TU Delft
Delft University of Technology
Faculty of Architecture
and the Built Environment

PHYSHADE-Net: Leveraging Geometric-Priors in Physics-Guided Neural Networks for Building Shadow Segmentation and Height Estimation

Lars C. Huizer

Delft University of Technology
Faculty of Architecture
and the Built Environment

Delft University of Technology

PHYSHADE-Net: Leveraging Geometric-Priors in Physics-Guided Neural Networks for Building Shadow Segmentation and Height Estimation

Master's Thesis - P5

To fulfill the requirements for the degree of
Master of Science in Geomatics
at Delft University of Technology under the supervision of
Dr. A. Rafiee (Geomatics, Delft University of Technology)
and
Ir. E. Verbree (Geomatics, Delft University of Technology)

Lars C. Huizer

June 19, 2025

Colophon

Title	*PHYSHADE-Net: Leveraging Geometric-Priors in Physics-Guided Neural Networks for Building Shadow Segmentation and Height Estimation*
Author	Lars Cornelis Huizer
Programme	Msc. Geomatics
University	Delft University of Technology
Supervisors	Dr. Azarakhsh Rafiee & Ir. Edward Verbree
Version	Final Version
Date	June 19, 2025

This document was typeset with LaTeXusing the Overleaf editor. The template for this document was adapted from Manvi Agarwal's template made available on the Overleaf website. The diagrams for the PHYSHADE pipelines were created using the draw.io software. Creation of various figures was done with Affinity Designer. The maps were created using QGIS. Countless of Python libraries not limited to Matplotlib, Pytorch, Pandas and GeoPandas among various SciPy modules were used to make this work possible. Zotero was used to manage the bibliography.

Acknowledgments

This masters thesis is the culmination of my academic career thus far, and I'm proud to have brought it to this point. Firstly, I would like to give my thanks to my supervisors Dr. Azarakhsh Rafiee and Ir. Edward Verbree for their support throughout this thesis. Secondly I would like to thank Dr. Weixiao Gao for generously agreeing to co-read my thesis on such short notice and for his thoughtful comments.

This thesis would not have been possible without the ongoing encouragement and help from my friends. I am especially grateful to Casper, Florens, Hjalmar, Lucas, Raisa, Tim, and Wiert, whose kindness and moral support kept me grounded during the more difficult stages of this project.

Finally, I would like to express my heartfelt gratitude to my brother and my parents for their unwavering belief in me, and to everyone who, in big or small ways, has contributed to my growth over the years.

Abstract

Building heights are important information for a variety of subjects, such as wind analysis, energy demand simulations and solar potential assessment, yet large-scale LiDAR scanning is costly. This thesis introduces PHYSHADE: a set of physics-guided U-Net-based models. It employs shadow projections derived from building footprints and solar geometry into an aerial image shadow segmentation pipeline, for the purposes of building shadow extraction and consequently the large-scale estimation of building heights. Thirty-five aerial images in the Netherlands were manually annotated for buildings and their associated buildings. By employing transfer learning, based on a general purpose shadow-segmentation model, a total of 130 models were trained, which can be categorized into three different implementations of PHYSHADE. Through these different configurations, the performance impact of the various methods of addition of pseudo-shadows to the models was ablated. Afterwards, the best-performing PHYSHADE configurations were used with a raycasting algorithm to convert shadow lengths and solar altitudes back into building heights. The inclusion of pseudo-shadows lifted the mean Dice scores from 0.53 to 0.85, with an average gain of 0.32 and statistical significance across different folds. Physics-guided loss, based on the pseudo-shadows, was not found to be significantly different in most cases, whilst hurting model performance in some cases compared to the pseudo-shadow enabled models. On six out-of-fold test tiles the best PHYSHADE variants retained Dice scores of 0.72 – 0.95, although recall declined in one winter scene. Finally, height estimation on these tiles using the inference from the best PHYSHADE variants resulted in mean RMSE of $\approx 1.9m$ and MAE of $\approx 1.5m$. While its application needs to be tested in broader contexts, PHYSHADE offers a viable low-cost complement to LiDAR for building height estimation.

Keywords: Building Height Estimation, Shadow Segmentation, Physics-Informed Neural Network, PHYSHADE, Aerial Imagery, U-Net, Remote Sensing

List of Abbreviations

In alphabetical order:

3D BAG	3D Basisregistratie Adressen en Gebouwen
AISD	Aerial Imagery dataset for Shadow Detection
AHN	Actueel Hoogtebestand Nederland
BCE	Binary Cross-Entropy
CNN	Convolutional Neural Network
CRF	Conditional Random Field (Dense CRF)
DSM	Digital Surface Model
DTM	Digital Terrain Model
DSSDNet	Deeply Supervised Shadow Detection Network
HYB	Hybrid configuration (RGBS input + physics-guided loss)
IoU	Intersection-over-Union metric
LiDAR	Light Detection and Ranging
MAE	Mean Absolute Error
PHYSHADE	Physics-guided Shadow-segmentation model suite
PINN	Physics-Informed Neural Network
RGB	Red–Green–Blue colour channels
RGBS	RGB plus Pseudo-Shadow channel
RMSE	Root Mean Square Error
U-Net	U-shaped Convolutional Network

Contents

List of Figures

List of Tables

1 Introduction

Building-height information is a useful metric in a variety of fields. It can be used to construct datasets such as the 3D BAG (Peters et al., 2022), extruding two-dimensional footprints into three dimensional structures by matching up the planar coordinates with heights from a digital surface model (DSM). In turn, these datasets can then be used to power e.g. energy demand simulations (León-Sánchez et al., 2021), solar potential assessment for solar panels (Apra et al., 2021) among other use cases where a digital twin may be useful. In broader societal terms, building height is a vital metric for the simulation of microclimates in urban environments through windflow analysis (Ng et al., 2011), which is an important topic of study in the global context of climate change.

A popular method for inferring surface heights for both buildings and terrain is through LiDAR. In the Netherlands, the "Actueel Hoogtebestand Nederland" (freely translated to "Current Dutch Elevation Record", often abbreviated to AHN) is a national dataset collected either by helicopter or airplane, with the fourth edition of the dataset sporting a height accuracy of up to 5 cm and an estimated average point density of 10-14 points per square meter (AHN, 2020).

While the AHN and similar point cloud datasets collected through airborne laser scanning work well for determining the height of broad and well-defined objects such as buildings, they are often comparatively expensive to collect at scale and require significant investment costs, meaning that if such data is not available publicly it is difficult for individuals to finance. While the Dutch government is able to freely provide such data, countries with financial resource limitations may find LiDAR as a method cost prohibitive, leading to less data being available. As such, finding new methods for the estimation of building heights at scale would promote economic equity, which in turn would provide opportunities for those economically disadvantaged to have a more direct impact at helping shape their communities through scientific discovery, informed policy, education and awareness raising (Craglia & Shanley, 2015).

Democratizing data access is therefore crucial, which results in a persistent need for new methods of data collection. Given the economic limitations inherent to building height data collection through LiDAR, an opportunity lies in the exploration of alternative methods that rely on cheaper and more accessible data sources. One such approach is through the inference and processing of building shadows from aerial imagery. By measuring the length of the shadow and multiplying it with the tangent of the sun's altitude, it becomes possible to calculate the height of the object that casts the shadow. Since data collection through e.g. aerial photography and satellite imagery is comparatively cheaper than airborne laser scanning, they can be a viable alternative for height estimation.

1.1 Problem Statement

In order to estimate building heights through shadows, the accurate segmentation of building shadows from aerial imagery is required. However, large scale shadow detection can struggle in terms of robustness. For example, classical filtering techniques which historically were the most performant (Adeline et al., 2013) may require manual selection of thresholding parameters to lead to accurate segmentation results, which turns labour intensive if large amounts of imagery need to be processed. Recent developments however in the field of Artificial Intelligence and the advent of deep-learning based solutions have led to significant improvements in shadow detection and segmentation accuracy, resulting in comparable accuracies to the filter-based methods (Luo et al., 2020). However, the accurate extraction of shadows from aerial imagery remains difficult for a number of reasons. For example, time of day and illumination have a large influence on contrast, colour and saturation which may lead to variations in segmentation performance if the method employed is not robust to them. Other

issues that degrade segmentation quality are occlusions from objects such as trees and street furniture, atmospheric effects such as fog that can reduce the contrast and reflectance characteristics of the surfaces in the imagery.

Given these challenges, the improvement of shadow segmentation could potentially be achieved by combining multiple data sources to compensate for the individual shortcomings of the data. Conventional neural networks (CNNs) generally rely on large-scale training using input-output data pairs and relies purely on statistical inference of the data in itself. However, these models have no situational context of the site or the conditions under which the data came to be. Since the existence of shadows is dependent on a combination of solar characteristics (i.e. height and direction) and the geometry of the object (i.e. the building) casting the shadows, it follows that the position and boundary of a shadow would become easier to predict if a CNN is given the context that allows a shadow to be formed.

Building on this notion, an opportunity lies in taking inspiration from Physics-Informed Neural Networks (PINNs) (Raissi et al., 2017) which use known rules and laws that describe a given dataset to regularize the learning and inference process to improve upon the generalisability of the models, while also requiring lower amounts of training data. For example, PINNs have been employed to solve fluid dynamics problems such as in the work by Jin et al. (2021), where promising results in flow simulation were delivered by giving the model knowledge of fluid dynamics through the inclusion of the Navier-Stokes equations.

For the segmentation of building shadows, the use of pre-calculated shadow projections based upon building footprints (henceforth *"geometrical priors"* or *"pseudo-shadows"*) is hypothesized to serve as a fitting method of regularization, thus incorporating real-world geometrical context into the model to reach more accurate and robust segmentation than normal segmentation would achieve.

As of writing, no papers have yet considered the explicit usage of building footprints as geometric priors for shadow segmentation. While Masquil et al. (2025) have a similar method where they predict shadows by raycasting based upon combining the solar azimuth, solar altitude and a DSM, it presupposes that one has access to a DSM which goes against the purpose of estimating heights through shadows as a low-cost alternative to other methods.

Thus, following the ever-existing need for alternative methods for data collection, this thesis will propose a new set of physics-guided CNNs named "PHYSHADE", which will use domain-informed geometrical priors in the form of pre-calculated shadow projections derived from building footprints and solar positions as an additional mode for regularization and as input. The end goal is for PHYSHADE to see an increase in quality of shadow inference compared to its non-physics-guided counterparts, with the idea that this in turn will lead to robust and usable height estimations at scale. For the height estimations, an algorithm based on raycasting within rasters is proposed that can be used to estimate building heights and relate them back to the relevant buildings.

1.2 Research Questions

To address the gap in research as posed above, this thesis will be guided by the following research questions:

Q Main. **How and to what extent does injecting geometric priors in the form of shadow masks derived from building footprints into a U-Net architecture improve the accuracy and robustness of building-shadow segmentation in aerial imagery?**

Q1. What is the baseline performance of an RGB U-Net trained on the Luo et al. (2020) data set when evaluated on Dutch aerial imagery and compared to the original dataset?

Q2. How does adding the pre-calculated shadow mask channel derived from building footprints affect segmentation accuracy across different urban morphologies, seasons, and solar geometries?

Q3. Which loss formulations and weighting schemes most effectively balance appearance-based learning with the geometric prior?

Q4. How effective are the inferred building shadows at estimating building height using the raster-based raycasting algorithm?

1.3 Research Scope

This thesis will focus on the creation of a Physics-Guided Neural Network designed to segment shadows from aerial photography and the subsequent estimation of building heights from these shadows. The study will be limited to:

- Adapting an existing CNN model (U-Net) for shadow segmentation, thereby using an existing neural architecture.
- The use of pre-calculated shadow masks based upon building footprints as site-specific priors for the informing and regularization of CNNs.
- The evaluation of performance of the newly proposed models using default metrics such as Dice score among others.
- Using aerial photography datasets with known metadata (solar angle, shadow position), in particular the imagery as provided by Beeldmateriaal (Nederland, 2023).
- The association of buildings with their cast shadows, thereafter using raycasting-based height estimation algorithms.

As such, this study **will not** focus on:

- Object classification except for shadow segmentation.
- Exploring physics-based constraints other than the usage of geometric priors that can be generated from building footprints for shadow segmentation, i.e. atmospheric scattering.

- Similarly, whereas true Physics-Informed Neural Networks make use of partial differential equations as a part of their loss functions, the models proposed in this thesis cannot be considered as ”true” PINNs because of the the absence of physics-grounded rules and equations integrated directly into the architecture. Instead, the proposed PHYSHADE models will be referred to as ”Physics-Guided” to reflect the added knowledge of solar positions combined with shadow-casting geometry.

2 Related Work

In order to contextualize the thesis, this section will review existing works and methods related to the estimation of object heights and shadow segmentation. Firstly, different methods for the segmentation of shadows from aerial imagery are discussed as these are a core component for the envisioned height estimation later on. Secondly, some relevant work in the field of AI computer vision is mentioned, followed by Physics-Informed Neural Networks. Finally, some height estimation and point cloud generation methods are discussed. In the end, a summary of the research gap is provided.

2.1 Classic Shadow Segmentation

According to Liasis and Stavrou (2016), the methods by which to detect shadows can be mostly described by two categories, namely the property-based methods and the model-based methods. With property-based methods, the spectral and spatial features of shadowed regions themselves are utilized, whereas model-based methods employ the use of additional site-specific information, such as the solar altitude and the object's geometry.

Filter methods are property-based methods that analyse shadows based on e.g. intensity values or textures. The most simple example are histogram-based techniques that work based on the assumption that sunlit and shadowed regions have a clear separation between them in terms histogram levels (Adeline et al., 2013). A threshold by which to segment the different areas may then either be set automatically through statistical means (Otsu, 1979) or may be selected manually (Yamazaki et al., 2009). Another example of such a filter includes Gabor filters (Granlund, 1978), which are linear filters consisting of a Gaussian function and a sine-wave sensitive to edges and ridges that are oriented in a particular direction.

Model-based approaches consider a priori knowledge about the site of study. If one has pre-gathered geometrical information about the object casting a shadow as well as knowledge on the then-current standing of the sun, segmentation would become easier since its shadows can be precisely calculated instead of needing to interpret images. An example of this is Volumetric Shadow Analysis as performed by Lee and Kim (2010) which assumes that the ground sample distance, elevation and solar azimuth are known. It works by generating a frame representing the building, and manually generating a shadow based on the frame. By adjusting the height of the frame, the simulated-projected shadow is matched up to the real shadow, with the frame height then serving as the actual height (see Figure 1).

2.2 Machine Learning-based Shadow Segmentation

In 2012, Adeline et al. (2013) conducted a comparative study where they evaluated and ranked the above mentioned shadow-detection techniques on their F-scores. The conclusion was that histogram thresholding using the methods of Nagao et al. (1979) performed the best, followed by a physics based method by Richter and Müller (2005), a support vector machine-based (SVM) method and respectively in last a K-means clustering method and the SMACC method.

	Property-based methods		**Physics-based methods**		**Machine-learning methods**
	Histogram thresholding (Nagao et al., 1979)	RGB combination model	Richter and Müller (2005) method	SMACC	SVM
Average F-score	92.5	87.5	90.0	83.9	87.7

Table 1: The results of the comparative study by Adeline et al. (2013)

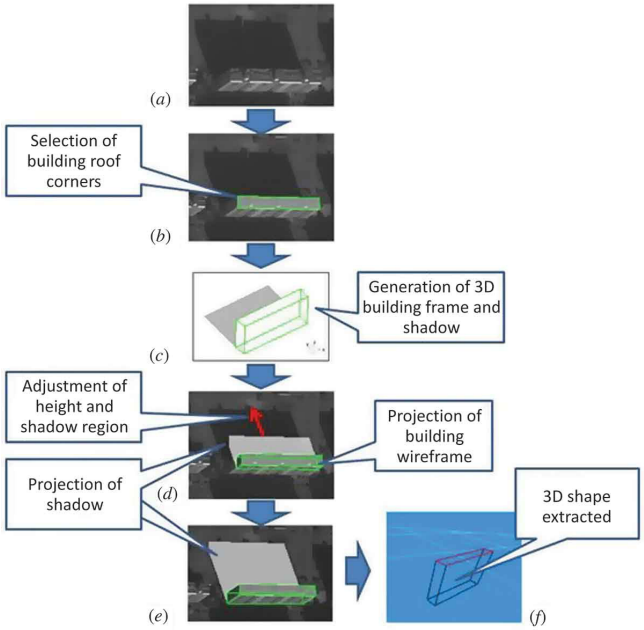

Figure 1: The processing steps used by VSA to extract building height. Image from Lee and Kim (2013)

Since then however, the traditional machine learning methods such as SVM and K-means have largely been outpaced by deep learning algorithms. Deep learning is a subset of machine learning where neural networks with multiple layers are used to interpret data. One of the foremost examples of deep learning methods is the Convolutional Neural Network (CNN), which are commonly employed for the purposes of computer vision. A CNN consists of multiple layers through which input data is fed, with each layer performing specific operations to extract and process features.

CNNs have been employed in various papers for shadow segmentation. For example, a widely used CNN architecture named U-Net (Ronneberger et al., 2015) was adapted by Jiao et al. (2020) for the segmentation of clouds and cloud shadows. This is then followed by Dense Conditional Random Field (Dense CRF) refinement, where the inference of a given pixel is not only dictated by its local context but by its global context (Krähenbühl & Koltun, 2011) as well. In another example, Luo et al. (2020) created a CNN named ”DSSDNET” in an encoder-decoder residual structure similar to U-Net, which was trained using an auxiliary supervision structure giving each level the ability to train directly on the ground truth, thus avoiding vanishing gradient issues. It differs from conventional CNNs and deep-supervision networks in the sense that the outputs of the intermediate auxiliary layers are combined and refined into a final prediction, instead of only fusing from the last levels of the network. In their testing, the model reached an average F-score of 91.78%, outcompeting other shadow detection methods such as U-Net (in its original form per Ronneberger et al. (2015)), which scored an F-score of 87.84%.

2.3 Building Footprint Segmentation

Since the goal of this thesis is to enhance the robustness of shadow segmentation by employing geometric priors consistent of building footprints combined with solar geometry, an ideal pipeline would involve the automatic segmentation of building footprints first, such that these building footprints may then help inform the shadow segmentation. As an example, in W. Li et al. (2019), automatic

building footprint extraction from aerial photography was explored using a U-Net based model architecture alongside some post-processing steps, reaching Dice scores of up to 0.704. Another paper by Kang et al. (2021) used a DeepLabV3+ based model and employed additional loss terms based on contrast, in the end reaching Dice scores of up to 0.8482 when segmenting for building footprints.

While these are promising results and could potentially be applied for the purposes of physics-guided building shadow segmentation as proposed in this thesis, the decision was made to instead manually annotate the building footprints. This way, potential downstream negatives of imperfect building footprints can be minimized whilst keeping the scope of the thesis tight. Further developments within the segmentation of building footprints could lead to a comprehensive pipeline combined with the physics-guided shadow segmentation methods where low-cost, large-scale building height estimation can be performed automatically.

2.4 Physics-Informed Neural Networks for Shadow Segmentation

In the current body of research, most of the work on shadow segmentation through neural networks is purely driven by a data-centric approach; that is, any inference or training is only based on the relations encoded within the data. However, if one has knowledge on the laws that this data has to abide by (often the laws of physics), it becomes possible to fine-tune the learning process to reach more robust and accurate results. This is the basis of Physics-Informed Neural Networks (PINNs), as originally proposed by Raissi et al. (2017).

Earlier mentioned in the comparative analysis by Adeline et al. (2013), one may have noticed that two physics-based methods were mentioned for the segmentation of shadows, with the method employed by Richter and Müller (2005) coming in second when compared to the other methods. Their method operates by the same philosophy as that a PINN would; both use physics-based constraints. However, they differ greatly since Richter and Müller (2005) in practice is a deterministic method that does not employ neural networks, whereas a PINN relies specifically on a neural network to generalize and infer without needing manual calibration under different circumstances.

As of writing, while PINNs have been used in various remote sensing tasks such as for the reconstruction of hyperspectral imagery from standard RGB images (Liu et al., 2022) or the estimation of surface temperatures in urban environments (Chen et al., 2022), its application to geospatial contexts remains limited. In this sense, the concept of employing building geometry and solar positions as a physical prior to regularize the segmentation of building shadows remains a novel application.

2.5 Structure-from-Motion

Structure-from-Motion (SfM) as originally proposed by Ullman and Brenner (1979) and brought in practice by Tomasi and Kanade (1992) is a method where by taking multiple images of the same object from different perspectives, sparse pointclouds can be generated by matching feature points seen in multiple images. Then, through Singular Value Decomposition (SVD) and awareness of the orthogonal nature of the imagery, the camera motions and 3D structure can be estimated.

Since then, SfM has been developed further and is able to produce relatively dense and accurate point clouds. However, the reconstruction of high quality 3D urban models and similar goals requires high amounts of multi-perspective imagery which more often than not is unavailable. Additionally, SfM pipelines often require a lot of computational power, making them unfeasible at larger scales. As such, while SfM does have a place in the reconstruction of urban geometry if accurate data is required, alternatives that have less data-intensive requirements and could work with single images such as the shadow-based height estimation proposed in this thesis remain relevant to explore.

2.6 Semi-Global Matching

Another method also relying on epipolar geometry like SfM is Semi-Global Matching (SGM) (Hirschmuller, 2008). This method allows for the estimation of depth of a given object by relying on stereo-aligned images; that is, the cameras taking the images are aligned such that corresponding points between the different images lie on the same horizontal line. For a set of corresponding pixels, their horizontal shift is calculated which can then be combined with the camera characteristics to return the depth of the object in relation to the cameras. In the Netherlands, this methodology is used to determine the heights in the Topographic Registry (BRT), Large-Scale Topographic Registry (BGT) and the building and addresses registry (BAG) (Kadaster, 2023).

While SGM does not have the same dense data sampling requirements like SfM, it still rests upon having an accurate set of cameras that are well calibrated. While this data is available in the Netherlands, SGM cannot be applied in contexts where only single images are available. Consequently, shadow-based height estimation from single images is still a valid alternative avenue.

2.7 Research Gap

Despite various methods both classical and through the usage of deep-learning exist for the segmentation of shadows, as of current there are no approaches where a deep-learning approach is combined with the physical constraints that presuppose the existence of shadows, i.e. building geometry and the position of the sun. The usage of physical laws and rules are applied effectively mostly for remote sensing tasks through PINNs, but its usage remains limited for the purposes of shadow segmentation.

Likewise, there are various methods for the height estimation of buildings, but factors such as operational complexity, computational power required and investment costs for data acquirement constitute barriers to entry. Considering the fact that height estimation from shadows can be done from relatively inexpensive aerial photography which is more widely accessible than other modes of data, there exists an incentive to explore avenues for improving such methodology. The niche this research aims to fill therefore is two-fold: through the improvement of the segmentation of building shadows, it is hoped that a more robust basis is provided to perform shadow-based height estimation upon.

3 Theoretical Background

This section will provide an outline for the theory driving the methodology. First of all, the concept of height estimation of objects from shadows will be explained as well as potential issues due to ambiguity. Then, the basis of Convolutional Neural Networks (CNNs) will be outlined in the context of image recognition. Building on this, the U-Net CNN architecture will be outlined, considering that the PHYSHADE model proposed in this thesis is based upon it. From there the theory of loss functions will be described, as these influence the way that CNNs train from data by punishing deviations and rewarding adherence to a given ground truth. Since the proposed methodology is inspired by Physics-Informed Neural Networks (PINNs), a short description of these will provided as well.

3.1 Shadow-based height estimation

The estimation of heights from shadows relies on the geometric relationship between shadow length (l_s), solar altitude angle ($\angle Solar_{alt}$) and object height (h_o).

$$h_o = l_s \cdot tan(\angle Solar_{alt}) \tag{1}$$

However, this formula makes the assumption that the shadow a given object casts falls on flat ground. As soon as the terrain becomes uneven, or the shadow falls on top of other objects additional measures need to be taken to ensure that the calculation is accurate. For example, adding the difference in height between the base of the building and the tip of shadow is required to come to an accurate height:

$$h_o = L \cdot tan(\angle Solar_{alt}) + (h_{tip} - h_{base}) \tag{2}$$

Note in the above equation that L is the planimetric horizontal length of a given shadow. To ensure an accurate height, L should be measured from the tip of the shadow to the point directly vertical from the highest point of the building. Considering that the goal of this research is to determine the height of a building without supplemental height information such as DSMs, the more simple equation (eq.1) of the two will be used, considering that the error the difference would introduce is likely to be small in a flat country such as the Netherlands. In order to find the length of the shadow (l_s), accurate segmentation must occur first.

3.2 Height Estimation of Buildings

3.2.1 Ambiguities from shadows

The estimation of heights from buildings follows the same mathematical principles as described in the section above. However, there are a few issues inherent to the way buildings cast shadows. For example, the geometry of a given building and its rooftop in combination with variations in solar altitude may lead to wildly different results. In Figure 8, it can be seen that with higher solar altitudes a shadow is cast based on the top of the roof, whereas with lower solar altitudes a shadow is cast based on the gutter of the roof. Unfortunately, this may lead the shadow-based height estimation unsuitable in cases where aerial imagery with higher solar altitudes is not available.

In the bottom example of Figure 8 with the building extension, it can be seen that there essentially are two shadows being cast from the same building, with shadow 1's length encoding the height information associated with the extension and shadow 2 the absolute top of the building.

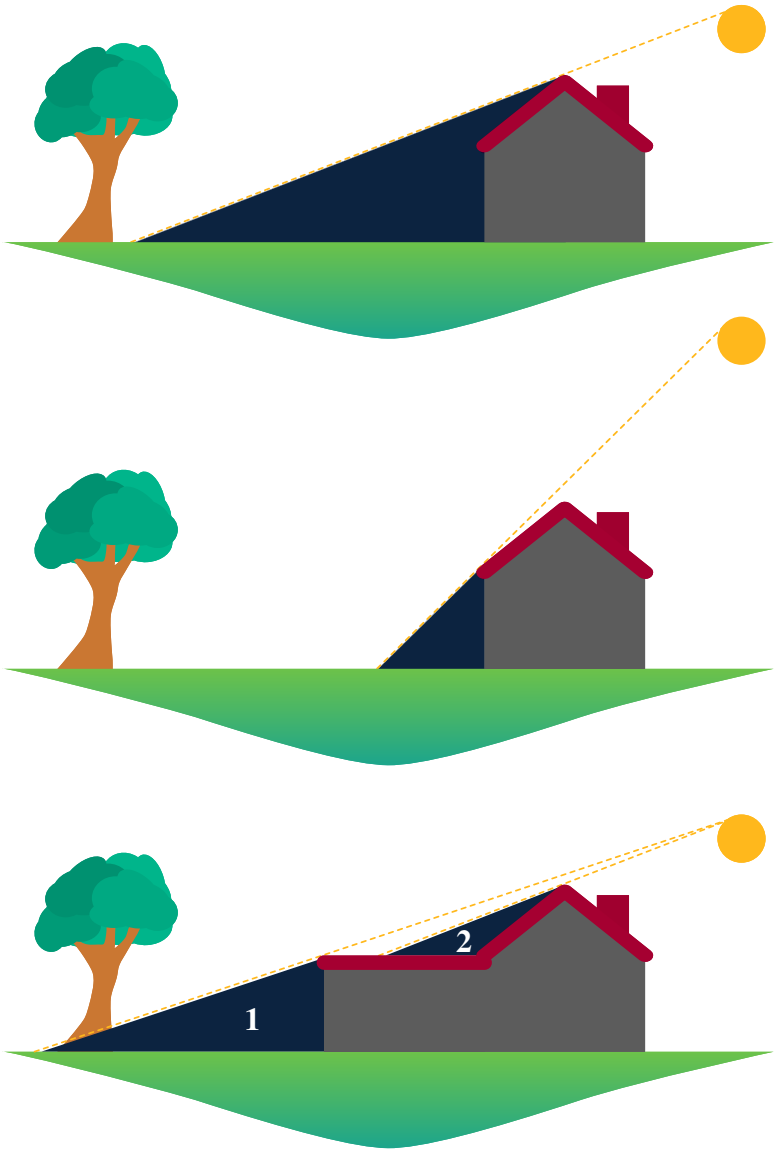

Figure 2: Image showcasing the difficulty with deriving building height from shadows depending on the interplay between building geometry and solar positions.

In addition, ambiguity may exist when trying to match a shadow to a given building, as more often than not shadows cast by buildings melt together into one continuous shadow. In some cases, overlapping shadows may even lead to the loss of information altogether. This concept is visualized in Figure 3 with a top-down schematic view of three pairs of buildings. While the buildings of pairs 1 and 2 share one continuous shadow encoding the heights encoded both by the bigger and smaller buildings, the bigger building of set 3 completely envelopes the smaller building, meaning that it is impossible to make a good estimation of its height without additional information.

For the buildings of set 1, if it can be assumed that the ambiguous orange areas do not have sporadic higher geometries (i.e. towers), the continuity of the shadow can be relatively easily resolved by taking the building footprint and sliding it from a distant point further down the solar azimuth back to the footprint origin, assigning the shadow pixels underneath with a unique ID associated with the building as it travels over the pixels back towards its origin (a more in-depth description of this procedure alongside clarifying images is given in subsubsection 4.6.2 and Figure 10). On the other hand, this operation for assigning shadows pixels to buildings would not work for building set 2, as

this would attribute the longer part of the ambiguous shadow region back to the smaller building since the associated building footprint passes over the ambiguous region last, which would be erroneous under the assumption that the buildings have constant heights. For more robust shadow-to-building classification, an alternative method would be required.

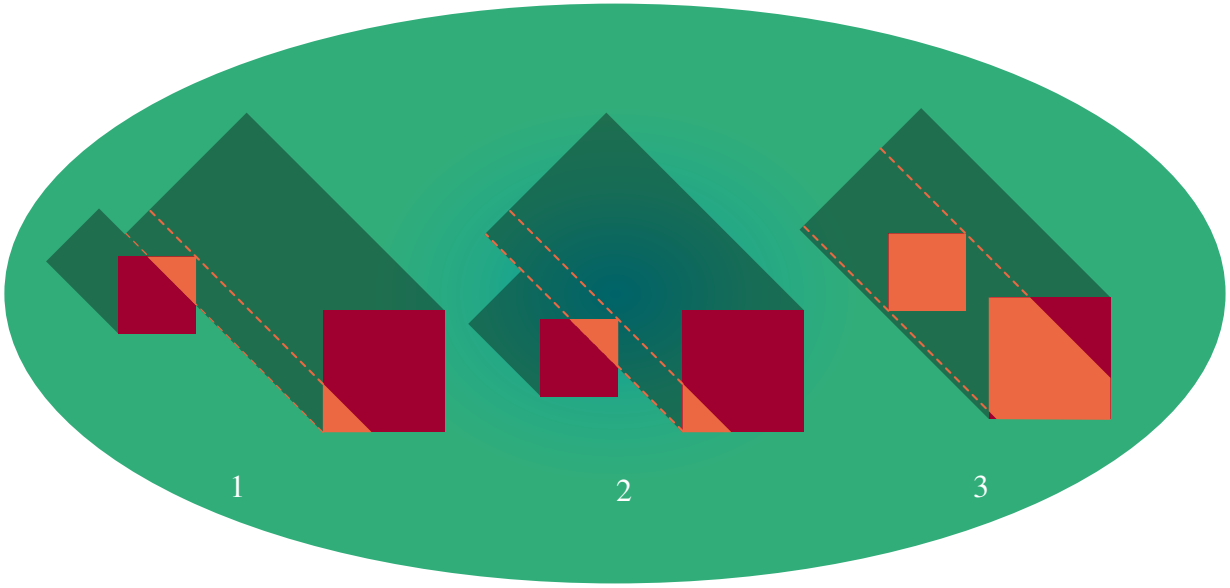

Figure 3: Image showcasing the loss of height information that occurs when shadows start to overlap one another. The orange areas of the buildings indicate about what part of the building suffers from ambiguity.

3.3 CNNs and the basis of Segmentation

A Convolutional Neural Network (CNN) is a deep-learning model that is often used for the purposes of computer vision, or any other field where innate structures in large datasets are exploited for inference. A CNN consists of multiple layers through which data is processed. Generally speaking, the following layers can be identified in a CNN (LeCun et al., 2015):

- **Convolutional Layers**: Layers where input data is scanned using filter banks to detect specific patterns such as textures and shapes. These filter banks are also called kernels.
- **Activation Layer**: The output of each individual node in a convolutional layer here is fed into an activation function, which calculates the output of a neuron. Depending on the type of function used by which to calculate the output, the main goal of this layer is to allow for the approximation of non-linear relations in the data.
- **Pooling Layer**: Downsamples the information coming in to reduce spatial dimension, removing redundant information and reducing computational complexity.
- **Dense Layer**: A fully connected layer that combines all extracted features to provide a final output (Lecun et al., 1998).

For example, an image inserted into a CNN is first processed by the convolutional layers where features such as edges and textures are detected. As these features flow further into the model through

more pooling and convolutional layers, the data is progressively abstracted. In the end, a dense fully connected layer maps these features back into specific output (Lecun et al., 1998; LeCun et al., 2015).

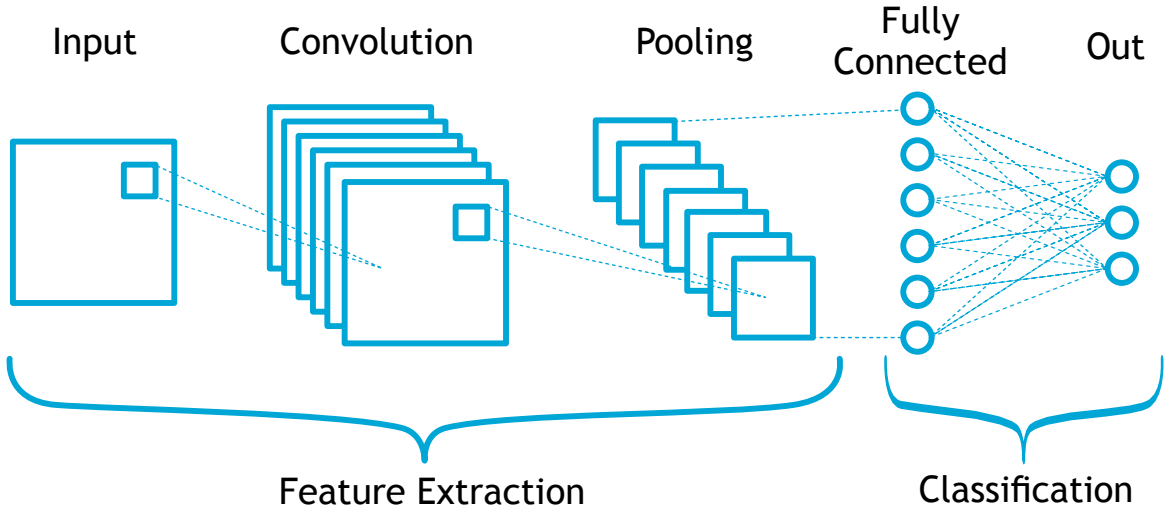

Figure 4: Schema of a typical CNN layout. Input is moved into the convolutional layer, where filters are applied to recognize patterns. Between input and convolution, an activation layer is used to enforce non-linearity. They are then connected to pooling layers that downsample the imagery, before then entering the fully connected layer that actively classifies the input. Schema adapted from Phung and Rhee (2018).

In order to train a basic CNN, a training dataset is fed into the network and the kernels in the convolutional layers are randomly initialized. After passing through the layers, the performance of the neural network is calculated through a loss function, which measures the difference between the predicted output and the ground truth. As an example of a loss function, in cases where the output is binary (i.e. a pixel is shadowed or non-shadowed), binary cross-entropy can be used which measures difference between the output and the ground truth:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(\hat{y}_i) + (1-y_i) log(1-\hat{y}_i)] \tag{3}$$

where i is a pixel in image N, y_i is the binary label of a given pixel i, and $\hat{y}_i$ the probability that pixel i belongs to the ground-truth label. Through a process called backpropagation, this loss function is then used to define how much each individual parameter in the convolutional layers has contributed to the loss. Based on the result of backpropagation, the weights are then finally adjusted through an algorithm like gradient descent to minimize the error (LeCun et al., 2015)

The ways in which a neural network can be trained can be split up into three categories depending on the data available; Firstly, supervised training relies on labeled datasets. This means that for a given input, the output is compared to a validation set. Secondly, unsupervised training relies on the algorithm to learn patterns from unlabeled data, meaning that there is no known mapping from input to output. Finally, there is semi-supervised learning which utilizes both labeled and unlabeled datasets (Nan, 2023). Such a method is advantageous, as fully supervised training requires large amounts of data which is either expensive or time consuming to procure, which is not feasible in all cases.

However, transfer learning offers an alternative method for reducing the costs of training. Originally introduced by Pratt et al. (1991), transfer learning relies on the concept of taking a neural network

originally made for a similar purpose and using the same weights instead of training a new model with randomly initialized weights. For example, in order to train a neural network that can recognize cats, it might make sense to start out with the weights of a CNN already trained for dogs, as they are morphologically relatively similar. From there, the model can be further trained and fine-tuned into specifically recognizing cats, and will require relatively less paired input-output data than if the weights were randomly initialized.

Still, in order to validate a model's performance, it is important to have a set of annotated data on which the model has not been trained. As such, in circumstances were data are ample, data is often split up into training and validation sets so that the model does not have a-priori knowledge of the ground-truths in the validation set. However, if the dataset is small, such a split may lead to poor training outcomes due to a lack of data. In these cases, an alternative procedure can be found in the concept of K-Fold Cross-Validation. The method involves dividing the dataset into k-subsets (or folds), where each time the model is trained on the remaining folds and validated on the current fold, resulting in the training of five different models. The error can then be estimated by averaging them across all of the folds, leading to an estimation of model performance as if it was trained on the full dataset (Goodfellow et al., 2016).

3.4 U-Net Architecture

U-Net is an architecture developed by Ronneberger et al. (2015) originally for use in the biomedical field. At the time, the typical use for CNNs was to do classification where the entirety of an image was classified as one single class, as opposed to local pixel-wise classification. Although there were models that were able to perform tasks like these, such as the one described by Ciresan et al. (2012), where a sliding window is used to provide local context (in the form of patches) to help classify pixels. This patching approach which is still typically used in modern training approaches, although effective as a way to augment a dataset if data is scarce and giving the model a means to learn local context, has a few drawbacks. For example, it is relatively slow considering that the model has to train off of each individual patch. In addition, the patch size is of heavy influence to the ability for a model to either be able to localize well or to see context.

U-Net improved upon Ciresan et al. (2012)'s architecture by introducing an encoder-decoder structure where the valid content of the feature maps learned in the encoder are concatenated to the decoder allowing the model to provide context in the upsampling stages (see figure 5). The end result was a model that was able to perform very well with only little training data.

3.5 Loss Functions

The way a CNN trains is by evaluating the difference between the ground truth and its output, and using this difference backpropagation can be performed to tune the parameters and minimize the error. This error is also called loss, with the functions describing these being titled as loss functions. Depending on the task at hand for a CNN, different loss functions will provide different outcomes and as such may be better suited than others. Section 4.4.1 will provide an overview of the different loss functions considered for this research.

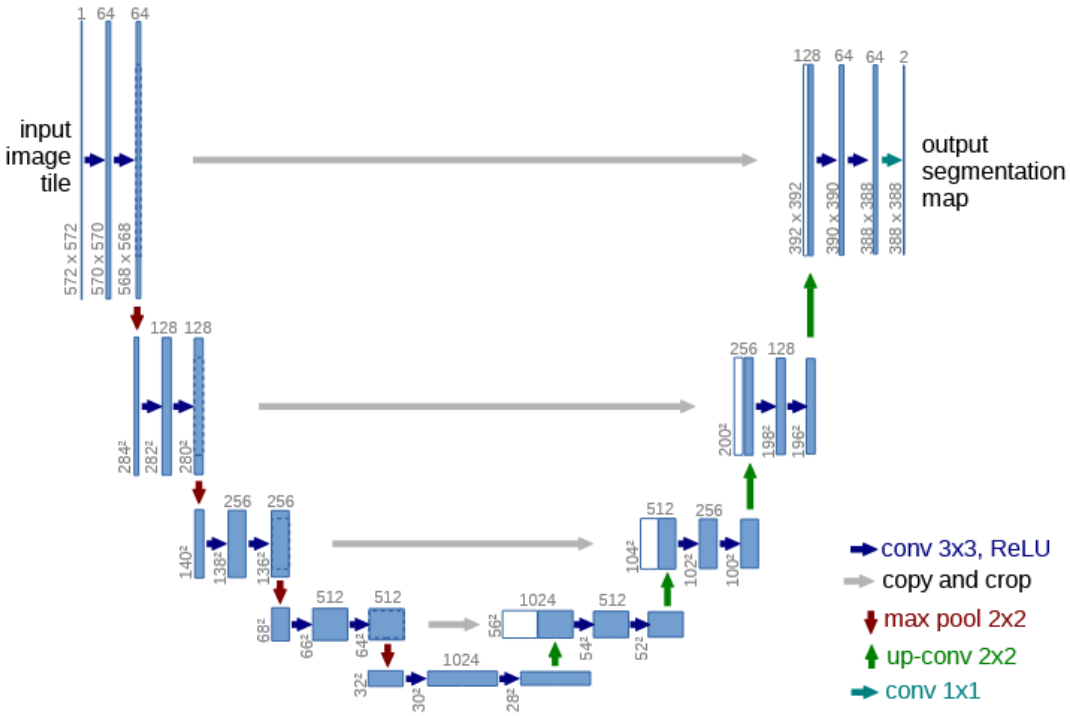

Figure 5: A schematic representation of the U-net architecture. The convolutional layers in the model are indicated with blue boxes, whereas copied layers are indicated with white boxes. On top of the convolutional layers the amount of channels (or feature maps) can be seen, with the size of these layers represented on the side. Note how the skip connections concatenate the feature maps from the encoder to the decoder. Diagram from Ronneberger et al. (2015)

3.6 Physics-Guided Loss Functions

A CNN can be turned into a Physics-Informed Neural Network (PINN) by integrating physical constraints into the loss function used during training. In the traditional sense, this means the integration of partial differential equations describing real-world phenomena. While the definition of physics-informed loss can differ greatly depending on the physical laws it describes, its implementation into an existing loss term for a model can be simple:

$$\mathcal{L}_{total} = \lambda_{BCE} \cdot \mathcal{L}_{BCE} + \lambda_{Phys} \cdot \mathcal{L}_{Phys} \tag{4}$$

where $\mathcal{L}_{Phys}$ is a loss function that penalizes based upon a partial differential equation. In the case of the shadow segmentation model with additional geometric priors, this loss can be based upon a precomputed shadow mask that abides by real-world constraints. In order to constrain or exaggerate the effects of the components that make up a loss term, i.e. $\mathcal{L}_{BCE}$ and $\mathcal{L}_{Phys}$, they are multiplied with a weighting term λ, which is considered to be a hyperparameter. A more in-depth description of the physics-guided loss function employed in this paper can be found in section 4.4.1.

4 Methodology

In this section, the methodology for creating and assessing the PHYSHADE models and height estimation pipeline will be outlined. Firstly, the case study area will be described and the collection and annotation of data for the new dataset will be described. Since this thesis uses transfer learning to minimize the issues caused by small dataset size, attention will be given to the training of the baseline Luo et al. (2020) U-Net model, as well as the dataset originally used to train this model. Some other strategies such as data augmentation and k-fold cross validation are then discussed, as these are helpful in combatting the issues caused by small-sized datasets. From here on, the architecture and learning procedures for the novel PHYSHADE models are proposed alongside the ablation strategy used to evaluate the performance of the added geometric priors. Finally, a description is given for the application the height estimation algorithm on the output of the PHYSHADE models. It is recommended to the reader to have a look at the pipeline in the appendices to gain a general idea of the full extent of the thesis first, Figure A.1 to Figure C.1.

4.1 Data Collection & Preprocessing

4.1.1 Case Study Area

The chosen case study area is split over 5 different areas, all of them in the Netherlands and most of them in the Midden-Delflanden. A map showing the exact locations of the areas can be seen in figure 6. Dense urban areas such as cityscapes were avoided, as they would pose a significant effort in terms of the volume of annotations necessary to create a dataset. Instead, the chosen areas of research have relatively simple spatial contexts which makes more accurate annotation easier and thus more time-effective.

Initially, a total of 206 512x512 tiles were initally roughly selected. As the urban morphology here can be best attributed to ribbon development and buildings being relatively sparse, a majority of the tiles were easily discardable. An overview of the procedures used to generate the tiles can be found in appendix D. These images were selected based upon the following criteria

- The image had at least one building with a clearly visible shadow.
- The majority of the shadows in the image are relatively clearly discernible.
- The image contains shadows that do not belong to buildings, as to provide a negative case as well.
- A few tiles including water were selected, as these are often misclassified as shadows.

Out of these 206 tiles, in the end a total of 42 were left after assessment of viability for the dataset. Out of these 42, six images were filtered out due to them containing no buildings for a total of 36. The initial reason for including them was to teach the model to disregard shadows not associated with a geometric prior. However, they may lead to negative accuracy outcomes with models using pixel-wise loss functions such as binary cross-entropy, as it may bias the model towards always predicting background. Since these issues would be exacerbated by the small size of the dataset, the decision was made to leave the images containing no buildings out.

The majority of these 36 images are summer-winter paired, meaning that there are images of the same geographic locations, but taken at different times during the year. This decision was made to ensure that the model is more robust to seasonal changes in lighting, vegetation, and is able to

generalize better with differing sun angles and shadow lengths. Specifically, this allows for the testing of the model generalisability under different solar geometries and environmental conditions. A full mosaic of the entire PHYSHADE dataset can be seen in Figure E.1. As can be seen there, there are a total of 15 summer-winter pairs and five singular images whose pairs got filtered out in the previous viability assessment.

Figure 6: An overview map showcasing the various locations used for data collection

4.1.2 Data Annotation

The data was then uploaded to Supervisely, which is an online platform that can be used to collaboratively annotate data for computer vision tasks. In total, five different people have helped to put together the dataset. In order to ensure consistency, an annotation guide was written to assist in labelling which can be found in Appendix F. To summarize, the following instructions were given to those labeling the aerial photography:

- **Buildings:** Always start with drawing buildings first. A building is considered to be a permanent (immobile) man-made structure that has a roof. Include any part of the building that is visible, including the facade. Only draw what is visible.
- **Shadow:** Label only shadows cast by buildings onto ground-level surfaces (e.g. streets, vegetation, parked cars), with no gap existing between the building and shadow labels. Exclude shadows from non-buildings or those falling from one building on another; the building label takes priority.
- **Precision/Consistency:** Try to be as pixel accurate as possible, carefully tracing all visible edges. Underestimate shadows rather than overestimate.

Labelling took place over the course of roughly a week. After annotation was completed, the masks were once more looked over to ensure that the labels were appropriate and seemed consistent. A few examples can be seen in figure G.1. One image was removed at this point due to a lack of annotation quality, finally dropping the main dataset size down to 35 images.

4.1.3 Baseline Dataset

Luo et al. (2020) provide a dataset for training named the AISD (Aerial Imagery dataset for Shadow Detection). It contains 412 images for training, 51 for validation and 51 for training, at sizes ranging from 256x2256 to 1688x1688, with the majority in 512x512. It is based upon the Inria dataset (Maggiori et al., 2017), containing aerial imagery in a wide gamut of contexts such as cities, forests and so on. The Inria dataset originally contained 360 images at a resolution of 5000x5000 pixels, each covering a surface of 1500x1500m. A table describing the geographical locations of the imagery within the AISD, as well as their distribution over the train, validation and test sets can be found in table 2.

As Luo et al. (2020) point out, the creation of the AISD was not easy, as the act of discriminating shadows from non-shadowed regions can turn out to be very difficult depending on the surfaces or objects. To counteract the creation of incorrect labels which in turn may lead to poor model performance, they carefully selected regions in the imagery that contained ”obvious” shadow regions.

Considering that the AISD dataset does not include any Dutch imagery it should be noted that the base-line model will effectively be used across domains, meaning that the accuracy of segmentation is not guaranteed due to differences in landscape and urban morphology. To establish a baseline performance of the base-line model in the case-study area, three images from the case study area were manually annotated for comparison.

Region	Austin, USA	Chicago, USA	Tyrol, Austria	Vienna, Austria	Innsbruck, Austria	Total
Number	150	127	100	79	58	514
Ratio	29.18%	24.71%	19.46%	15.37%	11.28%	100%
Train 80%	120	103	80	63	46	412
Val. 10%	15	12	10	8	6	51
Test 10%	15	12	10	8	6	51

Table 2: Table indicating the geographical spread and split of the AISD dataset. Adapted from Luo et al. (2020)

4.1.4 Out-of-Fold Dataset

For the purposes of allowing the best performing PHYSHADE models to be trained on the full dataset instead of individual folds and for the assessment of the height algorithm, six out-of-fold images were annotated as well. Four of these images (Tiles 7 and 8) were taken in the vicinity of Groningen/Eelde Airport, while the remaining two were taken over the city of Utrecht (Tile 6). See Figure K.1 for an overview of the images and Figure 7 for an overview of where the images were taken. Tiles six and seven were chosen for their generally unobstructed terrain, but with buildings types not seen in the in-fold dataset before to see whether PHYSHADE is able to generalize using the current dataset. Tile 8 was chosen for its relative similarity to the in-fold dataset while being in a completely different location,

to see whether or not any unseen variables unique to the in-fold domain could trouble PHYSHADE performance.

Figure 7: An overview of the locations where the aerial imagery of Tiles 6, 7 and 8 from the out-of-fold dataset were taken.

4.1.5 Data Augmentation

Since the original size of the dataset is small containing only 35 images, data augmentation was performed to stretch the dataset. For each image in the original dataset (i.e. 35 images), seven additional images were created using the set of operations found in Table 3. Through this augmentation, the size of the dataset was increased eightfold, from 35 to 280.

Table 3: An overview of the different augmentation operations, based upon a multiplicative factor.

Multiplication	Operation
1	Original Image
2	Horizontal Flip
3	Vertical Flip
4	Rotated 90 Degrees
5	Rotated 180 Degrees
6	Rotated 270 Degrees
7	Horizontal Flip, Rotated 90 Degrees
8	Vertical Flip, Rotated 90 Degrees

4.1.6 Data Splitting and Cross-Validation

To gain the most benefit out of a small dataset, five-fold cross-validation was used. This entails the randomized creation of five different training/validation folds, with each unique model as described in Appendix H.1 being trained and validated five times over these folds. Per fold, the validation and loss metrics were saved, and in the end averaged to generalize the performance over the different folds. Winter/Summer paired images were kept together within training/validation splits to ensure the model

would not be able to cheat by already having seen a similar ground truth.

4.2 Baseline Model Development and Transfer Learning

For this study, U-Net (Ronneberger et al., 2015) will be used as a base model before transfer learning is used, as it is a popular, tried and tested image classification model. In addition, Luo et al. (2020) in their study on shadow segmentation compared their own DSSDNet to U-Net, mentioning that although U-Net did not show significant errors during evaluation, it had the propensity to mistakenly identify dark surfaces such as dark cars and roofs as shadows. With this in mind, the decision to use U-Net as a base-model for transfer learning was deemed viable as the addition of geometric priors could help alleviate the issues Luo et al. (2020) found. In the pretraining phase, the model will be trained on their dataset to function as a baseline. The training loss function employed here will be typical for a U-Net model, using Binary Cross-Entropy (l_{BCE}, see eq.8):

$$\mathcal{L}_{pretrain} = \mathcal{L}_{BCE} \tag{5}$$

The training of the base U-Net model for the later purposes of transfer learning took place using the AISD dataset, which is described in the next subsection. The encoder weights from the here-trained U-Net baseline were transferred to be used as a starting point for the PHYSHADE models. An overview of the parameters used to train the base U-net model can be found in appendix I.1.

4.3 Proposed Model Structure: PHYSHADE

The methodology proposed in this thesis will integrate shadow priors into a standard convolutional segmentation pipeline in order to enhance the segmentation of building shadows. The shadow priors (henceforth **pseudo-shadows**) are simulated by combining the footprint of a building together with the solar azimuth and altitude. The underlying hypothesis is that the inclusion of context-specific information relevant to shadows, such as solar positioning and building geometry, can regularize the segmentation, especially in cases where shadows are degraded. In addition, the addition of pseudo-shadows may also bias the model towards only segmenting shadows that are attributed specifically to buildings. This can be advantageous in situations where building shadows need to be distinguished from those cast by other objects such as vegetation or cars. Likewise, the reverse may also be true where poor quality pseudo-shadow priors may lead to misses during segmentation.

To test the hypothesis, three different approaches were taken to implementing pseudo-shadows into a U-Net model:

- The first approach will be characterized by the addition of the pseudo-shadows as a 4th binary channel into an existing U-Net model.
- The second approach will implement the pseudo-shadows into the loss functions employed, following the philosophy behind Physics-Guided Neural Networks.
- The third and final model will be a hybrid of the first two, where the model is both regulated by a physics-constrained loss function and is able to see the 4th channel pseudo-shadow directly.

The derivative CNNs resultant of these methods will be evaluated in tandem and to one another to assess the the individual and combined effects of pseudo-shadows as input features, as constraints in the form of physics-guided loss, or both on segmentation accuracy.

4.3.1 Generation and Integration of Pseudo-Shadow Priors

To give the new model the ability to take into account the geometric priors of the building footprints into account, an algorithm was created that takes the original building footprint raster, solar azimuth and a given shadow length and constantly shifts it into the opposite direction of the solar azimuth, each time writing values to each cell overlapping with the shifting building footprint. The result is a smeared-out building footprint that simulates a shadow falling in a direction. The pseudocode for the shadow smearing algorithm can be found in the appendices as J.1 and a step-by-step visual in Figure 9.

The algorithm has a few parameters for defining the gradient of the footprint smear: *l_min* and *l_max*. *l_min* define the distance from the base of a footprint away from the solar azimuth where the pseudo-shadow raster is set to a solid value of 1. After a building footprint has been shifted an *l_min* amount of meters the gradient starts, which will diminish gradually until *l_max*. The reason a smooth gradient was selected, is because of the uncertainty of building heights beyond *l_min*. By expressing the confidence a shadow occurs at a given place using a gradient as opposed to a binary classification, where a value of 1 expresses high certainty down towards a 0 for no certainty, it is hoped that this provides the model a decision boundary where it has the freedom to not feel obligated to predict thereby minimizing the risk of false positives induced by the pseudo-shadows. A drawing showcasing this behaviour can be seen in figure Figure 8.

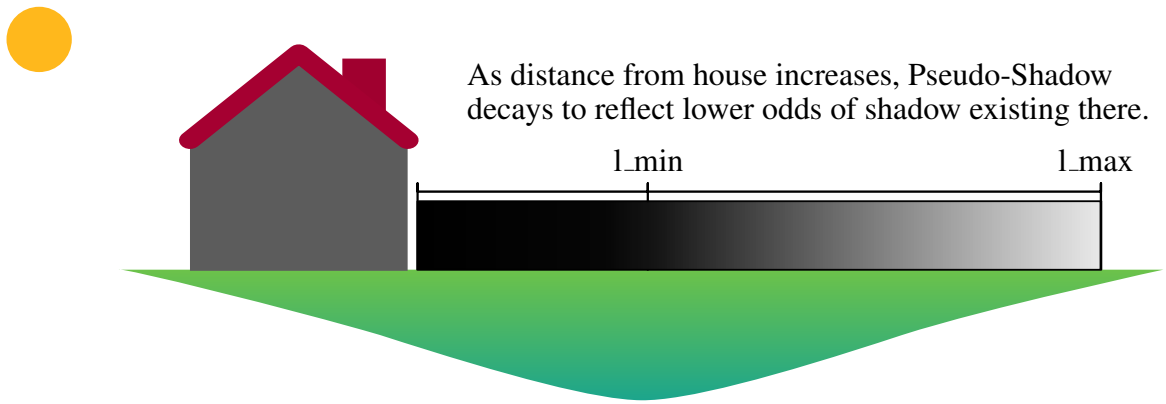

Figure 8: A schematic overview of the intensity of the pseudo-shadows decreasing as the distance increases from the base of the building. Note how the pseudo-shadow is solidly black from the base of the building to l_min; this reflects the assumption that building structures have a minimum height before being considered buildings.

In order to select appropriate values for *l_min* and *l_max*, the LoD 1.2 building data from the 3D BAG by tudelft3d and 3DGI, a dataset made by combining the Register of Buildings and Addresses (BAG) and National Height Model (AHN) of the Netherlands (Peters et al., 2022), was used to find the 95th percentile height of all buildings. This way, by removing tall outliers above the 95th percentile, relatively nationally representative pseudo-shadows can be generated that would fit for the most buildings in the Netherlands. Since the 3D BAG has its building heights determined through LiDAR scanning, the height used was the 70th percentile of all roof-surface points as this number is used to most reliably mitigate potential noise. In addition, it also foregoes any issues where small raised structures (e.g. access stairwells) on top of the roofs. Through this, it was found that the 95th percentile of all buildings was 42.90. To convert that into the appropriate *l_max*, the formula for estimating building heights as mentioned in Equation 1 is adapted:

(6) $$l\_max = \frac{42.90}{\tan(\angle Solar_{alt})}$$

Since the angle of the sun changes between images due to temporal differences, the final value of l_max can vary. The height value for l_min was chosen under the assumption that most structures that can be considered buildings are at least 2 meters high, therefore:

(7) $$l\_min = \frac{2}{\tan(\angle Solar_{alt})}$$

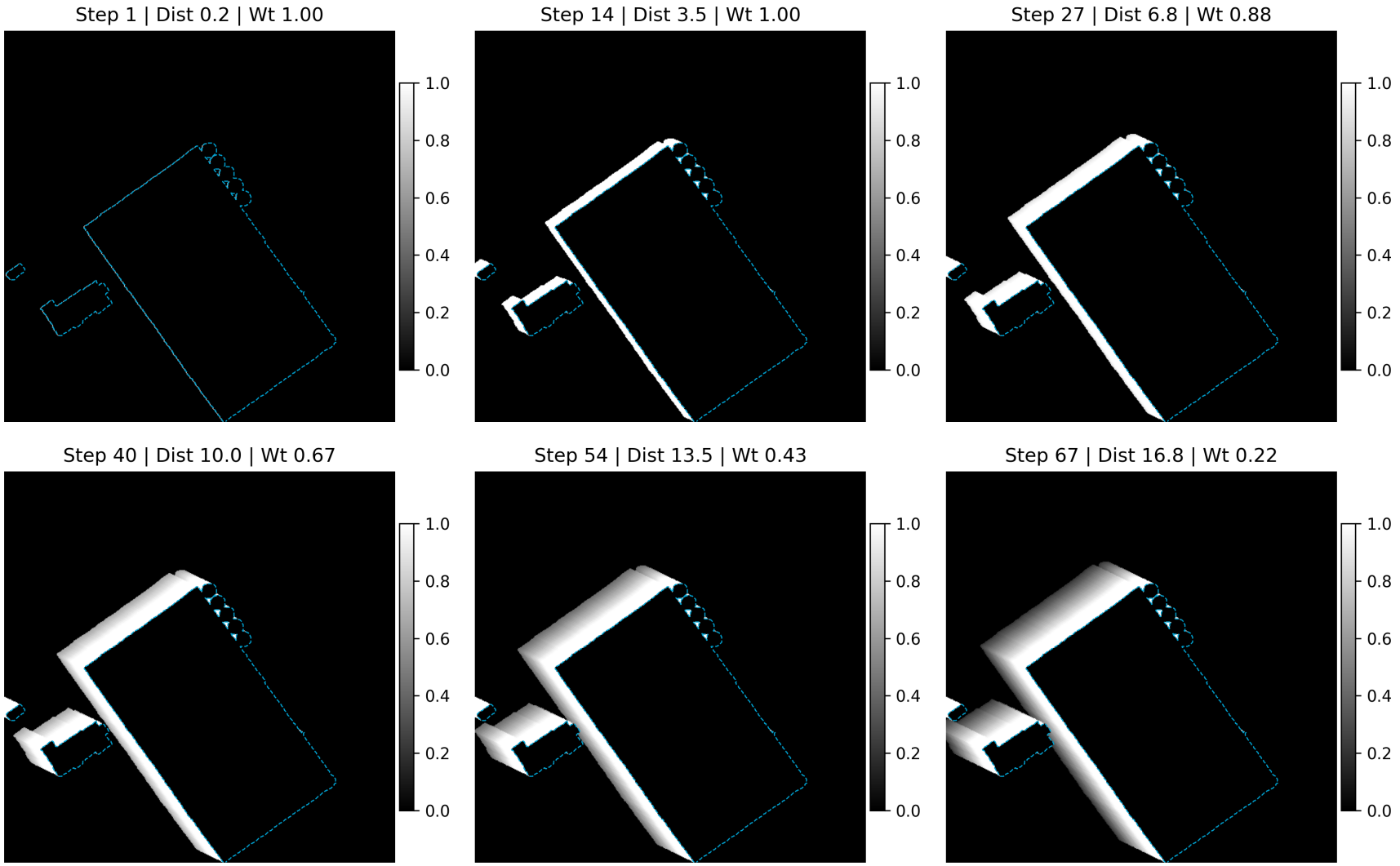

Figure 9: An overview of the smearing algorithm working step-by-step to generate pseudo-shadows from a building footprint. For showcasing purposes, $l_{min} = 5$ and $l_{max} = 20$, meaning that the gradient up until 5 meters (between steps 1 and 14) is solid, and slowly decaying thereafter until 20 meters. The brightness of the smear represents a confidence that aims to give the model a dynamic decision boundary.

4.3.2 Model Configurations

To evaluate the effects of the addition of pseudo-shadow priors on segmentation performance, a variety of models were trained, each with different hyperparameters so as to evaluate the impact of the additions through ablation. The model configurations can be broken up into four experimental subsets, each tackling to evaluate a different performance metric. An overview of the ran model training configurations can be found in Appendix H.1. In total, the subsets contain 26 different configurations, meaning with 5-fold cross validation, a total of 130 models were trained. A quick overview of the purpose of each experimental subset for the purposes of for ablation can be seen in table 13. What follows is a description of the purpose of each experimental subset, and how they will be interpreted:

Experimental Subset A: Baseline Performance
This subset serves to give a baseline performance metric against the other models that do contain some sort of pseudo-shadow priors. Thus, this model will be trained without any of the pseudo-shadows. It is hypothesized that this model is likely to perform poorly in segmenting building shadows and ignoring other shadows, as the dataset by itself is too small for it to learn the relatively advanced contexts (e.g. building geometry) needed to differentiate between them.

Experimental Subset B: 4th Channel Pseudo-Shadow Priors
This subset of models is meant to elucidate the effect of adding pseudo-shadow on segmentation performance when the model gets to freely interpret them. Models labelled as RGB will not see the pseudo-shadows, whereas models labelled as RGBS will. All models in this subsection will be using a combination of binary cross-entropy and Dice loss. A grid-search is performed on the hyperparameters balancing the weighting of BCE based loss and Dice loss, to see what performs best for the model. Half of the models will be able to a access the 4th channel, whereas the other half will run using the same parameters without access so that the influence of the pseudo-shadows on model performance can be considered in isolation by comparing the halves to one another (i.e. model B1_RGB is compared to model B1_RGBS as per Appendix H.1) It is hypothesized that the Dice-based models will have a higher performance than the BCE models due to their ability to handle class imbalance better.

Experimental Subset C: Physics-Guided Losses
These models will not directly be able to see the pseudo-shadow. Instead, the loss function will be modified in such a way that the pseudo-shadow information will be used to supervise the losses rather than directly inform the model as a feature. Like in the previous subsets, BCE and Dice loss will be used. A third model has been added as well which places extra weight on segmentation errors within the pseudo-shadow. This loss function differs from the first two models by pseudo-shadows not being interpreted as a label the model needs to take into account, but rather as an amplification of any existing loss that falls within the regions of the pseudo-shadow priors. Out of these three models, it is hypothesized that experiments PHYS_BCE and PHYS_DICE will end up with a lower validation score than the PHYS_ATT experiments, but a higher score than the BASE models.

Experimental Subset D: Hybrid Models
Finally, this subset of models will combine the components of subset B and C. The model will get direct access to the pseudo-shadows concatenated in the fourth channel as described in B, and will be additionally regulated by physics-guided losses described in C. Here it is hypothesized that the combination of access to the pseudo-shadows combined with the physics-guided loss will lead to increased performance and faster convergence time during training, as loss is weighed more heavily near the building shadow pixels.

Table 4: Summary table indicating the purpose of each individual ablation.

Ablation Subsets	Purpose
A & B	To establish the performance difference unique to the models' interpretation of the pseudo-shadows as an extra channel.
A & C	To establish the performance difference unique to the usage of physics-guided loss based on the pseudo-shadows without vision on the pseudo-shadows.
B & D	To establish the performance difference unique to the usage of various physics-guided losses based on the pseudo-shadows with vision on the pseudo-shadows.

4.4 Loss Functions

4.4.1 Standard Losses

As mentioned in the above section, a variety of different loss functions will be tested to see which will give the best performance for building shadow segmentation. This subsection will give a short overview of the mathematical definitions, as well as the reasoning behind their usage

Binary Cross Entropy (BCE)
Binary Cross-Entropy defines the error as the negative logarithm of the likelihood of belonging to a ground-truth label (Bishop, 2006). Its usage is commonplace in vision models that only need to do binary classification. Since the resultant gradients are smooth, they are a good fit for optimization since they respond predictably. However, when classes are imbalanced (i.e. too high of a difference between ground truth shadows and background) it can learn to learn to predict the majority class, which may lead to a loss of performance.

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1-y_i)\log(1-p_i)] \tag{8}$$

where p_i is the predicted probability, and y_i is the ground-truth label.

Dice Coefficient, Jaccard Index and F1-Score
Originally created by Dice (1945), the Dice Coefficient is a statistical instrument used to measure the similarity between two sets. For the purposes of classification, it is the overlap between ground truth and prediction divided over the total area of the sets. It is very similar to the Jaccard Index (also known as Intersection over Union, or IoU), and is defined as follows:

$$Dice(A,B) = \frac{2|A \cap B|}{|A| + |B|} \tag{9}$$

By comparison, the Jaccard Index is defined as:

$$IoU(A,B) = \frac{A \cap B}{A \cup B} = \frac{A \cap B}{|A| + |B| - |A \cap B|} \tag{10}$$

Both of these statistical measures give the overlap between datasets. However, by nature of the Jaccard Index having a larger denominator, it produces lower scores than the Dice coefficient which means that false positives and negatives are penalized harder. Inherently, this means that the Jaccard index is slightly more unstable and thus less fit for imbalanced data than the Dice coefficient which is relatively smoother. For this reason, the decision was made to perform training using the Dice coefficient.

A common statistic to evaluate model performance is the F1-Score, which is similar in definition to the Dice coefficient in binary classification tasks as can be seen in eq. 11 when compared to eq.9. However, to keep consistent terminology, we will continue to use Dice score to talk about the same thing.

$$F_1/Dice = \frac{2TP}{2TP+FP+FN} \tag{11}$$

In deep learning, Dice loss is implemented in the following form:

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum_i p_i y_i + \varepsilon}{\sum_i (p_i + y_i) + \varepsilon} \tag{12}$$

where p_i is the predicted probability, and y_i is the ground truth label.

4.4.2 Physics-Guided Losses

Besides BCE and Dice loss, a few PINN-inspired loss functions are introduced that will modulate the losses based on the pseudo-shadow. Firstly, the attentive variants of the physics-guided loss functions will weigh the error and subsequent loss of false negatives and false positives within the pseudo-shadows harder. It takes a slight redefinition to the old $\mathcal{L}_{BCE}$ and $\mathcal{L}_{Dice}$ functions:

$$\mathcal{L}_{\text{BCE}}^{\text{att}} = \frac{1}{N}\sum_{i=1}^{N}\left(1+\alpha s_i\right)\left[-y_i \log p_i - (1-y_i)\log(1-p_i)\right] \tag{13}$$

$$\mathcal{L}_{\text{Dice}}^{\text{att}} = 1 - \frac{2\sum_i (1+\alpha s_i)\, p_i y_i + \varepsilon}{\sum_i (1+\alpha s_i)\,(p_i + y_i) + \varepsilon} \tag{14}$$

Where α controls the extra weight given to the pixels within the pseudo-shadow, s_i is a weighting value between [0, 1] based on the pseudo-shadows generated by algorithm J.1, y_i the ground truth label and p_i the label predicted by the model. Setting the weight of α to 0.2 means that the pixels will be emphasized 20% more.

The other version of loss involves the usage of regular BCE and Dice loss, but with the addition of an extra loss term (which can be either BCE or Dice) that gives the CNN an extra objective to match its predictions to the pseudo-shadow channel by comparing the predictions to the pseudo-shadow map. In this sense, the difference between Equation 13 and Equation 14 versus Equation 15 lies in how the pseudo-shadow is used. The attentive losses emphasize learning within the pseudo-shadows through a more intense weighting of the regular losses within the region of the pseudo-shadow. The Physics-loss described in Equation 15 on the other hand adds an explicit loss term that discourages models from predicting outside of the pseudo-shadow regions, forcing it to respect the pseudo-shadow priors even if it conflicts with the ground-truth.

$$\mathcal{L}^{\text{Phys}}_{\text{BCE/Dice}} = \lambda_{BCE} \cdot \mathcal{L}_{BCE} + \lambda_{Dice} \cdot \mathcal{L}_{Dice} + \lambda_{Phys} \cdot \mathcal{L}_{Phys} \tag{15}$$

where λ_{BCE}, λ_{Dice}, and λ_{Phys} are weighting hyperparameters controlling the contribution of each loss component. The physics loss term, $\mathcal{L}_{Phys}$, measures the agreement between the model's predictions and the pseudo-shadow map using either BCE or Dice loss:

$$\mathcal{L}_{Phys} = -\frac{1}{N}\sum_{i=1}^{N}[s_i \log(p_i) + (1-s_i) log(1-p_i)] \tag{16}$$

or

$$\mathcal{L}_{Phys} = 1 - \frac{2\sum_i p_i s_i + \varepsilon}{\sum_i (p_i + s_i) + \varepsilon} \tag{17}$$

Where p_i is the predicted value and s_i is the pseudo-shadow value. Here it is important to note the distinction between the above two loss equations to their regular BCE and Dice counterparts; whereas the original loss formulations compare the predictions to the ground truth, the above physics-guided losses compare the predictions to the value of the pseudo-shadow.

4.5 Model Training

4.5.1 Training Protocol

All of the different configurations of PHYSHADE described in Table H.1 were trained using the AdamW optimizer, starting out with a learning rate of 1e-4 and weight decay of 1e-4. For a scheduler, the ReduceLROnPlateau scheduler was selected with a patience of 5 epochs and a minimum learning rate of 1e-6.

Training was conducted using a maximum of 150 epochs, with early stopping if the validation score plateaued for 25 epochs. The batch size was kept consistent over all models to a value of 8. To speed up training, mixed precision was enabled.

To ensure reproducibility, the seeds used to drive the randomization of the folds and data augmentations were kept fixed and reset every new cross validation to ensure that the models within and outside the subsets could be properly compared to one another. The training was ran on an Nvidia Blackwell RTX 5070Ti GPU with 16Gb of VRAM paired with an AMD Ryzen 5600X CPU on a Windows machine. Since this GPU as of writing is relatively new, a nightly build of PyTorch had to be used to ensure compatibility (Version 2.8.0.dev20250418+cu128). More details on other relevant parameters for reproducibility can be found in Appendix Table I.1.

4.5.2 Evaluation Metrics

For the evaluation of the performance of the different configurations of PHYSHADE, Dice (or F1) score will be used as it is a common metric by which to rate the performance of a segmentation model. For the height estimation algorithms, the ground truth building height will be defined by the 70th percentile of LiDAR hits, obtained from the 3D BAG. Because the 3D BAG's footprints do not line up one-to-one with the building footprints produced by the annotation efforts considering that the 3D BAG registers heights per administrative unit as opposed to entire buildings, a few data transformations

needed to be applied to gain representative ground truth heights. Firstly, each administrative unit was scaled up slightly to ensure spatial overlap. Then, a dissolve operation was performed to combine each administrative unit into one building unit. Since each administrative unit has their own 70th percentile height value, the most representative value was inferred by selecting the value corresponding to the unit with the largest area. Since the 70th percentile height value does not report building height (i.e. roof to ground) but rather the height above sea level, an additional operation was performed to obtain the true building height. A buffer of 5 meters was created around each building unit. Over the entire buffer, a DTM was sampled obtaining the 90th percentile ground height. Subtracting this value from the buildings resulted in the true building height. For the reporting of the performance of the height estimation algorithm, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and deviations of the residuals will be utilized.

4.6 Height Estimation from Segmented Shadows

4.6.1 Preprocessing Steps

Before the height estimation can be ran on the inference results, some preprocessing is required to ensure that the data is fit for input into the algorithm. For example, in some cases shadows from buildings will amalgamate into one continuous blob, requiring it to be broken up first so that height estimation can be ran on individual blobs that belong to specific buildings. To do so, an operation similar in spirit to the smearing algorithm was performed but in reverse: instead of smearing the building footprint outward from its origin in the direction the sun is shining, the building footprints are moved to an arbitrary distance outside of the grid. Then, is continually stepped back towards its origin. Every time a building footprint moves over a pixel in the inferred shadow mask, it is assigned a unique value that references the building footprint. When the footprints reached their origins, the relevant pixels for shadow estimation will have been marked, and an operation can be performed where continuous shadows representing multiple buildings can be spatially separated by setting each pixel with a unique id to zero, if one of its 8-direction neighbours has a different unique id. This way, unique separated shadow blobs are created which can be used as input for the height estimation algorithm. An overview of this assign-and-break operation can be found in Figure 10.

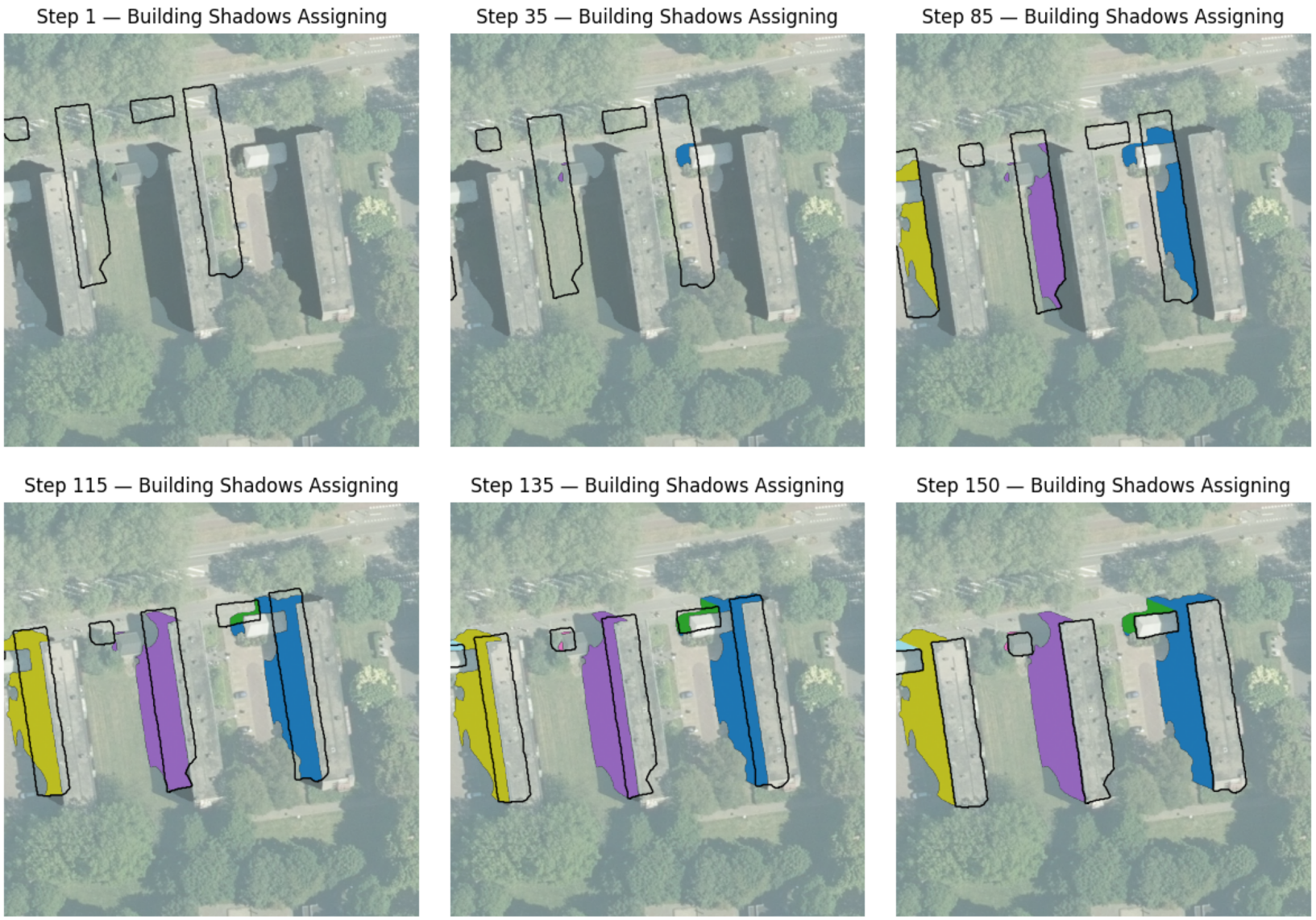

Figure 10: An overview of the assign-and-break algorithm assigning a unique ID to the shadows by moving the building footprint back towards its origin. Note how the smaller buildings reassign the shadows already touched by the larger buildings. After this processing step, a gap between the shadows can be created by using a 8-neighbour search, removing any pixel that has neighbours of a different class than itself

4.6.2 Shadow Length Estimation Algorithm

An algorithm was created based on ray-casting to retrieve shadow lengths, which in turn can be combined with the solar altitude to calculate back the size of a building. The pseudocode for this algorithm can be found in appendix J.2. In order to find a representative line that corresponds with the direction the sun is shining, the solar azimuth is first used to calculate a direction vector. In order to filter out potential blobs that do not match up with buildings, for each pixel in a given shadow blob, a ray is cast for five pixels opposite the direction vector. If no building is hit with this ray, the blob is thrown out (see Figure 11). In addition, any blobs that have an area of under 30 pixels are filtered out as well to get rid of potential noise. To get the height of the blob, the pixels on the edge facing the sun are found by stepping once from each pixel in the blob in the direction opposite the vector and checking the value of the pixel. If it is outside the blob, the previous pixel is marked as an edge pixel. Starting from each of the edge pixels, rays are then shot in the direction vector, stopping as soon as a ray steps outside of the shadow blob. When it does, the amount of steps taken (aka ray length) is appended to a list. From this list, an interpolated percentile is taken and finally multiplied by the pixel size to get the shadow length, from which it is possible to calculate back the object height through $h = l \cdot tan(alt)$. In cases where multiple blobs are linked back to the same building through the above method, for example in cases where the shadow is non-continuous, the final height of a building is defined by the blob with the biggest surface area.

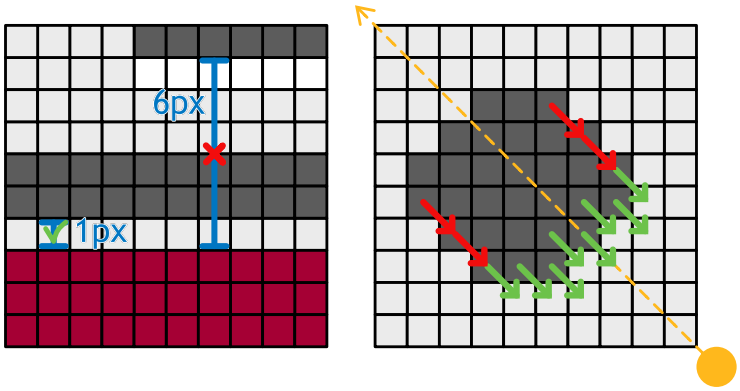

Figure 11: Image showcasing acceptance of close-by blobs and rejection of far-away blobs from buildings on the left, and the edge finding mechanism for finding the starting points for the height estimation ray-casts on the right.

5 Results

5.1 Baseline Model Performance

In order to establish a baseline performance for the standard RGB U-Net model trained on the (Luo et al., 2020) dataset (henceforth LUO U-NET) before transfer learning, shadows were inferred using two different images from the dataset. These were then compared to manually annotated ground truth masks using Dice, precision and recall. An overview of these statistics can be seen in Table 5, while inference results can be seen in Figure 12.

Although it should be minded that the test size here is small, the performance as tested is a downgrade from the Dice scores attained during training; whereas Luo et al. (2020) reported a Dice score of 0.8784, across domains the model loses out on accuracy, at worst by 0.4023 compared to data_winter_loc2_27. Looking at Figure 12, it is revealed that the model both under- and overpredicts, with false positives and negatives scattered throughout. There seems to be a low recall in situations that should be relatively simple to predict; the large horizontal building in the middle of data_winter_loc1_5 has the north side of its roof shadowed, but the model only catches part of it. Such performance degradation is expected in cross-domain situations, considering that the original model was trained on imagery from Austin, Texas and Vienna, Austria and does not recognize the Dutch contexts.

Image	Dice	Precision	Recall
data_winter_loc1_5	0.6587	0.6937	0.6271
data_winter_loc2_27	0.4761	0.5468	0.4215
Average	0.5674	0.6203	0.5243

Table 5: LUO U-NET performance evaluated over two fully-annotated images from the dataset.

5.2 Fold Statistics

Before looking at the performance of the models, it is important to mention some general statistics on how each separate fold performed during training. Since the folds were picked out semi-randomly while respecting winter/summer pairs, some folds may prove more difficult for the model to train and validate on than others depending on the complexity of the image. As an example, if one fold generally scores lower than others, it may be an indicator that the model has difficulty generalizing if the fold contains contexts that other folds does not have. It also becomes possible to analyse the consistency a model has over the various folds. Some models may perform more consistently, whereas others may have high error variance. Fold performance metrics also provide a basis for qualitative analysis, where the fold datasets can be compared to one another to see differences in landscapes or geometry. These differences can in turn help guide future dataset creation by identifying information gaps the models are suffering under.

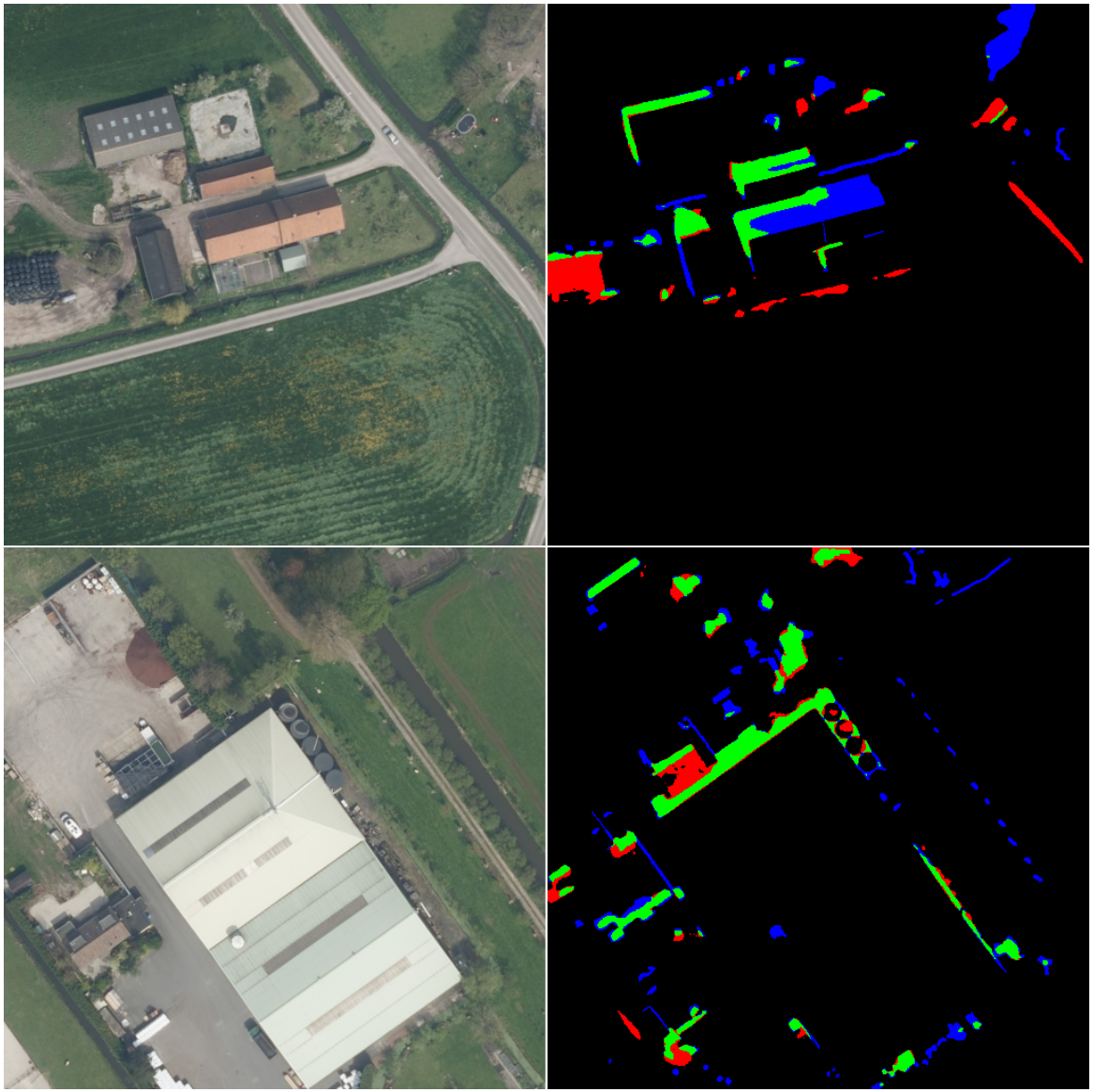

Figure 12: Side-by-side imagery showing the segmentation results for the U-Net model trained on the Luo et al. (2020) dataset, applied to the Dutch case study area. From top to bottom: data_winter_loc1_5 and data_winter_loc2_27, with the original RGB image on the left and on the right the inference results, with the true positives (green), the false positives (red) and the false negatives (blue).

5.3 Per-fold analysis

An overview of the per-fold statistics can be found in Table 6. As can be seen, fold 5 was the highest-performing fold over all trained models, suggesting that the training/validation combo was the most predictable. Fold 2 scored the worst scores over Dice and loss, meaning that the fold was more difficult to train on. Fold two also fields the highest standard deviation, meaning that the difference in performance between the different configurations of PHYSHADE is higher. Folds 1 and 2 represent the lowest-performing cases in the set, whereas folds 4 and 5 were the easiest to train/validate on.

To further elucidate the reason behind why some folds performed worse compared to the others, the training and validation images were manually assessed to see if anything obvious is noticeable at first sight. While no systemic errors were found between the folds such as size or training/validation leakage, qualitative assessment at first glance does not reveal any major difference in terms of environment or lighting conditions besides the impression that the validation set of fold 2 contains slightly more occlusions blocking out the building shadow annotations. Since these occlusions add to the complexity of a given shadow, annotation quality during dataset creation could potentially suffer which likewise would also influence PHYSHADE's ability to interpret the imagery. It is important to note however that no formal quantifiable differences were established between the folds; for example, annotation quality and per-fold annotation quality could be assessed by measuring the inter-annotator agreement, which could have been computed by comparing two annotations by different annotators over the same image using a statistical measure like intersection-over-union. However, due to time constraints such metrics are unfortunately out of scope.

Table 6: An overview of the average fold statistics over all training configurations.

Fold	Mean Dice	Std. Dice	Mean Precision	Std. Precision	Mean Recall	Std. Recall	Mean Loss	Std. Loss
1	0.6493	0.1633	0.6646	0.1562	0.5894	0.2024	0.2768	0.1468
2	0.6432	0.1825	0.5698	0.1880	0.6489	0.1922	0.2420	0.1381
3	0.6688	0.1406	0.6530	0.1259	0.6236	0.1977	0.2582	0.1288
4	0.6904	0.1473	0.7529	0.0910	0.5936	0.2247	0.2377	0.1252
5	0.7017	0.1714	0.7180	0.1485	0.6098	0.2504	0.2443	0.1427

5.4 Experimental Subset A: Baseline Performance

As predicted, the performance of both baseline models without any reference to the pseudo-shadows is relatively poor. BASE DICE was able to score ≈ 0.03 points higher on the Dice mean than BASE BCE, but considering the large overlap between their confidence intervals (0.4387-0.5569 and 0.4764-0.5598 ≈ 0.0805), no claims can be made that BASE DICE is actually better than BASE BCE.

Table 7: An overview of the averaged fold statistics for experimental subset A.

Experiment	Mean Dice	Std. Dice	Mean Precision	Std. Precision	Mean Recall	Std. Recall	Mean Loss	Std Loss	Mean Epochs Run	Dice CI95 Lower	Dice CI95 Upper
BASE BCE	0.4978	0.0674	0.5447	0.1145	0.3778	0.0335	0.0418	0.0136	39.4000	0.4387	0.5569
BASE DICE	0.5309	0.0331	0.5885	0.0944	0.4301	0.0610	0.4915	0.0566	46.8000	0.5019	0.5598

5.5 Experimental Subset B: 4th Channel Pseudo-Shadow Addition

With Subset B, the goal is to analyse the effect of adding a fourth channel to the RGB image, containing the pseudo-shadows as generated by the smearing algorithm. By comparing the RGB models to the RGBS models, the effect of the additional fourth channel containing the pseudo-shadows can be established in isolation. An overview of the statistics for Subset B can be seen in Table 8, with the ablation isolating the effect of the pseudo-shadows in Table 9. At first glance, the addition of pseudo-shadows majorly increases segmentation quality, leading to around ≈ 0.32 increase in mean Dice score. To statistically test the difference in Dice score between the RGB and RGBS models, a paired t-test was ran between over the individual folds. With all p-values testing below $\alpha = 0.05$, the null hypothesis that the means between the RGB and RGBS configurations are similar can be rejected. As such, the difference in Dice score is significant meaning that the RGBS models can be accepted as scoring higher in Dice than their RGB counterparts. Of the RGBS models, the BCE70 DICE30 scored the highest mean Dice score of 0.8487.

Table 8: An overview of the averaged fold statistics for Experimental Subset B, comparing the RGB vs the RGBS models.

Experiment ID	Mean Dice	Std. Dice	Mean Precision	Std. Precision	Mean Recall	Std. Recall	Mean Loss	Std. Loss	Dice CI95 Lower	Dice CI95 Upper
RGBS BCE30 DICE70	0.8475	0.0259	0.8435	0.0552	0.8239	0.0283	0.1245	0.0126	0.8248	0.8702
RGBS BCE50 DICE50	0.8455	0.0242	0.8488	0.0380	0.8172	0.0321	0.0971	0.0095	0.8242	0.8667
RGBS BCE70 DICE30	0.8487	0.0277	0.8470	0.0403	0.8118	0.0391	0.0672	0.0097	0.8244	0.8730
RGB BCE30 DICE70	0.5225	0.0313	0.5662	0.1033	0.4295	0.0754	0.3837	0.0323	0.4951	0.5499
RGB BCE50 DICE50	0.5208	0.0271	0.5578	0.0875	0.4186	0.0744	0.3007	0.0301	0.4971	0.5446
RGB BCE70 DICE30	0.5249	0.0230	0.5550	0.0955	0.4070	0.0349	0.2011	0.0219	0.5047	0.5451

Table 9: An overview of the Experimental Subset B ablation using paired t-testing, ran between RGB versus RGBS channels.

Experiment	RGBS Mean Dice	RGB Mean Dice	Delta Mean Dice	Delta Std. Dice	Delta Mean Precision	Delta Std. Precision	Delta Mean Recall	Delta Std. Recall	Delta Dice CI95 Lower	Delta Dice CI95 Upper	p-value (Dice)
BCE30 DICE70	0.8475	0.5225	0.3250	0.0264	0.2772	0.0556	0.3944	0.0709	0.2991	0.3509	0.0000
BCE50 DICE50	0.8455	0.5208	0.3247	0.0242	0.2910	0.0579	0.3986	0.0693	0.3009	0.3484	0.0000
BCE70 DICE30	0.8487	0.5249	0.3239	0.0280	0.2920	0.0649	0.4048	0.0415	0.2964	0.3513	0.0000

5.6 Experimental Subset C: Physics-Guided Loss

To establish the effect of Physics-Guided loss in isolation, the models of Experimental Subset C were compared to their equivalent baseline models from Subset A based on hyperparameters and loss used. As predicted in the methodology, the introduction of only a physics-guided loss without any access to the pseudo-shadows does not improve performance into acceptable levels. Over all the different configurations Subset C, none of them tested to score significantly differently to their non-physics-guided counterparts. This is unsurprising, as the model is not given any extra methods through which to distinguish between building shadows and other shadows.

One potential reason why BCE seems to have a slight edge over Dice loss in terms of delta Dice with the addition of physics-guided terms, may lie in the inherent differences in locality of both functions; whereas BCE works by looking at the pixels on an individual and local basis, Dice loss looks at the full mask globally instead. Based on this, Dice loss will struggle since it has to both agree with the ground-truth mask and the pseudo-shadow mask, which is impossible considering that the pseudo-shadow mask is an imperfect representation. This conflict in objectives then leads to a decrease in the model's performance. However, considering that none of the paired t-testing came back as significant, any claims made about the differences between the Dice and BCE variants in this case should be considered conjecture that would require further study to confirm.

Table 10: An overview of the ablation ran using Experimental Subset C.

Physics Config	Mean Dice	Std. Dice	Mean Precision	Std. Precision	Mean Recall	Std. Recall	Base Mean Dice	Delta Mean Dice	Delta Std. Dice	Delta Dice CI95 Lower	Delta Dice CI95 Upper	p-value	Significant?
ATT 0.1	0.5264	0.0177	0.5436	0.0783	0.4153	0.0591	0.5208	0.0055	0.0096	-0.0038	0.0149	0.3105	No
ATT 0.5	0.5394	0.0268	0.5465	0.1029	0.4119	0.0628	0.5208	0.0186	0.0156	0.0033	0.0338	0.0752	No
ATT 1.0	0.5325	0.0214	0.5578	0.0812	0.4307	0.0564	0.5208	0.0116	0.0135	-0.0016	0.0249	0.1597	No
BCE 10	0.5306	0.0341	0.5754	0.0965	0.4346	0.0365	0.4978	0.0328	0.0368	-0.0033	0.0689	0.1497	No
BCE 33	0.5234	0.0577	0.5543	0.1317	0.4161	0.0397	0.4978	0.0256	0.0203	0.0058	0.0455	0.0647	No
BCE 50	0.5108	0.0774	0.5076	0.1378	0.4384	0.0359	0.4978	0.0130	0.0136	-0.0003	0.0264	0.1275	No
DICE 10	0.5327	0.0247	0.5548	0.0914	0.4149	0.0517	0.5309	0.0019	0.0297	-0.0272	0.0310	0.9047	No
DICE 33	0.5267	0.0292	0.5479	0.0875	0.4330	0.0480	0.5309	-0.0041	0.0346	-0.0381	0.0298	0.8229	No
DICE 50	0.5396	0.0413	0.5108	0.0832	0.4413	0.0343	0.5309	0.0087	0.0429	-0.0333	0.0508	0.7052	No

5.7 Experimental Subset D: Effect of Hybrid Model

In subset D, the combination of the RGBS input together with the physics-guided loss is evaluated. For the ablation for this subset, the models were compared to the B3_RGBS model from Subset B, making it possible to see the effect of the addition of physics-guided loss. As can be seen in table 11, only two models tested as significantly different in Dice means compared to BCE50_DICE50. Both D3_BCE and D3_Dice saw drops in their mean Dice scores as well as increases in their standard deviations, meaning that the addition of physics-guided loss to models already having access to the pseudo-shadows is harming performance. In general for the different configurations of the Dice and BCE models in Subset D, a downward trend can be seen in Dice score the higher the weighting given to the physics-guided loss is.

5.8 Epoch Ablation

Since it was noted that the models of Subset B required more epochs to reach convergence, a general ablation based on epochs ran was performed to see whether any of the models saw any significant differences compared to their ablation counterparts, the results of which can be found in Table 12.

Table 11: An overview of the ablation ran using Experimental Subset D

Physics Config	Mean Dice	Std. Dice	Mean Precision	Std. Precision	Mean Recall	Std. Recall	Base Mean Dice	Delta Mean Dice	Delta Std. Dice	Delta Dice CI95 Lower	Delta Dice CI95 Upper	p-value	Significant?
BCE PHYS10	0.8523	0.0258	0.8498	0.0344	0.8320	0.0222	0.8455	0.0068	0.0031	0.0038	0.0099	0.0119	Yes
BCE PHYS30	0.8349	0.0270	0.8036	0.0402	0.8453	0.0253	0.8455	-0.0105	0.0075	-0.0179	-0.0032	0.0475	Yes
BCE PHYS50	0.8077	0.0327	0.7237	0.0599	0.8669	0.0288	0.8455	-0.0377	0.0097	-0.0472	-0.0282	0.0015	Yes
DICE PHYS10	0.8443	0.0266	0.8329	0.0494	0.8317	0.0277	0.8455	-0.0012	0.0029	-0.0040	0.0016	0.4546	No
DICE PHYS30	0.8447	0.0286	0.8139	0.0544	0.8540	0.0310	0.8455	-0.0008	0.0041	-0.0048	0.0032	0.7065	No
DICE PHYS50	0.8182	0.0267	0.6569	0.1132	0.8944	0.0185	0.8455	-0.0273	0.0087	-0.0358	-0.0187	0.0033	Yes
ATT 0.1	0.8472	0.0235	0.8433	0.0395	0.8205	0.0245	0.8455	0.0017	0.0019	-0.0002	0.0035	0.1476	No
ATT 0.5	0.8467	0.0277	0.8489	0.0378	0.8235	0.0333	0.8455	0.0012	0.0034	-0.0021	0.0045	0.5237	No
ATT 1.0	0.8417	0.0243	0.8403	0.0387	0.8189	0.0273	0.8455	-0.0038	0.0033	-0.0071	-0.0005	0.0847	No

Firstly, it becomes clear that the Addition of the pseudo-shadow in Subset B caused model convergence to take longer, with on average about 20 epochs more to reach training loss stagnation. Over the models in Subset C the difference in Delta Mean Epochs trained was tested as insignificant, meaning that the addition of physics-guided loss to models not having access to the pseudo-shadow channels led to no difference in epochs necessary to train to convergence. Finally, the models of Subset D were tested, with one model (RGBS_BCE50_DICE50 vs HYB_BCE50_DICE5) testing as the only model where the addition of physics-guided loss led to a meaningful decrease of 7.4 in the amount of epochs necessary to reach convergence. Since physics-guided loss weighs mistakes and correct predictions within the pseudo-shadows harder, it would follow that this would allow the model to converge faster. However, this was not established over the other models. In addition, it should be noted that in all cases the standard deviation also increases, meaning that there is more variability within the folds on epochs trained. This would suggest that while the addition of physics-guided loss in some cases is a good predictor for the ground truth, in other cases where it is more ambiguous they may lead to confusion for the model, increasing variability in training speed.

Table 12: An overview of the total epochs needed to train each model compared to their ablation counterparts.

Subset	Comparison	Group A Mean Epochs	Group B Mean Epochs	Delta Mean Epochs	Delta Std. Epochs	CI95 Lower	CI95 Upper	p-value	Significant?
B	RGB_BCE30_DICE70 vs RGBS_BCE30_DICE70	38.6	58.6	20	11.4368	8.792	31.208	0.025	Yes
B	RGB_BCE50_DICE50 vs RGBS_BCE50_DICE50	37.6	54.2	16.6	9.1782	7.6053	25.5947	0.0224	Yes
B	RGB_BCE70_DICE30 vs RGBS_BCE30_DICE70	35.2	58.6	23.4	14.3889	9.2989	37.5011	0.0313	Yes
C	RGB_BCE50_DICE50 vs PHYS_ATT_0.1	37.6	38.2	0.6	3.1369	-2.4741	3.6741	0.7215	No
C	RGB_BCE50_DICE50 vs PHYS_ATT_0.5	37.6	37.8	0.2	4.1665	-3.8832	4.2832	0.9281	No
C	RGB_BCE50_DICE50 vs PHYS_ATT_1.0	37.6	36.2	-1.4	6.375	-7.6475	4.8475	0.6832	No
C	BASE_BCE vs PHYS_BCE_10	39.4	41.4	2	3.5214	-1.4509	5.4509	0.3194	No
C	BASE_BCE vs PHYS_BCE_33	39.4	43.8	4.4	5.5353	-1.0246	9.8246	0.1871	No
C	BASE_BCE vs PHYS_BCE_50	39.4	43.8	4.4	7.4993	-2.9493	11.7493	0.3057	No
C	BASE_DICE vs PHYS_DICE_10	46.8	42	-4.8	7.8333	-12.4766	2.8766	0.2876	No
C	BASE_DICE vs PHYS_DICE_33	46.8	42	-4.8	11.3031	-15.877	6.277	0.4435	No
C	BASE_DICE vs PHYS_DICE_50	46.8	39.2	-7.6	11.9097	-19.2715	4.0715	0.2709	No
D	RGBS_BCE50_DICE50 vs HYB_BCE_PHYS10	54.2	58.2	4	8.7636	-4.5883	12.5883	0.413	No
D	RGBS_BCE50_DICE50 vs HYB_BCE_PHYS30	54.2	55.4	1.2	10.9618	-9.5425	11.9425	0.8374	No
D	RGBS_BCE50_DICE50 vs HYB_BCE_PHYS50	54.2	57.8	3.6	6.4062	-2.6781	9.8781	0.3239	No
D	RGBS_BCE50_DICE50 vs HYB_DICE_PHYS10	54.2	55.2	1	2.0976	-1.0557	3.0557	0.3943	No
D	RGBS_BCE50_DICE50 vs HYB_DICE_PHYS30	54.2	58.8	4.6	7.4993	-2.7493	11.9493	0.2872	No
D	RGBS_BCE50_DICE50 vs HYB_DICE_PHYS50	54.2	46.8	-7.4	4.5869	-11.8952	-2.9048	0.0321	Yes
D	RGBS_BCE50_DICE50 vs HYB_ATT_0.1	54.2	50.2	-4	7.9246	-11.7662	3.7662	0.3698	No
D	RGBS_BCE50_DICE50 vs HYB_ATT_0.5	54.2	55.2	1	6.8702	-5.7328	7.7328	0.7854	No
D	RGBS_BCE50_DICE50 vs HYB_ATT_1.0	54.2	52.6	-1.6	4.3174	-5.8311	2.6311	0.4997	No

5.9 Ablation Summary & Model Selection

In order to select a PHYSHADE model configuration that will be trained on the full dataset for the final height estimations, Table 13 was produced ranking all different experiments by their mean Dice scores. Overall, HYB BCE PHYS10 was the best scoring model overall (0.8523 ± 0.0258), followed by RGBS BCE70 DICE30 (0.8487 ± 0.0277). Five positions down from HYB BCE PHYS10, its ablational counterpart RGBS BCE50 DICE50 can be found which managed to score middle-of-the-pack between the subset B and D models.

For the purposes of cross comparison later on for the height estimation, the decision was made to train three different models on the full dataset, with validation on the out-of-fold dataset:

1. **HYB BCE PHYS10**, as this model scored the best Dice overall and includes the physics-guided loss.
2. **RGBS BCE50 DICE50**, as this model has the exact same architecture and BCE/Dice weighting parameters, and can be used to ablate the effect of physics-guided loss.
3. **RGBS BCE70 DICE30**, as this model was the best performing model not containing physics-guided loss

Table 13: Table of all experimental configurations trained, averaged and sorted by mean Dice score.

Experiment ID	Mean Dice	Std. Dice	Mean Precision	Std. Precision	Mean Recall	Std. Recall	Mean Loss	Std. Loss	Mean Epochs Run
HYB BCE PHYS10	0.8523	0.0258	0.8498	0.0344	0.8320	0.0222	0.1402	0.0214	58.2
RGBS BCE70 DICE30	0.8487	0.0277	0.8470	0.0403	0.8118	0.0391	0.0672	0.0097	53.2
RGBS BCE30 DICE70	0.8475	0.0259	0.8435	0.0552	0.8239	0.0283	0.1245	0.0126	58.6
HYB ATT 0.1	0.8472	0.0235	0.8433	0.0395	0.8205	0.0245	0.0975	0.0097	50.2
HYB ATT 0.5	0.8467	0.0277	0.8489	0.0378	0.8235	0.0333	0.0963	0.0127	55.2
RGBS BCE50 DICE50	0.8455	0.0242	0.8488	0.0380	0.8172	0.0321	0.0971	0.0095	54.2
HYB DICE PHYS30	0.8447	0.0286	0.8139	0.0544	0.8540	0.0310	0.2614	0.0241	58.8
HYB DICE PHYS10	0.8443	0.0266	0.8329	0.0494	0.8317	0.0277	0.1573	0.0147	55.2
HYB ATT 1.0	0.8417	0.0243	0.8403	0.0387	0.8189	0.0273	0.1018	0.0125	52.6
HYB BCE PHYS30	0.8349	0.0270	0.8036	0.0402	0.8453	0.0253	0.1947	0.0301	55.4
HYB DICE PHYS50	0.8182	0.0267	0.6569	0.1132	0.8944	0.0185	0.3782	0.0347	46.8
HYB BCE PHYS50	0.8077	0.0327	0.7237	0.0599	0.8669	0.0288	0.2433	0.0390	57.8
PHYS DICE 50	0.5396	0.0413	0.5108	0.0832	0.4413	0.0343	0.5432	0.0231	39.2
PHYS ATT 0.5	0.5394	0.0268	0.5465	0.1029	0.4119	0.0628	0.2841	0.0321	37.8
PHYS DICE 10	0.5327	0.0247	0.5548	0.0914	0.4149	0.0517	0.3451	0.0263	42.0
PHYS ATT 1.0	0.5325	0.0214	0.5578	0.0812	0.4307	0.0564	0.2810	0.0295	36.2
BASE DICE	0.5309	0.0331	0.5885	0.0944	0.4301	0.0610	0.4915	0.0566	46.8
PHYS BCE 10	0.5306	0.0341	0.5754	0.0965	0.4346	0.0365	0.3244	0.0244	41.4
PHYS DICE 33	0.5267	0.0292	0.5479	0.0875	0.4330	0.0480	0.4183	0.0136	42.0
PHYS ATT 0.1	0.5264	0.0177	0.5436	0.0783	0.4153	0.0591	0.2913	0.0275	38.2
RGB BCE70 DICE30	0.5249	0.0230	0.5550	0.0955	0.4070	0.0349	0.2011	0.0219	35.2
PHYS BCE 33	0.5234	0.0577	0.5543	0.1317	0.4161	0.0397	0.3242	0.0372	43.8
RGB BCE30 DICE70	0.5225	0.0313	0.5662	0.1033	0.4295	0.0754	0.3837	0.0323	38.6
RGB BCE50 DICE50	0.5208	0.0271	0.5578	0.0875	0.4186	0.0744	0.3007	0.0301	37.6
PHYS BCE 50	0.5108	0.0774	0.5076	0.1378	0.4384	0.0359	0.3570	0.0462	43.8
BASE BCE	0.4978	0.0674	0.5447	0.1145	0.3778	0.0335	0.0418	0.0136	39.4

5.10 Model Training on Full Dataset & Results

For the purposes of evaluating PHYSHADE's ability to generalize outside its current domain, the three model configurations posed in subsection 5.9 were trained on the full dataset of 35 images, similarly multiplied using data augmentation to a total of 280 training images. The out-of-fold dataset was used to validate the training, and to guide early stopping as soon the validation Dice starts to stagnate. Besides this, all other training parameters were kept identical to those posed in Table I.1.

Before looking at the individual model performance, the individual images were evaluated to see how they compare to each other in terms of Dice score, precision and recall. To do so, inference was ran on each individual image using all three models, after which the three resulting testing statistics were averaged. The results of this can be seen in Table 14. Here, a clear distinction can be seen between the Dice means between the summer and winter tiles, where winter scores lower Dice ranges and higher standard deviations. These winter tiles will be assessed during the qualitative analysis to see if and how they may differ in comparison to their summer counterparts, which is especially the case for winter Tile 7 even compared to the other winter tiles. To see whether there are any visible reasons for this, the tile will be evaluated during the qualitative analysis. Finally, an overview of the performance of the individual configurations of PHYSHADE on the out-of-fold dataset can be found in Table 15. For comparison's sake, the baseline model LUO UNET was added which was the model trained on the Luo et al. (2020) dataset and used for transfer learning. As can be seen, the performance of the models when applied to imagery outside of the domain of the PHYSHADE dataset drops, leading to a ≈ 0.11 decrease in Dice score. Interestingly enough, even though RGBS BCE70 DICE30 was ranked as the second best performing model based on the 5-fold cross validation, it scored considerably lower here when compared to RGBS BCE50 DICE50 which ranked four positions below before. The decrease in Dice score for this model can mostly be attributed to the considerable drop in precision and relatively smaller increase in recall, which would suggest that the model is predicting less conservatively at the cost of more false negatives. RGBS BCE50 DICE50 sees a similar pattern, but the trade-off works out to a favourable Dice score where although the share of false negatives increases, it is compensated with the highest mean recall score of all. Due to the domain shift, it may be possible that the features that PHYSHADE were using inside the domain may not be reliably found in the new domain. In this sense, it seems that the pseudo-shadow geometric priors are good at forcing the model to not predict shadows that fall outside them, but are less valuable in helping find actual shadows within the bounds of the pseudo-shadow. As expected, the precision of the LUO UNET model comparatively is poor, since the model is trained originally for predicting all shadows and not only building shadows leading to mismatches with the ground truth. However, its recall is the highest of all models, which either may lead to overestimation of heights if preprocessing is insufficient.

When compared to the results of the 5-fold cross validation, HYB BCE PHYS10 and RGBS BCE70 DICE30 saw increases in their precision, offset by a more or equally hefty decrease in recall. RGBS BCE50 DICE50 saw decreases in both precision and recall but not as severe, leading it to become the second-best performing model in the out-of-fold dataset.

Table 14: An overview of the per-image statistics in the out-of-fold dataset, calculated by averaging the statistics of HYB BCE PHYS10, RGBS BCE70 DICE30 and RGBS BCE50 DICE50.

Image	Dice Mean	Std. Dice	Precision Mean	Std. Precision	Mean Recall	Std. Recall
Summer Tile 6	0.8996	0.0149	0.9357	0.0135	0.8671	0.0385
Summer Tile 7	0.9526	0.0288	0.9749	0.0052	0.9321	0.0515
Summer Tile 8	0.8294	0.1012	0.7854	0.1338	0.8827	0.0695
Winter Tile 6	0.7819	0.1497	0.8614	0.0485	0.7517	0.2568
Winter Tile 7	0.2836	0.1895	0.8926	0.1348	0.1872	0.1475
Winter Tile 8	0.7159	0.1627	0.6557	0.2224	0.8273	0.1289

Table 15: An overview of the statistics when applying the final models to the out-of-fold dataset.

Experiment	Mean Dice	Std. Dice	Mean Precision	Std. Precision	Mean Recall	Std. Recall	Delta Dice	Delta Precision	Delta Recall
HYB BCE PHYS10	0.7636	0.2612	0.9327	0.043	0.7109	0.3196	-0.0887	0.0829	-0.1211
RGBS BCE70 DICE30	0.7213	0.3266	0.8654	0.1267	0.7192	0.3439	-0.1275	0.0184	-0.0926
RGBS BCE50 DICE50	0.7466	0.1897	0.7548	0.1951	0.7939	0.2279	-0.0989	-0.094	-0.0233
LUO UNET	0.4483	0.2049	0.3562	0.1828	0.8593	0.2443	-	-	-

5.11 Qualitative Analysis: Final Models

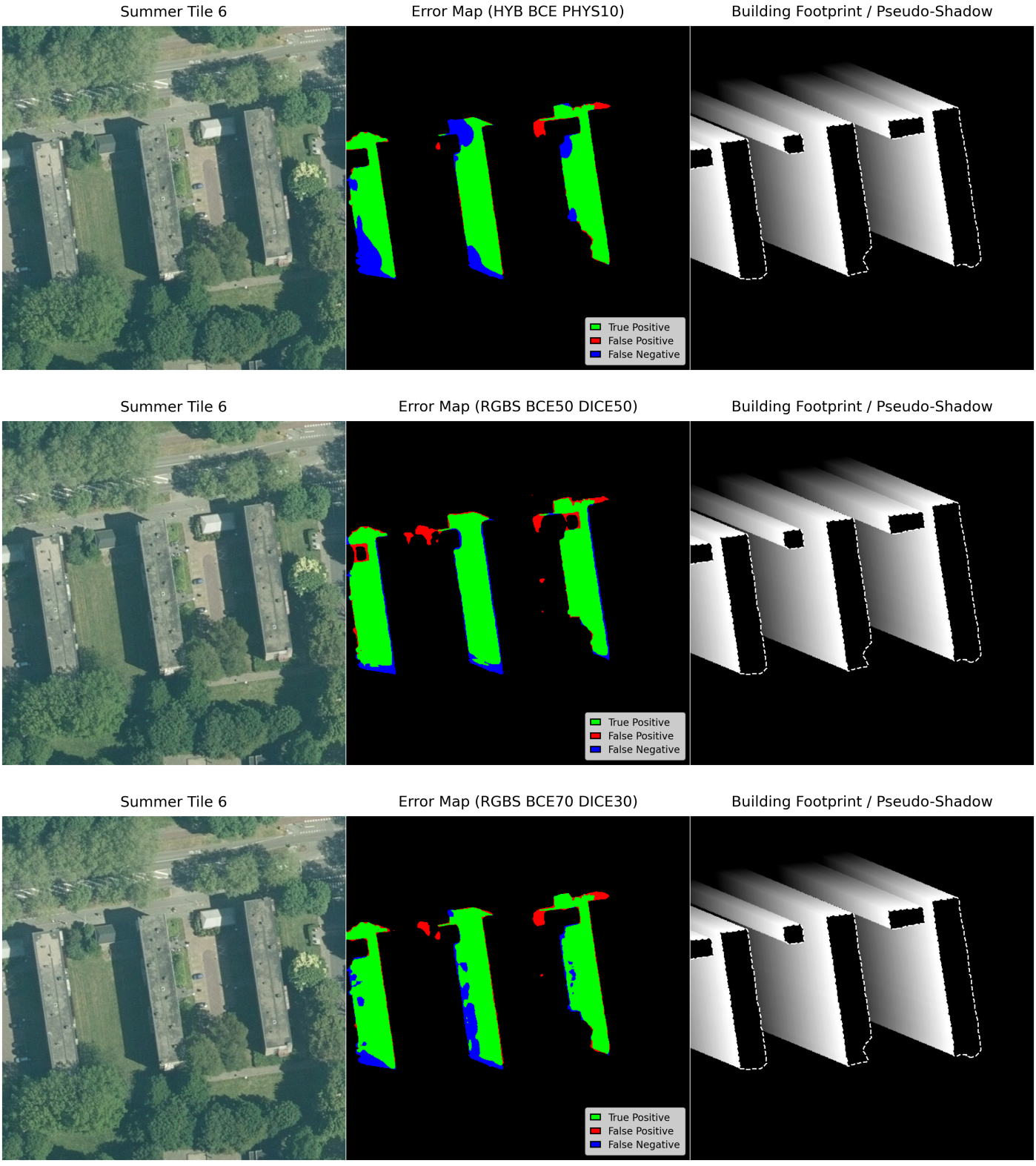

Figure 13: Overview of the results per configuration for Summer Tile 6. Overall, segmentation quality was relatively good across all three models. RGBS BCE50 DICE50 had the least false negatives but at the cost of higher false positives. Between HYB BCE PHYS10 and RGBS BCE50 DICE50, the addition of physics-guided loss helps avoid these false positives but slightly worsens recall. RGBS BCE70 DICE30 shows a good balance between precision and recall, although false negatives are distributed more sporadically throughout the shadow.

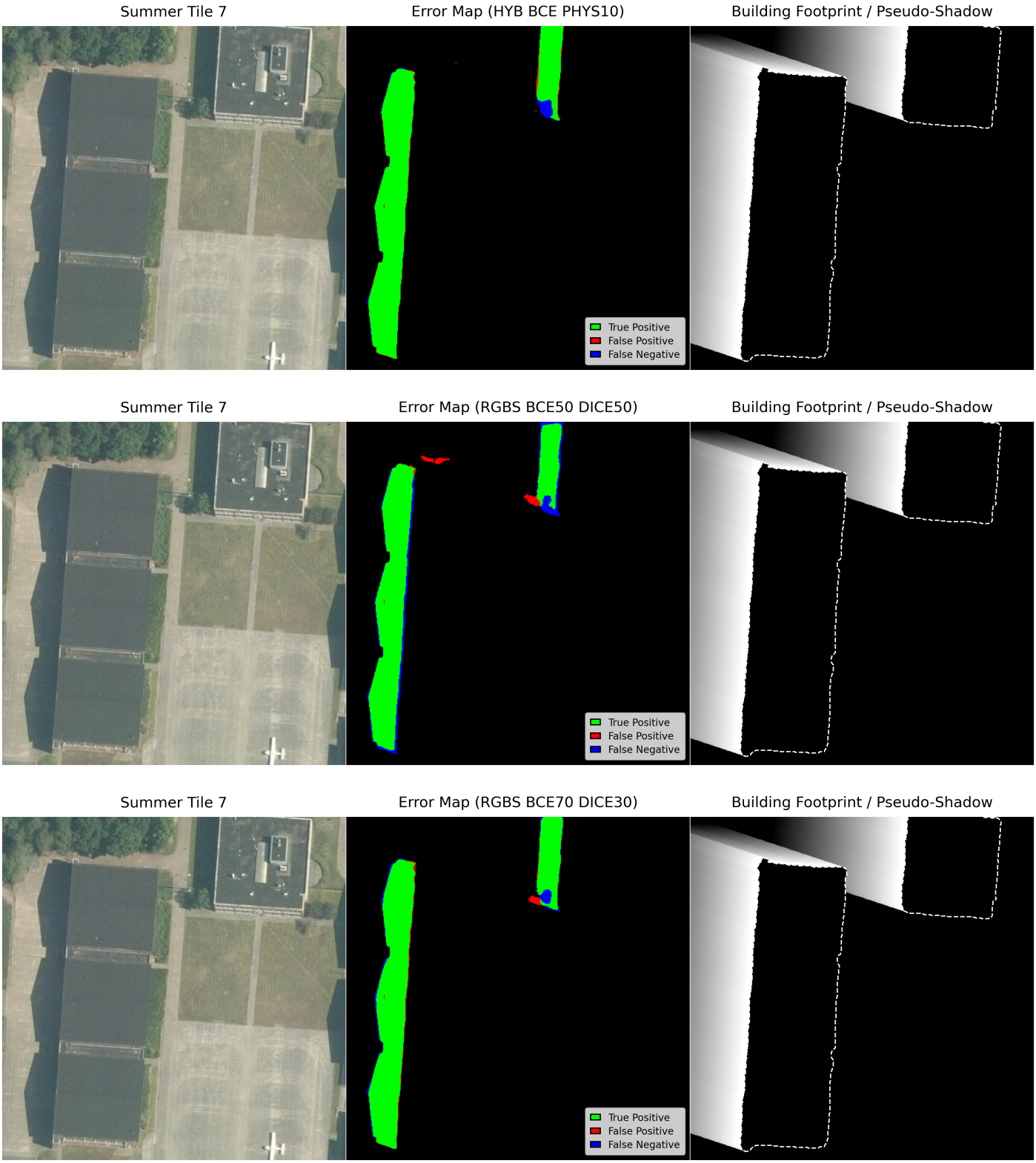

Figure 14: Overview of the results per configuration for Summer Tile 7. All models performed well here, following similar trends as described in Figure 13. In the hangar shadow on the left, a small black sliver is visible in each model's output. Upon inspection, this turned out to be a few missed pixels in the dataset, leading to an erroneous false positive.

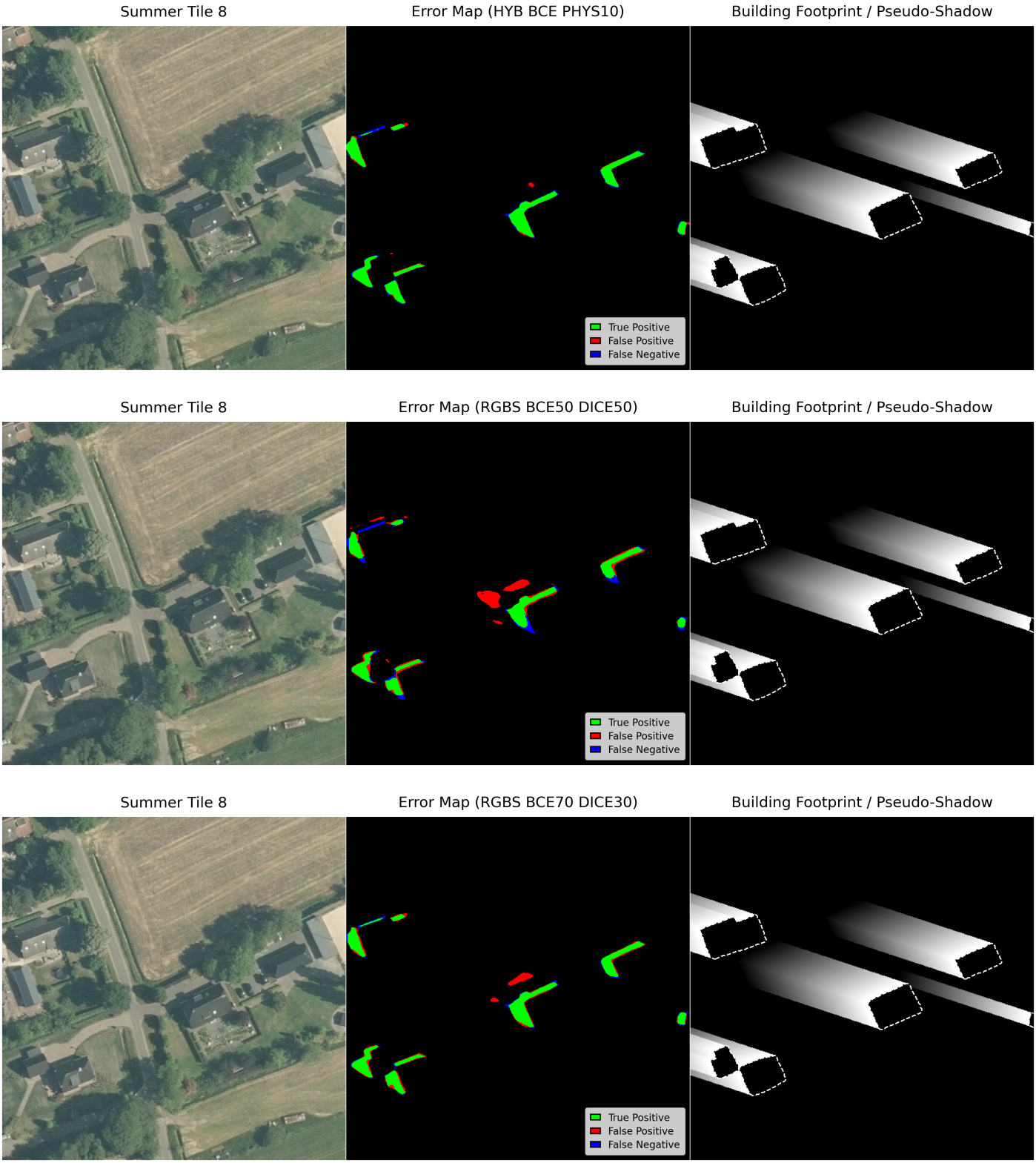

Figure 15: Overview of the results per configuration for Summer Tile 8. Here it can be seen that HYB BCE PHYS10 is the clear winner compared to the other two models; it has next to no false positives, and very little false negatives. In this sense, the addition of the benefits of physics-guided loss over RGBS BCE50 DICE50 are clear, which suffers from more false positives and negatives.

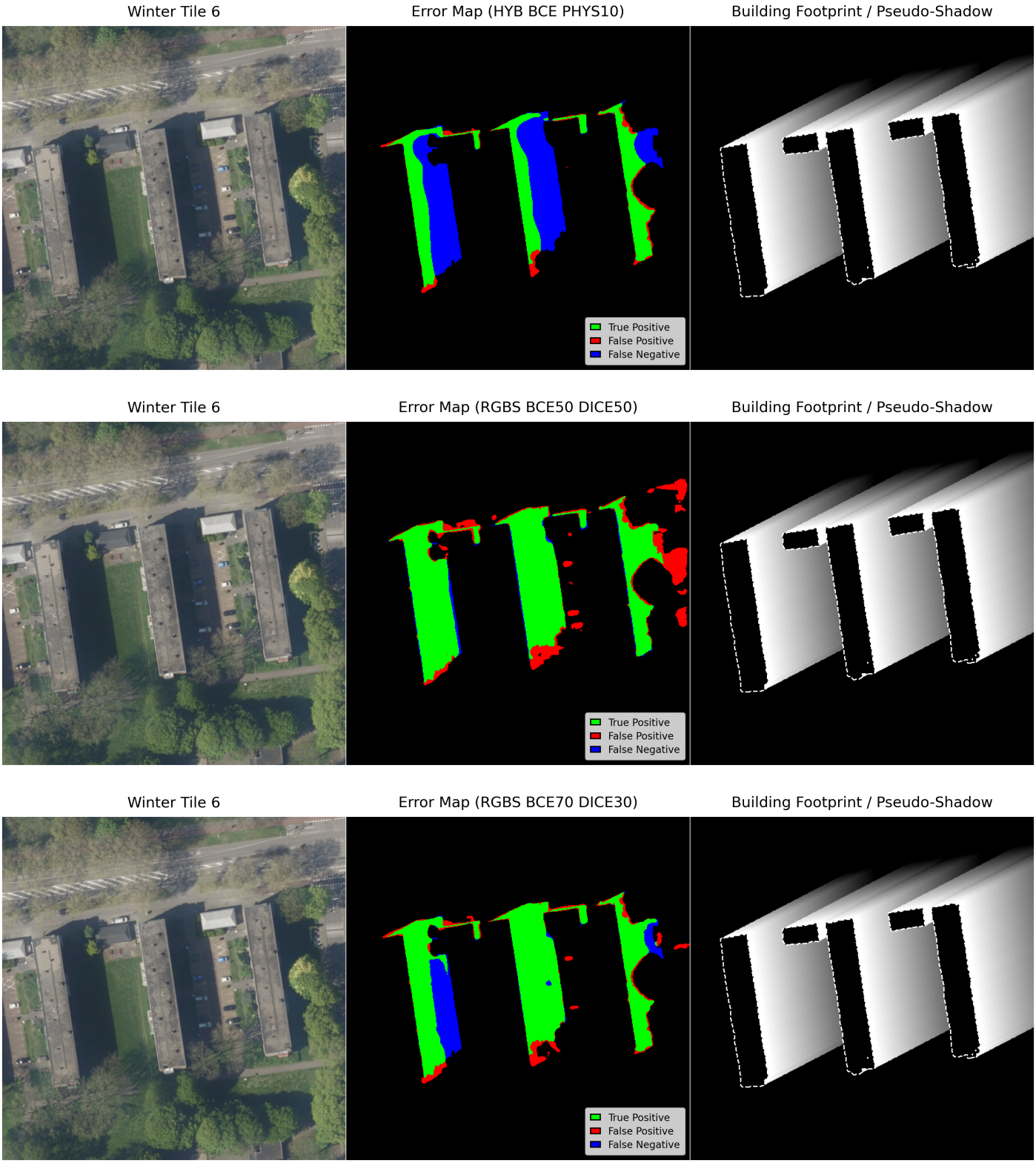

Figure 16: Overview of the results per configuration for Winter Tile 6. Compared to its summer equivalent, the performance for the HYB BCE PHYS10 drops drastically, as can be seen in the large vertical swaths of false negatives. The specific reason for this is unclear; the middle building has a patch of grass next to the paved parking lot that seems to loosely follow the TP/FN divide which could be explanatory, but this same pattern is not seen in the left building whose shadows fall over grassland, even though it has this same TP/FN division. For these images, RGBS BCE50 DICE50 seems to perform the best as it is not as conservative at guessing, getting almost all of the true positive. RGBS BCE70 DICE30 is the middle-ground model here; having both a relatively good true-positive rate whilst avoiding some (but not all) of the vertical swaths of false negatives.

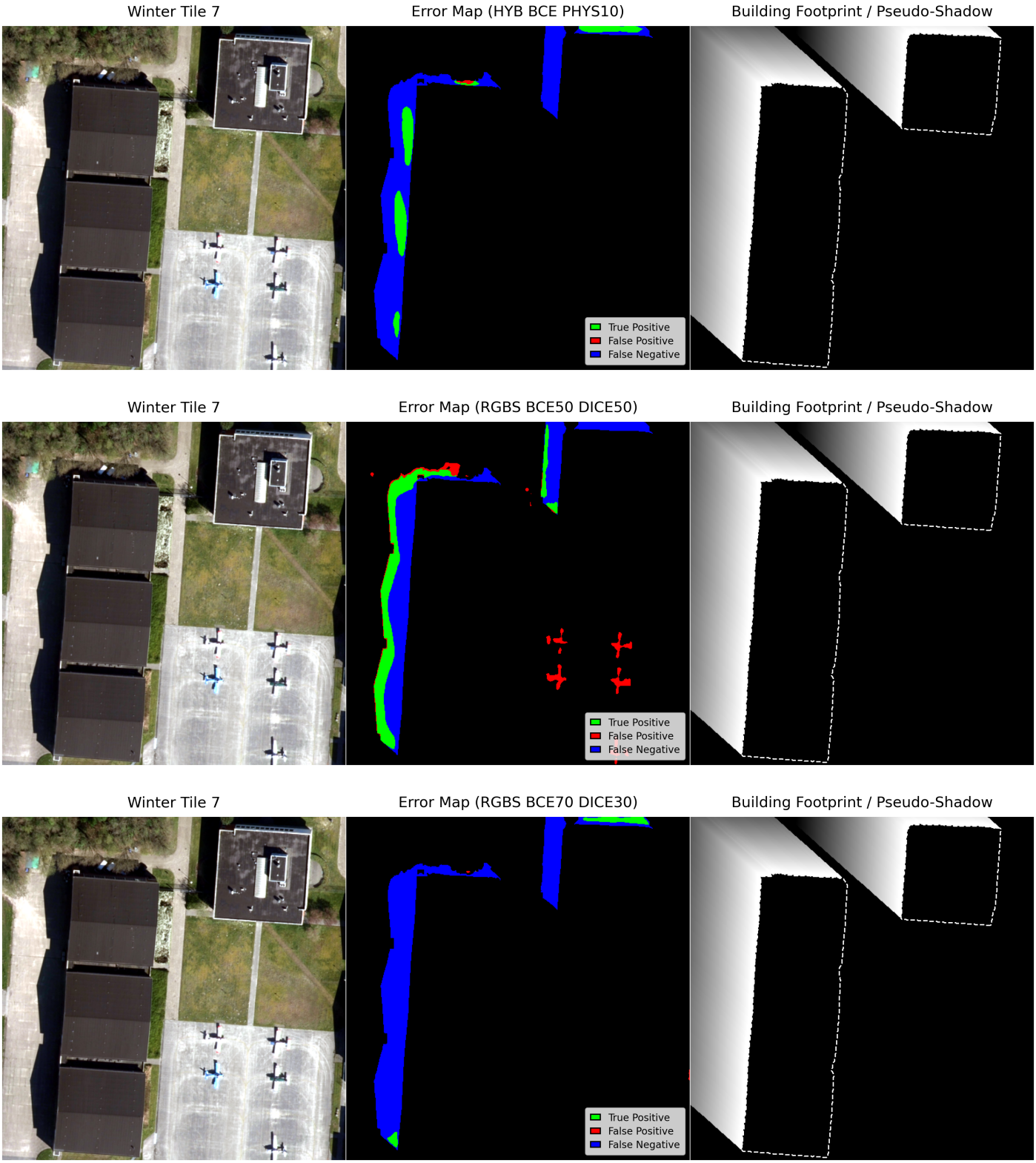

Figure 17: Overview of the results per configuration for Winter Tile 7. Of all tiles, the performance of all models was the worst here as can be seen by the huge share of false negatives. RGBS BCE70 DICE30 scored the lowest of all, making almost zero predictions. In terms of Dice scores, HYB BCE PHYS10 and RGBS BCE50 DICE50 are very similar, with the first being more conservative avoiding false positives and the latter more aggressive in its inference. Interestingly enough, RGBS BCE50 DICE50 predicts outside of the pseudo-shadows for the first time here, which is a phenomenon not yet seen in any of the other images or other models. One guess for why this image performed so poorly overall is the lighting conditions; it has a more yellowish tint than the images in the training dataset or the other validation imagery, meaning that the model is not generalizing well to all potential out-of-fold conditions despite augmentation.

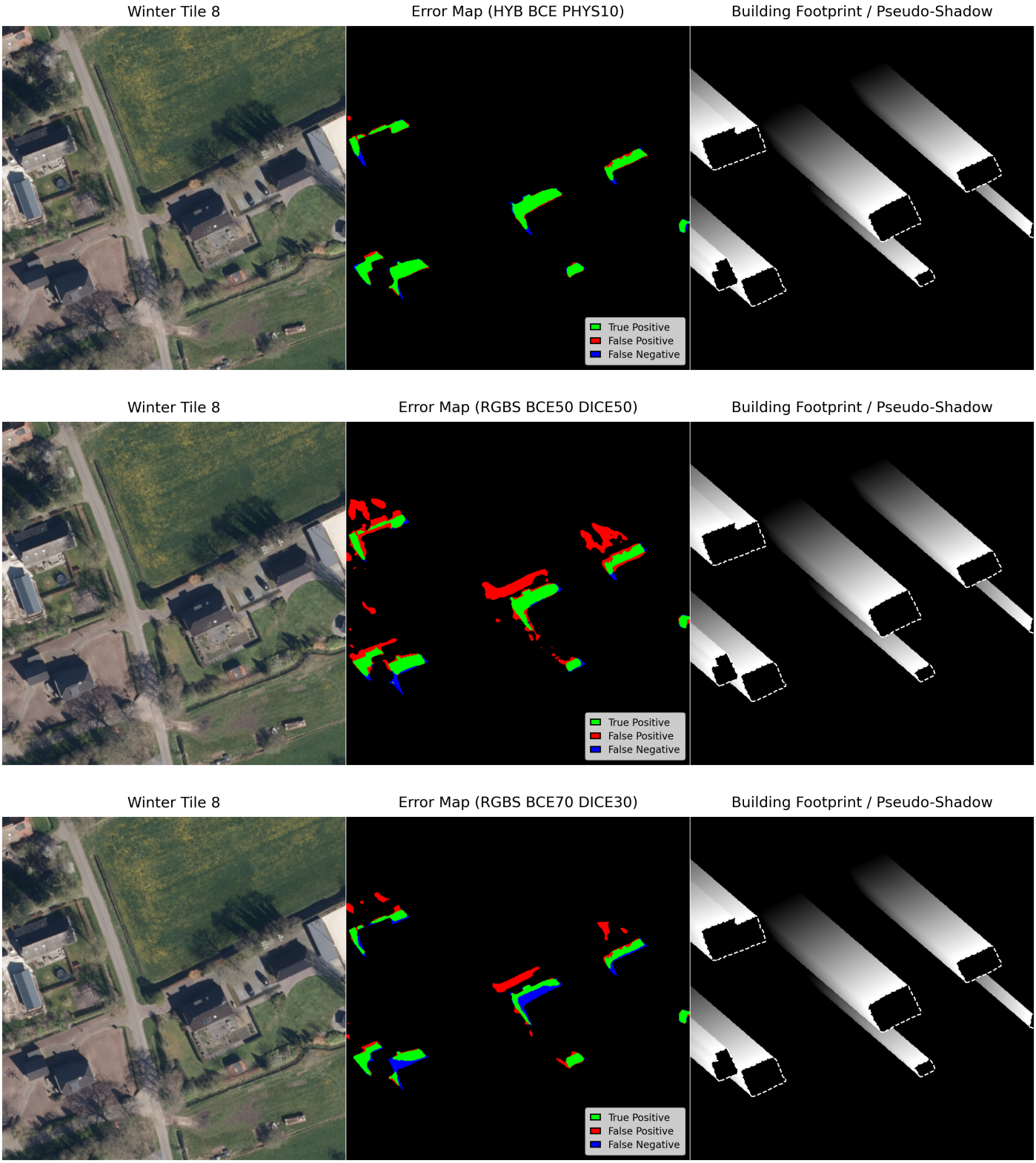

Figure 18: Overview of the results per configuration for Winter Tile 8. This image and the resulting model inference has a lot of similar charateristics to its summer counterpart, where HYB BCE PHYS10 performs the best, RGBS BCE50 DICE50 overestimates and RGBS BCE70 DICE30 falls in the middle ground between them. One difference between summer and winter inferences is that the false positive rate for both RGBS models has increased likely due to the different lighting conditions. However, HYB BCE PHYS10 seems to not have suffered much under the change in seasons. Once again, it can be seen here that RGBS BCE50 DICE50 seems to wrongly infer outside of the pseudo-shadows below the top-left building, which is not reflected in the other two models.

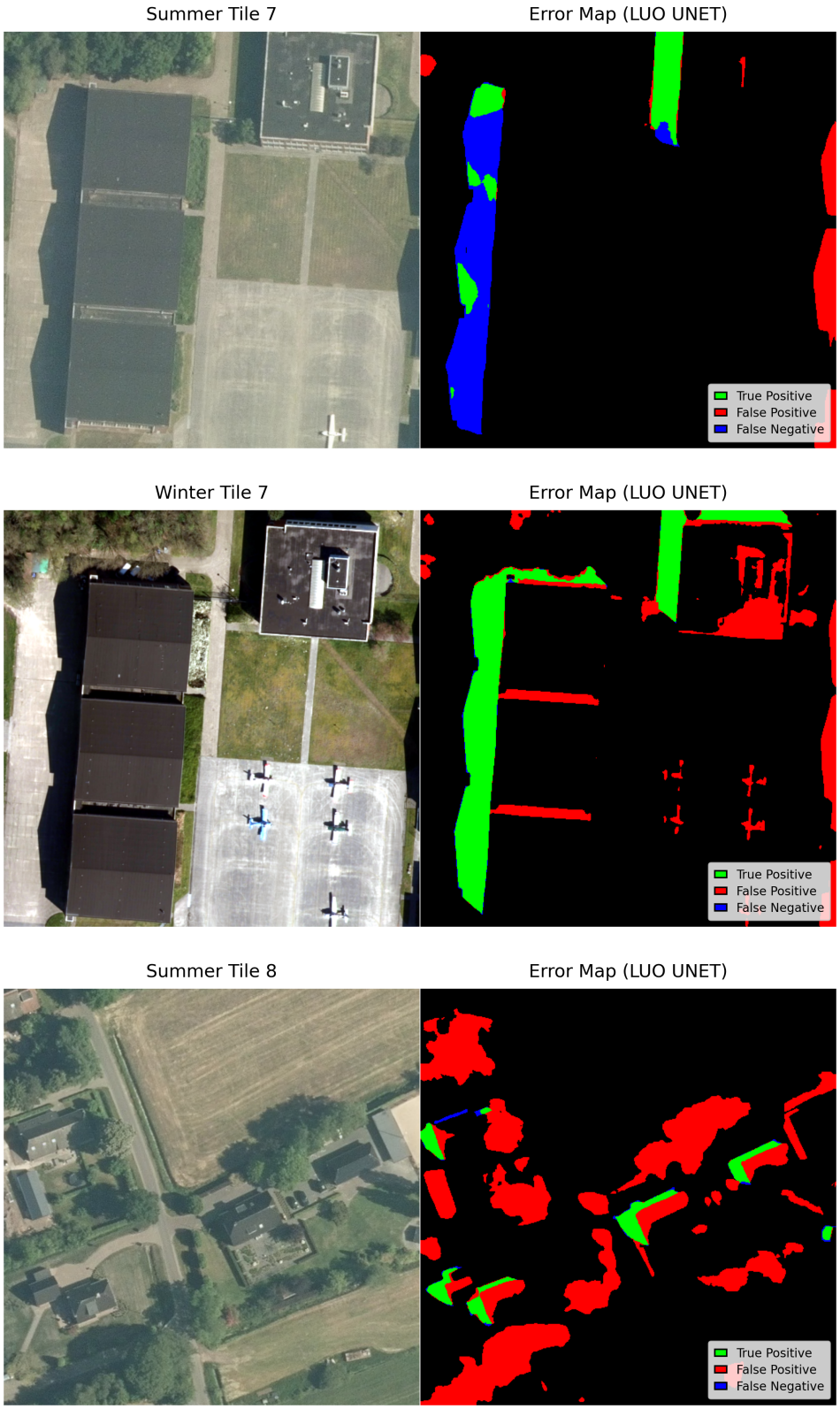

Figure 19: An assorted overview of results when applying the LUO UNET model to the dataset. Note that the ground truth here is still dictated by building shadows only; it is therefore not fully representative of the performance of the LUO UNET when applied for the purposes of detecting all shadows. Interestingly, what can be seen here is that the performance of the LUO UNET model is reversed for the Summer and Winter pairs of tile 7; whereas the PHYSHADE models all have trouble segmenting Winter Tile 7 for shadows, LUO UNET seems to have no issue and has a high recall as a result. At the same time, recall drops for the summer version, where the PHYSHADE models did better. Outside of LUO UNET being trained on data outside far outside the current domain, the specific explanation for this switch remains unclear. In Summer Tile 8, it can be seen that while the model has high recall, its precision is poor in some cases, segmenting objects such as trees erroneously in addition. This is a trend for the other images in the out-of-fold datset as well.

5.12 Height Estimation from Shadows

To estimate the heights using the inference from the PHYSHADE models, an algorithm based on raymarching was created to extrapolate the height back from a raster containing building shadows. The algorithm can be found described with pseudocode in the appendices as Algorithm J.2. Since the rays may have to traverse the grid outside of the 8 compass directions (i.e. not perfectly diagonal) depending on the solar azimuth, pixel stepping may lead to jagged paths which hurts the precision of the algorithm. At lower resolutions, these effects become more pronounced thus leading to higher errors.

5.12.1 Height Estimation Algorithm Baseline Error

To verify the error-proneness inherent to raster-based raymarching operations as described above, a controlled validation was first ran using building footprints and solar metadata to generate synthetic shadows with known lengths. By running these shadows with known lengths through the height estimation algorithm, it becomes possible to see what potential error is inherent within the algorithm itself as opposed to the errors introduced by the inference of real-world shadows, thus giving a better idea of what term of the error can be attributed to the algorithm and what term can be attributed to the PHYSHADE models. For evaluation, synthetic shadows were generated using the building footprints of the out-of-fold dataset at azimuths of 60, 120, 180, 240 and 300. Since shadows falling outside of the raster are unable to lead to accurate results as any shadow information outside the extents are lost, these were omitted for the benchmark. With the raymarching algorithm, selecting the ray that is most representative for the actual height is dependent on the spread of outliers. Due to this, the benchmark was ran at various ray length percentiles, which are defined as the accepted ray percentile from the group of all collected rays per a given shadow blob. The statistics by the benchmark can be found in Table 16.

Table 16: An overview of the error statistics when running the raster algorithm on synthetic shadows generated by the smearing algorithm.

Ray Length Percentile	Mean Error	RMSE	Std. Error	Median Error	P90 Error	P95 Error	Max Error
100	0.0169	0.0822	0.0807	0.0	0.0	0.0	0.5
99	0.0056	0.0317	0.0313	0.0	0.0	0.0	0.25
95	0.0236	0.0964	0.0938	0.0	0.0	0.25	0.75
90	0.0446	0.2113	0.2073	0.0	0.0	0.25	2.0
75	0.2183	0.8021	0.7744	0.0	0.325	1.4125	5.625

Looking at the Table 16, the best performing percentile parameter for the height estimation algorithm is 99, sporting the lowest RMSE, standard deviation and maximum error. The 99th percentile fits a good niche between 100 and 95 where its associated errors are minimized, making it a good sampling strategy when the algorithm is applied to synthetic shadows. That said, as can be seen for all of the rays, the median error is 0, meaning that the majority of rays cast matched with the ground truth. Considering the fact that the rasters on which the algorithm was ran are 25cm/pixel, the performance of the algorithm overall is very good and unlikely to lead to large error terms.

5.12.2 Height Estimation Quality on Out-of-Fold Dataset

After the statistics for the height estimation algorithm's error-proneness were produced using known synthetic data, the height estimator was then deployed on the out-of-fold domain dataset using the inferred shadows produced by the PHYSHADE models HYB BCE PHYS10, RGBS BCE50 DICE50 and RGBS BCE70 DICE30. To gain an impression of the height estimation error per image in the out-of-fold dataset, Table 17 was generated. These statistics were generated based upon valid blobs, meaning that they are the blobs that were able to get through the filtering process as described in the methodology and were the final singular largest blob picked per building. The blob weighted average takes into account the amount of blobs generated per tile, as opposed to simply combining the statistics per tile and dividing over six. In Table 17, it can be seen that the average error hovers at ≈ -0.3036, meaning that the model on average tends to slightly underestimate the building height as opposed to overestimation within the out-of-fold dataset. Generally, height estimation performance is linked in part to the performance of segmentation; Winter Tile 7 has the highest RMSE term, whereas Summer Tiles 7 and 8 have the lowest. Generally the RMSE averages at around ≈ 1.91 meters, with Summer Tile 6, Winter Tile 6 and Winter Tile 7 having a higher RMSE than average. During the qualitative analysis on the height estimation, these will be regarded in particular to see why this might be the case.

Table 17: An overview of the averaged height estimation performances per image produced by HYB BCE PHYS10, RGBS BCE50 DICE50 and RGBS BCE30 DICE70.

Image	Mean Error	Mean Absolute Error	RMSE	Std. Residuals	Mean Est. Height	Mean True Height	Min Est. Height	Max Est. Height
Summer Tile 6	1.8106	2.0248	2.6009	1.9214	9.818	8.0073	1.851	14.7363
Summer Tile 7	0.3654	0.7393	0.7725	0.786	8.8688	8.5034	6.6386	9.958
Summer Tile 8	-0.4708	0.7234	0.8748	0.7587	4.8645	5.3353	2.4865	6.4235
Winter Tile 6	-1.3822	1.561	2.7984	2.5186	7.6386	9.0209	2.3961	13.7778
Winter Tile 7	-3.5174	3.5174	3.7343	1.402	4.7626	8.28	2.8361	7.4953
Winter Tile 8	-0.2743	1.1711	1.3999	1.4067	4.6934	4.9677	2.397	7.3907
Blob Weighted Average	-0.3036	1.4899	1.9135	1.4892	6.5275	6.8311	2.5997	9.8472

The performance of the height estimation algorithm per individual model of PHYSHADE can be seen in Table 18. In addition, inference was ran using the LUO UNET basemodel to see whether or not PHYSHADE built to segment only building shadows would perform better than a general purpose shadow segmentation model. Of the PHYSHADE models, RGBS BCE50 DICE50 scored the lowest Mean Absolute Error. Considering the fact that the Dice score for that model was the lowest, it is interesting to see that it outperforms RGBS BCE70 DICE30 by a bit and HYB BCE PHYS10 by a lot, even though that model scored the best Dice score. A possible explanation for this is that although precision is an important metric if one wants accurate estimations of shadows themselves, the height estimation algorithm performs better in cases where the recall is high and is able to deal with the lower precision through the filtering steps taken. In addition, the standard deviation of the residual errors is low, meaning that there is a low variation in the error statistics produced by this model.

However, all of the PHYSHADE models were outperformed by the LUO UNET model. This likely has to do with the combination of high recall and the preprocessing steps taken to go from inference towards height estimation, as the preprocessing steps taken may be relatively robust to over-segmentation. Looking at Table M.1, it can be seen that the LUO Model has about equal non-highest blobs to the other models, meaning that the pre-processing was effective in getting rid of shadow blobs unrelated to the buildings. In addition, for each blob, the height is determined by taking the

99th percentile of all rays cast for that blob which could clamp down on outliers produced by height estimation from LUO UNET inference.

To see the differences in how the height estimation algorithm determined the heights per the different models of PHYSHADE and LUO UNET, visualisations of the representative raycast line were created as seen in figures 20 to 25. Additionally, Table 19 contains the individual statistics per model per image. Here, it can be seen why HYB BCE PHYS10 sees such an increase in RMSE: Winter Tile 6 and 7 were especially difficult to estimate, even though the other two model configurations were able to score much lower terms. That said, RGBS BCE50 DICE50 and RGBS BCE70 DICE30 seem to score better regardless; across all images except two, they outperformed the HYB BCE PHYS10 model, further strengthening the idea that recall is a more important metric than precision for the purposes of height estimation. LUO UNET seems to not have any major outliers when compared to the PHYSHADE models which contributes to its relative mean performance.

Table 18: An overview of the average performance of height estimation per model of PHYSHADE applied to the out-of-fold dataset.

Model	Mean Error	Mean Absolute Error	RMSE	Std. Residuals	Mean Est. Height	Mean True Height	Min Est. Height	Max Est. Height
HYB BCE PHYS10	-0.8976	1.9068	2.7489	2.6477	5.9569	6.8545	1.851	14.7363
RGBS BCE50 DICE50	0.3397	1.1832	1.6948	1.6909	7.2132	6.8735	2.5958	14.5126
RGBS BCE70 DICE30	-0.1453	1.2849	1.774	1.8031	6.6888	6.834	2.0889	14.5126
Blob Weighted Average	-0.2284	1.4571	2.0716	2.0458	6.6261	6.8545	2.1848	14.5872
LUO UNET	0.2616	1.0805	1.5315	1.5377	7.1161	6.8545	1.6559	15.2264

Table 19: An overview of the metrics of height estimation for each model of PHYSHADE per image.

Model	Image	Mean Error	Mean Absolute Error	RMSE	Std. Residuals	Mean Est. Height	Mean True Height	Min Est. Height	Max Est. Height
HYB BCE PHYS10	Summer Tile 6	1.6646	2.0255	2.5509	2.1175	9.6719	8.0073	1.851	14.7363
	Summer Tile 7	1.0823	1.0823	1.0823	-	9.958	8.8757	9.958	9.958
	Summer Tile 8	-0.4817	0.6269	0.8623	0.7834	4.8536	5.3353	2.4865	6.4235
	Winter Tile 6	-3.8292	3.8292	4.7869	3.2115	5.1916	9.0209	2.3961	8.3865
	Winter Tile 7	-4.1808	4.1808	4.1971	0.5226	3.9502	8.1311	2.8361	5.0644
	Winter Tile 8	-0.701	0.9971	1.1674	1.0082	4.2666	4.9677	2.397	5.6808
RGBS BCE50 DICE50	Summer Tile 6	2.303	2.303	2.9424	2.0063	10.3103	8.0073	5.4149	14.5126
	Summer Tile 7	-0.144	0.6038	0.6208	0.854	7.9871	8.1311	6.6386	9.3356
	Summer Tile 8	-0.5014	0.7366	0.8605	0.7661	4.8339	5.3353	2.9009	5.8848
	Winter Tile 6	0.0445	0.4186	0.4947	0.5508	9.0654	9.0209	2.5958	13.7778
	Winter Tile 7	-2.1753	2.1753	2.316	1.1242	5.9557	8.1311	4.4162	7.4953
	Winter Tile 8	0.4455	1.0346	1.2715	1.2864	5.4132	4.9677	3.196	7.3907
RGBS BCE70 DICE30	Summer Tile 6	1.4643	1.746	2.2649	1.8927	9.4717	8.0073	2.0889	14.5126
	Summer Tile 7	0.6673	0.6673	0.6673	-	9.543	8.8757	9.543	9.543
	Summer Tile 8	-0.4292	0.8067	0.9009	0.8677	4.9061	5.3353	2.9009	6.2287
	Winter Tile 6	-0.362	0.4353	0.5782	0.5041	8.6588	9.0209	2.3961	13.1788
	Winter Tile 7	-4.8748	4.8748	4.8748	-	4.0009	8.8757	4.0009	4.0009
	Winter Tile 8	-0.5672	1.4818	1.7029	1.7343	4.4004	4.9677	2.7965	6.5917
LUO UNET	Summer Tile 6	1.9179	2.3439	2.7906	2.2206	9.9253	8.0073	1.6559	15.2264
	Summer Tile 7	-0.9923	0.9923	0.9923	-	7.8834	8.8757	7.8834	7.8834
	Summer Tile 8	-0.4019	0.5789	0.8387	0.8064	4.9333	5.3353	2.4554	6.6307
	Winter Tile 6	-0.0865	0.5207	0.5949	0.6581	8.9344	9.0209	2.1965	13.7219
	Winter Tile 7	1.0598	1.0598	1.3285	1.1328	9.1909	8.1311	7.6452	10.7365
	Winter Tile 8	-0.3897	0.846	0.9333	0.916	4.5779	4.9677	2.7965	6.1642

Table 20: Results of Paired T-Testing between the LUO UNET baseline vs the PHYSHADE models for height estimation.

Comparison	Metric	LUO UNET Mean	vs. Mean	t-statistic	p-value	Significant?
LUO UNET vs HYB BCE PHYS10	Mean Error	0.1846	-1.0743	1.1423	0.3051	No
LUO UNET vs HYB BCE PHYS10	Mean Absolute Error	1.0569	2.1236	-1.5618	0.1791	No
LUO UNET vs HYB BCE PHYS10	RMSE	1.2464	2.4412	-1.5707	0.177	No
LUO UNET vs HYB BCE PHYS10	Std. Residuals	1.1468	1.5286	-0.6868	0.53	No
LUO UNET vs RGBS BCE50 DICE50	Mean Error	0.1846	-0.0046	0.3011	0.7755	No
LUO UNET vs RGBS BCE50 DICE50	Mean Absolute Error	1.0569	1.212	-0.7382	0.4936	No
LUO UNET vs RGBS BCE50 DICE50	RMSE	1.2464	1.4176	-0.9005	0.4091	No
LUO UNET vs RGBS BCE50 DICE50	Std. Residuals	1.1468	1.1468	0.0002	0.9998	No
LUO UNET vs RGBS BCE70 DICE30	Mean Error	0.1846	-0.6836	0.8183	0.4504	No
LUO UNET vs RGBS BCE70 DICE30	Mean Absolute Error	1.0569	1.6686	-0.921	0.3993	No
LUO UNET vs RGBS BCE70 DICE30	RMSE	1.2464	1.8315	-0.945	0.388	No
LUO UNET vs RGBS BCE70 DICE30	Std. Residuals	1.1503	1.2497	-0.3938	0.7201	No

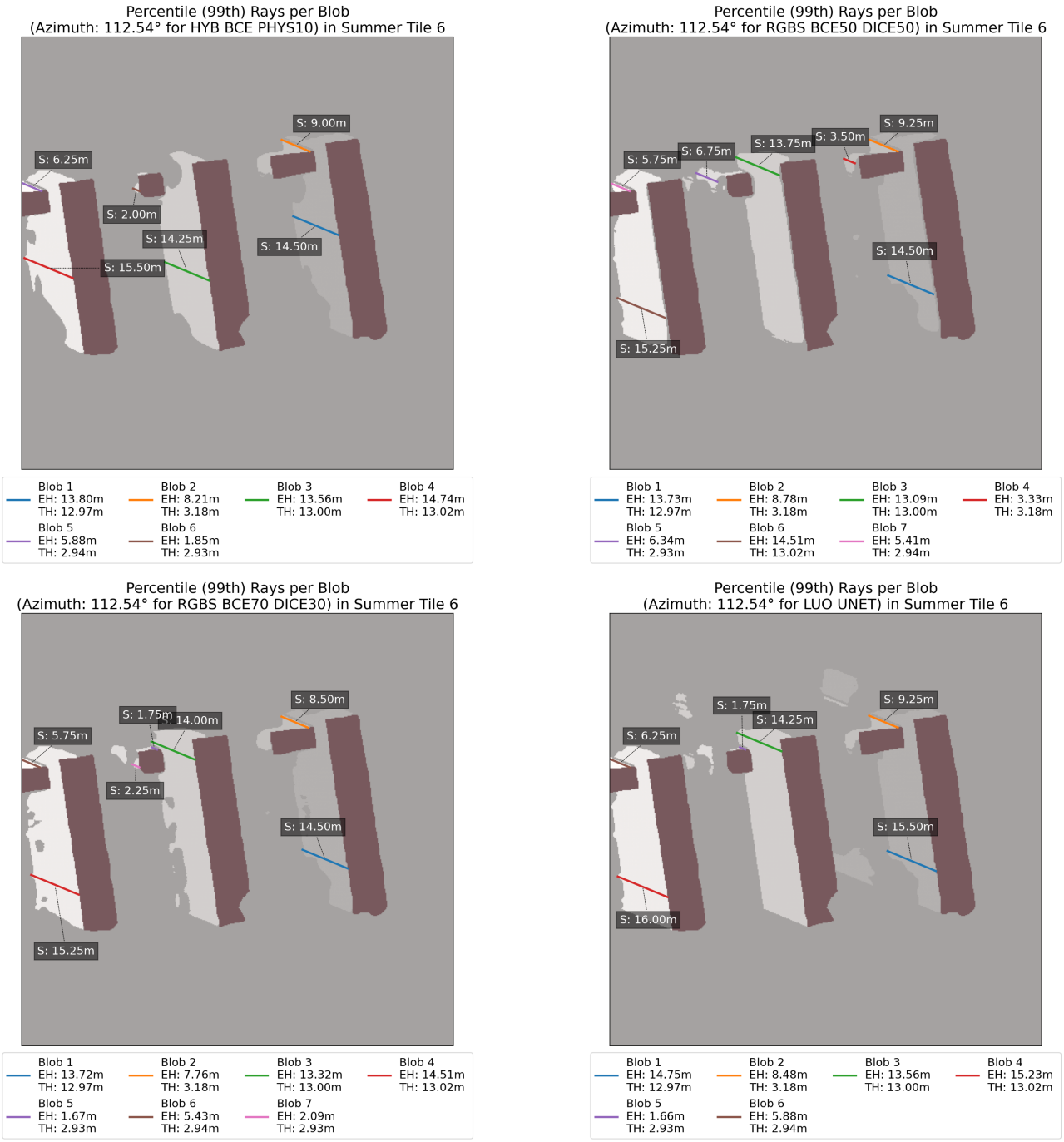

Figure 20: Overview of the lines fitted for height estimation on the different models of PHYSHADE and LUO UNET in Summer Tile 6. It can be seen here that generally speaking, all models performed comparably. As can be seen with blob 2, in some cases the height of a building can be vastly overestimated if the shadow of a neighbouring building falls over it. Attributing the correct parts of a shadow beyond what was done in the height-estimation pipeline already would be required to increase the robustness in such cases. Note how the shadow blobs after filtering for LUO UNET look very similar to the others, although with some additional blobs that were ignored by the height estimation algorithm; the selection process is robust enough to not be heavily influenced by these cases. LUO UNET performed the worst here in terms of inference, but is able to salvage this with a relatively good height estimation on blob 2.

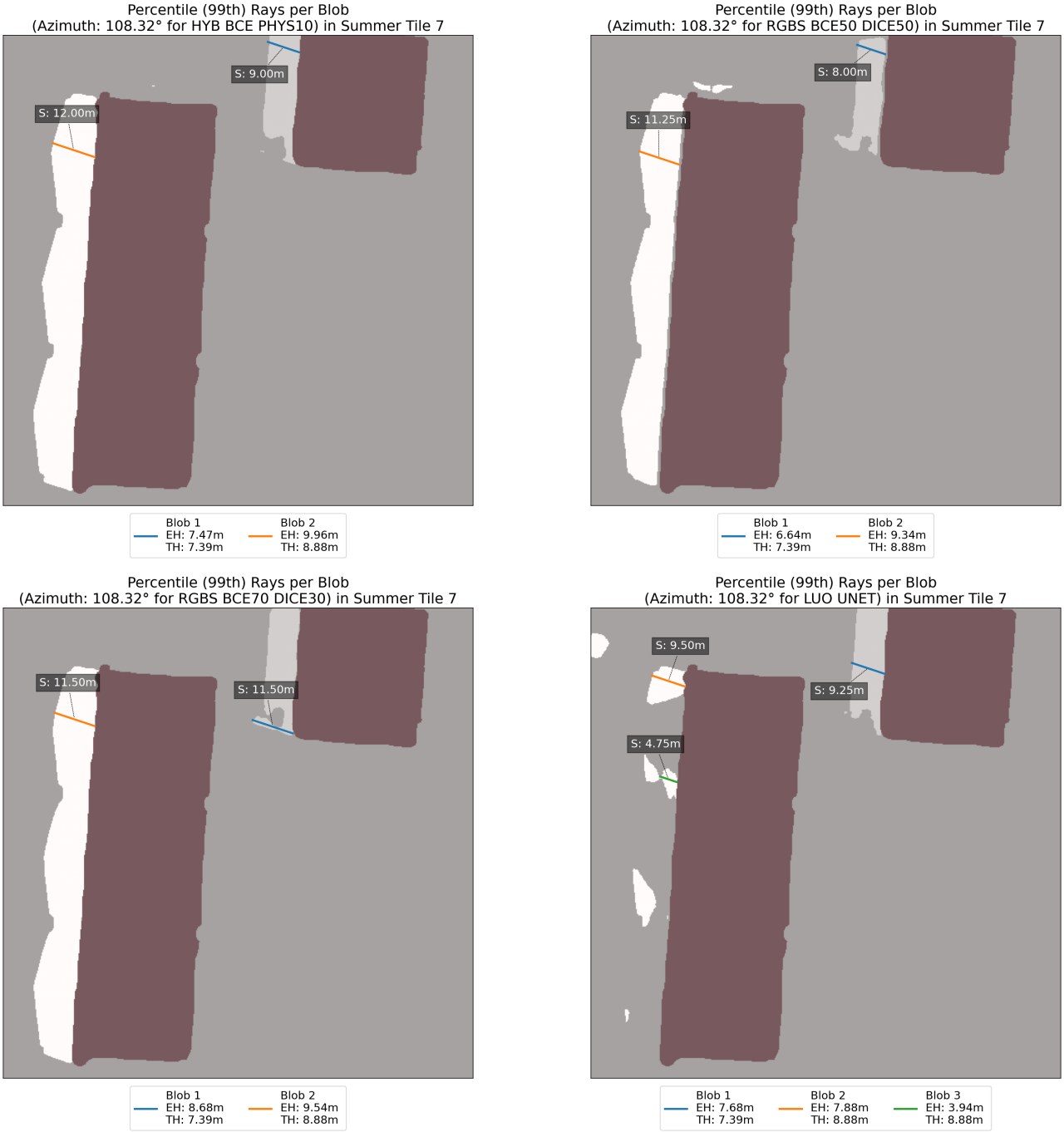

Figure 21: Overview of the lines fitted for height estimation on the different models of PHYSHADE in Summer Tile 7. Considering the simplicity of this image in terms of building number and geometry, the performance over all models was very good. HYB BCE PHYS10 performed the best for Blob 1, whereas RGBS BCE50 DICE50 did the best for Blob 2. RGBS BCE70 DICE30 saw a slight overestimation for Blob 1, leading to a higher-than-truth estimation. Overall though, the differences in most cases are only a few pixels off. The difference in True Height in Table 18 and Table 19 can be explained due to the building in the top right not being counted by HYB BCE PHYS10 and RGBS BCE70 DICE30 compared to RGBS BCE50 DICE50, since their shadow blobs were truncated as they were connected to the edge of the raster and thus filtered.

Figure 22: Overview of the lines fitted for height estimation on the different models of PHYSHADE in Summer Tile 8. Not much can be said about the differences between the rays cast here; Most of them are calculate into relatively good representations of building height, with the buildings having multiple blobs associated with them needing the extra step of filtering for the ”correct” blob by looking at what blob has the largest surface area. In all cases here, this corresponded with the building with the most correct line. Once again the LUO UNET model has the most noise from inference, but the blob-to-building processing is robust enough to guard against this.

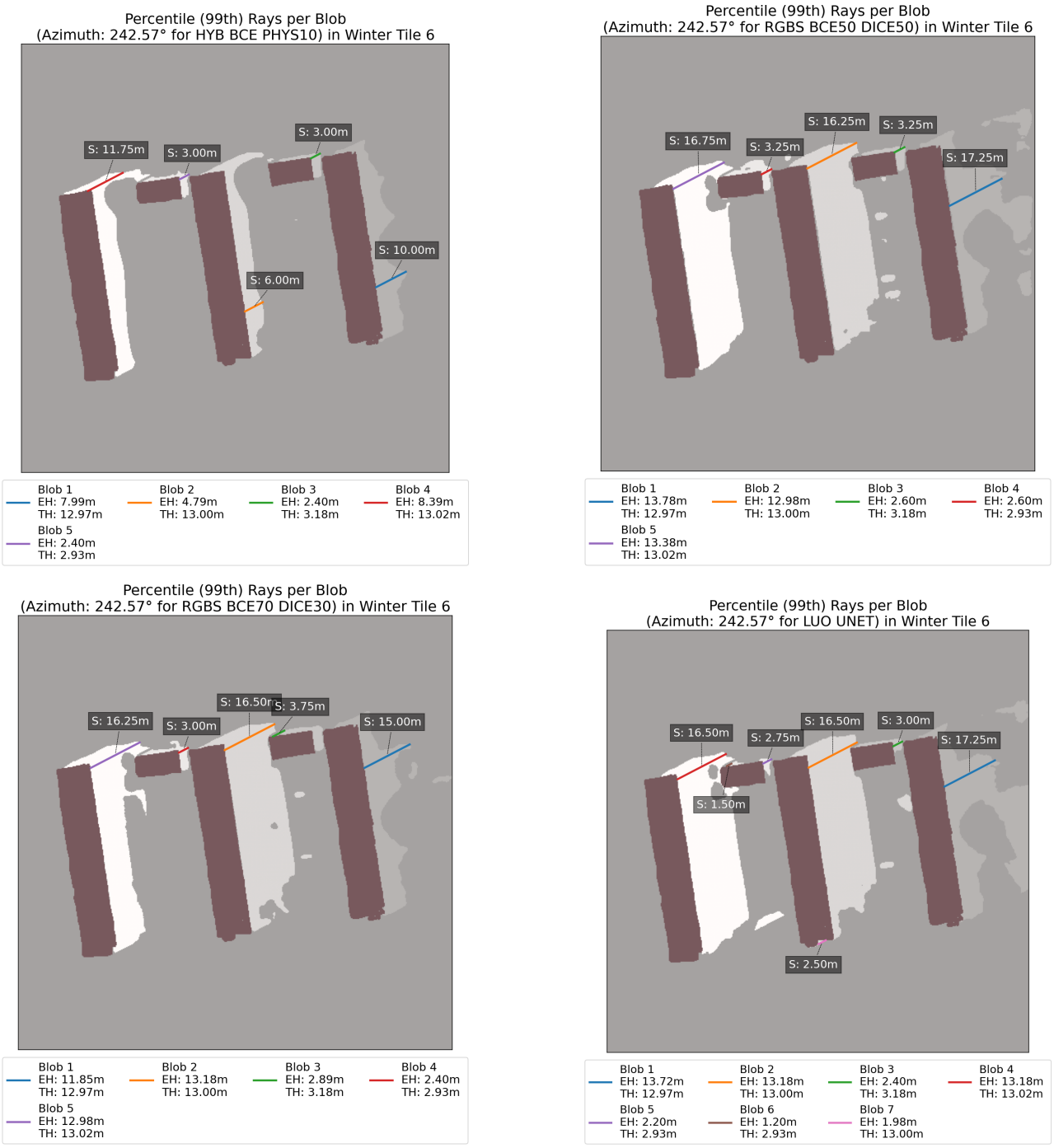

Figure 23: Overview of the lines fitted for height estimation on the different models of PHYSHADE and LUO UNET in Winter Tile 6. While RGBS BCE50 DICE50 and RGBS BCE30 DICE70 were able to score better on the height estimation due to their higher recalls, HYB BCE PHYS 10 suffered a lot on this imagery with the middle building under blob 2 underestimating the building height by around ≈ 9.2 meters. It should be mentioned here that the overeagerness for RGBS BCE50 DICE50 to predict led it close to overestimating more than it did; a large part of the shadow associated with blob 1 was a false positive, which could have negatively influenced the quality of height estimation if it were to fall in one continouous line from the building footprint. The inference from LUO UNET looks very similar to RGBS BCE50 DICE50 here, and as such they share a lot of similarities, although LUO UNET found two more blobs that were eligible for getting rays cast.

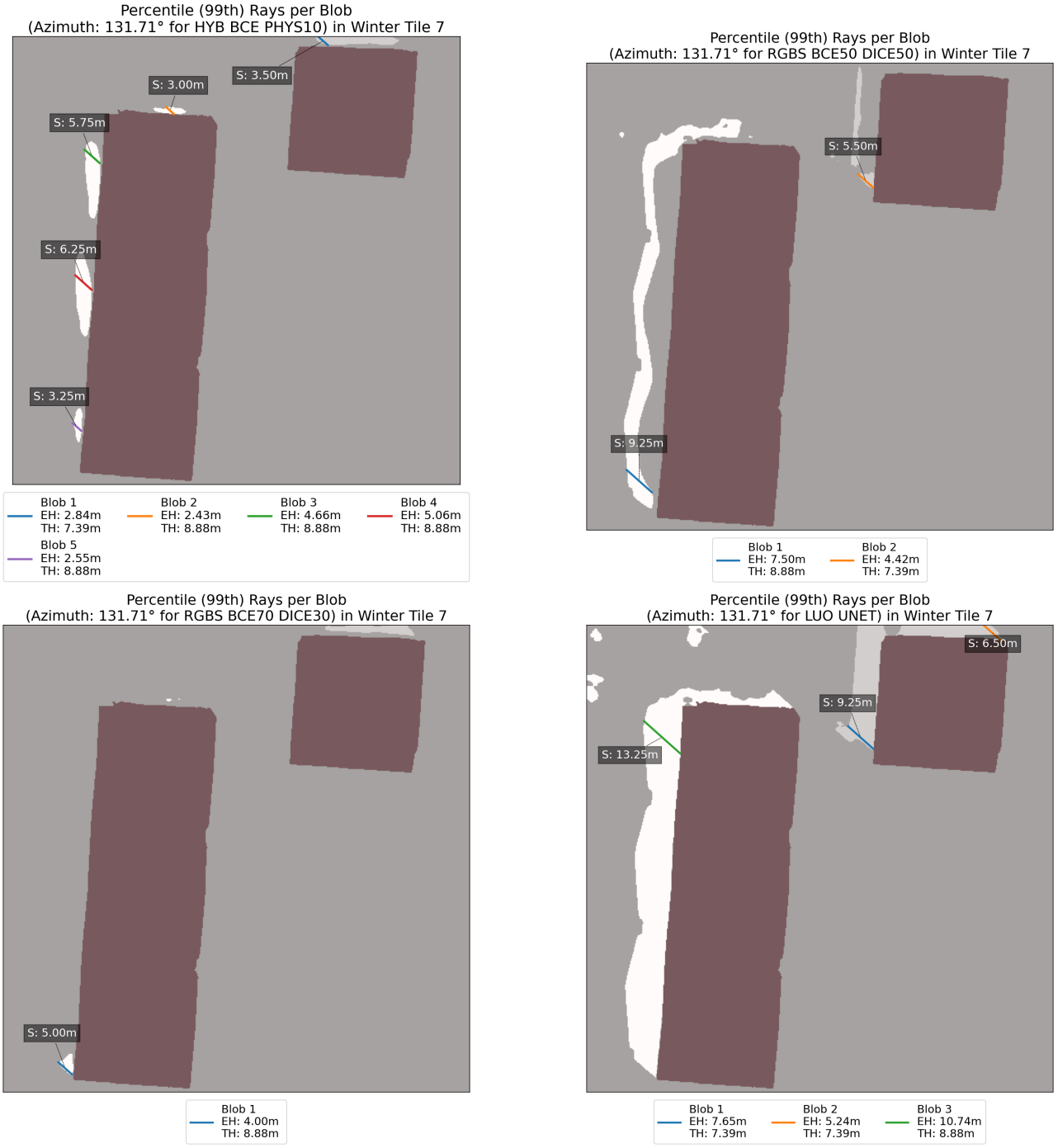

Figure 24: Overview of the lines fitted for height estimation on the different models of PHYSHADE and LUO UNET in Winter Tile 7. This image had the largest errors of the entire out-of-fold dataset for the PHYSHADE, as the inference over this image was weak. Here, it is the high recall of RGBS BCE50 DICE50 that help it still come to a relatively good estimate for building height with the error for blob 1 being less than two meters. In stark contrast, LUO UNET performed the best here as opposed to the PHYSHADE models due to inference quality being much better; although its precision is not great and is overestimating the blobs to the top left, it was able to nearly perfectly capture the building shadows leading to relatively accurate height estimations. As with Summer Tile 7, the difference in true height for this image when comparing to Table 17 and Table 18 in the statistics can be explained due to the building in the top right being filtered out in two of the images.

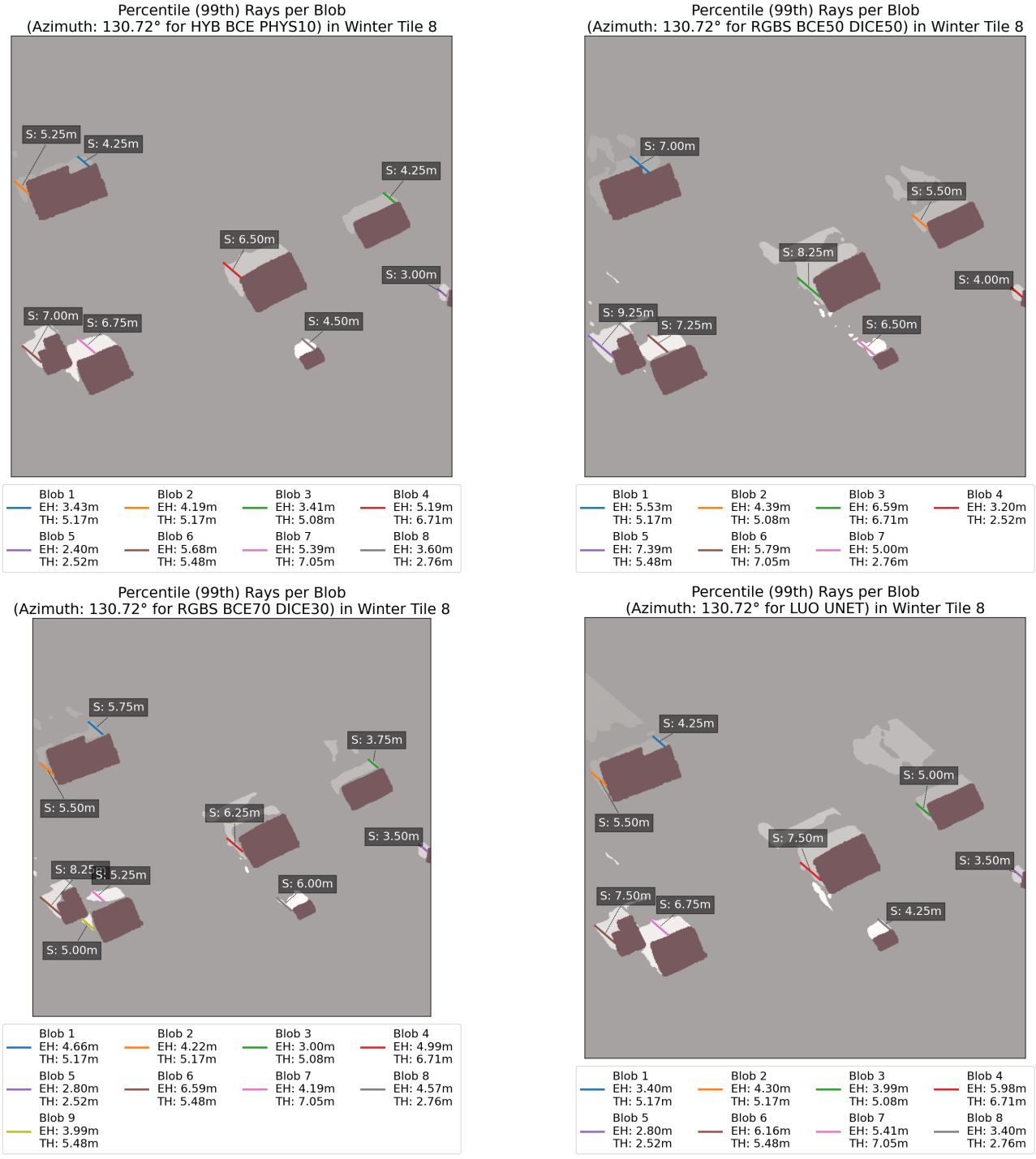

Figure 25: Overview of the lines fitted for height estimation on the different models of PHYSHADE and LUO UNET in Winter Tile 8. For this image, HYB BCE PHYS10 was able to perform best overall when compared to RGBS BCE50 DICE50, whose inference output was noisier in turn leading to more outliers.

6 Discussion and Conclusions

In this thesis, pseudo-shadows derived from building footprints were employed as geometric priors to help steer a U-Net based model into recognizing building shadows for the purposes of shadow inference and subsequent height estimation. Through this, it was hoped that the quality of segmentation aided by the pseudo-shadows would subsequently help increase the robustness of height estimation.

6.1 Discussion of Results

Q1: What is the baseline performance of an RGB U-Net trained on the Luo et al. (2020) dataset when evaluated on Dutch aerial imagery compared to the original dataset?
Before the different configurations of PHYSHADE were trained, its predecessor U-Net model used for transfer learning trained on the dataset by Luo et al. (2020) was evaluated on two intra-domain images where all shadows as opposed to only building shadows were annotated as ground truth. Since the original dataset was created from imagery of different countries and urban morphologies, this step was taken to assess the difference in performance . It was found that the cross-domain application caused deficiencies in performance; whereas Luo et al. (2020) reported Dice scores of ≈ 0.88, performance for the two fully annotated images of the intra-domain dataset dropped to ≈ 0.57. Although the sample size here is small and would require a larger fully annotated Dutch shadow dataset before this statistic can be considered reliable, it is indicative of cross-domain performance loss meaning that the model would need to be fine-tuned before it can be used to reliably detect shadows in the Dutch domain inhabited by the PHYSHADE dataset.

Q2/3: How does adding the pre-calculated shadow mask channel derived from building footprints affect segmentation accuracy across different urban morphologies, seasons, and solar geometries? What are the most effective formulations and weighting schemes to effectively balance appearance learning with the geometric prior?
After the application of transfer learning to turn the Luo et al. (2020) into PHYSHADE, it was found that the injection of geometric priors in the form of pseudo-shadows allowed for PHYSHADE to perform better when segmenting building shadows, leading to an increase in Dice score of ≈ 0.32 when comparing the models without access to fourth channel pseudo shadows (those found in Experimental Subset A) to the models with access to the pseudo-shadows (Experimental Subset B). As such, the model architecture after training is able to interpret the pseudo-shadows and learns to respect them by not inferring shadows that fall outside them, leading to much higher precision statistics.

Besides the addition of pseudo-shadows as a fourth channel to model input, the inclusion of pseudo-shadows through loss functions was explored as well. Here, it was found that between the baseline models configurations of Subset A and the models of Subset C, no significant difference could be found in Dice score. This result was unsurprising, as while the physics guided loss is pushing the model to weigh predictions surrounding the pseudo-shadows more heavily and in this sense should guide the model to focus there, it was unable to learn the context necessary for understanding that the ground truth shadows are specifically building shadows.

During the comparison of Subset B and D, the addition of physics-guided loss was examined for the PHYSHADE models that did have access to the pseudo-shadows as fourth channels in their input. With the added context of what can and cannot be considered a building shadow through the pseudo-shadow, it was hoped that the physics-guided loss would either help improve segmentation quality through increases in Dice score, or that it would help the models converge faster by focussing the models' attention on the relevant pixels within the pseudo-shadows. While the addition of physics-guided loss

did not lead to significant increases in Dice score (in fact, out of the four significantly tested models, three saw decreases in Dice score), one model saw an increase of 0.0068 in their mean Dice score when compared to its Subset B counterpart, making it the highest scoring model of all experimental configurations with a mean Dice score of 0.8523, precision of 0.8498 and recall of 0.8320. As for the difference in time needed for convergence when comparing the models from subsets B and D, HYB DICE HYS50 saw a significant decrease in epochs needed to converge, taking on average 7.4 epochs less to reach training completion compared to RGBS BCE50 DICE50. However, in the majority of cases the physics-guided loss term did not lead to significant decreases in training epochs required and in some cases even caused it to go up (albeit non-significantly). As for the weighting used in the configurations of subset B and D, it seems that in the RGBS models without physics-guided loss the performance is roughly similar for higher weightings of BCE loss vs Dice loss making them equally fit for the application of segmenting building shadows. For the hybrid models containing the physics-guided loss, it generally is true that the higher the weighting for the loss, the more it will negatively impact performance. Since the added physics-guided loss increased the performance for HYB BCE PHYS10 by only 0.0068, it may be a valid decision to use the simpler RGBS models over the hybrid models. That said, for the purposes of ablation and limitations in computational time available for experimentation, the weighting for the HYB models of Subset D was kept constantly at BCE 50 and DICE 50, which means that only part of the possible configurations for the hybrid models were tested. As such, there may be hyper-parameter configurations for the models that could result in higher performance than what was trained within the purviews of this thesis.

The results described above were derived through five-fold cross validation, where the dataset is divided into five parts (or folds). The dataset is then iterated over five times, each time using four of the folds for training the models and one fold for validation, such that each image gets trained on four times and validated on once. This way, the error statistics produced over each of the five folds can be averaged to get a more representative idea of the performance using the full dataset, since some (especially randomized) selections of validation imagery may not provide an adequate representation of the domain they depict.

To get a true idea of what the model performance might look like if it were trained on the full size of the dataset without splitting, six additional images were annotated outside of the domain. The PHYSHADE configurations selected were the top two performing models (HYB BCE PHYS10, RGBS BCE70 DICE30) alongside a model useful for ablating physics-guided loss terms (RGBS BCE50 DICE50). These models were then applied to the validation set to see how their performance would fare in a scenario not seen before. Here, it was found that despite the fact that the original PHYSHADE dataset contained paired summer and winter imagery, the performance over the winter tiles in the out-of-fold dataset was a lot worse compared to the summer tiles. This error was mostly found in Winter Tile 7, where all three models ran into the same issue of falsely missing out on labelling shadowed regions despite that these shadows are perhaps the most clear to the human eye of all imagery. The exact reason for this is unclear, but one thing immediately noticeable is that the increased contrast and yellowish hue is making it stand out from the other imagery. Either way, it is indicative of the fact that the augmentation strategy employed is not adequate for ensuring performance in these cases, and that PHYSHADE's performance is likely capped by the small size of the dataset leading to lower generalisability and robustness. Despite the winter images lowering the overall scores, some good performances were seen for the summer tiles, with the three final models of PHYSHADE obtaining average Dice scores around ≈ 0.89. For the sake of comparison, the original U-Net model trained on the Luo et al. (2020) dataset was applied to the images as well, and scored low on Dice due to its high recall but poor Dice scoring.

Q4: How effective are the inferred building shadows for estimating building height using the raster-based raycasting algorithm?
Before the application of the height estimation algorithm on the inference produced by the final PHYSHADE models and the LUO UNET baseline model, it was decided to test the algorithm in isolation with synthetic data to see if it is conducive to errors related to raster-based operations. Here, it was found that the overall error statistics inherent to the algorithm are acceptably small, where the RMSE for ray length percentiles set to 90 and above are within 0.2113 meters. Since the 99th percentile of rays performed best as a parameter, it was selected to be used for the subsequent height estimation using real data. When applied to the inference produced by employing the final PHYSHADE models and LUO UNET on the out-of-fold dataset, it becomes clear that while the mean absolute error and RMSE for the different models are around 1.46m and 2.07m respectively, they are outperformed by the LUO UNET baseline model which was able to score a 1.08m mean absolute error and a 1.53m RMSE. The reason for this is two-fold: shadow segmentation models that score higher on recall are able to get retrieve more of the shadow back, meaning that the subsequent height estimation based on this is more likely to be accurate. This same trend where higher recall models scored better on height estimation was also seen in the PHYSHADE models, where RGBS BCE50 DICE50 as a model suffered from comparatively lower precision but higher recall. leading it to perform similarly to LUO UNET on height estimation. Secondly, although LUO UNET suffered on precision which could lead to overestimation, some of the effects of this were mitigated due to the preprocessing steps used to filter out invalid blobs from height estimation. However, despite the fact that the LUO UNET Model scored better on height estimation than the rest, the paired t-tests comparing the models came back as insignificant, meaning that no real difference in the means was found between them.

6.1.1 Study Limitations

While the test statistics for height estimation in-vitro show that there is some potential for the height estimation to give good ballpark estimates of building heights, a larger-scale evaluation of the PHYSHADE models and method should be performed to see whether the performance works well in other scenarios. The out-of-fold dataset for shadow inference and height estimation is small, and represent a best-case scenario for the models; the buildings and subsequent shadows are mostly clear of obstructions, and contain little building-over-building obstructions where the shadow of one building completely envelops the other (although one such case can be seen in Figure 14 where the height of blob 2 was overestimated due the shadow of the building on the right).

In addition, although it can be seen that the addition of geometric priors in the form of pseudo-shadows is able to increase the performance of building shadow segmentations such as seen in the ablation of Subsets A and B, there are limitations to the methodology posed. For example, PHYSHADE's interpretation of the interplay between building geometry and solar positioning is reliant on the pseudo-shadow mapping. Since in its current form a large l_max is used to generate the extent of the pseudo-shadows based upon the 95th percentile of all buildings heights, a lot of granularity is lost due to the discrepancy in height between the average building and this number. While this has not been too problematic in the contexts of the current dataset as buildings are spread out, segmentation quality in more urban context could potentially degrade due to the increased density of buildings smearing out into pseudo-shadows that predict shadows everywhere. Although this may be mitigated in part through the tuning of l_max where an estimate is made per image on the building height first, it is unsure what effect different values for l_min and l_max would have as unfortunately, these were not ablated for in this study due to time constraints.

Finally, while the results show that the concept of PHYSHADE has value in the segmentation of

building shadows, an expansion in both the training and validation sets would be required to gain a better idea of the generalisability of the model and subsequent height estimations stemming from it. With this, the main goal should be to expand the breadth of domains (i.e. rural-urban, building types, varied times of day) further than the dataset presented in this thesis.

6.2 Conclusion

This thesis introduced the PHYSHADE models; a set of physics-guided U-Net based CNNs that combines RGB aerial imagery together with pseudo-shadows encoding shadow probabilities towards the improvement of building shadow segmentation and height estimation. Inspired by Physics-Informed Neural Networks, an algorithm was introduced that creates pseudo-shadow mappings based upon building footprints and the solar position and altitude of the sun, thereby giving PHYSHADE the ability to better understand the difference between building shadows and other shadows. By using data augmentation, k-fold cross-validation combined with transfer learning to get past the limitations of a small dataset, 130 different models were trained for the purposes of ablating PHYSHADE. Here, it was shown that the addition of pseudo-shadows as an extra channel to the RGB input provided a lot of value, leading to Dice score increases of up to 0.32 and average Dice scores of up to 0.847. In addition, the usage of pseudo-shadows to regularize training loss were explored as well, where although one model was found to be the best performing over all configurations, the physics-guided loss functions as posed in this thesis rarely provided added benefit whilst actively harming model performance in some cases.

Finally, height estimation was performed using a raycasting-based algorithm on the inference of four different models; the top two best scoring, one model for ablation of the first and the pre-transfer learning model as a control to assess the added benefit of the PHYSHADE models towards height estimation. Although it was found that the PHYSHADE models did not perform better as input for the height estimation tasks due to their relatively lower recall, the concept of height estimation through building shadow segmentation proved to be a viable alternative to more costly approaches such as LiDAR by scoring RMSE values around two meters and MAE values around 1.46 meters, provided that the use case is not for purposes where high accuracy is mission-critical.

There are limitations to the study however; the procedures used for the subdivision and assignment of shadows to buildings suffers from ambiguity in some cases, where building-over-building overlap may cause certain shadows to be attributed to the wrong building leading to erroneous height calculations. Such problems would be exacerbated in urban contexts where building density is higher. In addition, the employed dataset for this study was small, where the combination of the in-fold and out-of-fold dataset only yields 41 images which limits the claims that can be made about the generalisability of PHYSHADE and the subsequent statistics produced by inference and height estimation.

6.2.1 Future Work

Future work could focus on filling the gaps left by this research; while a proof-of-concept for physics-guided shadow segmentation through geometric priors was provided and applied in a small domain, an expansion in the dataset could reveal the viability of employing this model architecture on a larger scale. Advancements in the field of building segmentation; that is, the recognition of buildings in aerial photography would go hand-in-hand with PHYSHADE, potentially leading to a pipeline where building shadows can be segmented accurately at scale requiring minimal manual intervention. There are also architectural changes that could potentially be explored that forego usage of a raycasting algorithm and implement the height estimation as an additional objective directly into a model similar

to PHYSHADE. For example, X. Li et al. (2020) use multi-task learning to segment and do height estimations on buildings with good results. The inclusion of a building-shadow segmentation branch could potentially help inform the height-estimation branch of a CNN, which could potentially lead to performance gains. Another possible avenue of research would not only be the additional classification and linkage of building shadows to buildings through a CNN as opposed to just the segmentation, as this would potentially bypass the issues caused by the ambiguity of shadows within this research.

References

Adeline, K. R. M., Chen, M., Briottet, X., Pang, S. K., & Paparoditis, N. (2013). Shadow detection in very high spatial resolution aerial images: A comparative study. *ISPRS Journal of Photogrammetry and Remote Sensing*, *80*, 21–38. https://doi.org/10.1016/j.isprsjprs.2013.02.003

AHN. (2020, February 18). *Kwaliteitsbeschrijving* [AHN] [Publisher: AHN]. Retrieved December 13, 2024, from https://www.ahn.nl/kwaliteitsbeschrijving

Apra, I., Bachert, C., Cáceres Tocora, C., Tufan, Ö., Veselý, O., & Verbree, E. (2021). Inferring roof semantics for more accurate solar potential assessment. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLVI-4/W4-2021*, 33–37. https://doi.org/10.5194/isprs-archives-XLVI-4-W4-2021-33-2021

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Chen, L., Fang, B., Zhao, L., Zang, Y., Liu, W., Chen, Y., Wang, C., & Li, J. (2022). DeepUrbanDownscale: A physics informed deep learning framework for high-resolution urban surface temperature estimation via 3d point clouds. *International Journal of Applied Earth Observation and Geoinformation*, *106*, 102650. https://doi.org/10.1016/j.jag.2021.102650

Ciresan, D., Giusti, A., Gambardella, L., & Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in Neural Information Processing Systems*, *25*. Retrieved May 3, 2025, from https://papers.nips.cc/paper_files/paper/2012/hash/459a4ddcb586f24efd9395aa7662bc7c-Abstract.html

Craglia, M., & Shanley, L. (2015). Data democracy – increased supply of geospatial information and expanded participatory processes in the production of data [Publisher: Taylor & Francis eprint: https://doi.org/10.1080/17538947.2015.1008214]. *International Journal of Digital Earth*, *8*(9), 679–693. https://doi.org/10.1080/17538947.2015.1008214

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, *26*(3), 297–302. https://doi.org/10.2307/1932409

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

Granlund, G. H. (1978). In search of a general picture processing operator. *Computer Graphics and Image Processing*, *8*(2), 155–173. https://doi.org/10.1016/0146-664X(78)90047-3

Hirschmuller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(2), 328–341. https://doi.org/10.1109/TPAMI.2007.1166

Jiao, L., Huo, L., Hu, C., & Tang, P. (2020). Refined UNet: UNet-based refinement network for cloud and shadow precise segmentation [Number: 12 Publisher: Multidisciplinary Digital Publishing Institute]. *Remote Sensing*, *12*(12), 2001. https://doi.org/10.3390/rs12122001

Jin, X., Cai, S., Li, H., & Karniadakis, G. E. (2021). NSFnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations. *Journal of Computational Physics*, *426*, 109951. https://doi.org/10.1016/j.jcp.2020.109951

Kadaster. (2023, January 20). *4.2 Het genereren van de puntenwolken* [3D-Basisvoorziening]. Retrieved June 9, 2025, from https://3d.kadaster.nl/praktijkhandleiding/productbeschrijving/puntenwolkengenereren

Kang, J., Wang, Z., Zhu, R., Sun, X., Fernandez-Beltran, R., & Plaza, A. (2021). PiCoCo: Pixelwise contrast and consistency learning for semisupervised building footprint segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 10548–10559. https://doi.org/10.1109/JSTARS.2021.3119286

Krähenbühl, P., & Koltun, V. (2011). Efficient inference in fully connected CRFs with gaussian edge potentials. *Advances in Neural Information Processing Systems*, *24*. Retrieved February 16,

2025, from https://proceedings.neurips.cc/paper_files/paper/2011/hash/beda24c1e1b46055dff2c39c98fd6fc1-Abstract.html

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition [Conference Name: Proceedings of the IEEE]. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, T., & Kim, T. (2010). GENERATION OF 3d BUILDING MODELS FROM COMMERCIAL IMAGE DATABASE THROUGH SAHDOW ANALYSIS. *San Diego*.

Lee, T., & Kim, T. (2013). Automatic building height extraction by volumetric shadow analysis of monoscopic imagery. *International Journal of Remote Sensing*, *34*(16), 5834–5850. https://doi.org/10.1080/01431161.2013.796434

León-Sánchez, C., Giannelli, D., Agugiaro, G., & Stoter, J. (2021). TESTING THE NEW 3d BAG DATASET FOR ENERGY DEMAND ESTIMATION OF RESIDENTIAL BUILDINGS [Conference Name: ISPRS TC IV¡br¿6th International Conference on Smart Data and Smart Cities - 15–17 September 2021, Stuttgart, Germany Publisher: Copernicus GmbH]. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLVI-4-W1-2021*, 69–76. https://doi.org/10.5194/isprs-archives-XLVI-4-W1-2021-69-2021

Li, W., He, C., Fang, J., Zheng, J., Fu, H., & Yu, L. (2019). Semantic segmentation-based building footprint extraction using very high-resolution satellite images and multi-source GIS data [Number: 4 Publisher: Multidisciplinary Digital Publishing Institute]. *Remote Sensing*, *11*(4), 403. https://doi.org/10.3390/rs11040403

Li, X., Wang, L., & Fang, Y. (2020, September 22). Geometry-aware segmentation of remote sensing images via implicit height estimation. https://doi.org/10.48550/arXiv.2006.05848

Liasis, G., & Stavrou, S. (2016). Satellite images analysis for shadow detection and building height estimation. *ISPRS Journal of Photogrammetry and Remote Sensing*, *119*, 437–450. https://doi.org/10.1016/j.isprsjprs.2016.07.006

Liu, L., Li, W., Shi, Z., & Zou, Z. (2022). Physics-informed hyperspectral remote sensing image synthesis with deep conditional generative adversarial networks. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–15. https://doi.org/10.1109/TGRS.2022.3173532

Luo, S., Li, H., & Shen, H. (2020). Deeply supervised convolutional neural network for shadow detection based on a novel aerial shadow imagery dataset. *ISPRS Journal of Photogrammetry and Remote Sensing*, *167*, 443–457. https://doi.org/10.1016/j.isprsjprs.2020.07.016

Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*.

Masquil, E., Marí, R., Ehret, T., Meinhardt-Llopis, E., Musé, P., & Facciolo, G. (2025, April 21). S-EO: A large-scale dataset for geometry-aware shadow detection in remote sensing applications. https://doi.org/10.48550/arXiv.2504.06920

Nagao, M., Matsuyama, T., & Ikeda, Y. (1979). Region extraction and shape analysis in aerial photographs. *Computer Graphics and Image Processing*, *10*(3), 195–223. https://doi.org/10.1016/0146-664X(79)90001-7

Nan, L. (2023, February 15). *Introduction to machine learning**. Retrieved January 22, 2025, from https://3d.bk.tudelft.nl/courses/geo5017/handouts/01_notes_Introduction.pdf

Nederland, B. (2023, March 6). *Home* [Beeldmateriaal Nederland] [Publisher: Beeldmateriaal Nederland]. Retrieved May 7, 2025, from https://www.beeldmateriaal.nl/

Ng, E., Yuan, C., Chen, L., Ren, C., & Fung, J. C. H. (2011). Improving the wind environment in high-density cities by understanding urban morphology and surface roughness: A study in hong kong. *Landscape and Urban Planning*, *101*(1), 59–74. https://doi.org/10.1016/j.landurbplan.2011.01.004

Otsu, N. (1979). A threshold selection method from gray-level histograms [Conference Name: IEEE Transactions on Systems, Man, and Cybernetics]. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

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 [ISSN: 0099-1112 Issue: 3 Pages: 165–170 Publication Title: Photogrammetric Engineering and Remote Sensing Volume: 88]. https://doi.org/10.14358/PERS.21-00032R2

Phung, V. H., & Rhee, E. J. (2018). A deep learning approach for classification of cloud image patches on small datasets [Publisher: Array], *16*(3), 173–178. https://doi.org/10.6109/jicce.2018.16.3.173

Pratt, L. Y., Mostow, J., & Kamm, C. A. (1991). Direct transfer of learned information among neural networks. *Proceedings of the ninth National conference on Artificial intelligence - Volume 2*, 584–589.

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017, November 28). Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. https://doi.org/10.48550/arXiv.1711.10561

Richter, R., & Müller, A. (2005). De-shadowing of satellite/airborne imagery [Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01431160500114664]. *International Journal of Remote Sensing*, *26*(15), 3137–3148. https://doi.org/10.1080/01431160500114664

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation [Series Title: Lecture Notes in Computer Science]. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical image computing and computer-assisted intervention – MICCAI 2015* (pp. 234–241, Vol. 9351). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28

Tomasi, C., & Kanade, T. (1992). Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, *9*(2), 137–154. https://doi.org/10.1007/BF00129684

Ullman, S., & Brenner, S. (1979). The interpretation of structure from motion [Publisher: Royal Society]. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, *203*(1153), 405–426. https://doi.org/10.1098/rspb.1979.0006

Yamazaki, F., Liu, W., & Takasaki, M. (2009). Characteristics of shadow and removal of its effects for remote sensing imagery [ISSN: 2153-7003]. *2009 IEEE International Geoscience and Remote Sensing Symposium*, *4*, IV–426–IV–429. https://doi.org/10.1109/IGARSS.2009.5417404

A Overview of Pipeline: Preprocessing

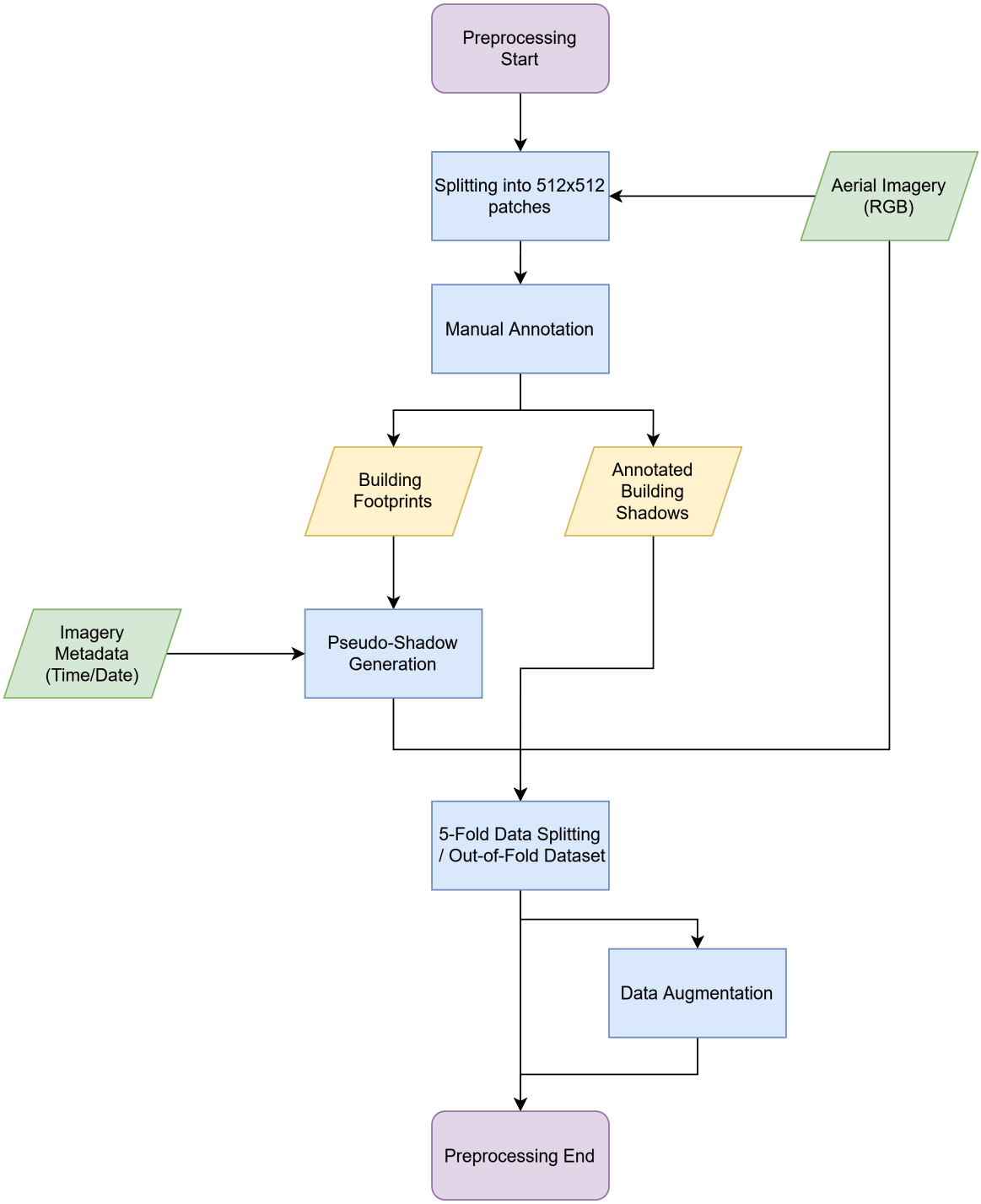

Figure A.1: An overview of the first part of the PHYSHADE pipeline, where the initial RGB imagery is taken and pseudo-shadows are generated using the time/date metadata. Afterwards, the original 35 images are split into 5 folds and applied data augmentation to, whereas the 6 Out-of-Fold images are kept as-is.

B Overview of Pipeline: Training

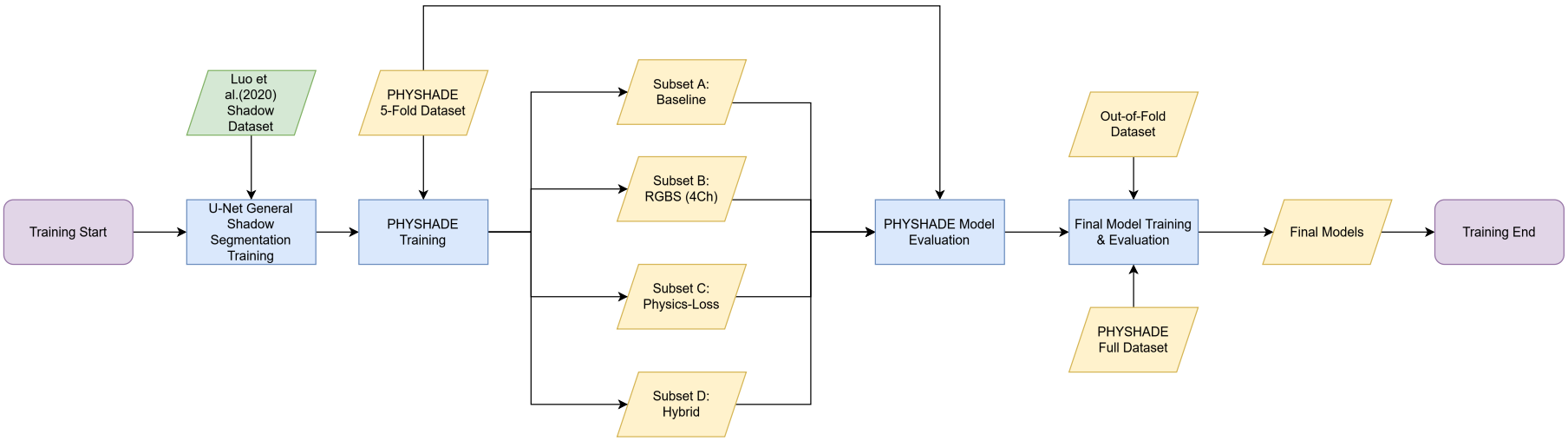

Figure B.1: An overview of the second part of the PHYSHADE pipeline where the baseline Luo U-Net model is trained, followed by the training of the various subsets of PHYSHADE models. For the initial round of evaluation of the PHYSHADE models, the cross-fold statistics are used, after which the best performing models are trained on the full dataset and validated on the out-of-fold dataset.

C Overview of Pipeline: Height Estimation

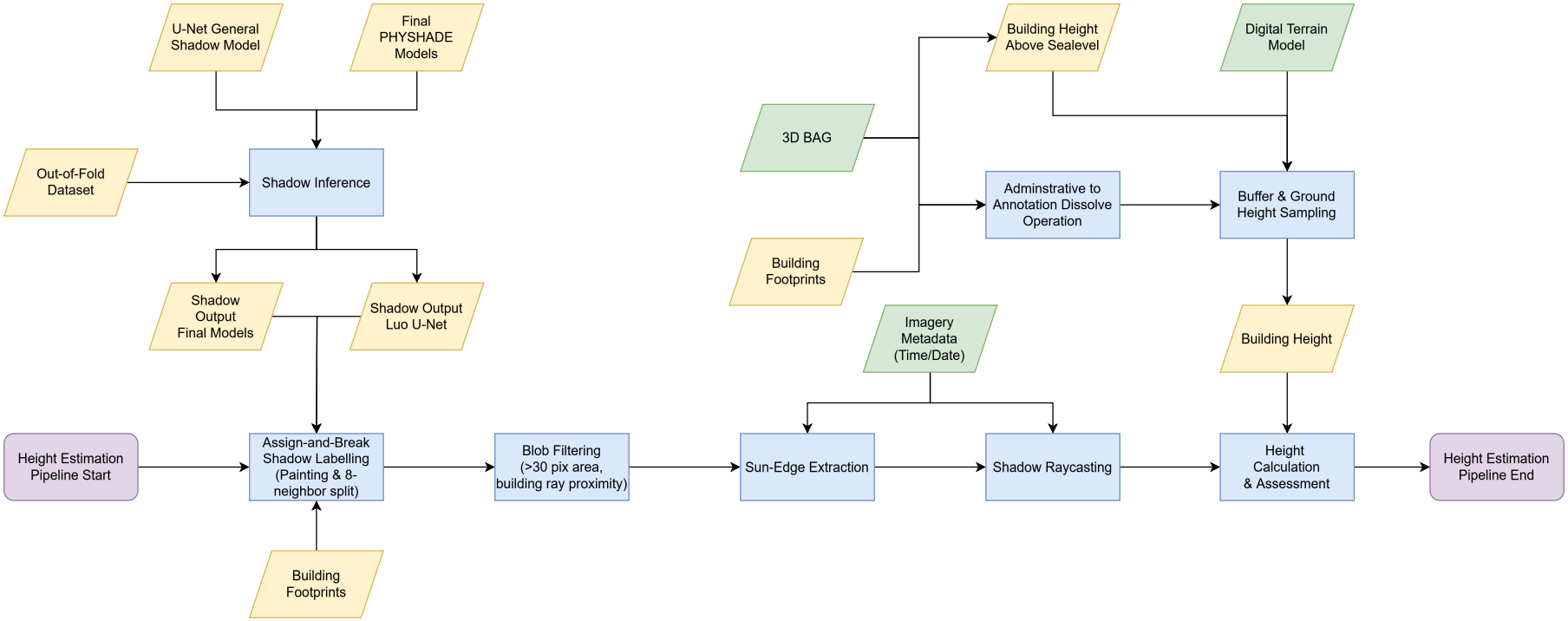

Figure C.1: An overview of the final part of the PHYSHADE pipeline where the height estimations are being performed. Firstly, using the inference from the final models and the Luo et al. (2020) general shadow segmentation model trained as a transfer learning base, shadows are inferred on the out-of-fold dataset. Afterwards, the inferred shadows are assigned to buildings and filtered. During sun-edge extraction, the pixels of the shadow blobs facing the sun are selected as starting rays for the shadow length raycasting, after which they are compared to the true building heights to gain the error statistics.

D Dataset Collection Procedure

In order to create a sample of equal sized 512x512 raster images for input in CNNs, the following steps were taken:

Figure D.1: During step 1, a bounding polygon is created over one of the areas of study

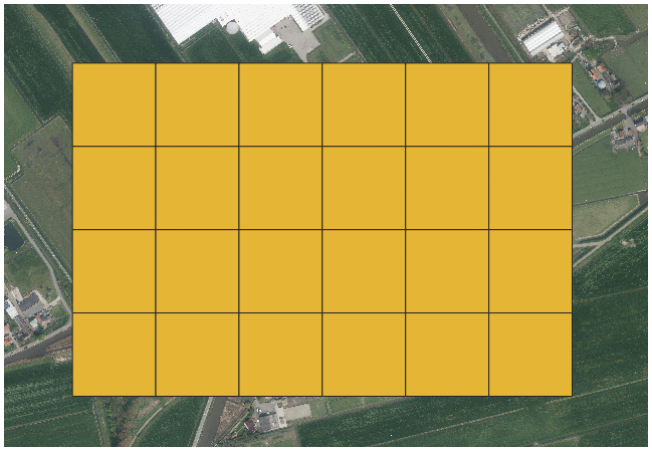

Figure D.2: During step 2, using the ”Create Grid” option from the QGIS Toolbox, a 512x512 grid is overlaid. To do so, spacing was set to 512x0.25 (25cm resolution) to get 128x128 meter grid cells. From here, the underlying raster can be clipped by iterating over each polygon within the grid layer individually while outputting.

E Mosaic of PHYSHADE Dataset

Figure E.1: A mosaic of the full dataset used for the training of PHYSHADE, consisting of 15 summer/winter pairs and 5 singular images.

F Annotation Guidelines

Aim for precise, pixel-level contours around buildings and shadow regions. Zoom in and carefully trace object outlines—we want to capture as much detail as possible without cutting off parts of the features. Consistent boundary handling is crucial: always delineate the true edges of buildings (including any visible façades) and the exact extent of shadows on the ground. Avoid leaving gaps or overlaps between classes; inconsistency in boundaries can confuse the model during training.

Buildings

Always start with drawing buildings first. A building is considered to be a permanent (immobile) man-made structure that has a roof.

What to draw

- Entire roof area of each building, up to the clearly visible edge. If the side of the building is visible (on the side away from the sun), include that side as well—the building's side in shadow is still part of the building, not a ground shadow.
- Any part of a building façade that is visible.
- Any type of building; from large structures to small sheds to storage silos. If it is immobile, man-made and has a roof, label it.
- Canopies/awnings or transparent roofs should be marked as building.

What not to draw

- Shadows on the building itself: if part of a building's roof is dark due to shadow from another structure (or another part of the same building), that area is still building (just not sunlit) and should remain class 2. Do not carve out shadows that fall on a roof or wall; only exclude shadows on the ground. We are not differentiating sunlit vs. shadowed building pixels.
- Ground areas between buildings: e.g., if a building has a courtyard, it is not a building. Only draw the actual structure.
- Non-buildings: bridges, roads, other man-made structures that don't have roofs.
- If a building is occluded (i.e., hiding behind something else), only draw what is visible of the building.

Shadow

Specifically, we are looking for shadows cast by buildings onto adjacent ground, i.e. terrain or other ground-level surfaces.

What to draw

- Shadows cast by buildings onto adjacent ground: typically appearing as a dark shape extending from the base of the building opposite to the sun.
- Shadows cast by one building onto another ground-level structure or surface: e.g., a building shadow falling across a parking lot, cars, street, low vegetation, etc.
- In case of partial overlap with other shadows: please try to only label the portion of the shadow that is confidently cast by the building.
- Ensure that there is no gap between building and shadow classifications.

What not to draw

- Shadows cast by non-buildings: trees, poles, vehicles, any object that isn't a building. Do not include it if it is not a building-cast shadow.
- Shadows on top of buildings: if one building's shadow falls on another, do not label that as a shadow but as building instead. The shadow label is strictly for shadows on ground surfaces.

- Interior shadows on building roofs: sometimes structures will cast shadows within themselves on the same roof; these are not ground shadows and should be marked as building.
- Very faint or ambiguous shadows: underestimate rather than overestimate, as this will help the model later.

G Visual Results & Qualitative Examples

Figure G.1: Three images showcasing the original RGB on the left, with the manually annotated masks on the right. Purple for building shadows, teal for building footprints.

H Model Training Configuration

Table H.1: An overview of the trained models, with varying hyperparameters for the purpose of ablation and parameter tuning.

Experiment ID	Name	Hyperparameters	4th Channel Pseudo-Shadow Enabled
BASE BCE	BCE	-	False
BASE DICE	Dice	-	False
RGB BCE30 DICE70	BCE / Dice	BCE: 0.3, Dice: 0.7	False
RGB BCE50 DICE50	BCE / Dice	BCE: 0.5, Dice: 0.5	False
RGB BCE70 DICE30	BCE / Dice	BCE: 0.7, Dice: 0.3	False
RGBS BCE30 DICE70	BCE / Dice	BCE: 0.3, Dice: 0.7	True
RGBS BCE50 DICE50	BCE / Dice	BCE: 0.5, Dice: 0.5	True
RGBS BCE70 DICE30	BCE / Dice	BCE: 0.7, Dice: 0.3	True
PHYS ATT 1.0	Attentive BCE / Dice	Attention: 1, BCE: 0.5, Dice: 0.5	False
PHYS ATT 0.5	Attentive BCE / Dice	Attention: 0.5, BCE: 0.5, Dice: 0.5	False
PHYS ATT 0.1	Attentive BCE / Dice	Attention: 0.1, BCE: 0.5, Dice: 0.5	False
PHYS BCE 33	Physics BCE	Physics: 0.3, BCE: 0.3, Dice: 0.3	False
PHYS BCE 10	Physics BCE	Physics: 0.1, BCE: 0.45, Dice: 0.45	False
PHYS BCE 50	Physics BCE	Physics: 0.5, BCE: 0.25, Dice: 0.25	False
PHYS DICE 33	Physics Dice	Physics: 0.3, BCE:0.3, Dice: 0.3	False
PHYS DICE 10	Physics Dice	Physics: 0.1, BCE: 0.45, Dice: 0.45	False
PHYS DICE 50	Physics Dice	Physics: 0.5, BCE: 0.25, Dice: 0.25	False
HYB BCE PHYS30	Physics BCE	Physics: 0.3, BCE: 0.3, Dice: 0.3	True
HYB BCE PHYS50	Physics BCE	Physics: 0.5, BCE: 0.25, Dice: 0.25	True
HYB BCE PHYS10	Physics BCE	Physics: 0.1, BCE: 0.45, Dice: 0.45	True
HYB DICE PHYS30	Physics Dice	Physics, 0.3, BCE: 0.3, Dice: 0.3	True
HYB DICE PHYS50	Physics Dice	Physics: 0.5, BCE: 0.25,Dice: 0.25	True
HYB DICE PHYS10	Physics Dice	Physics: 0.1, BCE: 0.3, Dice: 0.3	True
HYB ATT 1.0	Attentive BCE / Dice	Physics: 1, BCE: 0.5, Dice: 0.5	True
HYB ATT 0.5	Attentive BCE / Dice	Physics: 0.5, BCE: 0.5, Dice: 0.5	True
HYB ATT 0.1	Attentive BCE / Dice	Physics: 0.1, BCE: 0.5, Dice: 0.5	True

I Overview of Training Hyperparameters

Table I.1: An overview of the used hyperparameters to train the baseline model by Luo et al. (2020). These settings were also used for the the training of the derivative transfer-trained models, albeit with different loss functions.

Name	Value	Description
Batch Size	8 (Base), 16 (PHYSHADE)	Number of samples per training batch
Stride	128	The stride used to extract patches from images
Patch_size	512	The size of the patches cropped from the imagery
Seed	15	Seed for the purposes of reproducibility
Epochs	150	The maximum number of training epochs
Early Stop Patience	25	Epochs to wait without improvement to Dice loss before stopping
Optimizer	AdamW	Optimizer used for training
Learning Rate	1e-4	Initial learning rate
Weight Decay	1e-4	Modulates training regularization
Scheduler	ReduceLROnPlateau	Reduces the learning rate when validation loss plateaus
Factor	0.5	Learning rate is reduced by this factor
Patience	5	The number of epochs without improvement before educing the learning rate
Min_lr	1e-6	Lower bound on learning rate
Mode	'min'	Reduce learning rate
Loss function	BCEWithLogitsLoss	Binary Cross Entropy with logits

J Smearing and Height Estimation Algorithms

Algorithm J.1 Pseudocode describing the smearing algorithm responsible for generating pseudo-shadows from building footprints

function SMEAR($M, \theta, (l_{\min}, l_{\max}), T,$ fade) ▷ M: binary mask, θ: azimuth, T: transform
  Compute unit vector: $(u_x, u_y) \leftarrow (\sin(\theta+\pi), \cos(\theta+\pi))$
  Extract pixel size: $(r_x, r_y) \leftarrow$ resolution from T
  Compute Step Distance: $r_{\text{eff}} \leftarrow \sqrt{(u_x r_x)^2 + (u_y r_y)^2}$
  Compute number of steps: $N \leftarrow \lceil l_{\max}/r_{\text{eff}} \rceil$
  Calculate Step Distance: $\Delta\ell \leftarrow l_{\max}/N$
  Initialize empty matrix S, same size as M
  for $iteration = 1, 2, ..., N$ **do**
    $\ell_i \leftarrow i \cdot \Delta\ell$
    Compute decay weight w_i based on ℓ_i:

$$w_i = \begin{cases} 1, & \text{if fade = solid} \\ 1, & \text{if } \ell_i \le l_{\min} \\ 0, & \text{if } \ell_i \ge l_{\max} \\ 1 - \dfrac{\ell_i - l_{\min}}{l_{\max} - l_{\min}}, & \text{otherwise} \end{cases}$$

    Compute shift:

$$\Delta x = \frac{u_x \cdot \ell_i}{r_x}, \quad \Delta y = -\frac{u_y \cdot \ell_i}{r_y}$$

    Shift mask: $M_i \leftarrow \text{warp}(M, \Delta x, \Delta y)$
    Accumulate: $S \leftarrow \max(S, w_i \cdot M_i)$
  end for
  if fade is gradient **then**
    Normalize $S \leftarrow S/\max(S)$
  end if
  return S
end function

Algorithm J.2 Pseudocode describing the raymarching-based height estimation method using subpixel interpolation and percentile-based aggregation.

function ESTIMATEHEIGHT($M, B, \theta, e, p, \Delta, N$) ▷ M: shadow mask, B: building mask, θ: azimuth, e: elevation, p: percentile, Δ: pixel size
  Compute direction unit vector: $(u_x, u_y) \leftarrow (\sin(\theta+\pi), -\cos(\theta+\pi))$
  Create binary masks: $M_b \leftarrow M > 0.5$, $B_b \leftarrow B > 0.5$
  Label shadow blobs: $L_M, n_M \leftarrow \text{label}(M_b)$
  Label combined connectivity: $L_C \leftarrow \text{label}(M_b \vee B_b)$
  Identify connected blob IDs:

$$C \leftarrow \{\, i \mid \exists (x,y) \in M_b \text{ with } L_C[x,y] \in L_C[B_b > 0] \,\}$$

  Initialize height map $H \leftarrow 0$ and blob ID map $I \leftarrow 0$
  $k \leftarrow 1$ ▷ Blob counter
  for all $i \in C$ **do**
    Extract blob mask: $S_i \leftarrow L_M == i$
    if area$(S_i) < 30$ **then** ▷ Helps ignore sliver artifacts from smearing
      continue
    end if
    Collect edge starting coordinates: $P \leftarrow \{(x,y) \in S_i : (x - \text{sign}(u_x), y - \text{sign}(u_y)) \notin S_i\}$
    Initialize ray length list: $R \leftarrow [\,]$
    for all $(x_0, y_0) \in P$ **do**
      Initialize ray value list: $v \leftarrow [\,]$
      for $j = 1, 2, ..., N$ **do**
        $x \leftarrow x_0 + u_x \cdot j$, $y \leftarrow y_0 + u_y \cdot j$
        if (x,y) out of bounds **then**
          break
        end if
        $v_j \leftarrow \text{interpolate}(M, (x,y))$
        if $v_j < 0.25$ **then** ▷ Min. ray length, change depending on resolution
          break
        end if
        Append v_j to v
      end for
      Append $|v|$ to R
    end for
    if $R \neq \emptyset$ **then**
      $\ell \leftarrow \text{percentile}(R, p)$ ▷ Prevents outliers common outside 8-point compass dir.
      $h \leftarrow \ell \cdot \tan(e)$
      Set $H[S_i] \leftarrow h$
      Set $I[S_i] \leftarrow k$
      $k \leftarrow k + 1$
    end if
  end for
  return H, I
end function

K Mosaic of Out-of-Fold Dataset

Figure K.1: A mosaic of the out-of-fold dataset used to assess the final models of PHYSHADE and to do the height estimation on. From left to right: Tile 6, Tile 7 and Tile 8.

L Model Training Results

Table L.1: Raw output of inference showing per fold statistics and averages on the intra-domain dataset.

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
BASE BCE	1	39	0.4621	0.4987	0.3332	0.0602	BCE	None	FALSE
BASE BCE	2	43	0.4012	0.3757	0.3937	0.0235	BCE	None	FALSE
BASE BCE	3	37	0.5144	0.5633	0.3849	0.0414	BCE	None	FALSE
BASE BCE	4	42	0.5729	0.6753	0.4198	0.0477	BCE	None	FALSE
BASE BCE	5	36	0.5384	0.6107	0.3572	0.0362	BCE	None	FALSE
BASE BCE	avg	39.4	0.4978	0.5447	0.3778	0.0418	BCE	None	FALSE
BASE DICE	1	50	0.5196	0.5837	0.4406	0.5551	Dice	None	FALSE
BASE DICE	2	73	0.5151	0.4595	0.5148	0.4882	Dice	None	FALSE
BASE DICE	3	45	0.5608	0.6265	0.4485	0.4368	Dice	None	FALSE
BASE DICE	4	29	0.5687	0.7169	0.3931	0.4351	Dice	None	FALSE
BASE DICE	5	37	0.49	0.5557	0.3533	0.5423	Dice	None	FALSE
BASE DICE	avg	46.8	0.5309	0.5885	0.4301	0.4915	Dice	None	FALSE
RGB BCE30 DICE70	1	35	0.5077	0.5518	0.4217	0.4218	BCE / Dice	Weight Bce=0.3 Weight Dice=0.7	FALSE
RGB BCE30 DICE70	2	46	0.4741	0.4009	0.5479	0.3877	BCE / Dice	Weight Bce=0.3 Weight Dice=0.7	FALSE
RGB BCE30 DICE70	3	45	0.5473	0.5886	0.4476	0.3416	BCE / Dice	Weight Bce=0.3 Weight Dice=0.7	FALSE
RGB BCE30 DICE70	4	31	0.5414	0.6788	0.371	0.3621	BCE / Dice	Weight Bce=0.3 Weight Dice=0.7	FALSE
RGB BCE30 DICE70	5	36	0.542	0.6111	0.3593	0.4055	BCE / Dice	Weight Bce=0.3 Weight Dice=0.7	FALSE
RGB BCE30 DICE70	avg	38.6	0.5225	0.5662	0.4295	0.3837	BCE / Dice	Weight Bce=0.3 Weight Dice=0.7	FALSE
RGB BCE50 DICE50	1	37	0.4877	0.5299	0.3906	0.3523	BCE / Dice	Weight Bce=0.5 Weight Dice=0.5	FALSE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
RGB BCE50 DICE50	2	44	0.4977	0.435	0.5244	0.2802	BCE / Dice	Weight Bce=0.5 Weight Dice=0.5	FALSE
RGB BCE50 DICE50	3	45	0.5445	0.5423	0.4662	0.2802	BCE / Dice	Weight Bce=0.5 Weight Dice=0.5	FALSE
RGB BCE50 DICE50	4	30	0.5266	0.6595	0.3527	0.2896	BCE / Dice	Weight Bce=0.5 Weight Dice=0.5	FALSE
RGB BCE50 DICE50	5	32	0.5475	0.6222	0.3589	0.3011	BCE / Dice	Weight Bce=0.5 Weight Dice=0.5	FALSE
RGB BCE50 DICE50	avg	37.6	0.5208	0.5578	0.4186	0.3007	BCE / Dice	Weight Bce=0.5 Weight Dice=0.5	FALSE
RGB BCE70 DICE30	1	35	0.4931	0.5463	0.392	0.2386	BCE / Dice	Weight Bce=0.7 Weight Dice=0.3	FALSE
RGB BCE70 DICE30	2	42	0.5074	0.4085	0.4644	0.1838	BCE / Dice	Weight Bce=0.7 Weight Dice=0.3	FALSE
RGB BCE70 DICE30	3	33	0.5433	0.5535	0.4148	0.1882	BCE / Dice	Weight Bce=0.7 Weight Dice=0.3	FALSE
RGB BCE70 DICE30	4	35	0.5402	0.6699	0.3859	0.1952	BCE / Dice	Weight Bce=0.7 Weight Dice=0.3	FALSE
RGB BCE70 DICE30	5	31	0.5403	0.5966	0.3781	0.1999	BCE / Dice	Weight Bce=0.7 Weight Dice=0.3	FALSE
RGB BCE70 DICE30	avg	35.2	0.5249	0.555	0.407	0.2011	BCE / Dice	Weight Bce=0.7 Weight Dice=0.3	FALSE
RGBS BCE30 DICE70	1	61	0.8234	0.8541	0.7838	0.1276	BCE / Dice	Weight Bce=0.3 Weight Dice=0.7	TRUE
RGBS BCE30 DICE70	2	46	0.8359	0.7577	0.8386	0.1344	BCE / Dice	Weight Bce=0.3 Weight Dice=0.7	TRUE
RGBS BCE30 DICE70	3	77	0.8362	0.8288	0.8272	0.1371	BCE / Dice	Weight Bce=0.3 Weight Dice=0.7	TRUE
RGBS BCE30 DICE70	4	46	0.8521	0.8729	0.8114	0.1163	BCE / Dice	Weight Bce=0.3 Weight Dice=0.7	TRUE
RGBS BCE30 DICE70	5	63	0.8901	0.9039	0.8586	0.1073	BCE / Dice	Weight Bce=0.3 Weight Dice=0.7	TRUE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
RGBS BCE30 DICE70	avg	58.6	0.8475	0.8435	0.8239	0.1245	BCE / Dice	Weight Bce=0.3 Weight Dice=0.7	TRUE
RGBS BCE50 DICE50	1	60	0.8297	0.8648	0.7683	0.0988	BCE / Dice	Weight Bce=0.5 Weight Dice=0.5	TRUE
RGBS BCE50 DICE50	2	51	0.8401	0.81	0.8404	0.0974	BCE / Dice	Weight Bce=0.5 Weight Dice=0.5	TRUE
RGBS BCE50 DICE50	3	49	0.8225	0.8078	0.8065	0.1114	BCE / Dice	Weight Bce=0.5 Weight Dice=0.5	TRUE
RGBS BCE50 DICE50	4	56	0.8507	0.8685	0.821	0.0922	BCE / Dice	Weight Bce=0.5 Weight Dice=0.5	TRUE
RGBS BCE50 DICE50	5	55	0.8844	0.8929	0.8498	0.0857	BCE / Dice	Weight Bce=0.5 Weight Dice=0.5	TRUE
RGBS BCE50 DICE50	avg	54.2	0.8455	0.8488	0.8172	0.0971	BCE / Dice	Weight Bce=0.5 Weight Dice=0.5	TRUE
RGBS BCE70 DICE30	1	48	0.8304	0.8602	0.7702	0.0764	BCE / Dice	Weight Bce=0.7 Weight Dice=0.3	TRUE
RGBS BCE70 DICE30	2	46	0.8534	0.8054	0.8321	0.0602	BCE / Dice	Weight Bce=0.7 Weight Dice=0.3	TRUE
RGBS BCE70 DICE30	3	58	0.8231	0.804	0.8199	0.0769	BCE / Dice	Weight Bce=0.7 Weight Dice=0.3	TRUE
RGBS BCE70 DICE30	4	53	0.843	0.8732	0.7747	0.0675	BCE / Dice	Weight Bce=0.7 Weight Dice=0.3	TRUE
RGBS BCE70 DICE30	5	61	0.8938	0.8923	0.8621	0.0552	BCE / Dice	Weight Bce=0.7 Weight Dice=0.3	TRUE
RGBS BCE70 DICE30	avg	53.2	0.8487	0.847	0.8118	0.0672	BCE / Dice	Weight Bce=0.7 Weight Dice=0.3	TRUE
PHYS ATT 0.1	1	38	0.5003	0.5296	0.4032	0.3317	Attentive BCE / Dice	Attention Weight=0.1 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 0.1	2	39	0.5171	0.4334	0.501	0.263	Attentive BCE / Dice	Attention Weight=0.1 Weight Bce=0.5 Weight Dice=0.5	FALSE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
PHYS ATT 0.1	3	45	0.5424	0.5385	0.4417	0.2703	Attentive BCE / Dice	Attention Weight=0.1 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 0.1	4	33	0.5314	0.6518	0.3855	0.2882	Attentive BCE / Dice	Attention Weight=0.1 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 0.1	5	36	0.5406	0.5646	0.3454	0.3034	Attentive BCE / Dice	Attention Weight=0.1 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 0.1	avg	38.2	0.5264	0.5436	0.4153	0.2913	Attentive BCE / Dice	Attention Weight=0.1 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 0.5	1	34	0.5148	0.5364	0.3992	0.332	Attentive BCE / Dice	Attention Weight=0.5 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 0.5	2	45	0.5087	0.3914	0.4981	0.2514	Attentive BCE / Dice	Attention Weight=0.5 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 0.5	3	39	0.5486	0.5594	0.447	0.2668	Attentive BCE / Dice	Attention Weight=0.5 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 0.5	4	35	0.5717	0.6794	0.381	0.2701	Attentive BCE / Dice	Attention Weight=0.5 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 0.5	5	36	0.5532	0.5659	0.3343	0.3003	Attentive BCE / Dice	Attention Weight=0.5 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 0.5	avg	37.8	0.5394	0.5465	0.4119	0.2841	Attentive BCE / Dice	Attention Weight=0.5 Weight Bce=0.5 Weight Dice=0.5	FALSE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
PHYS ATT 1.0	1	33	0.5115	0.5342	0.3998	0.3274	Attentive BCE / Dice	Attention Weight=1 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 1.0	2	46	0.5089	0.4392	0.5173	0.2464	Attentive BCE / Dice	Attention Weight=1 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 1.0	3	33	0.5367	0.5561	0.4583	0.2705	Attentive BCE / Dice	Attention Weight=1 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 1.0	4	37	0.5555	0.6565	0.3936	0.2764	Attentive BCE / Dice	Attention Weight=1 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 1.0	5	32	0.5497	0.6028	0.3844	0.2841	Attentive BCE / Dice	Attention Weight=1 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS ATT 1.0	avg	36.2	0.5325	0.5578	0.4307	0.281	Attentive BCE / Dice	Attention Weight=1 Weight Bce=0.5 Weight Dice=0.5	FALSE
PHYS BCE 10	1	41	0.5344	0.5815	0.4396	0.3642	Physics BCE	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	FALSE
PHYS BCE 10	2	45	0.4729	0.4254	0.4302	0.3005	Physics BCE	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	FALSE
PHYS BCE 10	3	45	0.5537	0.5807	0.4822	0.3204	Physics BCE	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	FALSE
PHYS BCE 10	4	39	0.5583	0.6952	0.3799	0.3101	Physics BCE	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	FALSE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
PHYS BCE 10	5	37	0.5335	0.5943	0.441	0.3266	Physics BCE	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	FALSE
PHYS BCE 10	avg	41.4	0.5306	0.5754	0.4346	0.3244	Physics BCE	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	FALSE
PHYS BCE 33	1	46	0.5126	0.5571	0.425	0.3763	Physics BCE	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	FALSE
PHYS BCE 33	2	47	0.427	0.3619	0.4211	0.2892	Physics BCE	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	FALSE
PHYS BCE 33	3	47	0.5533	0.5579	0.4604	0.3503	Physics BCE	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	FALSE
PHYS BCE 33	4	36	0.5631	0.7341	0.3513	0.3038	Physics BCE	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	FALSE
PHYS BCE 33	5	43	0.5611	0.5603	0.4226	0.3016	Physics BCE	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	FALSE
PHYS BCE 33	avg	43.8	0.5234	0.5543	0.4161	0.3242	Physics BCE	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	FALSE
PHYS BCE 50	1	33	0.4671	0.4422	0.4169	0.4124	Physics BCE	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	FALSE
PHYS BCE 50	2	44	0.4006	0.3221	0.392	0.3095	Physics BCE	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	FALSE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
PHYS BCE 50	3	41	0.5258	0.5055	0.4385	0.4002	Physics BCE	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	FALSE
PHYS BCE 50	4	48	0.5835	0.6855	0.4607	0.3352	Physics BCE	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	FALSE
PHYS BCE 50	5	53	0.5772	0.5829	0.4837	0.3274	Physics BCE	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	FALSE
PHYS BCE 50	avg	43.8	0.5108	0.5076	0.4384	0.357	Physics BCE	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	FALSE
PHYS DICE 10	1	41	0.5035	0.529	0.3938	0.3856	Physics Dice	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	FALSE
PHYS DICE 10	2	55	0.5106	0.4387	0.4984	0.3295	Physics Dice	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	FALSE
PHYS DICE 10	3	45	0.5393	0.5225	0.4285	0.3313	Physics Dice	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	FALSE
PHYS DICE 10	4	33	0.5603	0.681	0.3875	0.3216	Physics Dice	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	FALSE
PHYS DICE 10	5	36	0.55	0.6026	0.3664	0.3574	Physics Dice	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	FALSE
PHYS DICE 10	avg	42	0.5327	0.5548	0.4149	0.3451	Physics Dice	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	FALSE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
PHYS DICE 33	1	47	0.4982	0.5293	0.4186	0.4386	Physics Dice	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	FALSE
PHYS DICE 33	2	47	0.4923	0.4288	0.5084	0.4175	Physics Dice	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	FALSE
PHYS DICE 33	3	44	0.5431	0.5307	0.4491	0.4159	Physics Dice	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	FALSE
PHYS DICE 33	4	37	0.5451	0.6691	0.3886	0.4004	Physics Dice	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	FALSE
PHYS DICE 33	5	35	0.555	0.5813	0.4002	0.4189	Physics Dice	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	FALSE
PHYS DICE 33	avg	42	0.5267	0.5479	0.433	0.4183	Physics Dice	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	FALSE
PHYS DICE 50	1	40	0.5147	0.4685	0.4312	0.5597	Physics Dice	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	FALSE
PHYS DICE 50	2	45	0.4882	0.4209	0.4831	0.5633	Physics Dice	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	FALSE
PHYS DICE 50	3	44	0.5321	0.4743	0.4685	0.556	Physics Dice	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	FALSE
PHYS DICE 50	4	37	0.5848	0.6288	0.4254	0.5128	Physics Dice	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	FALSE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
PHYS DICE 50	5	30	0.5781	0.5614	0.3983	0.5241	Physics Dice	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	FALSE
PHYS DICE 50	avg	39.2	0.5396	0.5108	0.4413	0.5432	Physics Dice	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	FALSE
HYB BCE PHYS10	1	70	0.8384	0.855	0.8065	0.1553	Physics BCE	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	TRUE
HYB BCE PHYS10	2	49	0.8495	0.83	0.824	0.1263	Physics BCE	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	TRUE
HYB BCE PHYS10	3	63	0.826	0.8055	0.8268	0.17	Physics BCE	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	TRUE
HYB BCE PHYS10	4	46	0.8534	0.8615	0.8357	0.1287	Physics BCE	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	TRUE
HYB BCE PHYS10	5	63	0.8943	0.8968	0.8668	0.1206	Physics BCE	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	TRUE
HYB BCE PHYS10	avg	58.2	0.8523	0.8498	0.832	0.1402	Physics BCE	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	TRUE
HYB BCE PHYS30	1	43	0.8187	0.7882	0.8294	0.2213	Physics BCE	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	TRUE
HYB BCE PHYS30	2	56	0.8154	0.7636	0.8216	0.1759	Physics BCE	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	TRUE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
HYB BCE PHYS30	3	62	0.8188	0.7754	0.8436	0.2331	Physics BCE	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	TRUE
HYB BCE PHYS30	4	66	0.8426	0.8324	0.845	0.1741	Physics BCE	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	TRUE
HYB BCE PHYS30	5	50	0.8792	0.8583	0.8871	0.1691	Physics BCE	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	TRUE
HYB BCE PHYS30	avg	55.4	0.8349	0.8036	0.8453	0.1947	Physics BCE	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	TRUE
HYB BCE PHYS50	1	56	0.7876	0.7121	0.835	0.2819	Physics BCE	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	TRUE
HYB BCE PHYS50	2	57	0.7878	0.7091	0.8615	0.2164	Physics BCE	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	TRUE
HYB BCE PHYS50	3	60	0.7822	0.6365	0.8648	0.2898	Physics BCE	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	TRUE
HYB BCE PHYS50	4	52	0.8222	0.7823	0.8594	0.2178	Physics BCE	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	TRUE
HYB BCE PHYS50	5	64	0.8589	0.7783	0.9139	0.2106	Physics BCE	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	TRUE
HYB BCE PHYS50	avg	57.8	0.8077	0.7237	0.8669	0.2433	Physics BCE	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	TRUE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
HYB DICE PHYS10	1	61	0.8297	0.8512	0.7945	0.155	Physics Dice	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	TRUE
HYB DICE PHYS10	2	50	0.8367	0.7724	0.8574	0.1653	Physics Dice	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	TRUE
HYB DICE PHYS10	3	54	0.8169	0.7903	0.8145	0.1778	Physics Dice	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	TRUE
HYB DICE PHYS10	4	56	0.8523	0.8624	0.8333	0.1475	Physics Dice	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	TRUE
HYB DICE PHYS10	5	55	0.886	0.8884	0.8588	0.1408	Physics Dice	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	TRUE
HYB DICE PHYS10	avg	55.2	0.8443	0.8329	0.8317	0.1573	Physics Dice	Weight Phys=0.1 Weight Bce=0.45 Weight Dice=0.45	TRUE
HYB DICE PHYS30	1	63	0.826	0.8242	0.8127	0.2492	Physics Dice	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	TRUE
HYB DICE PHYS30	2	46	0.8375	0.7406	0.885	0.2852	Physics Dice	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	TRUE
HYB DICE PHYS30	3	57	0.8167	0.7808	0.8395	0.2894	Physics Dice	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	TRUE
HYB DICE PHYS30	4	56	0.8537	0.8441	0.8482	0.2462	Physics Dice	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	TRUE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
HYB DICE PHYS30	5	72	0.8894	0.8799	0.8846	0.237	Physics Dice	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	TRUE
HYB DICE PHYS30	avg	58.8	0.8447	0.8139	0.854	0.2614	Physics Dice	Weight Phys=0.3 Weight Bce=0.3 Weight Dice=0.3	TRUE
HYB DICE PHYS50	1	48	0.7905	0.6927	0.8724	0.3606	Physics Dice	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	TRUE
HYB DICE PHYS50	2	50	0.8235	0.4857	0.9104	0.4186	Physics Dice	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	TRUE
HYB DICE PHYS50	3	42	0.7921	0.6112	0.8941	0.4126	Physics Dice	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	TRUE
HYB DICE PHYS50	4	52	0.8328	0.7149	0.8802	0.3546	Physics Dice	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	TRUE
HYB DICE PHYS50	5	42	0.8522	0.7801	0.9151	0.3443	Physics Dice	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	TRUE
HYB DICE PHYS50	avg	46.8	0.8182	0.6569	0.8944	0.3782	Physics Dice	Weight Phys=0.5 Weight Bce=0.25 Weight Dice=0.25	TRUE
HYB ATT 0.1	1	48	0.8321	0.8583	0.7856	0.1017	Attentive BCE / Dice	Attention Weight=0.1 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 0.1	2	50	0.8447	0.796	0.8343	0.0944	Attentive BCE / Dice	Attention Weight=0.1 Weight Bce=0.5 Weight Dice=0.5	TRUE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
HYB ATT 0.1	3	59	0.8242	0.8099	0.8186	0.1119	Attentive BCE / Dice	Attention Weight=0.1 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 0.1	4	46	0.8498	0.8604	0.8129	0.0935	Attentive BCE / Dice	Attention Weight=0.1 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 0.1	5	48	0.8851	0.8918	0.851	0.0861	Attentive BCE / Dice	Attention Weight=0.1 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 0.1	avg	50.2	0.8472	0.8433	0.8205	0.0975	Attentive BCE / Dice	Attention Weight=0.1 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 0.5	1	58	0.8278	0.8651	0.7821	0.1015	Attentive BCE / Dice	Attention Weight=0.5 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 0.5	2	52	0.8406	0.8036	0.8328	0.0919	Attentive BCE / Dice	Attention Weight=0.5 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 0.5	3	55	0.8224	0.8166	0.813	0.1143	Attentive BCE / Dice	Attention Weight=0.5 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 0.5	4	46	0.8504	0.8639	0.8161	0.0942	Attentive BCE / Dice	Attention Weight=0.5 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 0.5	5	65	0.8922	0.8952	0.8732	0.0797	Attentive BCE / Dice	Attention Weight=0.5 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 0.5	avg	55.2	0.8467	0.8489	0.8235	0.0963	Attentive BCE / Dice	Attention Weight=0.5 Weight Bce=0.5 Weight Dice=0.5	TRUE

Table L.1 continued from previous page

Experiment	fold	Epochs Ran	Dice	Precision	Recall	Loss	Loss Function	Loss Hyperparameters	Use Prior Channel
HYB ATT 1.0	1	59	0.8211	0.8365	0.7796	0.1115	Attentive BCE / Dice	Attention Weight=1 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 1.0	2	49	0.8372	0.7994	0.8374	0.0925	Attentive BCE / Dice	Attention Weight=1 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 1.0	3	55	0.8238	0.8118	0.8082	0.1175	Attentive BCE / Dice	Attention Weight=1 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 1.0	4	52	0.8446	0.8571	0.8194	0.0993	Attentive BCE / Dice	Attention Weight=1 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 1.0	5	48	0.8815	0.8969	0.8502	0.088	Attentive BCE / Dice	Attention Weight=1 Weight Bce=0.5 Weight Dice=0.5	TRUE
HYB ATT 1.0	avg	52.6	0.8417	0.8403	0.8189	0.1018	Attentive BCE / Dice	Attention Weight=1 Weight Bce=0.5 Weight Dice=0.5	TRUE

M Height Estimation Results

Table M.1: An overview of the raw height estimation statistics per model, image and blob.

Image	Blob ID	Azimuth	Solar Elevation	Closest Real Ray	Interp. Ray Length	Shadow Length	Est. Height	True Height	Error	Is Largest Blob?
Summer_Tile6	1	112.5393	43.5808	58	58	14.5	13.7989	12.9739	0.825	TRUE
Summer_Tile6	2	112.5393	43.5808	36	34.5	8.625	8.208	3.1765	5.0314	TRUE
Summer_Tile6	3	112.5393	43.5808	57	57	14.25	13.561	12.9956	0.5654	TRUE
Summer_Tile6	4	112.5393	43.5808	62	61.94	15.485	14.7363	13.0244	1.7118	TRUE
Summer_Tile6	5	112.5393	43.5808	25	24.7	6.175	5.8764	2.9397	2.9367	TRUE
Summer_Tile6	6	112.5393	43.5808	8	7.78	1.945	1.851	2.9339	-1.0829	TRUE
Summer_Tile7	2	108.3214	39.6869	48	48	12	9.958	8.8757	1.0823	TRUE
Summer_Tile8	2	108.2517	39.6533	23	23	5.75	4.7658	5.1738	-0.4079	TRUE
Summer_Tile8	3	108.2517	39.6533	21	21	5.25	4.3514	5.0823	-0.7309	TRUE
Summer_Tile8	4	108.2517	39.6533	31	31	7.75	6.4235	6.7105	-0.2869	TRUE
Summer_Tile8	5	108.2517	39.6533	12	12	3	2.4865	2.5192	-0.0327	TRUE
Summer_Tile8	6	108.2517	39.6533	29	28.54	7.135	5.9138	5.4782	0.4356	TRUE
Summer_Tile8	8	108.2517	39.6533	25	25	6.25	5.1803	7.0477	-1.8674	TRUE
Winter_Tile6	1	242.5711	38.6148	40	40	10	7.9871	12.9739	-4.9867	TRUE
Winter_Tile6	2	242.5711	38.6148	24	24	6	4.7923	12.9956	-8.2033	TRUE
Winter_Tile6	3	242.5711	38.6148	12	12	3	2.3961	3.1765	-0.7804	TRUE
Winter_Tile6	4	242.5711	38.6148	47	42	10.5	8.3865	13.0244	-4.638	TRUE
Winter_Tile6	5	242.5711	38.6148	12	12	3	2.3961	2.9339	-0.5377	TRUE
Winter_Tile7	1	131.7112	39.018	14	14	3.5	2.8361	7.3864	-4.5504	TRUE
Winter_Tile7	4	131.7112	39.018	25	25	6.25	5.0644	8.8757	-3.8113	TRUE
Winter_Tile8	2	130.7157	38.6246	21	21	5.25	4.1947	5.1738	-0.9791	TRUE
Winter_Tile8	3	130.7157	38.6246	17	17.08	4.27	3.4117	5.0823	-1.6706	TRUE

Table M.1 continued from previous page

Image	Blob ID	Azimuth	Solar Elevation	Closest Real Ray	Interp. Ray Length	Shadow Length	Est. Height	True Height	Error	Is Largest Blob?
Winter_Tile8	4	130.7157	38.6246	26	26	6.5	5.1934	6.7105	-1.517	TRUE
Winter_Tile8	5	130.7157	38.6246	12	12	3	2.397	2.5192	-0.1222	TRUE
Winter_Tile8	6	130.7157	38.6246	28	28.44	7.11	5.6808	5.4782	0.2027	TRUE
Winter_Tile8	7	130.7157	38.6246	27	27	6.75	5.3932	7.0477	-1.6545	TRUE
Winter_Tile8	8	130.7157	38.6246	18	18	4.5	3.5955	2.762	0.8335	TRUE
Summer_Tile8	1	108.2517	39.6533	12	12	3	2.4865	5.1738	-2.6872	FALSE
Summer_Tile8	7	108.2517	39.6533	12	12	3	2.4865	7.0477	-4.5612	FALSE
Winter_Tile7	2	131.7112	39.018	12	12	3	2.4309	8.8757	-6.4448	FALSE
Winter_Tile7	3	131.7112	39.018	23	23	5.75	4.6593	8.8757	-4.2165	FALSE
Winter_Tile7	5	131.7112	39.018	13	12.6	3.15	2.5525	8.8757	-6.3232	FALSE
Winter_Tile8	1	130.7157	38.6246	17	17.18	4.295	3.4317	5.1738	-1.7421	FALSE
Summer_Tile6	1	112.5393	43.5808	62	62	15.5	14.7505	12.9739	1.7767	TRUE
Summer_Tile6	2	112.5393	43.5808	37	35.65	8.9125	8.4816	3.1765	5.305	TRUE
Summer_Tile6	3	112.5393	43.5808	57	57	14.25	13.561	12.9956	0.5654	TRUE
Summer_Tile6	4	112.5393	43.5808	64	64	16	15.2264	13.0244	2.2019	TRUE
Summer_Tile6	5	112.5393	43.5808	7	6.96	1.74	1.6559	2.9339	-1.278	TRUE
Summer_Tile6	6	112.5393	43.5808	25	24.7	6.175	5.8764	2.9397	2.9367	TRUE
Summer_Tile7	2	108.3214	39.6869	38	38	9.5	7.8834	8.8757	-0.9923	TRUE
Summer_Tile8	2	108.2517	39.6533	23	23	5.75	4.7658	5.1738	-0.4079	TRUE
Summer_Tile8	3	108.2517	39.6533	22	22	5.5	4.5586	5.0823	-0.5237	TRUE
Summer_Tile8	4	108.2517	39.6533	32	32	8	6.6307	6.7105	-0.0797	TRUE
Summer_Tile8	5	108.2517	39.6533	12	11.85	2.9625	2.4554	2.5192	-0.0638	TRUE
Summer_Tile8	6	108.2517	39.6533	29	29	7.25	6.0091	5.4782	0.5309	TRUE
Summer_Tile8	7	108.2517	39.6533	25	25	6.25	5.1803	7.0477	-1.8674	TRUE
Winter_Tile6	1	242.5711	38.6148	69	68.72	17.18	13.7219	12.9739	0.748	TRUE

Table M.1 continued from previous page

Image	Blob ID	Azimuth	Solar Elevation	Closest Real Ray	Interp. Ray Length	Shadow Length	Est. Height	True Height	Error	Is Largest Blob?
Winter_Tile6	2	242.5711	38.6148	66	66	16.5	13.1788	12.9956	0.1831	TRUE
Winter_Tile6	3	242.5711	38.6148	12	12	3	2.3961	3.1765	-0.7804	TRUE
Winter_Tile6	4	242.5711	38.6148	66	66	16.5	13.1788	13.0244	0.1543	TRUE
Winter_Tile6	5	242.5711	38.6148	11	11	2.75	2.1965	2.9339	-0.7374	TRUE
Winter_Tile7	1	131.7112	39.018	37	37.74	9.435	7.6452	7.3864	0.2588	TRUE
Winter_Tile7	3	131.7112	39.018	53	53	13.25	10.7365	8.8757	1.8608	TRUE
Winter_Tile8	2	130.7157	38.6246	22	21.55	5.3875	4.3046	5.1738	-0.8692	TRUE
Winter_Tile8	3	130.7157	38.6246	20	20	5	3.995	5.0823	-1.0874	TRUE
Winter_Tile8	4	130.7157	38.6246	30	29.92	7.48	5.9765	6.7105	-0.734	TRUE
Winter_Tile8	5	130.7157	38.6246	14	14	3.5	2.7965	2.5192	0.2773	TRUE
Winter_Tile8	6	130.7157	38.6246	30	30.86	7.715	6.1642	5.4782	0.686	TRUE
Winter_Tile8	7	130.7157	38.6246	27	27.1	6.775	5.4132	7.0477	-1.6345	TRUE
Winter_Tile8	8	130.7157	38.6246	17	17	4.25	3.3957	2.762	0.6337	TRUE
Summer_Tile7	3	108.3214	39.6869	19	19	4.75	3.9417	8.8757	-4.934	FALSE
Summer_Tile8	1	108.2517	39.6533	11	10.84	2.71	2.2462			FALSE
Winter_Tile6	6	242.5711	38.6148	6	6	1.5	1.1981	2.9339	-1.7358	FALSE
Winter_Tile6	7	242.5711	38.6148	10	9.9	2.475	1.9768	12.9956	-11.0188	FALSE
Winter_Tile7	2	131.7112	39.018	26	25.85	6.4625	5.2366	7.3864	-2.1498	FALSE
Winter_Tile8	1	130.7157	38.6246	17	17	4.25	3.3957	5.1738	-1.778	FALSE
Summer_Tile6	1	112.5393	43.5808	58	57.69	14.4225	13.7251	12.9739	0.7513	TRUE
Summer_Tile6	2	112.5393	43.5808	37	36.91	9.2275	8.7813	3.1765	5.6048	TRUE
Summer_Tile6	3	112.5393	43.5808	55	55	13.75	13.0851	12.9956	0.0895	TRUE
Summer_Tile6	5	112.5393	43.5808	27	26.66	6.665	6.3427	2.9339	3.4089	TRUE
Summer_Tile6	6	112.5393	43.5808	61	61	15.25	14.5126	13.0244	1.4882	TRUE
Summer_Tile6	7	112.5393	43.5808	23	22.76	5.69	5.4149	2.9397	2.4752	TRUE

Table M.1 continued from previous page

Image	Blob ID	Azimuth	Solar Elevation	Closest Real Ray	Interp. Ray Length	Shadow Length	Est. Height	True Height	Error	Is Largest Blob?
Summer_Tile7	1	108.3214	39.6869	32	32	8	6.6386	7.3864	-0.7478	TRUE
Summer_Tile7	2	108.3214	39.6869	45	45	11.25	9.3356	8.8757	0.4599	TRUE
Summer_Tile8	2	108.2517	39.6533	22	22	5.5	4.5586	5.1738	-0.6151	TRUE
Summer_Tile8	3	108.2517	39.6533	22	21.57	5.3925	4.4695	5.0823	-0.6128	TRUE
Summer_Tile8	4	108.2517	39.6533	28	28.4	7.1	5.8848	6.7105	-0.8257	TRUE
Summer_Tile8	5	108.2517	39.6533	14	14	3.5	2.9009	2.5192	0.3817	TRUE
Summer_Tile8	6	108.2517	39.6533	28	28	7	5.8019	5.4782	0.3237	TRUE
Summer_Tile8	8	108.2517	39.6533	26	26	6.5	5.3875	7.0477	-1.6602	TRUE
Winter_Tile6	1	242.5711	38.6148	69	69	17.25	13.7778	12.9739	0.8039	TRUE
Winter_Tile6	2	242.5711	38.6148	65	65	16.25	12.9791	12.9956	-0.0165	TRUE
Winter_Tile6	3	242.5711	38.6148	13	13	3.25	2.5958	3.1765	-0.5807	TRUE
Winter_Tile6	4	242.5711	38.6148	13	13	3.25	2.5958	2.9339	-0.338	TRUE
Winter_Tile6	5	242.5711	38.6148	67	67	16.75	13.3784	13.0244	0.354	TRUE
Winter_Tile7	1	131.7112	39.018	37	37	9.25	7.4953	8.8757	-1.3804	TRUE
Winter_Tile7	2	131.7112	39.018	22	21.8	5.45	4.4162	7.3864	-2.9703	TRUE
Winter_Tile8	1	130.7157	38.6246	28	27.66	6.915	5.525	5.1738	0.3513	TRUE
Winter_Tile8	2	130.7157	38.6246	22	22	5.5	4.3945	5.0823	-0.6879	TRUE
Winter_Tile8	3	130.7157	38.6246	33	33	8.25	6.5917	6.7105	-0.1188	TRUE
Winter_Tile8	4	130.7157	38.6246	16	16	4	3.196	2.5192	0.6768	TRUE
Winter_Tile8	5	130.7157	38.6246	37	37	9.25	7.3907	5.4782	1.9125	TRUE
Winter_Tile8	6	130.7157	38.6246	29	29	7.25	5.7927	7.0477	-1.255	TRUE
Winter_Tile8	7	130.7157	38.6246	26	25.04	6.26	5.0017	2.762	2.2397	TRUE
Summer_Tile6	4	112.5393	43.5808	14	14	3.5	3.3308	3.1765	0.1542	FALSE
Summer_Tile8	1	108.2517	39.6533	13	13	3.25	2.6937	5.1738	-2.48	FALSE
Summer_Tile8	7	108.2517	39.6533	12	12	3	2.4865	7.0477	-4.5612	FALSE

Table M.1 continued from previous page

Image	Blob ID	Azimuth	Solar Elevation	Closest Real Ray	Interp. Ray Length	Shadow Length	Est. Height	True Height	Error	Is Largest Blob?
Summer_Tile6	1	112.5393	43.5808	58	57.65	14.4125	13.7156	12.9739	0.7418	TRUE
Summer_Tile6	2	112.5393	43.5808	34	32.6	8.15	7.7559	3.1765	4.5794	TRUE
Summer_Tile6	3	112.5393	43.5808	56	56	14	13.3231	12.9956	0.3275	TRUE
Summer_Tile6	4	112.5393	43.5808	61	61	15.25	14.5126	13.0244	1.4882	TRUE
Summer_Tile6	6	112.5393	43.5808	23	22.84	5.71	5.4339	2.9397	2.4942	TRUE
Summer_Tile6	7	112.5393	43.5808	9	8.78	2.195	2.0889	2.9339	-0.845	TRUE
Summer_Tile7	2	108.3214	39.6869	46	46	11.5	9.543	8.8757	0.6673	TRUE
Summer_Tile8	3	108.2517	39.6533	22	22	5.5	4.5586	5.1738	-0.6151	TRUE
Summer_Tile8	4	108.2517	39.6533	21	21	5.25	4.3514	5.0823	-0.7309	TRUE
Summer_Tile8	5	108.2517	39.6533	29	29	7.25	6.0091	6.7105	-0.7014	TRUE
Summer_Tile8	6	108.2517	39.6533	14	14	3.5	2.9009	2.5192	0.3817	TRUE
Summer_Tile8	7	108.2517	39.6533	31	30.06	7.515	6.2287	5.4782	0.7506	TRUE
Summer_Tile8	9	108.2517	39.6533	26	26	6.5	5.3875	7.0477	-1.6602	TRUE
Winter_Tile6	1	242.5711	38.6148	60	59.34	14.835	11.8489	12.9739	-1.125	TRUE
Winter_Tile6	2	242.5711	38.6148	66	66	16.5	13.1788	12.9956	0.1831	TRUE
Winter_Tile6	3	242.5711	38.6148	15	14.48	3.62	2.8913	3.1765	-0.2852	TRUE
Winter_Tile6	4	242.5711	38.6148	12	12	3	2.3961	2.9339	-0.5377	TRUE
Winter_Tile6	5	242.5711	38.6148	65	65	16.25	12.9791	13.0244	-0.0454	TRUE
Winter_Tile7	1	131.7112	39.018	20	19.75	4.9375	4.0009	8.8757	-4.8748	TRUE
Winter_Tile8	1	130.7157	38.6246	23	23.32	5.83	4.6581	5.1738	-0.5156	TRUE
Winter_Tile8	3	130.7157	38.6246	15	15	3.75	2.9962	5.0823	-2.0861	TRUE
Winter_Tile8	4	130.7157	38.6246	25	25	6.25	4.9937	6.7105	-1.7168	TRUE
Winter_Tile8	5	130.7157	38.6246	14	14	3.5	2.7965	2.5192	0.2773	TRUE
Winter_Tile8	6	130.7157	38.6246	33	33	8.25	6.5917	5.4782	1.1135	TRUE
Winter_Tile8	7	130.7157	38.6246	21	21	5.25	4.1947	7.0477	-2.853	TRUE

Table M.1 continued from previous page

Image	Blob ID	Azimuth	Solar Elevation	Closest Real Ray	Interp. Ray Length	Shadow Length	Est. Height	True Height	Error	Is Largest Blob?
Winter_Tile8	8	130.7157	38.6246	24	22.89	5.7225	4.5722	2.762	1.8102	TRUE
Summer_Tile6	5	112.5393	43.5808	7	7	1.75	1.6654	2.9339	-1.2685	FALSE
Summer_Tile8	1	108.2517	39.6533	12	12	3	2.4865	5.1738	-2.6872	FALSE
Summer_Tile8	2	108.2517	39.6533	5	5	1.25	1.0361	5.1738	-4.1377	FALSE
Summer_Tile8	8	108.2517	39.6533	14	13.7	3.425	2.8388	7.0477	-4.2089	FALSE
Winter_Tile8	2	130.7157	38.6246	22	21.12	5.28	4.2187	5.1738	-0.9551	FALSE
Winter_Tile8	9	130.7157	38.6246	20	20	5	3.995	7.0477	-3.0527	FALSE