Belton, D. (2018). CLASSIFICATION AND REPRESENTATION OF COMMONLY USED ROOFING MATERIAL USING MULTISENSORIAL AERIAL DATA. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLII-1*, 217–224. https://doi.org/10.5194/isprs-archives-XLII-1-217-2018

Karimi, D., & Salcudean, S. E. (2020). Reducing the Hausdorff Distance in Medical Image Segmentation with Convolutional Neural Networks. *IEEE Transactions on Medical Imaging*, *39*(2). https://doi.org/10.1109/TMI.2019.2930068

Kim, B. J., Choi, H., Jang, H., & Kim, S. W. (2023). Resolution-Aware Design of Atrous Rates for Semantic Segmentation Networks. *arXiv e-prints*, arXiv:2307.14179. https://doi.org/10.48550/arXiv.2307.14179

Kim, J., Bae, H., Kang, H., & Lee, S. G. (2021). CNN Algorithm for Roof Detection and Material Classification in Satellite Images. *Electronics*, *10*(13). https://doi.org/10.3390/electronics10131592

King, G., & Zeng, L. (2001). Logistic Regression in Rare Events Data. *Political Analysis*, *9*, 137163.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W. Y., Dollár, P., & Girshick, R. (2023). Segment Anything. *Proceedings of the IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV51070.2023.00371

Krówczyńska, M., Raczko, E., Staniszewska, N., & Wilk, E. (2020). Asbestos—Cement Roofing Identification Using Remote Sensing and Convolutional Neural Networks (CNNs). *Remote Sensing*, *12*(3). https://doi.org/10.3390/rs12030408

Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*, *30*(2), 195–215. https://doi.org/10.1023/A:1007452223027

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, *18*(17), 1–5. http://jmlr.org/papers/v18/16-365.html

Li, H., Xiong, P., An, J., & Wang, L. (2018). Pyramid Attention Network for Semantic Segmentation. *arXiv e-prints*, arXiv:1805.10180. https://doi.org/10.48550/arXiv.1805.10180

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2020). Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(2), 318–327. https://doi.org/10.1109/TPAMI.2018.2858826

Ling, C. X., & Li, C. (1998). Data mining for direct marketing: problems and solutions. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 73–79.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. *arXiv e-prints*, arXiv:2201.03545. https://doi.org/10.48550/arXiv.2201.03545

Loshchilov, I., & Hutter, F. (2017). Decoupled Weight Decay Regularization. *arXiv e-prints*, arXiv:1711.05101. https://doi.org/10.48550/arXiv.1711.05101

Lynn, T. (2023). Launch: Label Data with Segment Anything in Roboflow. https://blog.roboflow.com/label-data-segment-anything-model-sam/

Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 3226–3229. https://doi.org/10.1109/IGARSS.2017.8127684

Makarova, A., Shen, H., Perrone, V., Klein, A., Faddoul, J. B., Krause, A., Seeger, M., & Archambeau, C. (2022, September). Automatic Termination for Hyperparameter Optimization. In I. Guyon, M. Lindauer, M. van der Schaar, F. Hutter & R. Garnett (Eds.), *Proceedings of the first international conference on automated machine learning* (pp. 7/1–21, Vol. 188). PMLR. https://proceedings.mlr.press/v188/makarova22a.html

Mañas, O., Lacoste, A., Giró-I-Nieto, X., Vazquez, D., & Rodriguez, P. (2021). Seasonal Contrast: Unsupervised Pre-Training from Uncurated Remote Sensing Data. *Proceedings of the IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV48922.2021.00928

Mantas, D. (2024a, October). RoofSense. https://github.com/DimitrisMantas/RoofSense

Mantas, D. (2024b, October). RoofSense Dataset. https://universe.roboflow.com/my-workspace-lg4pq/roofsense-3

Mather, P. M., & Koch, M. (2022, April). *Computer Processing of Remotely-Sensed Images* (5th). Wiley-Blackwell.

Maxar Technologies. (2024a). WorldView-2 Data Sheet. https://resources.maxar.com/data-sheets/worldview-2

Maxar Technologies. (2024b). WorldView-3 Data Sheet. https://resources.maxar.com/data-sheets/worldview-3

Norman, M., Shafri, H. Z. M., Mansor, S., Yusuf, B., & Radzali, N. A. W. M. (2020). Fusion of multispectral imagery and LiDAR data for roofing materials and roofing surface conditions assessment. *International Journal of Remote Sensing*, *41*(18), 7090–7111. https://doi.org/10.1080/01431161.2020.1754493

Oke, T. R. (1982). The energetic basis of the urban heat island. *Quarterly Journal of the Royal Meteorological Society*, *108*(455), 1–24. https://doi.org/https://doi.org/10.1002/qj.49710845502

Osińska-Skotak, K., & Ostrowski, W. (2015). Use of satellite and ALS data for classification of roofing materials on the example of asbestos roof tile identification. *Technical Sciences*, *18(4)*.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., & Brunk, C. (1994). Reducing Misclassification Costs. In W. W. Cohen & H. Hirsh (Eds.), *Machine learning proceedings 1994* (pp. 217–225). Morgan Kaufmann. https://doi.org/https://doi.org/10.1016/B978-1-55860-335-6.50034-9

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., & Stoter, J. (2022). Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands. https://doi.org/10.14358/PERS.21-00032R2

Raczko, E., Krówczyńska, M., & Wilk, E. (2022). Asbestos roofing recognition by use of convolutional neural networks and high-resolution aerial imagery. Testing different scenarios. *Building and Environment*, *217*, 109092. https://doi.org/https://doi.org/10.1016/j.buildenv.2022.109092

Rajaraman, A., & Ullman, J. D. (2011). Data Mining. In *Mining of massive datasets* (pp. 1–17). Cambridge University Press.

RIEGL Laser Measurement Systems GmbH. (2019, April). *LAS Extrabytes Implementation in RIEGL Software* (tech. rep.).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv e-prints*, arXiv:1505.04597. https://doi.org/10.48550/arXiv.1505.04597

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., & Breitkopf, U. (2012). THE ISPRS BENCHMARK ON URBAN OBJECT CLASSIFICATION AND 3D BUILDING RECONSTRUCTION. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *I-3*, 293–298. https://doi.org/10.5194/isprsannals-I-3-293-2012

Santos, C. L. B., Medina, R. P., & Taylar, J. V. (2023). Classification of Roof Construction Materials using Satellite Images with Convolutional Neural Network. *2023 International Conference on Digital Applications, Transformation & Economy (ICDATE)*, 1–5. https://doi.org/10.1109/ICDATE58146.2023.10248935

Sechidis, K., Tsoumakas, G., & Vlahavas, I. (2011). On the stratification of multi-label data. *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III*, 145–158.

Shafri, H. (2013). Development of a Generic Model for the Detection of Roof Materials Based on an Object-Based Approach Using WorldView-2 Satellite Imagery. *Advances in Remote Sensing*, *2*, 312–321. https://doi.org/10.4236/ars.2013.24034

Shanmugam, D., Blalock, D., Balakrishnan, G., & Guttag, J. (2020). Better Aggregation in Test-Time Augmentation. *arXiv e-prints*, arXiv:2011.11156. https://doi.org/10.48550/arXiv.2011.11156

Solovyev, R. (2020). Roof material classification from aerial imagery. *arXiv e-prints*, arXiv:2004.11482. https://doi.org/10.48550/arXiv.2004.11482

Stewart, A. J., Lehmann, N., Corley, I. A., Wang, Y., Chang, Y.-C., Braham, N. A. A., Sehgal, S., Robinson, C., & Banerjee, A. (2023). SSL4EO-L: Datasets and

Foundation Models for Landsat Imagery. https://arxiv.org/abs/2306.09424

Stewart, A. J., Robinson, C., Corley, I. A., Ortiz, A., Lavista Ferres, J. M., & Banerjee, A. (2022). TorchGeo: Deep Learning With Geospatial Data. *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 1–12. https://doi.org/10.1145/3557915.3560953

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10553 LNCS*. https://doi.org/10.1007/978-3-319-67558-9{_}28

Szabo, S., Burai, P., Kovács, Z., Szabó, G., Kerényi, A., Fazekas, I., Paládi, M., Buday, T., & Szabo, G. (2014). Testing algorithms for the identification of asbestos roofing based on hyperspectral data. *Environmental engineering and management journal*, *13*. https://doi.org/10.30638/eemj.2014.323

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). Going Deeper with Convolutions. *arXiv e-prints*, arXiv:1409.4842. https://doi.org/10.48550/arXiv.1409.4842

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the Inception Architecture for Computer Vision. *arXiv e-prints*, arXiv:1512.00567. https://doi.org/10.48550/arXiv.1512.00567

Szymański, P., & Kajdanowicz, T. (2017a). A network perspective on stratification of multi-label data. *First International Workshop on Learning with Imbalanced Domains: Theory and Applications*, 22–35.

Szymański, P., & Kajdanowicz, T. (2017b). A scikit-based Python environment for performing multi-label classification. *arXiv e-prints*, arXiv:1702.01460. https://doi.org/10.48550/arXiv.1702.01460

Tommasini, M., Bacciottini, A., & Gherardelli, M. (2019). A QGIS Tool for Automatically Identifying Asbestos Roofing. *ISPRS International Journal of Geo-Information*, *8*(3). https://doi.org/10.3390/ijgi8030131

Trevisiol, F., Lambertini, A., Franci, F., & Mandanici, E. (2022). An Object-Oriented Approach to the Classification of Roofing Materials Using Very High-Resolution Satellite Stereo-Pairs. *Remote Sensing*, *14*(4). https://doi.org/10.3390/rs14040849

Unal, R., & Dean, E. B. (1990). Taguchi approach to design optimization for quality and cost: an overview. *1991 Annual conference of the international society of parametric analysts*.

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2019). ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. *arXiv e-prints*, arXiv:1910.03151. https://doi.org/10.48550/arXiv.1910.03151

Wang, Y., Braham, N. A. A., Xiong, Z., Liu, C., Albrecht, C. M., & Zhu, X. X. (2023). SSL4EO-S12: A large-scale multimodal, multitemporal dataset for self-supervised learning in Earth observation [Software and Data Sets]. *IEEE Geoscience*

and Remote Sensing Magazine, *11*(3). https://doi.org/10.1109/MGRS.2023.3281651

Watanabe, S. (2023). Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance. *arXiv e-prints*, arXiv:2304.11127. https://doi.org/10.48550/arXiv.2304.11127

Wightman, R. (2019). PyTorch Image Models. https://doi.org/10.5281/zenodo.4414861

Wightman, R., Touvron, H., & Jégou, H. (2021). ResNet strikes back: An improved training procedure in timm. *arXiv e-prints*, arXiv:2110.00476. https://doi.org/10.48550/arXiv.2110.00476

Willbo, M., Pirinen, A., Martinsson, J., Listo Zec, E., Mogren, O., & Nilsson, M. (2024). Impacts of Color and Texture Distortions on Earth Observation Data in Deep Learning. *arXiv e-prints*, arXiv:2403.04385. https://doi.org/10.48550/arXiv.2403.04385

Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I. S., & Xie, S. (2023). ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders. *arXiv e-prints*, arXiv:2301.00808. https://doi.org/10.48550/arXiv.2301.00808

Wu, Y., & He, K. (2018). Group Normalization. *arXiv e-prints*, arXiv:1803.08494. https://doi.org/10.48550/arXiv.1803.08494

Wyard, C., Beaumont, B., Grippa, T., Nys, G.-A., & Hallot, E. (2022, May). *Mapping roof materials using WV3 imagery and a state-of-the-art OBIA processing chain: application over Liège, Belgium*.

Wyard, C., Beaumont, B., Marion, R., Roupioz, L., Grippa, T., & Hallot, E. (2021, March). *Roof Material Mapping: Application Over Liège Using Open-Source Object-Based Supervised Classification Algorithms*.

Wyard, C., Fauvel, H., Palmaerts, B., Beaumont, B., & Hallot, E. (2023). From DL approach conception to operational product design : identifying roof materials for policy makers. *2023 Joint Urban Remote Sensing Event (JURSE)*, 1–4. https://doi.org/10.1109/JURSE57346.2023.10144142

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., & Sun, J. (2018). Unified Perceptual Parsing for Scene Understanding. *arXiv e-prints*, arXiv:1807.10221. https://doi.org/10.48550/arXiv.1807.10221

Yang, Y., Hallman, S., Ramanan, D., & Fowlkes, C. C. (2012). Layered Object Models for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(9), 1731–1743. https://doi.org/10.1109/TPAMI.2011.208

Yun, S., Han, D., Chun, S., Oh, S. J., Yoo, Y., & Choe, J. (2019). CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6022–6031. https://doi.org/10.1109/ICCV.2019.00612

Zevenbergen, L. W., & Thorne, C. R. (1987). Quantitative analysis of land surface topography. *Earth Surface Processes and Landforms*, *12*(1), 47–56. https://doi.org/https://doi.org/10.1002/esp.3290120107

Zhang, R. (2019). Making Convolutional Networks Shift-Invariant Again. *arXiv e-prints*, arXiv:1904.11486. https://doi.org/10.48550/arXiv.1904.11486

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2016). Pyramid Scene Parsing Network. *arXiv e-prints*, arXiv:1612.01105. https://doi.org/10.48550/arXiv.1612.01105

Colophon

This document was typeset in LaTeXusing the `scrbook` class of the KOMA-Scriptbundle by Frank Neukam, Markus Kohm, and Axel Kielhorn. The typeface used is Computer Modern by Donald Knuth.

TU Delft Delft University of Technology