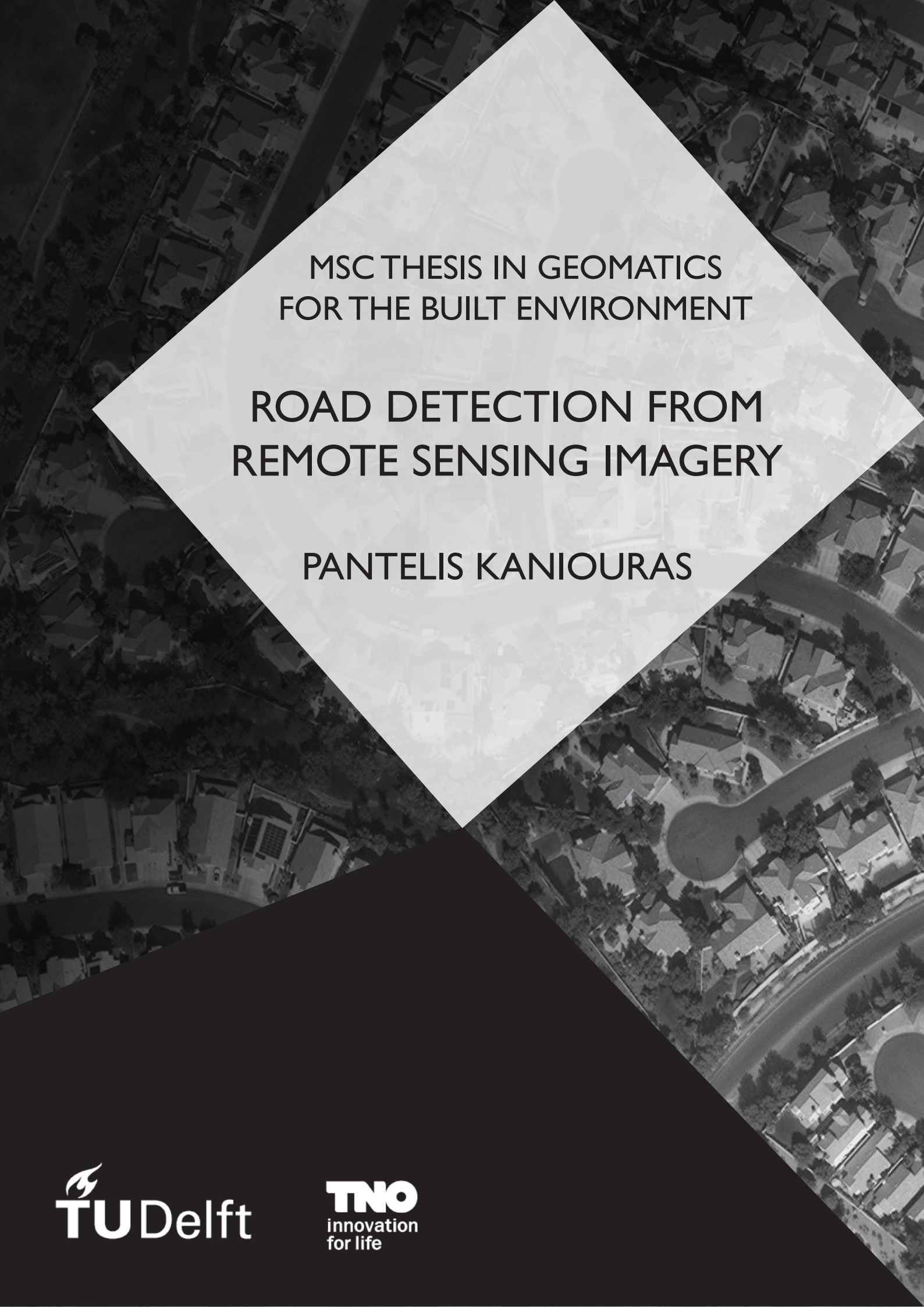


MSC THESIS IN GEOMATICS
FOR THE BUILT ENVIRONMENT

ROAD DETECTION FROM REMOTE SENSING IMAGERY

PANTELIS KANIOURAS

TU Delft

TNO innovation for life

MSc thesis in Geomatics

Road Detection from Remote Sensing Imagery

Pantelis Kaniouras

June 2020

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

Cover designer : Catherine Raad

The work in this thesis was carried out in the:

3D geoinformation group
Department of Urbanism
Faculty of the Built Environment & Architecture
Delft University of Technology

In cooperation with:

Supervisors:	Dr. Liangliang Nan
	Dr. Roderik Lindenbergh
	Frido Kuijper (TNO)
Co-reader:	Dr. Jesús Balado Frías

Abstract

Road network maps facilitate a great number of applications in our everyday life. However, their automatic creation is a difficult task, and so far, published methodologies cannot provide reliable solutions. The common and most recent approach is to design a road detection algorithm from remote sensing imagery based on a Convolutional Neural Network (CNN), followed by a result refinement post-processing step. In this project I proposed a deep learning model that utilized the Multi-Task Learning (MTL) technique to improve the performance of the road detection task by incorporating prior knowledge constraints. MTL is a mechanism whose objective is to improve a model's generalization performance by exploiting information retrieved from the training signals of related tasks as an inductive bias, and, as its name suggests, solve multiple tasks simultaneously. Carefully selecting which tasks will be jointly solved favors the preservation of specific properties of the target object, in this case, the road network. My proposed model is a Multi-Task Learning U-Net with a ResNet34 encoder, pre-trained on the ImageNet dataset, that solves for the tasks of Road Detection Learning, Road Orientation Learning, and Road Intersection Learning. Combining the capabilities of the U-Net model, the ResNet encoder and the constrained MTL mechanism, my model achieved better performance both in terms of image segmentation and topology preservation against the baseline single-task solving model. The project was based on the publicly available SpaceNet Roads Dataset.

Acknowledgements

I would like to thank the following people, without whom I would not have been able to complete this research, and I would not have made it through my masters degree:

My supervisors Dr. Liangliang Nan and Dr. Roderik C. Lindenbergh at Delft University of Technology. They were always available whenever I had a question about my research, providing guidance and feedback throughout this project.

The amazing people at TNO, who always made me feel more than welcome with their unwavering support during this stressful year. Especially, I am deeply indepted to my external supervisor Frido Kuijper for his expert guidance, incessant encouragement, and constructive criticism. I couldn't ask for a better supervisor in my first job abroad.

My family for all the support you have shown me through these years. Thank you for the daily video calls, the funny pictures and the extra virgin Greek olive oil.

My favorite person in the world, Roxani Gkavra. Without your support I would have stopped these studies a long time ago. You have been amazing, and I will now be able to cook more for you!

Finally, I would like to thank my friends for always keeping an eye on me and for sending long, drunk and loud motivational messages, even during the night, at around 3 a.m...

Contents

List of Figures

List of Tables

Acronyms

1 Introduction

1.1 Motivation

The road network is the core and essential mode of transportation in our society. The possession of detailed and reliable digital road network datasets can support a wide number of applications, such as traffic management, road monitoring, vehicle navigation, urban planning, updating of geographic information systems, crisis response, disaster management and many more.

Digital road network datasets (example shown in Figure 1.1) are usually created by manual extraction [50]. It is a time consuming, expensive and labor-intensive procedure. As a consequence, it is unfeasible to satisfy the needs of modern times with manual work, given the popularity of location-based services and applications. In order for those activities to maintain their continuous functionality, an automated method is needed to acquire and update digital datasets of the dynamic road network structure.

Technological advances in the field of remote sensing offer the opportunity for spatial information extraction from an abundance of high-resolution image data that faithfully depict the earth's surface [58]. Although scientists have been trying to solve the problem of road detection for more than 30 years now [3], there hasn't been a flawless software that can generalize the desired output under all different situations that occur in the built environment. The reason is that road networks are intricate structures. Their observation in remote sensing imagery can be either locally blocked by a variety of objects, like trees, buildings, cars and other street furniture or confused by a neighboring location that has similar texture. Furthermore, their intensity pixel values can vary due to the difference in atmospheric conditions, seasonality of data acquisition and shadows of objects. A plethora of factors and their interrelation has to be taken into consideration to create a robust road detection algorithm.

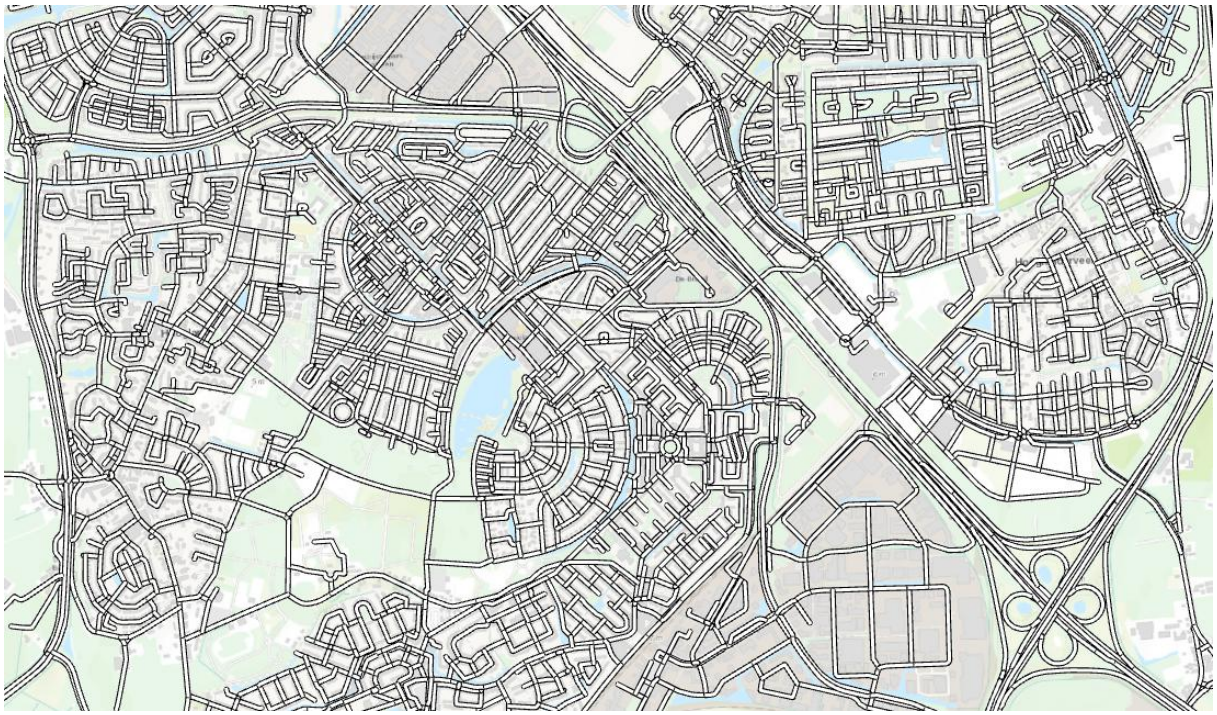


Figure 1.1: Road map of region Amersfoort, Netherlands

Recently, as a state-of-the-art machine learning technique, deep learning [42] made a major breakthrough in conventional computer vision tasks such as image classification, object detection, semantic segmentation and instance segmentation [78, 56, 28, 57, 24, 25, 71, 89, 12, 11, 10, 37, 59, 64, 55]. As a result, a large number of researchers started using deep learning techniques to solve remote sensing problems [48], including the road detection task [81]. Because of its supremacy of modeling complex nonlinear relationships between variables, deep learning surpassed conventional road detection algorithms [75]. However, the entire automation of the road extraction procedure is still not feasible. Extracted road networks using deep learning techniques frequently contain noise, artifacts, isolated road segments or miss information, making them inadequate for real-world applications [5].

Figure 1.2: Semantic segmentation on an image from the Cityscapes dataset

Road extraction using deep learning techniques and remote sensing data is usually posed as either a semantic segmentation task (Figure 1.2), where each pixel of a remote sensing image is classified according to the class that it belongs -road or non-road-, or as a road centerline vector graph extraction, both followed by post-processing steps to refine their results. The main advantage of semantic segmentation approaches is the preservation of geometric properties of the road network, since every classified road image pixel corresponds to a specific spatial extent, while the main advantage of vector graph extraction approaches is the preservation of topological properties, owing to their design to preserve road connectivity. Apart from the aforementioned difficulties and properties of the road network structure that complicate the procedure of road detection, inaccuracies using deep learning techniques appear for two more, but very important reasons. First, because topology is generally ignored during the pixel-wise semantic segmentation task, meaning that every pixel is handled individually and second, because ground truth or reference data used to train the deep learning models contain inaccuracies (label scarcity, omission, noise) [67, 53], which unfortunately confuse neural networks into making incorrect estimations.

Road networks are structures for which prior knowledge exists. For example, it is known that roads have locally consistent width, concrete as their surface material and are continuous (i.e. any location of the network should be able to be reached from any other location of the network). My focus on this study is to utilize as much prior knowledge or as many properties of the road network as possible meliorating current road detection algorithms. The collection of each property is equally important and essential for all applications. When combined, they can provide a structured and detailed representation of the road network, useful for safe localization and movement planning. In order to achieve that, I created a novel deep learning model inspired by the MTL mechasnism [46], that improves the task of road detection by sharing representations among related tasks and incorporating constraints. Enriching the amount of context that a deep learning model receives for its training leads to an improvement of precedent road detection results.

1.2 Problem Formulation

Before introducing more difficult concepts, the definition of the road detection task is given, allowing the reader to understand the fundamental concept and aim of this work. Therefore, let I be a remote sensing image, either a satellite or an aerial image (usually an RGB color image), and M its pairing road map image. $M(i, j)$ is equal to 1 whenever $I(i, j)$ corresponds to a road pixel and 0 otherwise. The purpose of a semantic segmentation deep learning model designed to detect the road network locations in a remote sensing image is to compute W_k, a set of learnable 2-D filters of size k, or else, a set of weights, that define the transformation $U_{W_k} : I \rightarrow M$ (Figure 1.3). Deep learning models are not infallible. The output of a deep learning model computes a transformation $U'_{W_k} : I \rightarrow M'$, where ${W'}_k \neq W_k$ and $M \neq M'$. M' is the prediction probability matrix, or otherwise, the model's output. During the procedure of a model's training, the goal is to minimize the differences between M and M' to obtain the optimal road map.

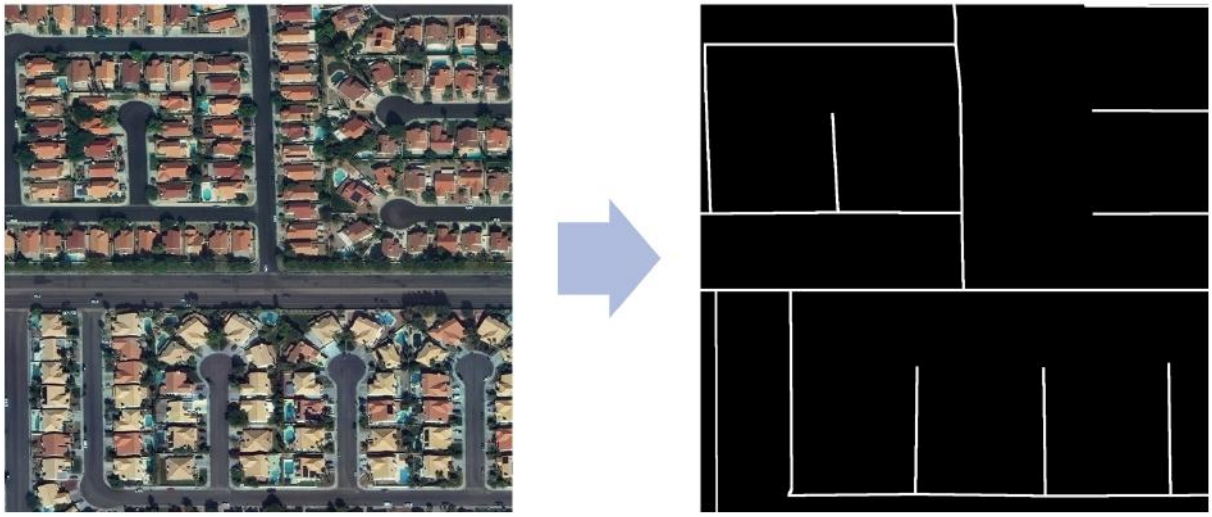


Figure 1.3: Transformation of a remote sensing image to a road map image

As stated in Section 1.1, two main categories of road detection methods exist. The technique described above belongs to the category of semantic segmentation approaches. The second category involves road centerline vector graph extraction approaches. Although many methods from the second cateogory exist, in the majority of published studies, researchers first exploited the capabilities of semantic segmentation, retrieving a byproduct, that would then be used by vector graph extraction techniques [77, 4, 44, 83, 49], finally yielding a vector graph that depicts the road network. Therefore, since semantic segmentation is the cornerstone of most studies, I decided to focus my research primarily on developing a novel model that utilizes the method of semantic segmentation and investigate how to improve its capabilities.

Apart from what it is shown in Figure 1.3, there are other ways to represent the road network with an image. For instance, the width of the road could be wider or narrower (e.g. only depicting its centerline), and could be represented with a constant or a non-constant width. A common and successful approach in most of the related work is to use a constant two meter wide road representation [21], mainly due to the lack of highly detailed ground truth for the whole extent of the road network and the differences that occur in every region. In my project I followed the same approach. The road network can also be represented with a vector graph, mainly consisting of line segments and nodes. A vector graph representation is outside the scope of this study.

1.3 Research Questions

The goal of this research is to examine how road network prior knowledge can assist the remote sensing task of road detection using deep learning techniques. In order to do that, a new deep learning model was designed, following the MTL paradigm, a system able to exchange and leverage additional information between related sub-tasks, thereby benefiting from the capabilities of each sub-task.

The main research question for this study is:

> ***To which extent is it possible to utilize road properties to improve road detection from remote sensing imagery using deep learning techniques?***

Underlying elements of the main research question are highlighted by the following sub-questions:

- *Can prior knowledge be incorporated as a constraint into a deep learning model? If yes, how?*
- *What are the limitations of a model that combines concepts of different models into one, unified model?*
- *Can prior knowledge improve road detection?*

Figure 1.4: Example of Road Detection with Semantic Segmentation over a region in the Netherlands

1.4 Thesis Overview

This thesis consists of five chapters. Chapter 2 is a record of scientific research related to the subject; in particular, road detection from remote sensing imagery -satellite or aerial imagery- using conventional methods or deep learning techniques and recently published work involving MTL deep learning models applied in various domains. Chapter 3 contains the proposed methodology. That includes explanations about the new Multi-Task Learning model and its building blocks, the creation of the learning tasks and the task-weighting scheme that was used. In Chapter 4, results derived from multiple experiments are presented and then used to assess the capabilities and performance of the proposed model. The thesis concludes with a discussion in Chapter 5, where research questions are answered and recommendations for future work are given.

2 Related work

As pointed out in the previous chapter, numerous algorithms have been developed in the past years to extract road networks from remote sensing imagery. In the first section of the current chapter, conventional methods for road extraction, as well as previous studies which explored deep learning techniques and are the current state of knowledge on the topic of road detection, are presented. Next, the concept of MTL is defined, accompanied with study projects that exploited it in various fields.

2.1 Road Detection

Classical methods for road detection can be classified into classification-based methods, knowledge-based methods, mathematical morphology methods, active contour model methods, and dynamic programming approaches. The scope of the literature review was confined only to the most referred works that utilized conventional methods and was more focused on previous studies using deep learning techniques, as the effectiveness and superiority of deep learning over traditional methods have already been demonstrated. For a more complete review of conventional road extraction methods, read [75].

2.1.1 Conventional Methods

Song et al. [68] created a two-step road extraction method. During the first stage, a support-vector machine was used to do a binary classification of the input image. The two output classes were the road class and the non-road class. During the second stage, a region growing algorithm was used to segment the road pixels to geometrically homogeneous objects based on a similarity criterion, giving higher importance to the objects' shapes rather than their spectral information. Post-processing steps of thresholding and thinning operations were used to refine the end results. Zhang et al. [85] focused on solving the misclassification issue of roads and parking lots by proposing an integrated approach that first segments the input images using a traditional k-means clustering algorithm and then automatically identifies the road cluster using a fuzzy logic classifier. A post-processing step to refine the result was applied here as well. They used shape descriptors of angular texture signature to differentiate objects that were mistakenly classified as road objects. Wegner et al. [76] used higher-order conditional random fields to reconstruct the road network by first segmenting overhead images into superpixels, and then identifying paths to connect these superpixels (Figure 2.1).

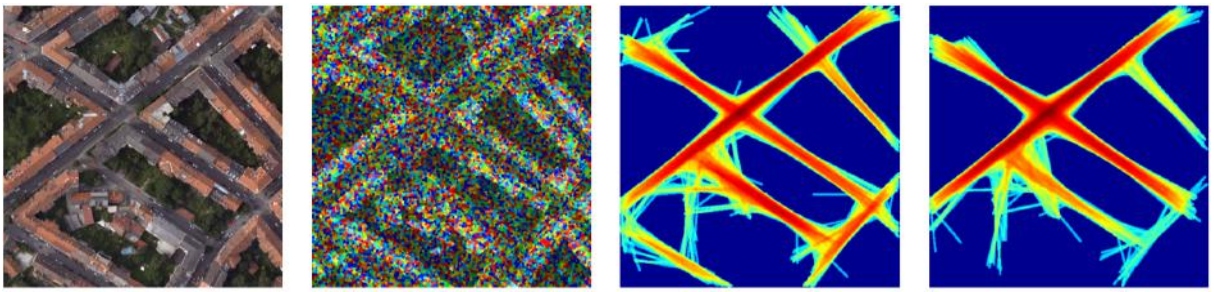


Figure 2.1: From Left to Right: aerial image of Graz, aerial image overlaid with the superpixel segmentation result, the density image of network cliques and the density image of junction cliques. The red color depicts higher density values and blue low density values. Retrieved from Wegner et al. [76]

Das et al. [18] also created a multi-stage algorithm to extract roads from remote sensing images. They exploited two principal features of roads, namely, distinct spectral contrast and locally linear trajectory, and utilized multiple techniques, like the Medial-Axis-Transformation, to refine their end result. Hu et al. [32] presented another two-step approach. They detect road footprints with a shape classification technique and combine them into a road tree, until no other candidate road footprints can be added. Afterwards, using a Bayes decision model, paths leaking outside of the road network are pruned. Laptev et al. [41] proposed a road detection model, only with a few parameters, based on the multi-scale detection of the road network, assisted by edge extraction using snakes. The output is created after extracting road intersections and combining them into a functional network (Figure 2.2). Hinz et al. [31] proposed a road detection technique for urban areas, integrating knowledge about the road network utilizing scale-dependent models. The implemented a strategy that automatically decides when and how context and road knowledge are optimally exploited.

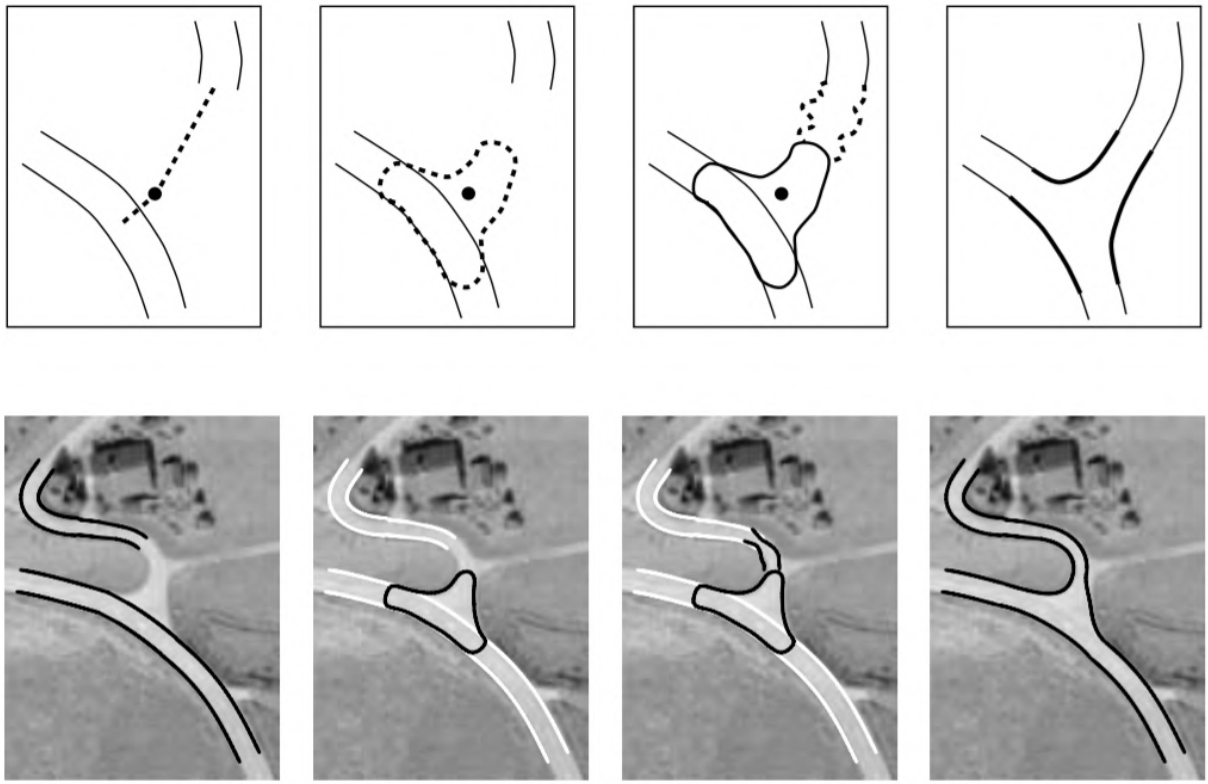


Figure 2.2: Road Intersection extraction. From Left to Right: Selection of starting hypothesis, approximation of the intersection, verification of geometry and connectivity, re-construction. Retrieved from Laptev et al. [41]

2.1.2 Deep Learning Methods

Cheng et al. [14] created a cascaded end-to-end CNN that both detects and extracts road center-lines. Their method is divided into two networks, one dealing with the road detection and one with the center-line extraction exploiting information collected during the inferring process of the first network. In the final step, a thinning algorithm is applied to refine the extracted center-line network. Mattyus et al. [49] created a method that improves road connectivity with post-processing steps after the prediction stage, which is based on a CNN segmentation, applying heuristics to missing connections or isolated road segments. Although it is a method that produces excellent results when the prediction output is accurate, it doesn't perform well when it has many errors, which is the usual case due to occlusion, shadows and reasons explained above (Figure 2.3).

Figure 2.3: Connection hypotheses generated by the $A*$ search during the post-processing step in DeepRoadMapper. Retrieved from Mattyus et al. [49]

Zhang et al. [86] created a Deep Residual U-Net model, by extending the U-Net architecture [59], adding short-cut connections between the CNN layers, producing a semantic segmentation output depicting the road network. Mosinska et al. [54] proposed a topology-aware loss function to assist the regular pixel-wise loss function (e.g. cross entropy loss), which is incapable of maintaining topological structures. Instead of creating a function that computes and compares topology, they used the response of chosen filters from a pre-trained VGG network [66] to construct it. The chosen filters favor elongated shapes, a property that decreases broken connections. However, it is very difficult to generalize this method to more complicated scenes with connections of random shapes and, even when their loss function is zeroed, their segmentation does not guarantee optimal topology. Alshehhi et al. [2] created a modified patch-based CNN architecture. Their model classifies every image pixel by using the sliding window technique, leading to direct extraction of roads and buildings. Even though CNN-based model perform adequately, the inefficiency of the sliding window approach makes them unsuitable for the road segmentation task. Bastani et al. [4] created an algorithm that directly constructs a graph representation of the road network using an iterative search procedure controlled by a CNN-based decision function. The construction of the road network graph starts from a seed location known to be on the road network, and sequential points are added according to the search procedure. The decision function is invoked at each step of the search to figure out the next best action to take: either add and walk towards a new node to the road network, or return to the previous node and continue from there. Their algorithm tends to fail at complex road intersections and on roads with high length or curvature. Once the CNN-based decision function makes a mistake, it is really hard to correct it and proceed with a valid road vector graph (Figure 2.4).

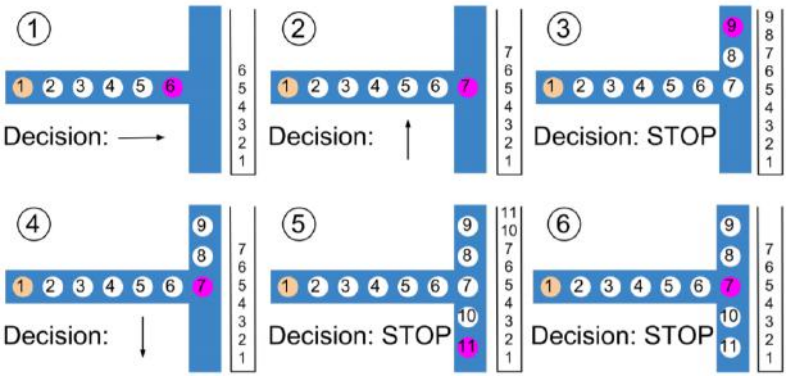


Figure 2.4: Exploring a junction with a CNN-based decision function. Retrieved from Bastani et al. [4]

Costea et al. [43] designed a three-step process to extract the road network in remote sensing images. In the first step, they combined multiple U-Nets to extract the initial road results. Then, they used a CNN to regain the details of the output results from the first step. In the final and third step, they refined isolated road segments by reasoning to improve the road extraction accuracy. Another study that utilized the U-Net architecture is the work of Sun et al. [69], who created a model using stacked U-Nets with multiple outputs. They also incorporated a hybrid loss function to confront the issue of unbalanced classes of training data. To improve performance, they also implemented post-processing steps, like shortest path search with hierarchical thresholds. Zhou et al. [88] designed a semantic segmentation model, called D-LinkNet, based on an encoder-decoder structure, dilated convolution to enlarge the receptive field of the feature points, maintaining resolution and pre-trained encoder, especially for the road detection task (Figure 2.5). The D-LinkNet was the winner of the DeepGlobe 2018 road challenge [19].

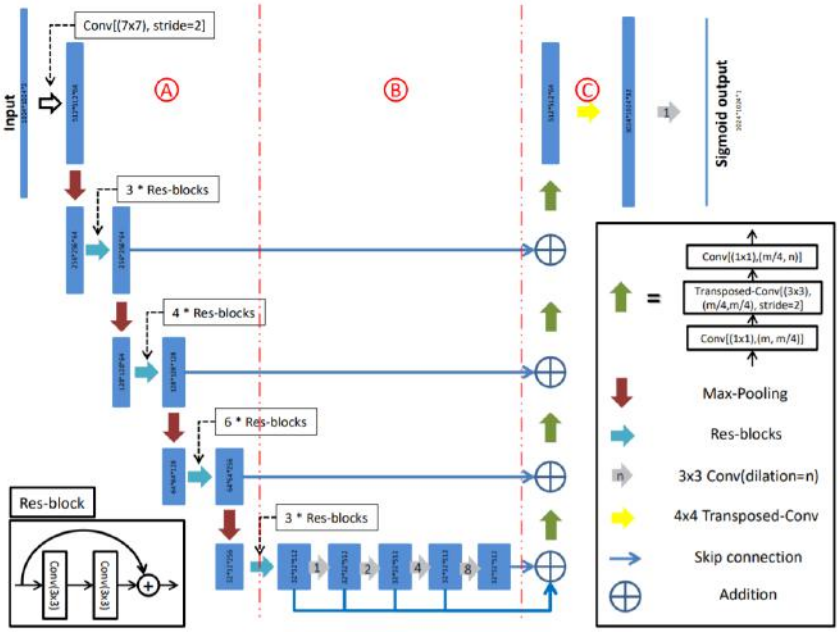


Figure 2.5: The D-LinkNet architecture. Retrieved from Zhou et al. [88]

Ventura et al. [73] designed a similar method to that of [4]. They constructed an iterative graph construction approach that adds segments to the network via a series of steps in a search process. On every step, a CNN determines whether a new node should be connected with the current graph and towards which direction. Both methods, [4] and [73], can improve connectivity despite their weaknesses, but they are also an order of magnitude slower than semantic segmentation approaches. Buslaev et al. [8] proposed a fully convolutional neural network consisting of a ResNet-34 pre-trained on ImageNet encoder and a decoder similar to that from a vanilla U-Net model. Additionally, they designed a loss function that simultaneously considers binary cross entropy loss and intersection over union (IoU) loss to improve their predictions, applied data augmentation and emphasized the importance of high quality ground truth masks in order to reach high prediction accuracy (Figure 2.6).

Li et al. [44] proposed PolyMapper, an algorithm for direct extraction of topological maps as a collection of building footprints and roads, using raw aerial imagery. They combined the principle of a maze solving algorithm, commonly known as the left-hand or right-hand rule, with a CNN-RNN deep learning model that first detects the points of interest, like intersection points or seed points, and then sequentially connects them according to their conditional probability distribution to belong to the road network. Batra et al. [5] developed a two stage road detection method to enhance the connectivity of the extracted road network. During the first stage, a joint learning module by stacking multi-branch encoder-decoder structure is implemented, aiming to allow the flow of information between two related tasks, those of per-pixel road segmentation and road orientation. During the second stage, a connectivity refinement model is applied, to connect small gaps and remove false positive occurrences.

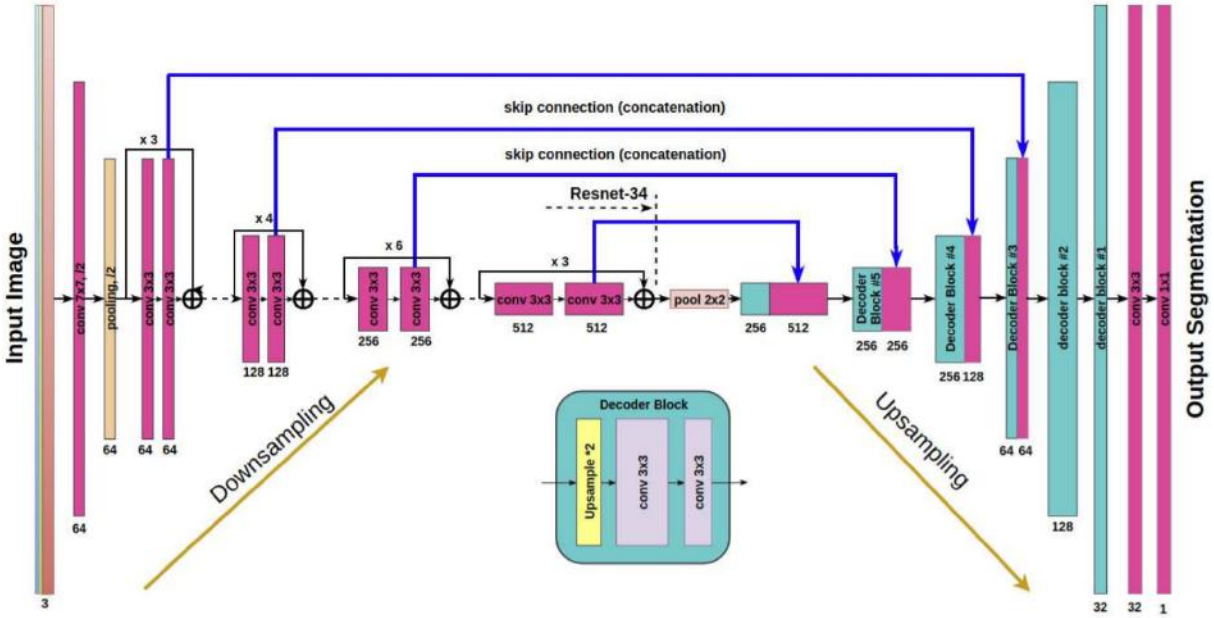


Figure 2.6: A Fully Convolutional Segmentation network. It follows a U-Net like architecture with a pre-trained ResNet-34 encoder. Each box in the figure corresponds to a multi-channel feature map. Retrieved from Buslaev et al. [8]

Yang et al. [82] created the RCNN unit and integrated it into the U-Net architecture, forming the RCNNUNet. This unit keeps detailed low-level spatial characteristics and improves inaccuracies caused by occlusion, complex background or noise. Wei et al. [77] also created a road network graph extraction that benefited from both segmentation and tracing methods to achieve better results. They proposed multiple starting points to avoid loss of connectivity in unreached areas. He et al. [27] improved road extraction by incorporating atrous spatial pyramid pooling (ASPP) with an encoder-decoder network. ASPP has the ability to extract multi-scale features, and, in combination with the Encoder-Decoder architecture, it can extract detailed features. They additionally implemented a structural similarity (SSIM) loss function and metric (Figure 2.7). Xu et al. [80] created a complete pipeline to extract a road network vector graph from satellite imagery of urban locations. They based their method on an architecture similar to the SegNeXt architecture [23], added a novel loss function and finally, a post-processing step to vectorize their segmentation result. Their method performs on par with other vector graph extraction methods.

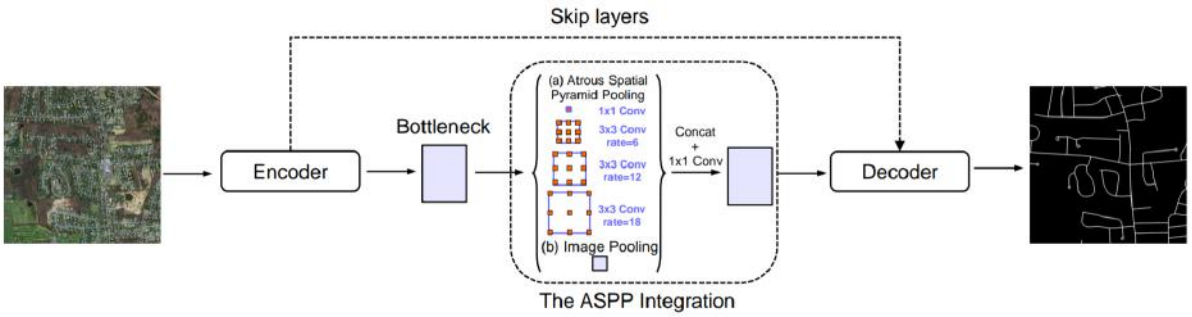
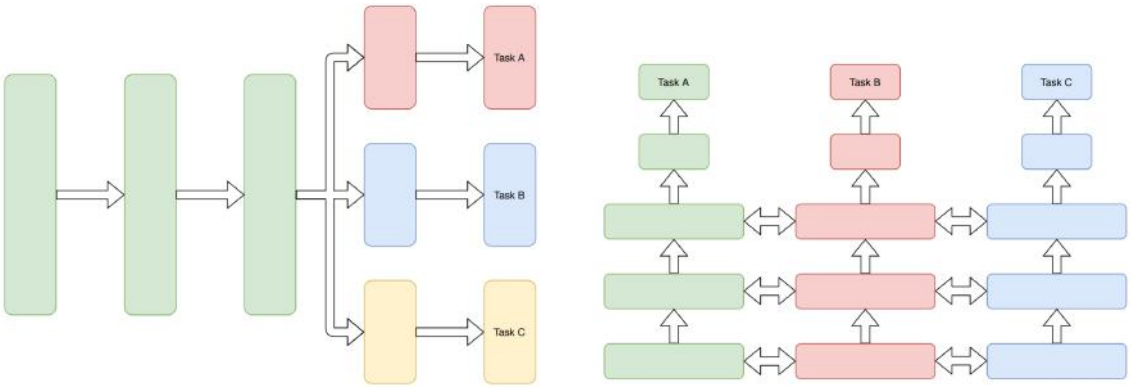


Figure 2.7: The improved U-net with an integrated atrous spatial pyramid pooling (ASPP). Retrieved from He et al. [27]

2.2 Multi-Task Learning

In common machine learning problems, the aim is to learn one task at a time and optimize for a particular metric. By doing so, information contained in the training signals of other tasks drawn from the same domain is neglected. As mentioned in Chapter 1, the goal of this project is to utilize as much prior knowledge as possible to assist the remote sensing task of road detection. One technique to accomplish that is the MTL approach, that uses information retrieved from related tasks to improve generalization, transferring knowledge as an inductive bias. This is achieved by learning tasks in parallel, using a shared representation. Sharing knowledge obtained for each task can assist other tasks improve their performance [9].

MTL is used in many disciplines, such as computer vision [46, 38, 52], natural language processing [47, 17], speech processing [79, 63] and reinforcement learning [36, 62]. There are two main ways of building a MTL system, hard parameter sharing and soft parameter sharing. In a hard parameter sharing model, hidden layers are shared across all tasks, but the model maintains separate task-specific output layers (e.g. fully-connected layers), as shown in Figure 2.8a. The most important limitation of this technique is that most often the final layers are responsible for learning representations that have to satisfy the needs of every participating task, which is rather difficult to optimize [60]. In a typical soft parameter sharing network each task has its own model with its own parameters, as shown in figure 2.8b. Representations are shared across different tasks with direct connections. The biggest limitation of this architecture is that the model's complexity grows linearly by increasing the number of participating tasks, leading to a less computationally efficient model [39]. Apart from improving generalization, MTL models have additional benefits. Because of their design, they reduce the amount of data required for training (all participating tasks use the same input), they can solve multiple tasks at once and, finally, they offer the possibility to avoid over-fitting [6].



(a) A Hard parameter sharing example model.

(b) A Soft parameter sharing example model.

Figure 2.8: Examples of Multi-Task Learning architectures

Although the fundamental concept of MTL models offers the possibility to achieve higher performance, it also comes with a cost. In a common MTL architecture, the objective loss function of every task is added into a unified loss function, which is optimized for the whole network, considering all participating tasks equal components of the network with a linear combination of each loss. Furthermore, especially in soft parameter sharing networks, it is difficult to understand which layers or which features should be shared and which are harmful for the network's performance. In order to overcome these issues, weights can be applied to the individual losses to distinguish which tasks are more important and reduce the amount a task influences the network to update the gradients towards a false direction or a representation learning scheme can be constructed that decides which layers won't compete each other and will contribute to a better outcome. Selecting these weights or constructing such a scheme can be difficult and expensive [52] (Figure 2.9). One of the early studies is the work of Kendall et al. [16],

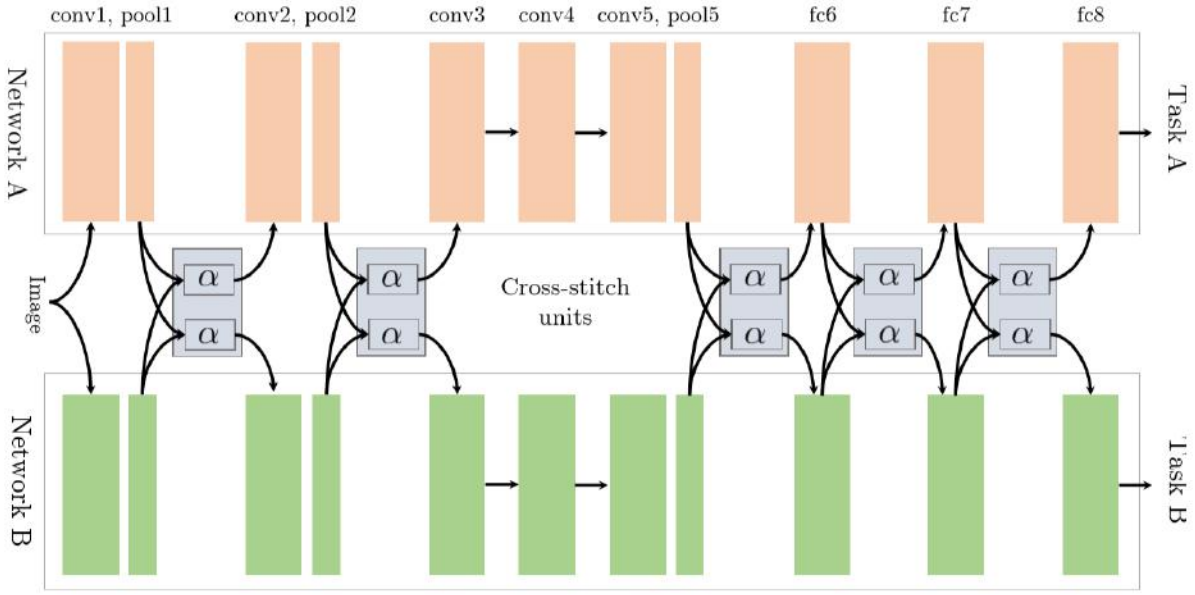


Figure 2.9: Cross-stitch networks for solving two tasks. Retrieved from Misra et al. [52]

who weighed the losses of a MTL model according to the homoscedastic uncertainty of each task (Figure 2.10).

A different weighting approach is the work of Chen et al. [13]. They chose to automatically balance the procedure of training by dynamically tuning gradient magnitudes. In other direction, Guo et al. [26] designed a method where more difficult tasks were prioritized. Zhao et al. [87] noticed that when two participating tasks in a MTL model have weak relevance, they might divert each other during parallel training and lead to a performance decline. As a solution they proposed a general modulation module to encourage feature sharing of related tasks and discourage the learning of unrelated tasks.

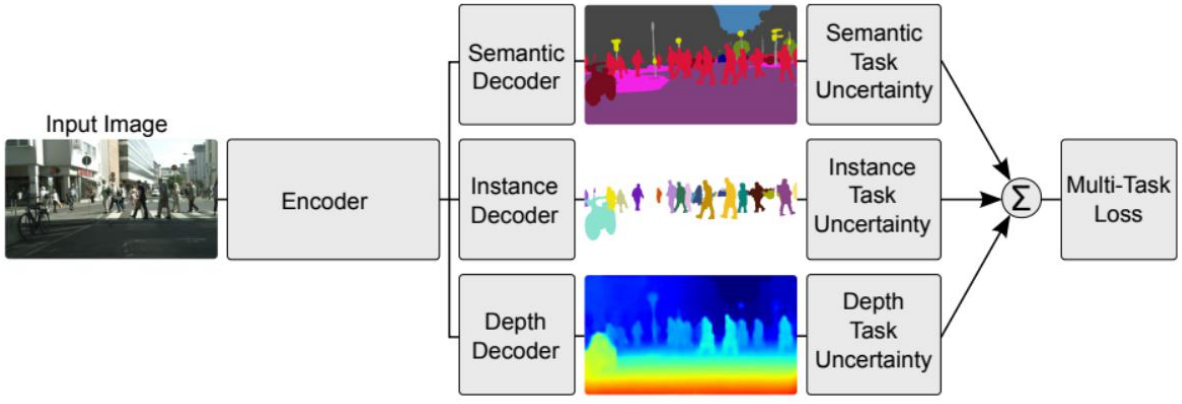


Figure 2.10: Multi-Task deep learning model with a Multi-Task Loss that combines multiple regression and classification loss functions. Retrieved from Cipolla et al. [16]

3 Methodology

Road network detection from remote sensing imagery is challenging due to the fact that road features visible in imagery are affected by multiple factors, such as the image collecting sensor type, image spectral and spatial resolution, weather conditions, light variations, landscape characteristics and many more. As a consequence, road networks are too difficult to be retrieved or modeled using a general procedural model. In order to build a system that reliably detects road networks, it is essential to analyze and understand what is considered as a road feature in remote sensing imagery.

In the work of Vosselman and Knecht [74], it is explained that road features in remote sensing imagery can be divided into 4 categories. The first category describes the geometric features. Roads are depicted as elongated elements with consistent width, that can only gradually change and their intersections have predefined geometric shapes, enforcing road construction rules. Another geometric road feature or property is the ratio between the road length and width, which is significantly large. The second category describes the photometric or spectral features. Road pixels have mainly gray intensity values (in urban environments, where dust roads are not present), are also consistent and can change slowly. Their edge gradient is most frequently larger, especially when adjacent non-road areas or objects are vegetation, cars and so on. The third category describes topological road properties. Generally, a road network consists of multiple road objects that interconnect and the transition from any point within the road network to another is feasible or not interrupted. The fourth and last category describes the texture features, which in image processing are considered patterns in an image. Translated into road detection, it means identifying the spatial distribution of road pixels in an area.

Many road detection algorithms use more than just one road feature in their implementation (see Chapter 2). Despite this, shadows, occlusion, difference in lighting and a plethora of other reasons impede the identification of the aforementioned road properties, making the task of road detection even more difficult (Figure 3.1). The focus of this study is to utilize as many road features as possible to create a more efficient road detection algorithm and propose ideas for further improvements in future work. In this chapter, the proposed deep learning model for detecting roads in remote sensing imagery is illustrated, together with its building blocks.

Figure 3.1: Landscapes with ambiguous semantics where road detection is difficult

3.1 Building blocks of Proposed Model

As mentioned both in Chapters 1 and 2, the proposed methodology of this research project includes the construction of a MTL model to solve the task of road network detection. Therefore, a model architecture is needed to be used as a basis, able to take advantage of the profitable capabilities of the MTL technique. In the following sections, a detailed description of the U-Net model with a ResNet34 encoder is given, which was chosen as the cornerstone model of this project due to its proven effectiveness. Further explanations and details about the model's capabilities and why it was selected are given in the following sections.

3.1.1 U-Net Architecture

The U-Net is a CNN created for biomedical imagery segmentation at the Computer Science Department of the University of Freiburg, Germany [59], and together with its variants, it is considered a standard model for semantic segmentation tasks. The U-Net architecture consists of a contracting path that captures context and a symmetric expanding path that enables precise localization. This sequence creates a u-shaped graph, which was the inspiration for the model's naming. The contracting or downsampling path is a CNN with repeated convolutions, carried on by a Rectified Linear Unit (ReLU) and a max pooling layer. During the contracting or downsampling path, spatial information is lessened and feature information is risen. The expansive or up-sampling path fuses information through a succession of up-convolutions and concatenations with high-resolution features from the corresponding levels of the contracting path.

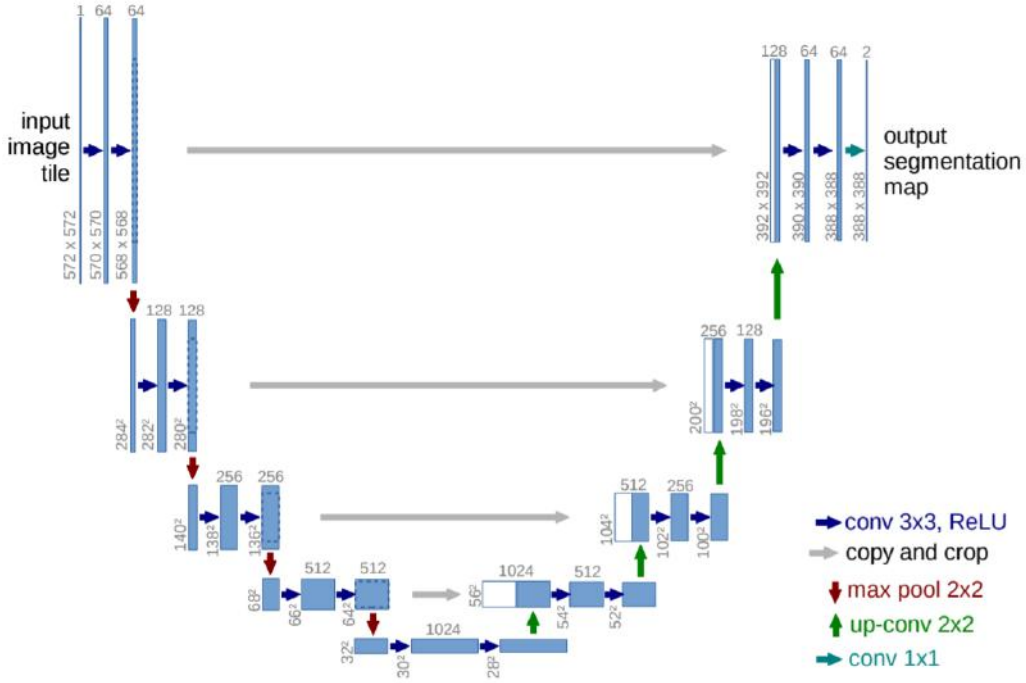


Figure 3.2: An example U-net model. Retrieved from Ronneberger et al. [59]

3.1.2 Residual Block

Neural networks with many hidden layers are capable of achieving higher performance on a chosen task than shallower networks. However, they are more complex and difficult to train. To address this problem, He et al. [29] introduced a deep residual learning framework, consisting of many stacked

residual blocks and won the 1st place on the ILSVRC 2015 classification task [61]. Each residual block can be expressed as a sequence of the following equations:

$$y_l = h(x_l) + F(x_l, W_l) \tag{3.1}$$

$$x_{l+1} = f(y_l) \tag{3.2}$$

where x_l and x_{l+1} are the input and the output of the $l - th$ block, F is a residual function, $h(x_l)$ is an identity mapping function and f is a ReLU function. The main idea behind this sequence is to learn the additive residual function F with respect to the $h(x_l)$, taking advantage of an identity mapping function $h(x_l) = x_l$. To make this happen, a skip connection or a 'shortcut' is attached.

The same group of researchers published another paper later that year [30], where they further analyzed deep residual networks and discovered that information propagation within residual blocks, but also to the rest of the network becomes easier if both $h(x_l)$ and $f(y_l)$ are identity mappings. To formulate an identity mapping $f(y_l) = y_l$ they considered activation functions ReLU and Batch Normalization (BN) [35] as the 'pre-activation' of the weight layers, while traditional techniques considered them as 'post-activation'. As a result they designed a new residual block, shown in Figure 3.3, and achieved even better results on the ImageNet challenge.

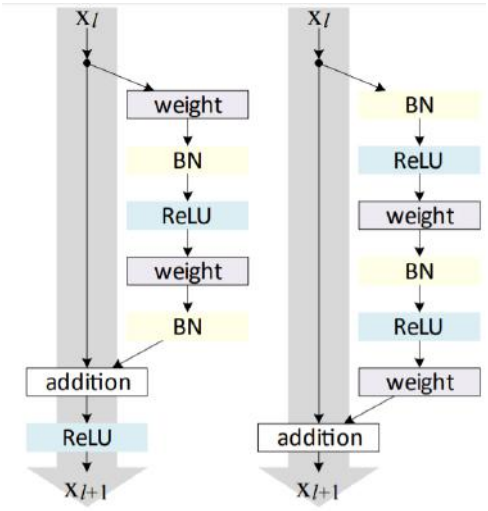


Figure 3.3: Left: original Residual Block. Right: improved Residual Block. Retrieved from He et al. [30]

3.1.3 U-Net with a ResNet34 Encoder

To take advantage of the capabilities of both previous techniques, many researchers replaced convolutions in U-Net on each level with Residual Blocks and achieved greater performance than with plain U-Net architecture [86]. In this project, I created a U-Net with a ResNet34 encoder as a baseline model. This network receives an RGB image as input and produces either a binary or a multi-class single-channel output of the same size. The model's architecture begins with a convolution with a kernel size of $7x7$ and stride of 2 to decrease the resolution of the input by half, followed by a BN, a ReLU and a max-pooling layer with a stride of 2. After that, stacked residual blocks execute the down-sampling path of the network. Each residual block consists of a double sequence of a convolution layer, a BN layer and a ReLU layer. The decoder part of the network consists of repeating up-sampling and regular U-Net convolutional blocks without "local" skip connections, that double the spatial resolution of the activation's output and decrease the number of feature channels by half. The final layer consists of a fully connecting layer with an activation function. Depending on the number of the output classes, it can either be a sigmoid or a softmax activation function.

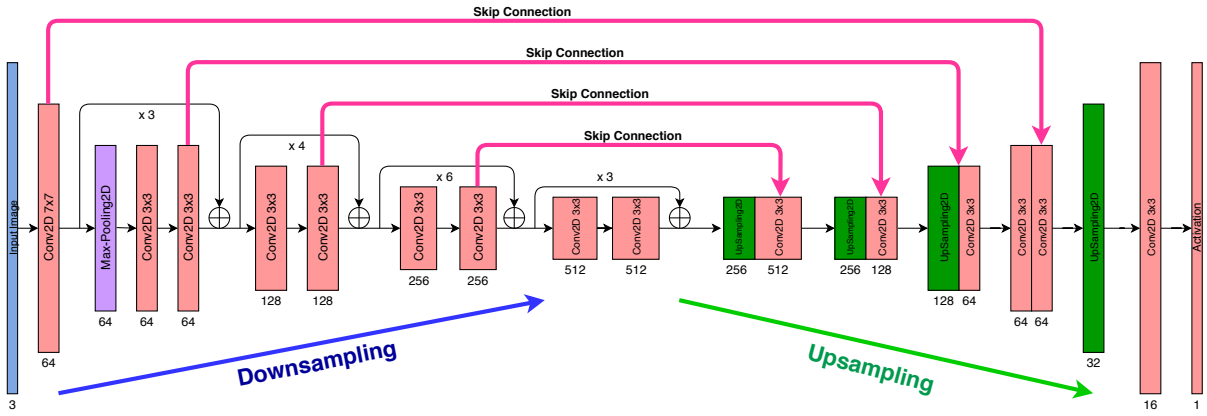


Figure 3.4: A U-net with a ResNet34 encoder. Each rectangle portrays a multi-channel feature map. The height of each rectangle symbolizes the resolution of a feature map, while its width the number of channels (also written bellow the rectangle). The pink arrows represent skip-connections, where information is shared from the encoder to the corresponding decoder level.

This network is efficient for numerous reasons. First, all convolutional operations are operated with a 3x3 filter and with the *SAME* padding, keeping the size of the feature maps the same at the same levels of the contracting and corresponding expanding path. The *SAME* padding boundary information is preserved and more convolutions can be added. Furthermore, because the feature size remains the same at a specific level for both the down-sampling and up-sampling paths, there is no need for cropping of the feature map before concatenating, and hence, no information is lost. Apart from the "long" skip connections between every level of contracting and expanding paths, "local" skip connections between convolutions on each level assist on receiving smooth loss curves and avoid gradient disappearance or explosion, like in the work of He et al. [30].

3.2 Proposed model: Multi-Task U-Net with a ResNet34 Encoder

In recent years, the development of deep learning and CNN achieved great performance in the field of computer vision [70, 33]. The common approach is to construct a neural network aiming to solve for a particular task using a case-specific architecture that usually cannot generalize good results under different circumstances. However, there are tasks, like visual scene understanding, that require outputs of more than just one network to accomplish that. MTL is a machine learning technique that solves multiple tasks simultaneously and overcomes this limitation, making it a rather popular technique in recent years. The second reason why it is useful is that it often achieves higher performance on a particular task, than regular single-task solving networks. This is happening because of the intrinsic dependencies of related tasks that are shared during training and operate as an inductive bias (see Chapter 2).

This study explores the usefulness of MTL to solve the problem of road network detection making good use of its second advantage. Therefore, a MTL model was constructed, using the neural network described in 3.1.3 as a building block model. The proposed model is a Multi-Task U-Net with a ResNet34 Encoder, able to solve 2, 3 or 4 tasks simultaneously. The architecture is similar to the U-Net with a ResNet34 Encoder until the beginning of the decoder path, where it splits into multiple branches according to the number of tasks it solves, forming its up-sampling paths. If two tasks are solved in parallel, then two upsampling branches will be created. If three tasks are solved in parallel, then three upsampling branches will be created, and so on. Each branch becomes the decoder of the selected task. To maintain the advantages of the baseline method, long skip connections were attached between the encoder path and the corresponding levels of the decoder path of every branch to share representations. The selection of the U-Net with a ResNet34 Encoder model as the building block of the proposed methodology was based on the fact that a variation of this architecture won the SpaceNet Road Detection and Routing Challenge [22] and is considered one of the state-of-the-art architectures.

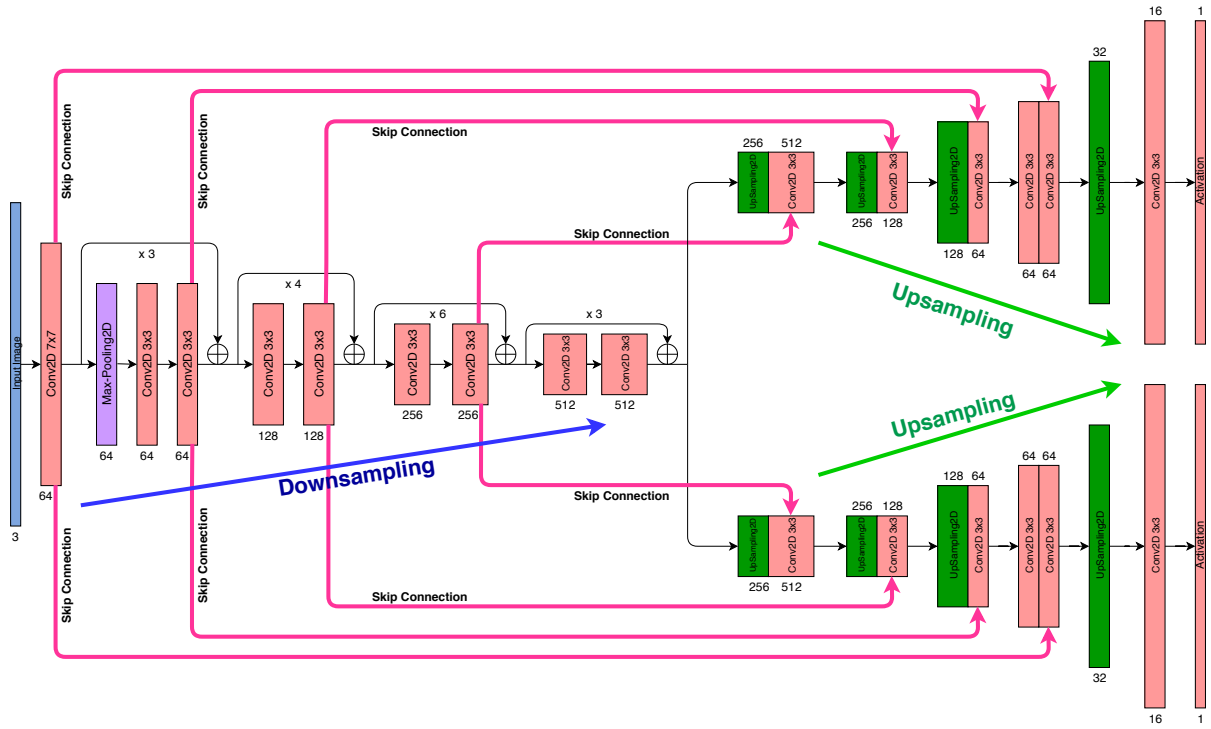


Figure 3.5: An example of the proposed Multi-Task U-Net model with a ResNet34 Encoder that solves two tasks. Each rectangle portrays a multi-channel feature map. The height of each box symbolizes the resolution of a feature map, while its width the number of channels (also written next to each rectangle). The pink arrows represent skip-connections, where information is shared from the shared encoder to each decoder branch of each task.

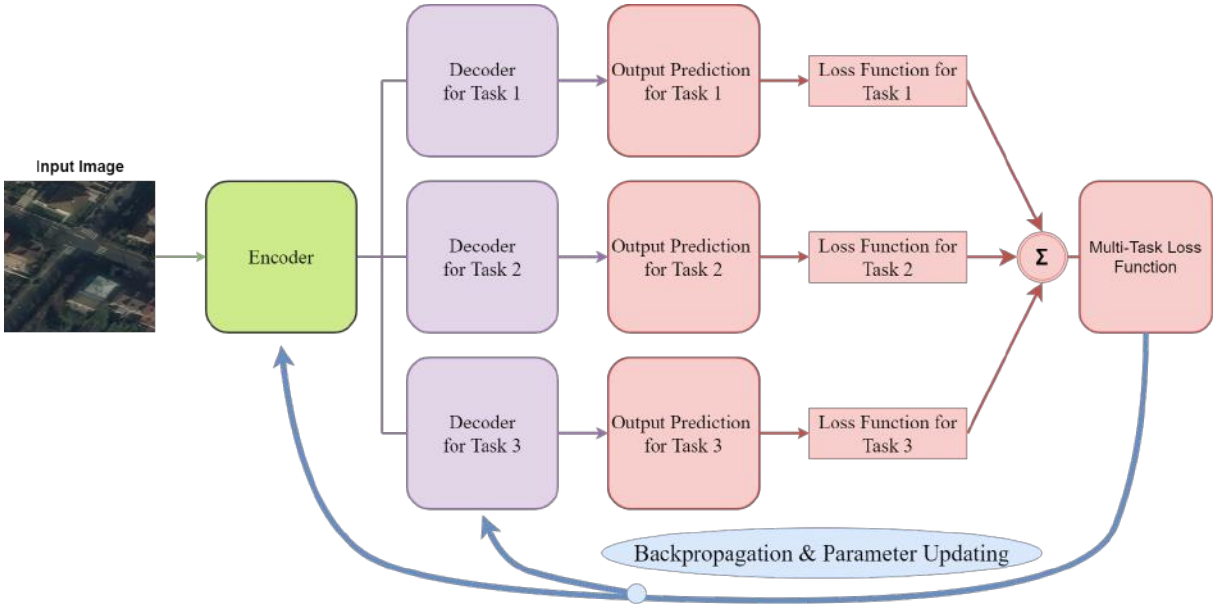


Figure 3.6: Illustration of the training procedure of an example of the proposed Multi-Task U-Net model with a ResNet34 Encoder solving 3 tasks simultaneously

3.2.1 Learning Tasks

According to relevant studies, when neural networks are used to detect the road network in remote sensing imagery, they mainly learn from geometric, photometric and texture features, such as pixel intensity values, road shapes and patterns. The features that are less obvious in an image to learn are the topological properties, like connectivity. Creating tasks that favor the preservation of topological properties, in combination with tasks that incorporate geometrical, photometrical and texture features, can cope with this weakness and boost performance. Therefore, I created four tasks related to the problem of road detection from the same ground truth graphs of the input dataset and trained numerous combinations of tasks with the proposed MTL model to investigate which combinations enhance the performance of the road detection task. The created tasks are presented bellow.

- Constant two meter wide Road Detection Learning task (TWO). The selection of the road width is a controversial topic. Depending on the landscape that an image dataset depicts, the selected road width can either be too wide to constrain a deep learning model into identifying useful features or too thin, making it inefficient to noise or pixel variety. A two meter road buffer seems to be suitable for the purposes of this project, since it satisfies the needs of every city included in the used dataset (described in section 4.1.1).

Figure 3.7: Constant two meter wide Road Detection Learning task. From Left to Right: satellite image, corresponding ground truth mask

- Road Orientation Learning task (ORIENTATION). The creation of this task is inspired by the work of Batra et al. [5]. Learning road orientation poses a connectivity constraint in the encoded representation. To create the orientation ground truth masks, the linear strings of the input vector road network were used, computing a unit vector between every consecutive pair of nodes and converting it into the polar coordinate system to acquire the orientation angles. The ground truth masks contain 36 orientation angles (a different orientation class for every 10 degrees turn) and a background class for the pixels that don't belong to the surface of the road.

Figure 3.8: Road Orientation Learning task. From Left to Right: satellite image, corresponding ground truth mask

- Road Intersection Learning task (INTERSECTION). Road junctions or intersections can improve the preservation of the road network's connectivity by forcing the network to learn topologically important locations. To create the ground truth masks for this task, a sliding window algorithm was used to identify which locations satisfy the requirement of an intersection, a dead-end or a perpendicular turn in a 3x3 kernel window.

Figure 3.9: Road Intersection Learning task. From Left to Right: satellite image, corresponding ground truth mask

- Gaussian Road Mask Learning task (GAUSSIAN). This task helps improve semantic segmentation accuracy at locations where the road boundaries are ambiguous (e.g, dirt roads blending into concrete roads) [7]. Ground truth masks are computed by a distance transformation from the vector reference road data.

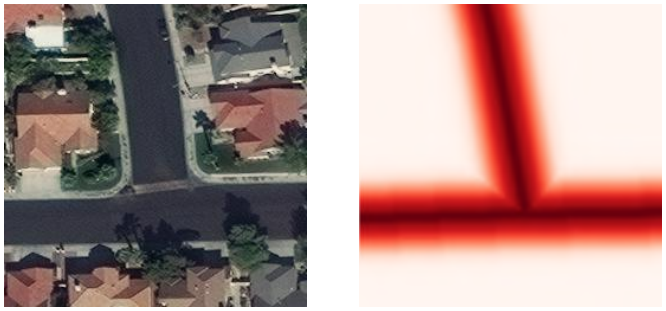


Figure 3.10: Gaussian Road Mask Learning task. From Left to Right: satellite image, corresponding ground truth mask

3.2.2 Multi-Task weighting

In this work, a simple uniform loss weighting scheme was applied. Due to the complexity of the topic of multi-task loss weighting and the time limitations of the current project, it wasn't attempted to incorporate state-of-the-art task weighting schemes, except for investigating the possibilities to improve the capabilities of the proposed model, as mentioned in Chapter 2.

4 Results and Analysis

To investigate the capabilities of the proposed method, MTL with specially designed learning tasks, extensive experiments to detect roads from remote sensing images have been conducted using the dataset created for the SpaceNet Challenge: Road Extraction and Routing [22], described in this Chapter. I compared my proposed method with the baseline single-task solving architecture. The experimental setup, results and analysis are illustrated in the following sections.

4.1 Experimental Settings

4.1.1 Dataset

In this study, the SpaceNet 3 Road Dataset is used, created for the SpaceNet Challenge: Road Extraction and Routing [22]. The dataset contains 2567 satellite images, collected by DigitalGlobe's satellite from four different cities: Paris, Las Vegas, Shanghai, and Khartoum. Their format is GeoTiff (16-bit), with size 1300 x 1300 pixels and ground resolution of 30cm/pixel. Their corresponding road network ground truth data is provided in the form of vector GeoJson file data (line-strings), representing the center line of roads. Furthermore, the road labels correspond to different road types (Motorway, Primary, Secondary, Tertiary, Residential, Unclassified, Cart Tracks) from the four cities. Each image may have multiple line-strings and each line-string consists of pixel coordinates X Y depicting road center line points in the 2D image plane, assuming top-left corner as the origin.

For this study, the dataset was divided into splits for training, validation and testing, after converting all images from 16-bit to 8-bit format. The division of the dataset was done in a way that each city would equally contribute to training, validation and testing (80% -10% -10% respectively). To artificially enrich the dataset with multiple images, initial images were clipped into samples with a size of 256 x 256 pixels with 50% overlap with their neighboring samples. The vector road network information was rasterized to create ground truth masks creating pairs with the image crops.

Figure 4.1: Example Images from the SpaceNet 3 Road Dataset [22]

4.1.2 Implementation details

All networks were trained using the Adam optimizer [40] with learning rate 0.001, which was reduced when performance would stop improving after two consecutive epochs by a factor of 0.1. The models were trained with a batch size of 15 images, the most that the available GPU could handle. Because the chosen dataset has a limited size and the ground truth vector linear strings contained many errors, the evaluation of the prediction during training using metrics such as Intersection over Union (IoU) is not totally representative with a model's capabilities, and as a result, fruitful callbacks were ineffectual. ResNet34 was initialized pre-trained on the ImageNet dataset. In previous studies, it has been proven that transfer learning can accelerate network convergence and improve network performance [34].

All experiments were performed on a cluster of Linux compute servers with a lot of processing power and memory for running large or long jobs, provided by the Intelligent Systems Department of Delft University of Technology. More specifically, all models were trained on an NVIDIA GeForce GTX 1080 Ti GPU paired with 11 GB GDDR5X memory. The deep learning framework that was used was Keras [15] using the TensorFlow backend [1].

4.1.3 Evaluation Methods

To evaluate the road detection results, the classical pixel-wise segmentation metrics with Precision (P), Recall (R), F1 Score (F1) and IoU were followed as well as the clDice metric (clDice) designed for topology preservation evaluation. A brief explanation of those metrics is given in the following sections. However, it is worth noting that visual inspection remains the main method to perform assessment and for that reason, dozens of images were randomly selected from every region to be examined against the predictions of every tested model.

Confusion Matrix

A confusion matrix calculates the classification performance of a classification model or a classifier with respect to a set of test data with known true values. It is a two-dimensional matrix, where one dimension represents the instances of the predicted class and the second dimension the instances of the actual or ground truth values [72]. Although the confusion matrix is not a performance measure itself, it is used by many performance measurements, and especially, as it is described in the following sections, its individual cell values.

n = 206	Predicted: NO	Predicted: YES	
Actual: NO	TN = 42	FP = 56	98
Actual: YES	FN = 8	TP = 100	108
	50	156	

Figure 4.2: An example of a two-class confusion matrix

Precision, Recall and F1 Score

The definitions of the associated terms within the Confusion Matrix that are used for the computation of the metrics of P, R and F1 are the following:
True Positives (TP): cases when both the prediction and the actual class are equal to 1 (True)
True Negatives (TN): cases when both the prediction and the actual class are equal to 0 (False)
False Positives (FP): cases when the actual class is 0 (False) but the prediction is 1 (True)
False Negatives (FN): cases when the actual class is 1 (True) but the prediction is 0 (False)

A "perfect" classifier should give 0 FP and 0 FN, but in real life there hasn't been a model that was 100% accurate all the times, under any circumstances.

P expresses the proportion of pixels that were classified as road and truly belong to the road class (their ground truth is also road) and can be computed as:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4.1}$$

On the other hand, R expresses the proportion of pixels that actually belong to the road class and were predicted as road class. It can be computed with:

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4.2}$$

It is obvious that if the aim is to minimize FN, R should be maximized and if the aim is to minimize the FP, P should be maximized. Usually, by increasing one, the other one suffers. This is the fundamental trade-off between precision and recall. When building a classifier, both measures should be considered and the correct balance between precision and recall should be determined according to the characteristics of the problem.

F1 is a metric that kind of represents both P and R with a single score. It is a balanced score metric because it utilizes the harmonic mean. As a consequence, if one number between precision and recall is really small, the F1 metric will get closer to the smaller value than the bigger one, giving a more representative score rather than what the arithmetic mean of precision and recall would give. The F1 score metric can be computed with the following formula:

$$\mathrm{F1} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}} \tag{4.3}$$

Intersection over Union

The IoU metric, also referred to as the Jaccard index, is an additional metric that quantifies the proportion of overlap between the prediction image and its corresponding ground truth image. Essentially, it measures the similarity between the two finite sample sets of prediction images and ground truth images, using the number of "common" pixels within the ground truth and the prediction images that describe the road surface divided by the total amount of pixels that describe the road in both images. It can be computed using the following formula:

$$\mathrm{IoU} = \frac{prediction \cap ground\ truth}{prediction \cup ground\ truth} \tag{4.4}$$

clDice metric

Accurate pixel-wise segmentation of network-like structures, like roads, is relevant to many fields of research. For such structures, topology or connectivity is their most important characteristic. In the paper [65], a novel similarity metric named centerline-in-mask-dice-coefficient or clDice was introduced, prooving to be a reliable evaluation metric for topological correctness. It is computed on the intersection of the segmentation masks and their morphological skeletons.

To be more specific, assuming there are two binary masks: a ground truth mask (GT) and its corresponding prediction mask (PR). In the first step of the computation, the skeletons S_{GT} and S_{PR} are created from GT and PR respectively. Next, the fraction of S_PR that falls withing GT is computed, which the authors called *Topology Precision* or $T_{prec}(S_{PR}, GT)$, and vice-a-versa, the *Topology Sensitivity* or $T_{sens}(S_{GT}, PR)$ and finally clDice is defined as:

$$clDice(GT, PR) = 2 * \frac{Tprec(S_{PR}, GT) * Tsens(S_{GT}, PR)}{Tprec(S_{PR}, GT) + Tsens(S_{GT}, PR)} \tag{4.5}$$

4.1.4 Loss Function

The task of road detection belongs to the category of unbalanced classification problems. In imbalanced classification problems, the distribution of examples across the known classes is unequal, as the name suggests. Usually roads occupy a very small part of the whole extent of an image, as roads are most often very thin objects in comparison to buildings, city squares, forests, etc. There are two common ways to cope with this issue. The first solution is to ”balance” the dataset by modifying what the network sees during training (e.g. feed the network only with images that have a higher road representation ratio than a chosen threshold). The second solution is the mindful selection of the loss function[20].

Semantic segmentation is usually practiced with a pixel-level cross-entropy loss, which compares the class prediction of each individual pixel with its ground truth and averages the result of the whole. Every pixel is separately and equally evaluated. In the task of road detection, the training dataset is heavily unbalanced, and the class considered as 'background' is trained a lot more times than the 'road' class, resulting in parameters that favor the 'background' class every time. To deal with this, I tested several common semantic segmentation loss functions and discovered that categorical cross-entropy loss function in combination with the IoU metric is the most beneficial solution for almost all the tasks tested. Other loss functions, such as Dice loss [51], Focal loss [45] or other variations, are shown in the next section in tables generated for quantitative assessment to assist the loss function selection procedure. Bellow, you can find a short description of the selected loss function.

The Cross-Entropy Loss is defined by the following formula:

$$CE = -\sum_{i}^{C} t_i \log(s_i) \tag{4.6}$$

,where t_i and s_i are the ground truth and the prediction result of class C respectively. According to the number of classes of each task, the Cross-Entropy Loss was converted to either Categorical Cross-Entropy (CCE) Loss with a Softmax activation function or Binary Cross-Entropy (BCE) Loss with a Sigmoid activation function. The formulas of both losses are presented bellow:

$$CCE = -log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right) \tag{4.7}$$

$$BCE = -\sum_{i=1}^{C'=2} t_i log(f(s_i)) = -t_1 log(f(s_1)) - (1 - t_1) log(1 - f(s_1)) \tag{4.8}$$

,where s_p is the prediction score of the positive class and $f(s_i)$ refers to the activation function used.

To combine one of these loss functions with the IoU presented above, I added the sum of $1 - IoU$ as a second term in the loss function, defining the final loss function as:

$$Loss = \lambda_1 * CE + \lambda_2 * (1 - IoU) \tag{4.9}$$

,where CE refers to the Cross-Entropy Loss used according to the task under optimization, IoU refers to the IoU , and λ_1, λ_2 refer to the weight parameters which was set equal to 1 for both instances. Due to the time limitation of the numerous tests performed, different loss weighting values were not tested.

4.2 Results and Analysis

4.2.1 Single-Task solving models

The first stage of the experimentation contains the training of each individual task on its own and the analysis of its accuracy. By doing so, it is easier to understand the limitations of each proposed task and decide how they can be combined in a MTL model.

The first task that was tested was the task of ORIENTATION. The results show (see Figure 4.3) that it can indeed be a useful task, since the predictions contain valuable information of the road network. In many cases, it would even be sufficient to train only for the task of ORIENTATION, because merging all the pixels from every orientation angle class in the prediction output would produce a fine segmentation result of the road network. Although the IoU and F1 scores in Table 4.1 show excellent performance, it is worth noting that these measures do not represent the actual capabilities of the corresponding model. Road Orientation Learning ground truth masks contain 37 classes (36 orientation angles and 1 class for background), but only a maximum of 7 or 8 orientation angle classes are usually present per image. The remaining classes have 0% representation in images, which means that if the prediction image does not contain them, it will be evaluated with a perfect score for these classes, leading to an increase of the metric. The efficiency of the model was therefore evaluated mainly using visual inspection of the prediction results. A second important note is that a trained model on the task of ORIENTATION learns to predict better orientation angles than what the ground truth masks of this task depict, which is a factor that also affects the final estimation of the model's accuracy. The limitation of this task is that it tends to fail in wide areas, where it is possible to expand into multiple different orientations.

Table 4.1: Performance of the Road Orientation Learning task with different loss functions on a small subset of the dataset. Bold letters represent the selected loss function. For more information about the loss function read 4.1.4

Loss function	**IoU Score**	**F1-Score**
CCE	0.926	0.930
CCE + IoU	0.889	0.895
CCE + Dice	0.855	0.862
Dice + Focal	0.923	0.925

The second task investigated was the task of GAUSSIAN. The results show (see Figure 4.4) that it can be a useful task as well. Although the metrics have very low scores, meaningful information useful for the task of road detection is expressed in the prediction masks. A model trained for the Gaussian Road Mask Learning task, learns how to predict output masks with a gradual convergence to road centerlines, failing to specifically predict the final classes that mainly belong to road centerlines (see Figure 4.4), leading to low evaluation metrics. Getting advantage of pixels that belong to classes closer to the road centerline can produce a fine image segmentation that includes the biggest portion of the road network. Weak predictions occur on locations where the scene depicted on an image is mostly covered by concrete or wide drivable surfaces, like parking lots, public squares or highway intersections. Another limitation

Table 4.2: Performance of the Gaussian Road Mask Learning task with different loss functions on a small subset of the dataset. Bold letters represent the selected loss function. For more information about the loss function read 4.1.4

Loss function	IoU Score	F1-Score
CCE	0.166	0.167
CCE + Dice	0.179	0.190
CCE + IoU	0.180	0.191
Dice + Focal	0.194	0.222

of this task occurs when roads are intersecting in an ambiguous or unique way (that has never been seen as an example inside the dataset). Under these circumstances, the prediction output stops sharply, because there are numerous ways to produce an output, and the model cannot decide which one is the optimal.

The third task investigated was the task of INTERSECTION. The results show (see Figure 4.5) that they contain meaningful information relevant to road detection. The biggest advantage of this task is that the model learns to detect reliable seed points on the image boundaries in almost every situation. On the other hand, it doesn't predict intersection points very accurately. One reason for that is that the ground truth masks are not suitable to describe what the model needs to learn. They are constructed from the provided road vector linear strings using a sliding window algorithm, aiming to detect points that satisfy the topology of crossroads, dead-ends or perpendicular turns. Unfortunately, the dataset contains errors, and therefore, there are many ground truth masks that also contain ambiguities or incorrect intersection nodes. Results of poor quality occur on locations with big openings or wide road areas where the location of an intersection is uncertain.

Table 4.3: Performance of the Road Intersection Learning task with different loss functions on a small subset of the dataset. Bold letters represent the selected loss function. For more information about the loss function read 4.1.4

Loss function	IoU Score	F1-Score
CCE	0.718	0.770
CCE + Dice	0.724	0.782
CCE + IoU	0.734	0.790
Dice + Focal	0.733	0.790

The last task investigated was the task of TWO, where the road is depicted with a constant rasterized line of two meter width. This task is the primary task that is attempted to be optimized by being jointly trained with related sub-tasks to improve its final segmentation output. According to the results (see Figure 4.6), when it is trained on its own it suffers from common errors and limitations of the road detection task, as described in Chapter 2. Isolated road segments, missing information, inaccurate geometries and false predictions are shown in the prediction results.

Table 4.4: Performance of the Road Detection Learning task with different loss functions on a small subset of the dataset. Bold letters represent the selected loss function. For more information about the loss function read 4.1.4

Loss function	IoU Score	F1-Score
CCE	0.700	0.759
CCE + Dice	0.719	0.784
CCE + IoU	0.723	0.787
Dice + Focal	0.715	0.782

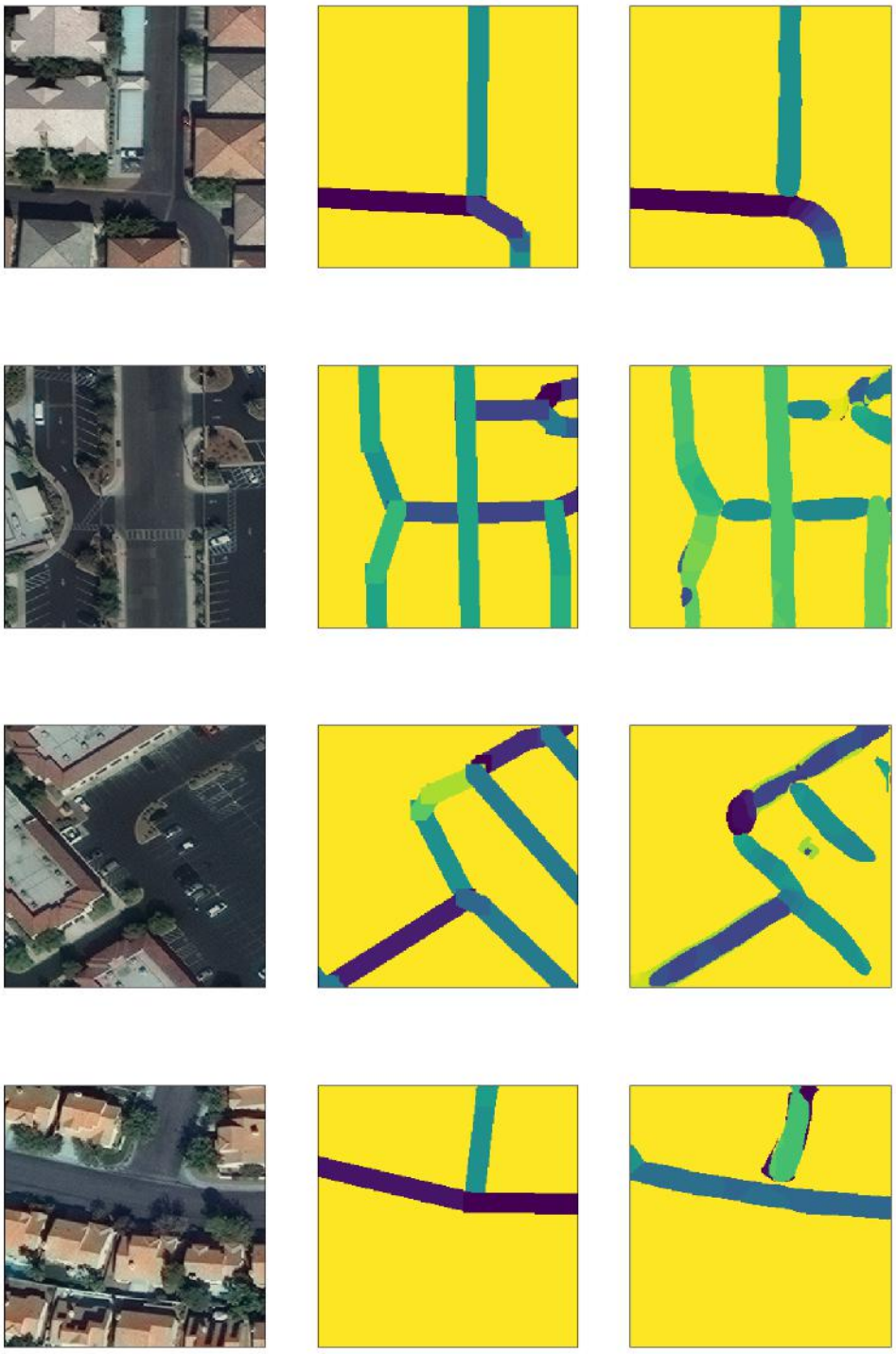


Figure 4.3: Example results of the Road Orientation Learning task. From Left to Right: original image, ground truth image, prediction image

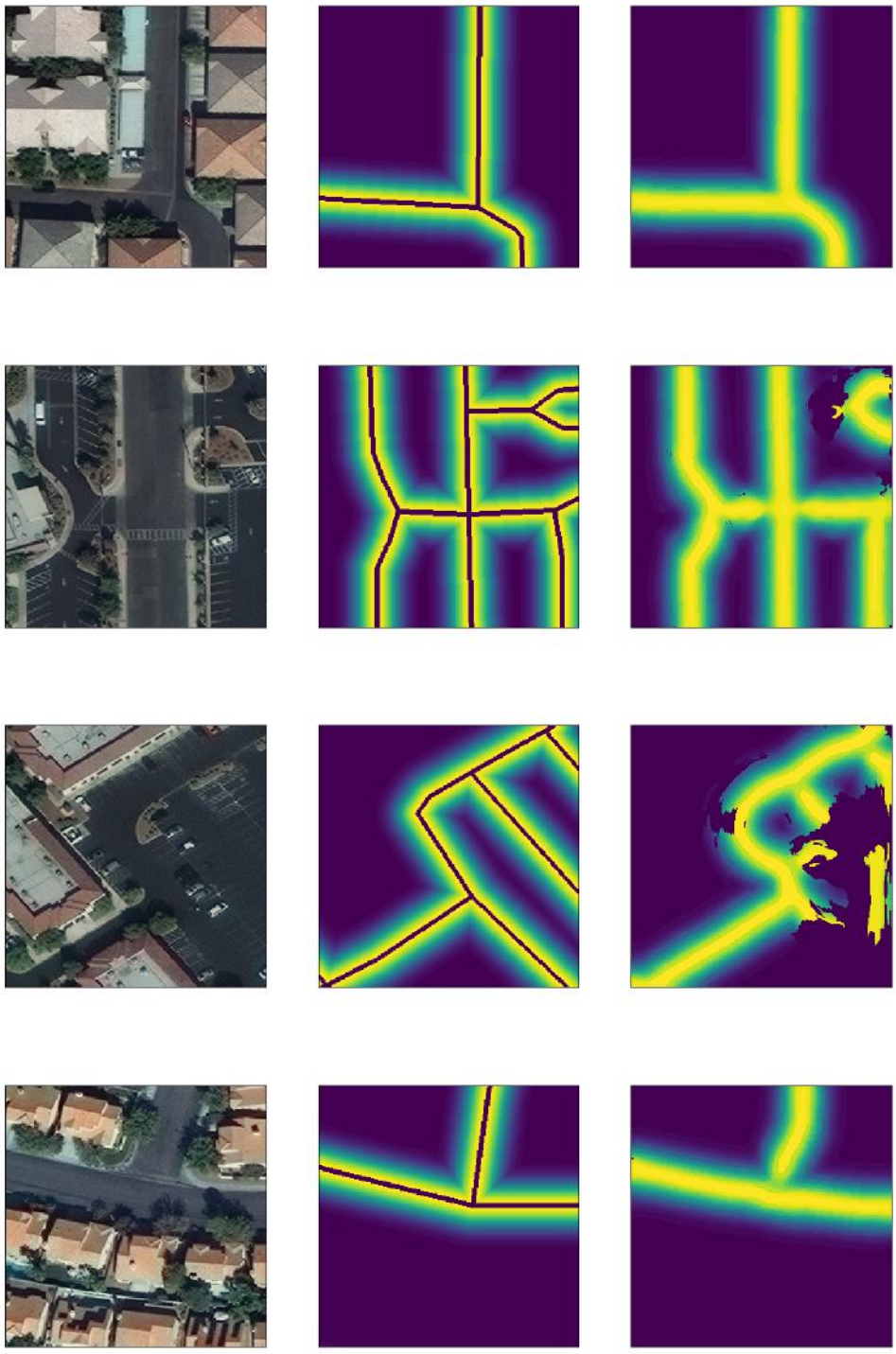


Figure 4.4: Example results of the Road Gaussian Mask Learning task. From Left to Right: original image, ground truth image, prediction image

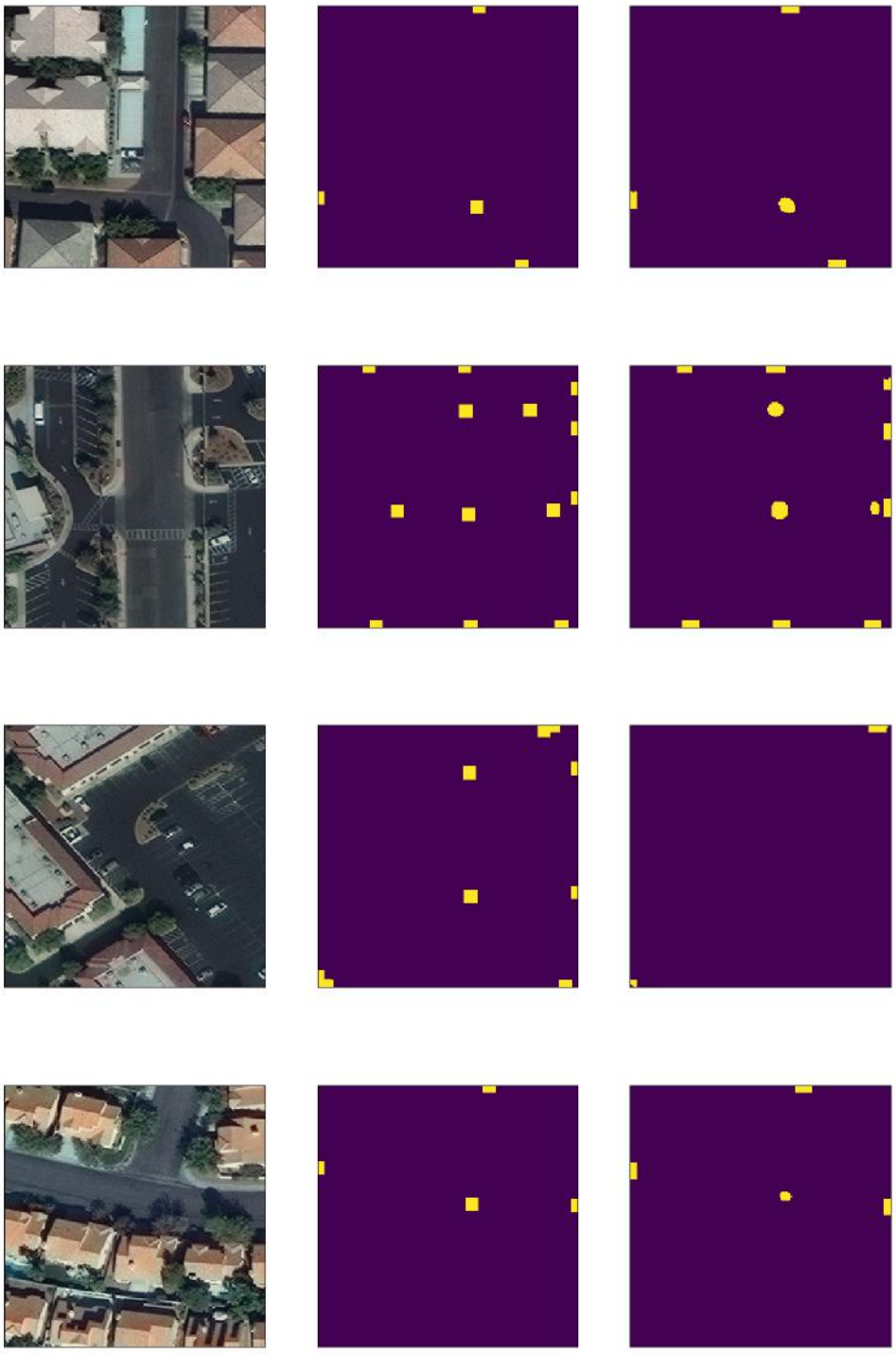


Figure 4.5: Example results of the Road Intersection Learning task. From Left to Right: original image, ground truth image, prediction image

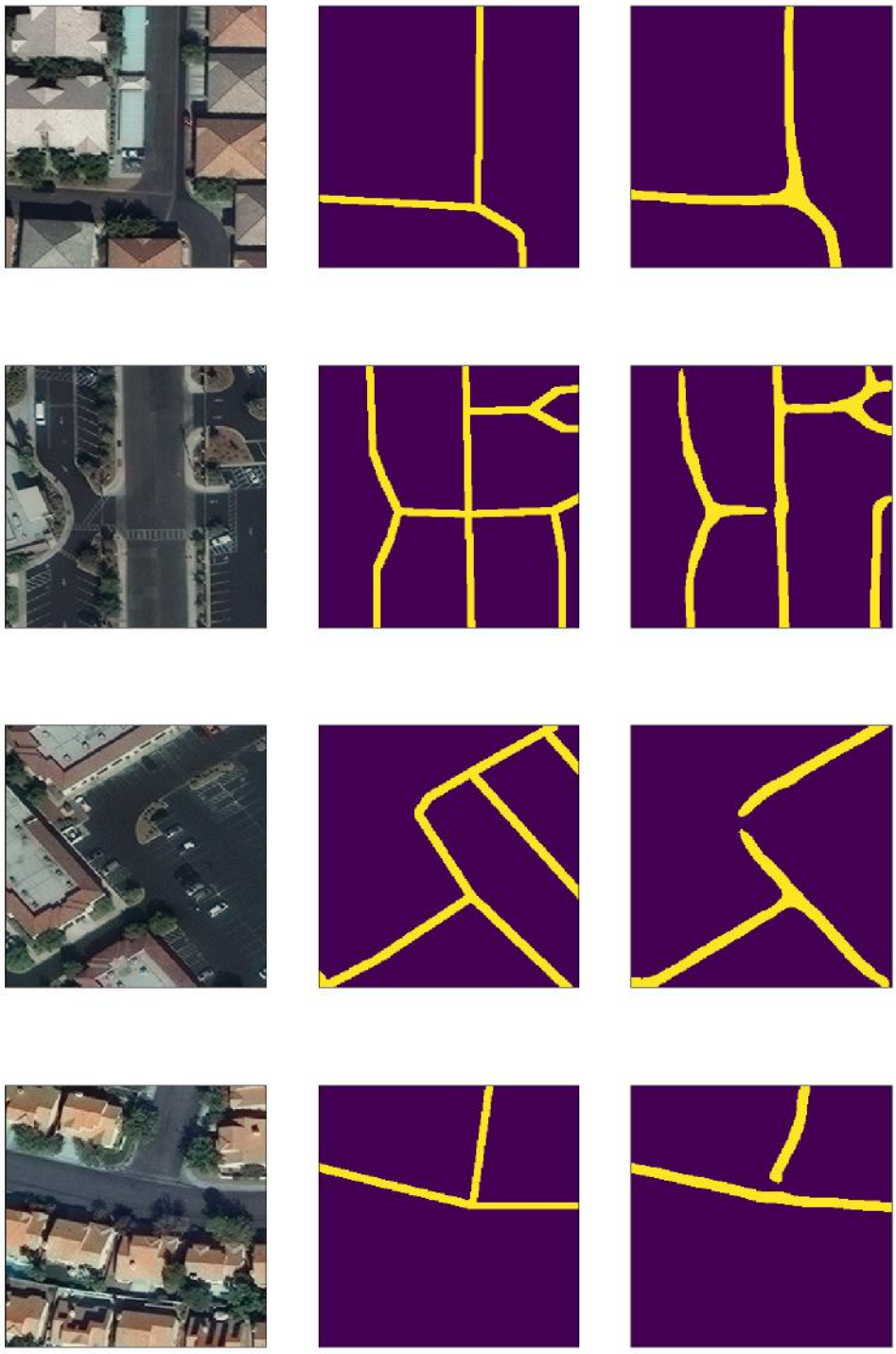


Figure 4.6: Example results of the Constant two meter wide Road Detection Learning task. From Left to Right: original image, ground truth image, prediction image

4.2.2 Comparison with Proposed Multi-Task solving models

To compare the performance of the proposed method with the performance of the single-task solving model, all combinations of tasks were trained with the proposed MTL architecture using the same hyper-parameters and loss functions as the best performing single-task solving model. The evaluation metric scores of each model are shown in table 4.5. The proposed MTL model that combines three tasks, namely the TWO, the INTERSECTION and ORIENTATION, performs better than the baseline single-tasked model solving for road detection. Both in terms of image segmentation and topology preservation, the proposed MTL model outperforms the single-task model. In the same table, almost all the other combinations of MTL models perform either on par with the single-task model or have worse performance.

Both types of models, single-tasked or MTL models, favor from the symmetric 'U' shape architecture and the residual blocks, effectively learning how to output acceptable predictions in most of the landscapes that don't include ambiguous or unique combinations of road objects. The MTL method with the highest metric scores provides better predictions on intersections, preserving the road network's connectivity, increasing the clDice by almost 3 %, and usually, provides also a more complete road surface segmentation result. An improved segmentation and topology output translates into a better road detection algorithm, with a smaller need for output refinement post-processing procedures.

The limitation of the proposed methodology is that it fails to predict or produces less accurate predictions than the single-task trained model, in cases where the auxiliary tasks cannot provide a realistic solution. For example, in the case of the model with the highest metric scores, whenever the prediction fails to provide accurate intersection locations or accurate orientation angles, the road detection task also tends to provide weak predictions. Another limitation of the proposed methodology is that it requires more time to train than the single-task model, since the MTL model consists of more parameters than the single-task model. In addition, modifying a MTL network is twice as hard as modifying a single-task solving network. In the proposed model, long skip connections had to be connected from the encoder path of the network to each decoder path of every participating task. A more complex modification will increase the complexity factor during implementation or design.

Table 4.5 contains the proof that the MTL method works, even without a sophisticated task-weighting scheme or a specially designed representation sharing network. Relevant literature proved that with the addition of an advanced enhancement, these results could achieve even higher rates (see Chapter 2). An additional benefit of the proposed methodology is the fact that related tasks to the task of TWO also managed to increase their performance by being trained in parallel with other tasks. The increase of their performance is highlighted in the tables 4.6, 4.8 and 4.7.

Table 4.5: Assessment of the Road Detection Learning task. Comparative evaluation of the proposed Multi-Task Learning models and the baseline Single-Task Learning model. The results are computed using all the images in the test set. A higher value indicates a better performance.

TWO	INTERSECTION	ORIENTATION	GAUSSIAN	Mean IoU	Mean F1	Mean clDice
✓	-	-	-	0.684	0.732	0.671
✓	✓	-	-	0.677	0.725	0.661
✓	-	-	✓	0.679	0.726	0.661
✓	-	✓	-	0.680	0.729	0.676
✓	✓	✓	-	**0.691**	**0.740**	**0.698**
✓	✓	-	✓	0.681	0.727	0.657
✓	-	✓	✓	0.668	0.712	0.621
✓	✓	✓	✓	0.673	0.716	0.621

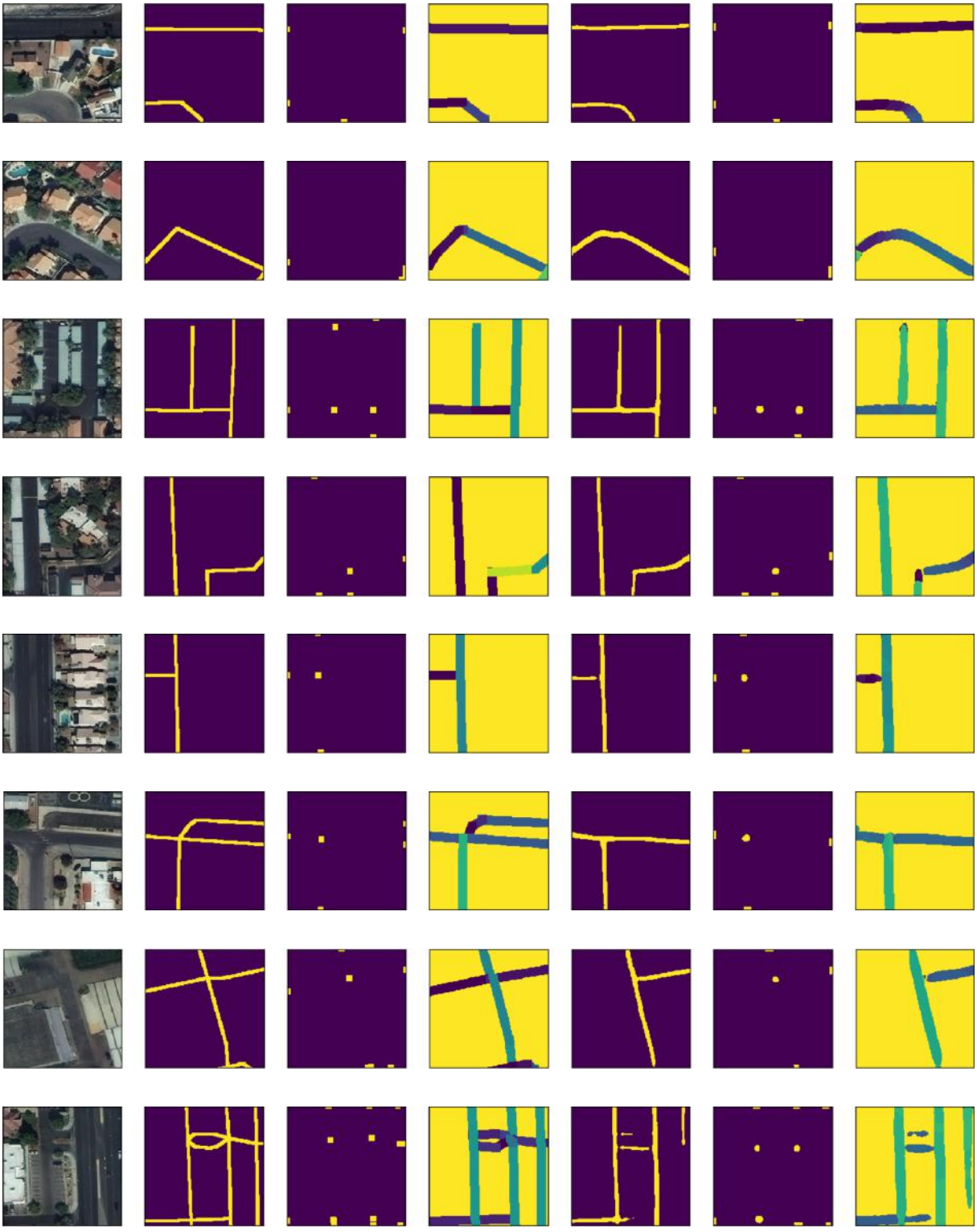


Figure 4.7: Example results of the best-performing proposed Multi-Task Learning model. From Left to Right: original image, ground truth image of Road Detection Learning task, ground truth image of Road Intersection Learning task, ground truth image of Road Orientation Learning task, prediction image of Road Detection Learning task, prediction image of Road Intersection Learning task, prediction image of Road Orientation Learning task.

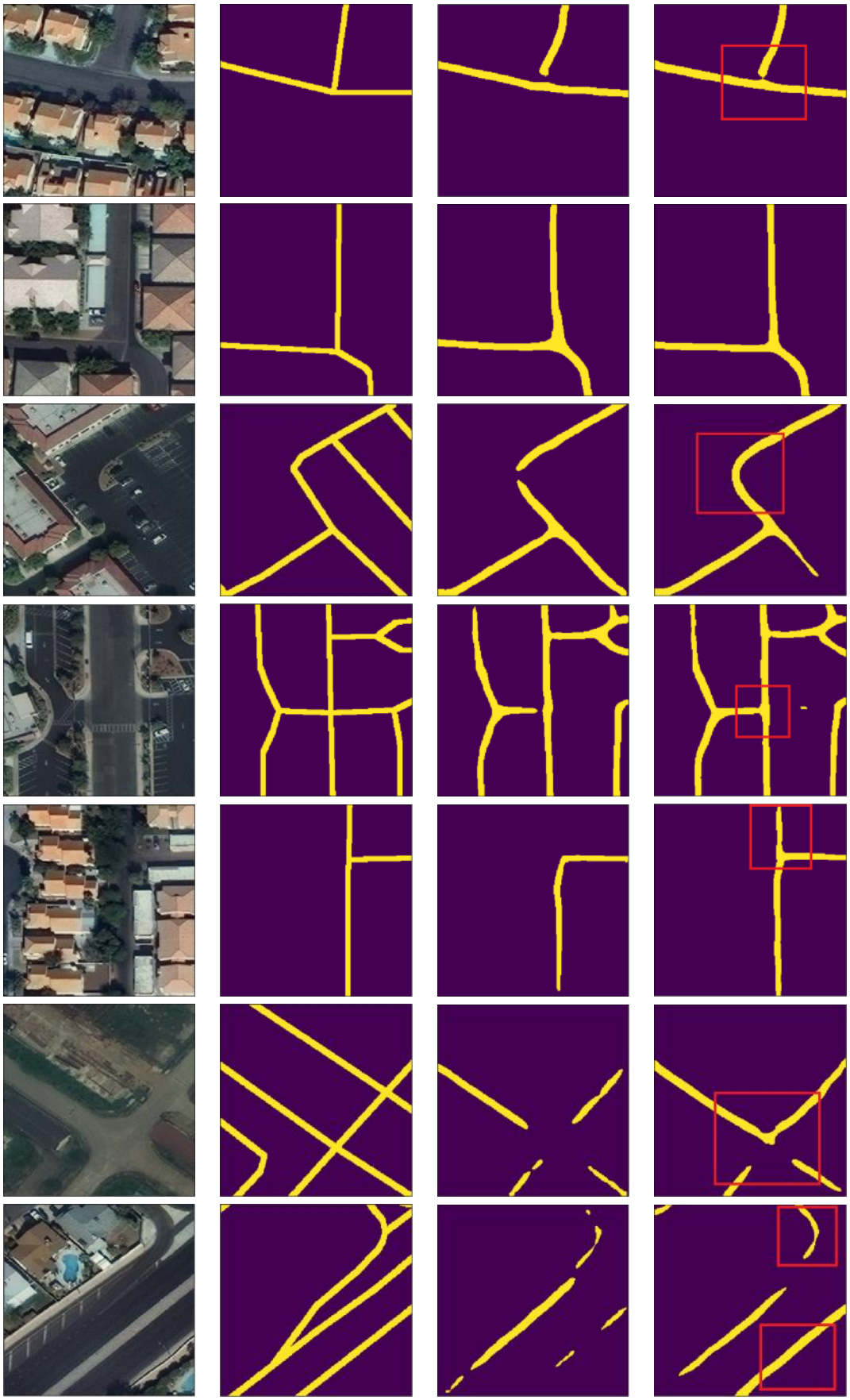


Figure 4.8: Visual inspection of the proposed model's performance. From left to right: satellite image, ground truth mask, single-task Road Detection Prediction, Multi-Task Road Detection Prediction

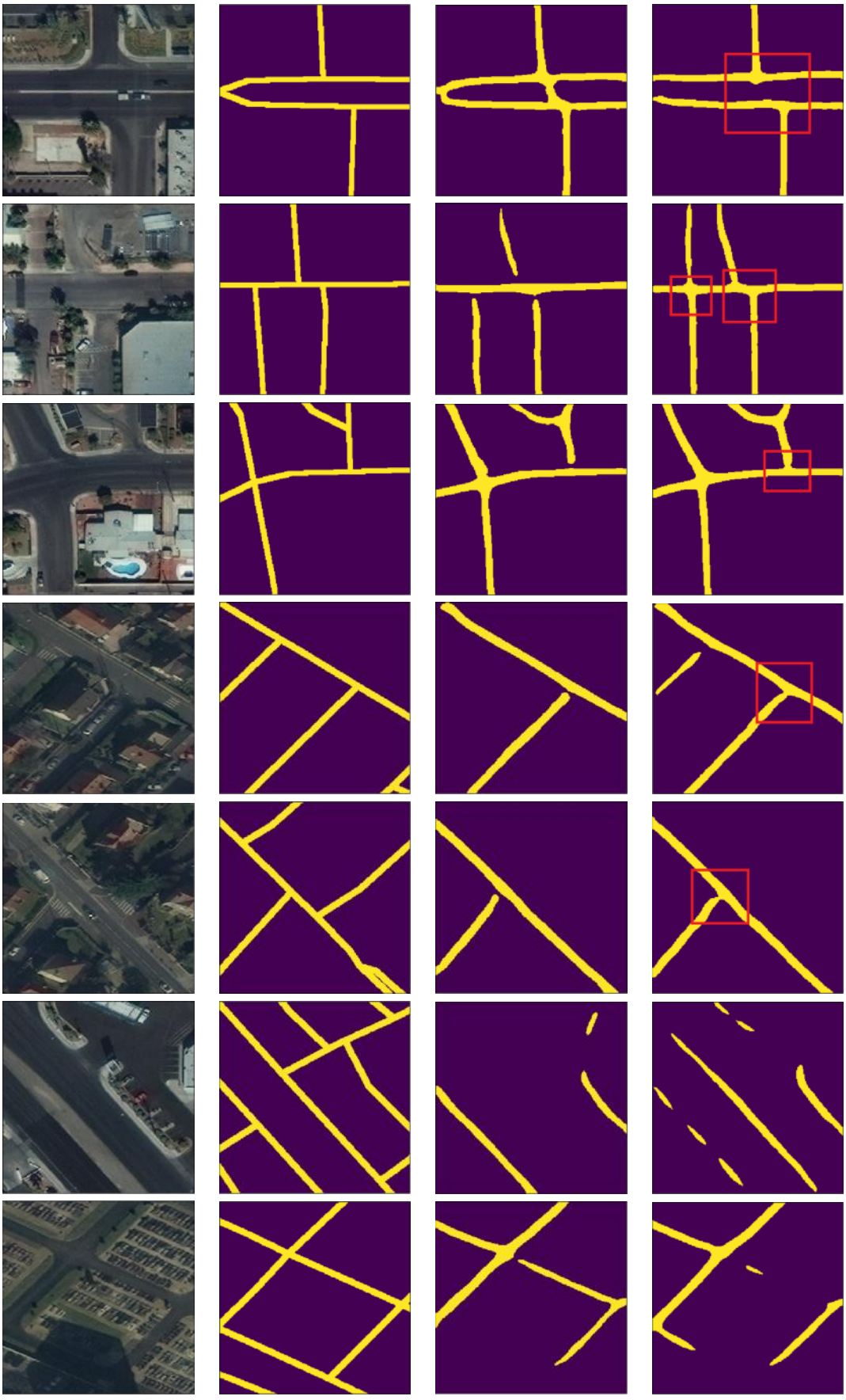


Figure 4.9: Visual inspection of the proposed model's performance. From left to right: satellite image, ground truth mask, single-task Road Detection Prediction, Multi-Task Road Detection Prediction

Table 4.6: Assessment of the Gaussian Road Mask Learning task. Comparative evaluation of the proposed Multi-Task Learning models and the baseline Single-Task Learning model. The results are computed using all the images in the test set. A higher value indicates a better performance.

ORIENTATION	TWO	GAUSSIAN	INTERSECTION	**Mean IoU Score**	**Mean F1 Score**
✓	-	-	-	0.891	0.896
✓	✓	-	-	0.680	0.729
✓	✓	-	✓	0.886	0.890
✓	✓	✓	-	0.894	0.898
✓	✓	✓	✓	**0.896**	**0.900**

Table 4.7: Assessment of the Road Orientation Learning task. Comparative evaluation of the proposed Multi-Task Learning models and the baseline Single-Task Learning model. The results are computed using all the images in the test set. A higher value indicates a better performance.

GAUSSIAN	TWO	ORIENTATION	INTERSECTION	**Mean IoU Score**	**Mean F1 Score**
✓	-	-	-	0.681	0.720
✓	✓	-	-	0.687	0.728
✓	✓	✓	-	0.691	0.729
✓	✓	-	✓	**0.697**	**0.739**
✓	✓	✓	✓	0.601	0.603

Table 4.8: Assessment of the Road Intersection Learning task. Comparative evaluation of the proposed Multi-Task Learning models and the baseline Single-Task Learning model. The results are computed using all the images in the test set. A higher value indicates a better performance.

INTERSECTION	TWO	ORIENTATION	GAUSSIAN	**Mean IoU Score**	**Mean F1 Score**
✓	-	-	-	0.681	0.720
✓	✓	-	-	0.687	0.728
✓	✓	-	✓	0.691	0.729
✓	✓	✓	-	**0.697**	**0.739**
✓	✓	✓	✓	0.601	0.603

5 Conclusions

In this project, I proposed a novel deep learning method to solve the task of road detection. Essentially, I proposed a MTL U-Net variation model for image semantic segmentation proposing specially designed auxiliary learning tasks. Jointly training multiple related tasks, combining both the capabilities of the U-Net model with the capabilities of the MTL paradigm, lead to an improved accuracy of road detection. Extensive experiments were conducted based on a publicly available dataset and the results demonstrated that my proposed method achieved better performance on all the evaluation metrics against the baseline single-task solving model. Multiple image examples depict its superiority both in terms of image segmentation and topology preservation. An additional finding is that MTL also affected the performance of the proposed auxiliary related tasks, which were also improved after joint training. The analysis of the experiments showed that the model can benefit from a more reliable input dataset and a task-weighting scheme and provide even better output predictions.

5.1 Discussion

In the following paragraphs, I review the research questions defined in Section 1.3. A short answer is given for each question, supported by experimental results presented in Chapter 4.

a. Can prior knowledge be incorporated as a constraint into a deep learning model? If yes, how?

It is possible to incorporate prior knowledge as a constraint (or multiple constraints) into a deep learning model by enforcing it to solve tasks that preserve a desired property or element. In the present study, multiple experiments were held to examine the performance of every proposed task related to the problem of road detection by solving it individually. The results of the experiments showed that the goal of the task determines the properties of the target object that will be kept. For example, the task of ORIENTATION verified that it favors the preservation of the road's connectivity. The task of INTERSECTION had a similar effect.

b. Can prior knowledge improve road detection?

Yes. As demonstrated in Chapter 4, a neural network was trained to maintain different road properties by being jointly trained on multiple tasks (task of TWO, ORIENTATION, task of INTERSECTION, etc.). Each task urged the model on modifying its parameters to improve its predictions, resulting in a selection of weights that is capable on solving multiple problems at once, or otherwise, generalize better in all tasks simultaneously. If one task aims on preserving property A and a second task aims on preserving property B, a model capable on solving both tasks at once is capable of maintaining both properties.

c. What are the limitations of a model that combines concepts of different models into one, unified model?

The biggest limitation of a neural network that combines concepts of multiple and/or different models is the increased level of complexity. Training a neural network requires time and increasing its performance requires experience and complete understanding of its functionality. The more complex a neural network is, the more difficult it is to fine-tune it. In the present case, a more complex model required more hours for implementation, training and fine-tuning.

d. To which extent is it possible to utilize road properties to improve road detection from remote sensing imagery using deep learning techniques?

The present study shows that by incorporating road properties as constraints during a neural network's training, a better solution in the problem of road detection can be provided. However, it does not eliminate every error or limitation of the road detection problem. According to relevant literature, the accuracy of the proposed network can be further enhanced by integrating modules that increase the performance of MTL neural networks, which couldn't be applied due to time limitations of the project. However, it is important to clarify that a constraint cannot guarantee a flawless preservation of a desired property. Bibliography shows that a property-preserving loss function or a specially-designed neural network architecture (or a combination of all the above) can be more effective.

5.2 Future Work

An experiment using the proposed model usually requires more than two days to finish a complete run. Therefore, due to lack of time, many ideas and tests were not implemented. In this section, I propose different mechanisms and approaches for future work, that according to my experience, point towards the right direction to improve the current methodology.

For example, one suggestion is to investigate if there are any other related learning tasks to the task of TWO, that could be used to assist the training procedure and further improve the model's performance, like the degree of a road's connectivity proposed by Sun et al [69]. Apart from further exploiting the input dataset and creating new learning tasks, it would be valuable to understand if different combinations of related tasks are more beneficial for a specific landscape. Moreover, as stated in the thesis, the selection of a constant road width value is a controversial topic. As a result, it would be interesting to investigate the effect of different road widths in the prediction result.

In the current project, training images had a size of 256x256 pixels. Different input image sizes might affect the performance of the network, as the receptive field of the model would be enlarged. Related to the previous suggestion, it would also be more beneficial to enrich the dataset using more data augmentation techniques, such as image rotations, flips and bigger overlap percentages. Although it is not considered as a model's enhancement, test time augmentation could further improve the final segmentation result.

MTL systems are complex and difficult to implement. Related literature has proved that it is not rare to use MTL and end up with worse results than what a single-task solving model can achieve, due to "competing" learning tasks, unweighted loss functions or learning tasks or even due to the inability of a neural network architecture to capture the necessary representations and features. The integration of a task-weighting or loss function weighting scheme (or a task-priority training method) that optimizes the MTL mechanism could provide better generalization than what the proposed method achieved. Accompanied with the previous proposal, an investigation of whether other neural network architectures are more suitable to make good use of auxiliary tasks for road network detection would be advantageous as well.

Lastly, if the goal is to only preserve the road network's connectivity, a methodology completely transferred to the vector domain should be preferred. For example, a procedural model that receives the outputs of the proposed model as input and provides a vector road network by looking for "confident" junctions and road segments that satisfy adjacency rules, connecting them through a shortest path algorithm could maximize connectivity, resulting in a vector graph of the road network. Inspiration for this idea came from the work of Marcos et al. [84]

A Reproducibility self-assessment

A.1 Marks for each of the criteria

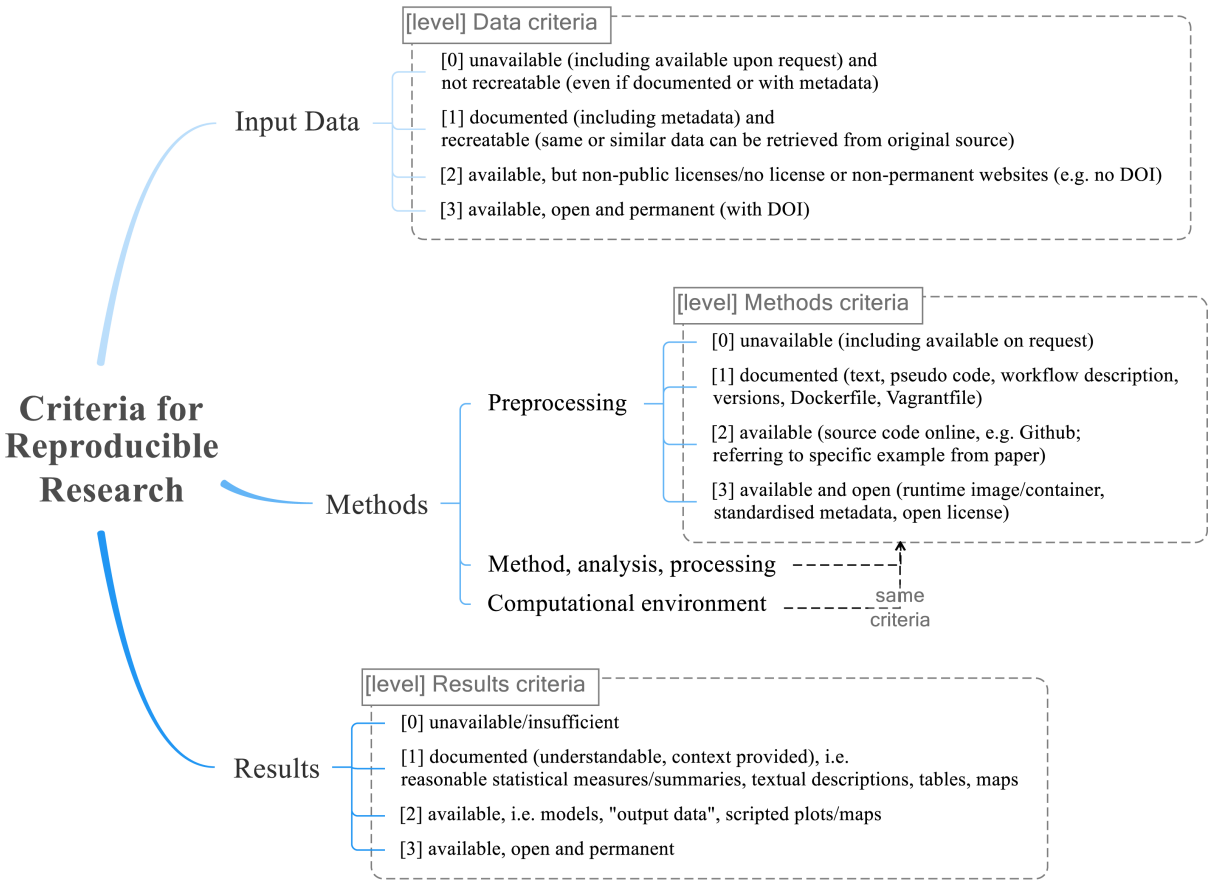


Figure A.1: Reproducibility criteria to be assessed.

The Reproducibility scores of my research project, according to the criteria chosen above, are shown in Table A.1.

Table A.1: Assessment of Reproducibility criteria

Criterion	Score
Input Data	3
Preprocessing	2
Methods	2
Computational environment	2
Results	2

A.2 Self-reflection

All my project is hosted on the following Github repository page: https://github.com/ntelo007/road_detection_mtl.git. I used the publicly available SpaceNet 3 Dataset, which is a corpus of commercial satellite imagery and labeled training data offered to be used for machine learning research. It is currently hosted as an Amazon Web Services (AWS) Public Dataset. All the scripts used for data preprocessing, model creation, training, and testing are well-documented with easy-to-follow instructions for execution. The Python virtual environment used can also be found inside the repository, containing its dependencies. Apart from that, a requirements.txt file is also provided, specifying what python packages are required to run the project. Example results are given to describe the capabilities of the proposed methodology. Finally, weights of the trained model are also offered to avoid re-training the model and save up time.

B Multi-Task Learning Models

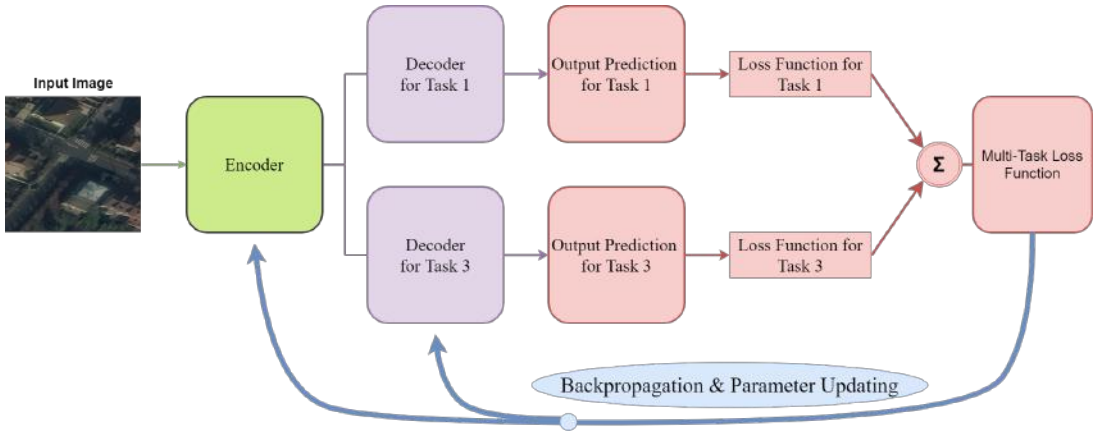


Figure B.1: Illustration of the training procedure of an example of the proposed Multi-Task U-Net model with a ResNet34 Encoder solving 2 tasks simultaneously

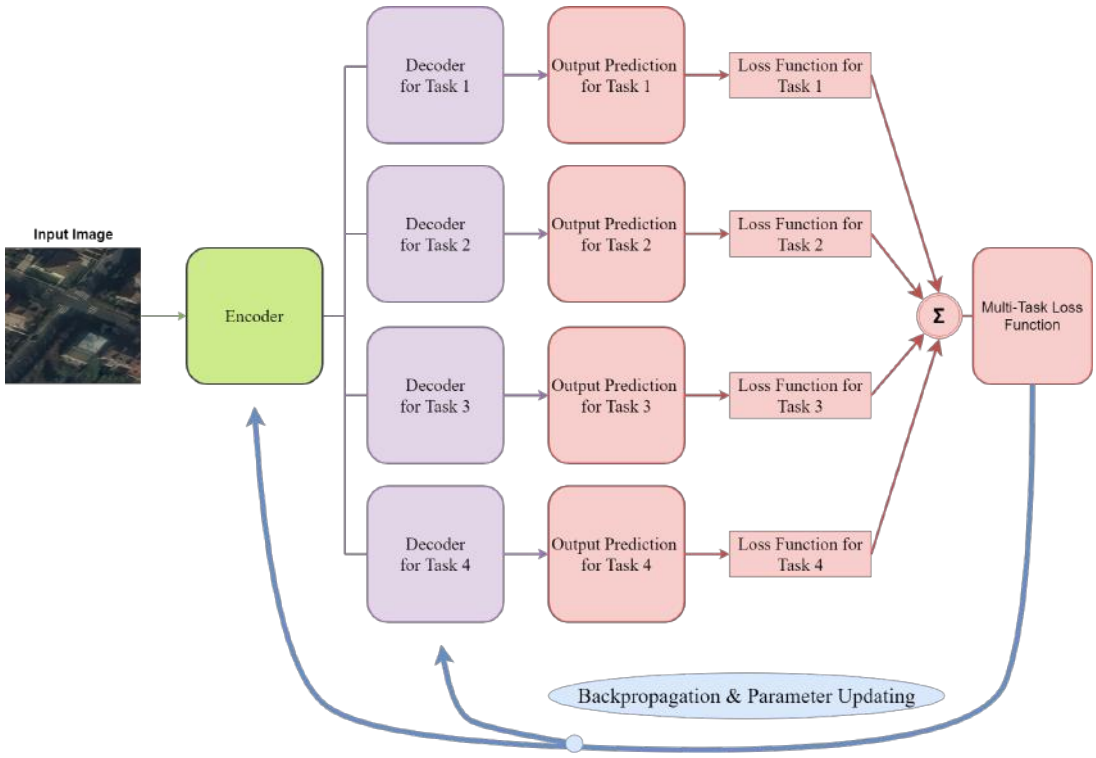


Figure B.2: Illustration of the training procedure of an example of the proposed Multi-Task U-Net model with a ResNet34 Encoder solving 4 tasks simultaneously

C Additional Results

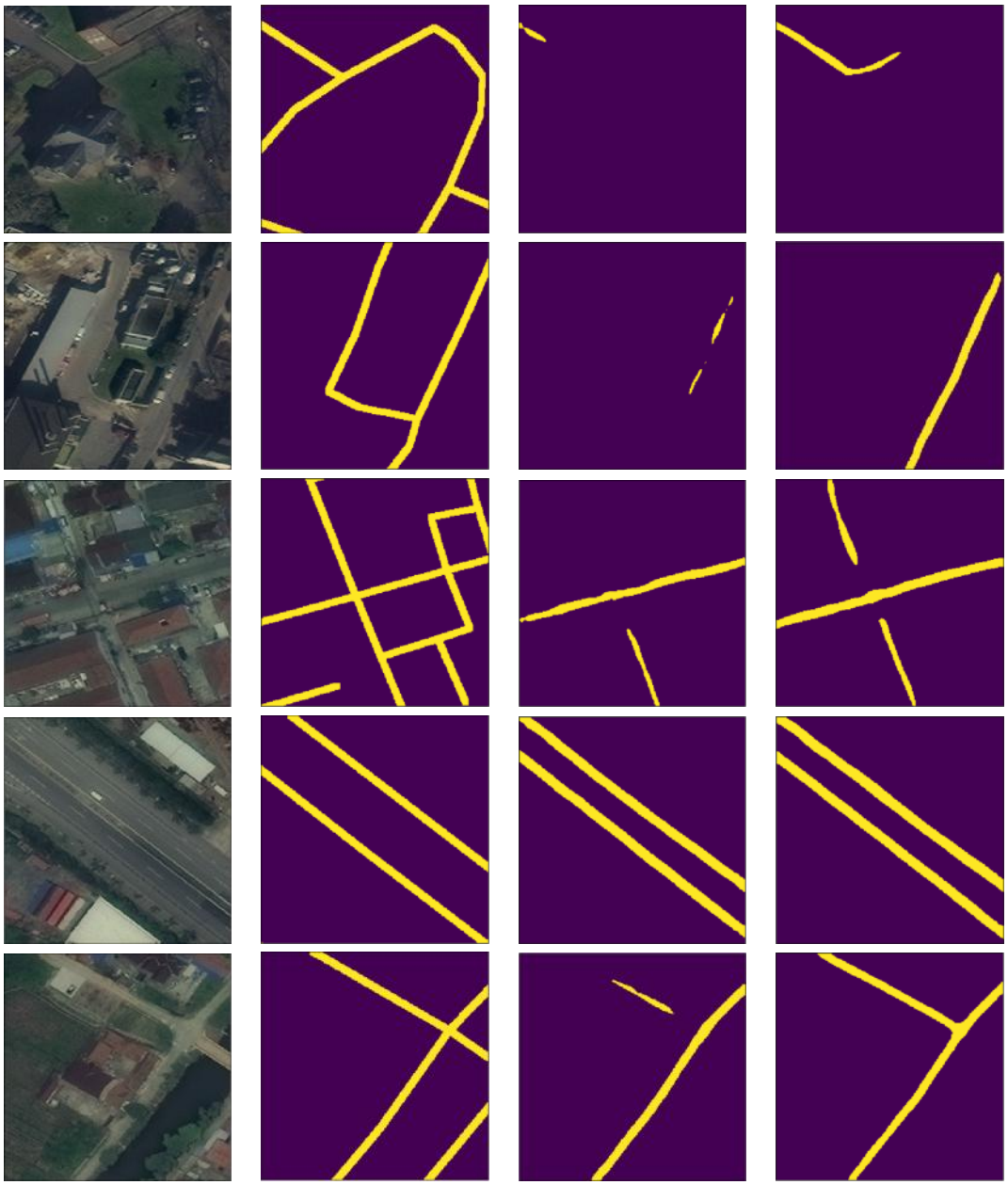


Figure C.1: Additional results for visual inspection of the proposed model's performance. From left to right: satellite image, ground truth mask, single-task learning model prediction, Multi-Task learning model prediction

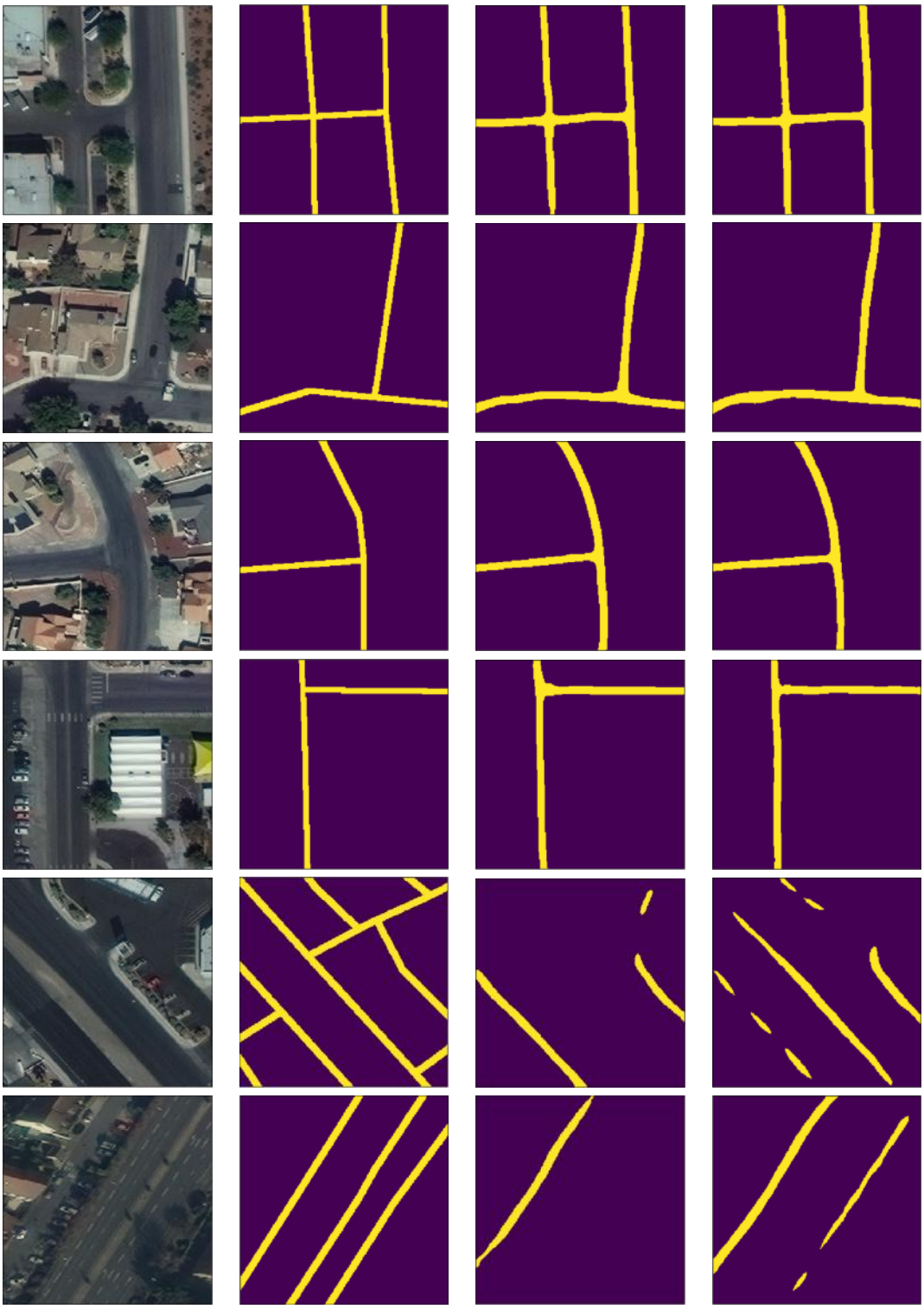


Figure C.2: Additional results for visual inspection of the proposed model's performance. From left to right: satellite image, ground truth mask, single-task learning model prediction, Multi-Task learning model prediction

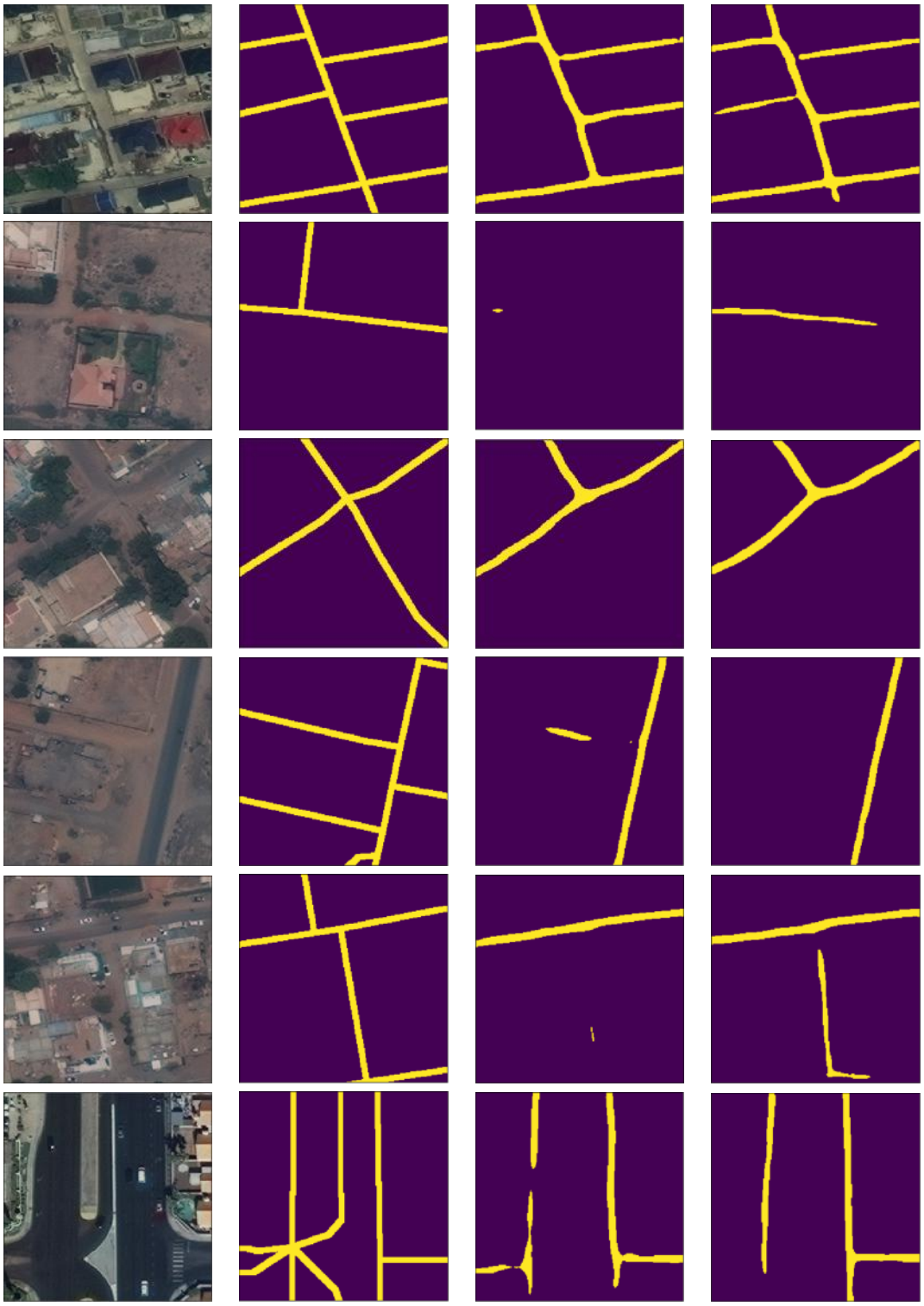


Figure C.3: Additional results for visual inspection of the proposed model's performance. From left to right: satellite image, ground truth mask, single-task learning model prediction, Multi-Task learning model prediction

Figure C.4: Additional results for visual inspection of the proposed model's performance. From left to right: satellite image, ground truth mask, single-task learning model prediction, Multi-Task learning model prediction

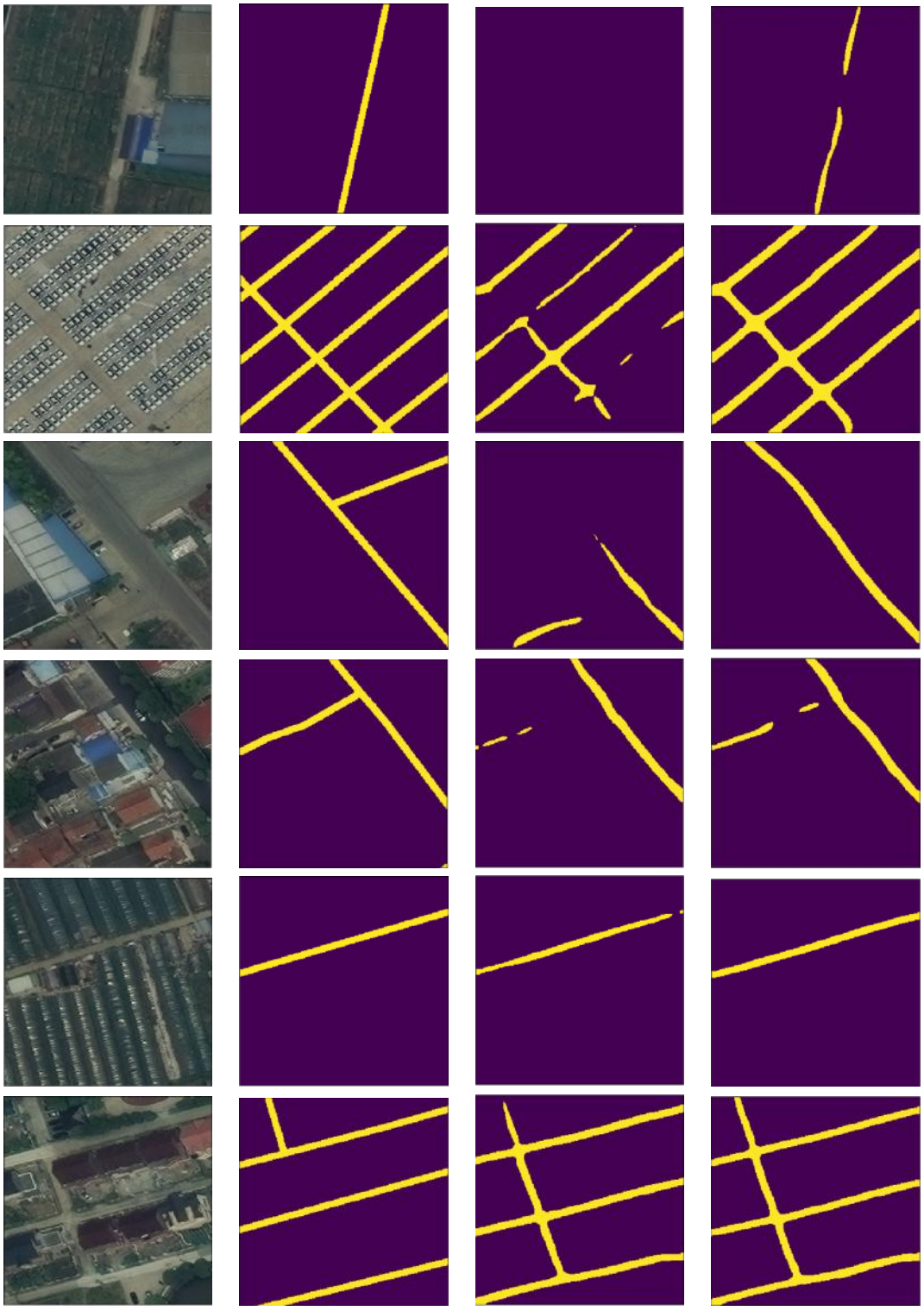


Figure C.5: Additional results for visual inspection of the proposed model's performance. From left to right: satellite image, ground truth mask, single-task learning model prediction, Multi-Task learning model prediction

Bibliography

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[2] Alshehhi, R., Marpu, P. R., Woon, W. L., and Mura, M. D. (2017). Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 130:139 – 149.

[3] Bajcsy, R. and Tavakoli, M. (1976). Computer recognition of roads from satellite pictures. *IEEE Transactions on Systems, Man, and Cybernetics*, 6:623–637.

[4] Bastani, F., He, S., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Madden, S., and DeWitt, D. (2018). Roadtracer: Automatic extraction of road networks from aerial images. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[5] Batra, A., Singh, S., Pang, G., Basu, S., Jawahar, C., and Paluri, M. (2019). Improved road connectivity by joint learning of orientation and segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] Baxter, J. (1997). A bayesian/information theoretic model of learning to learn viamultiple task sampling. *Mach. Learn.*, 28(1):7–39.

[7] Bischke, B., Helber, P., Folz, J., Borth, D., and Dengel, A. (2019). Multi-task learning for segmentation of building footprints with deep neural networks. *2019 IEEE International Conference on Image Processing (ICIP)*.

[8] Buslaev, A., Seferbekov, S., Iglovikov, V., and Shvets, A. (2018). Fully convolutional network for automatic road extraction from satellite imagery. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 197–1973.

[9] Caruana, R. (1993). Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Morgan Kaufmann.

[10] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915.

[11] Chen, L., Papandreou, G., Schroff, F., and Adam, H. (2017a). Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587.

[12] Chen, L., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611.

[13] Chen, Z., Badrinarayanan, V., Lee, C.-Y., and Rabinovich, A. (2017b). Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks.

[14] Cheng, G., Wang, Y., Xu, S., Wang, H., Xiang, S., and Pan, C. (2017). Automatic Road Detection and Centerline Extraction via Cascaded End-to-End Convolutional Neural Network. *IEEE Transactions on Geoscience and Remote Sensing*, 55(6):3322–3337.

[15] Chollet, F. et al. (2015). Keras. https://keras.io.

[16] Cipolla, R., Gal, Y., and Kendall, A. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[17] Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA. Association for Computing Machinery.

[18] Das, S., Mirnalinee, T. T., and Varghese, K. (2011). Use of salient features for the design of a multistage framework to extract roads from high-resolution multispectral satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, 49(10):3906–3931.

[19] Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., and Raska, R. (2018). Deepglobe 2018: A challenge to parse the earth through satellite images. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

[20] Dong, Q., Gong, S., and Zhu, X. (2019). Imbalanced deep learning by minority class incremental rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(6):1367–1381.

[21] Etten, A. V. (2019). City-scale road extraction from satellite imagery v2: Road speeds and travel times.

[22] Etten, A. V., Lindenbaum, D., and Bacastow, T. M. (2018). Spacenet: A remote sensing dataset and challenge series. *CoRR*, abs/1807.01232.

[23] Forbes, T. and Poullis, C. (2018). Deep autoencoders with aggregated residual transformations for urban reconstruction from remote sensing data. *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 23–30.

[24] Girshick, R. B. (2015). Fast R-CNN. *CoRR*, abs/1504.08083.

[25] Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524.

[26] Guo, M., Haque, A., Huang, D.-A., Yeung, S., and Fei-Fei, L. (2018). Dynamic task prioritization for multitask learning. In *ECCV (16)*, pages 282–299.

[27] He, H., Yang, D., Wang, S., Wang, S., and Li, Y. (2019). Road extraction by using atrous spatial pyramid pooling integrated encoder-decoder network and structural similarity loss. *Remote Sensing*, 11(9). cited By 8.

[28] He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. (2017). Mask R-CNN. *CoRR*, abs/1703.06870.

[29] He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[30] He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*.

[31] Hinz, S. and Baumgartner, A. (2003). Automatic extraction of urban road networks from multi-view aerial imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58(1):83 – 98. Algorithms and Techniques for Multi-Source Data Fusion in Urban Areas.

[32] Hu, J., Razdan, A., Femiani, J., Cui, M., and Wonka, P. (2007). Road network extraction and intersection detection from aerial images by tracking road footprints. *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):4144–4157. cited By 225.

[33] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[34] Iglovikov, V. and Shvets, A. (2018). Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation.

[35] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167.

[36] Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. (2016). Reinforcement learning with unsupervised auxiliary tasks.

[37] Jégou, S., Drozdzal, M., Vázquez, D., Romero, A., and Bengio, Y. (2016). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. *CoRR*, abs/1611.09326.

[38] Jou, B. and Chang, S.-F. (2016). Deep cross residual learning for multitask visual recognition. *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*.

[39] Kawakami, R., Yoshihashi, R., Fukuda, S., You, S., Iida, M., and Naemura, T. (2019). Cross-connected networks for multi-task learning of detection and segmentation. *2019 IEEE International Conference on Image Processing (ICIP)*.

[40] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

[41] Laptev, I., Mayer, H., Lindeberg, T., Eckstein, W., Steger, C., and Baumgartner, A. (2000). Automatic extraction of roads from aerial images based on scale space and snakes. *Machine Vision and Applications*, 12(1):23–31. cited By 204.

[42] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521:436–44.

[43] Li, Y., Peng, B., He, L., Fan, K., Li, Z., and Tong, L. (2019a). Road extraction from unmanned aerial vehicle remote sensing images based on improved neural networks. *Sensors (Switzerland)*, 19(19). cited By 0.

[44] Li, Z., Wegner, J., and Lucchi, A. (2019b). Topological map extraction from overhead images. volume 2019-October, pages 1715–1724. cited By 1.

[45] Lin, T., Goyal, P., Girshick, R. B., He, K., and Dollár, P. (2017). Focal loss for dense object detection. *CoRR*, abs/1708.02002.

[46] Liu, S., Johns, E., and Davison, A. J. (2019). End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880.

[47] Luong, M.-T., Le, Q. V., Sutskever, I., Vinyals, O., and Kaiser, L. (2015). Multi-task sequence to sequence learning.

[48] Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., and Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152:166 – 177.

[49] Mattyus, G., Luo, W., and Urtasun, R. (2017). Deeproadmapper: Extracting road topology from aerial images. In *The IEEE International Conference on Computer Vision (ICCV)*.

[50] Miller, G. (2017). The huge, unseen operation behind the accuracy of google maps.

[51] Milletari, F., Navab, N., and Ahmadi, S. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. *CoRR*, abs/1606.04797.

[52] Misra, I., Shrivastava, A., Gupta, A., and Hebert, M. (2016). Cross-stitch networks for multi-task learning. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[53] Mnih, V. and Hinton, G. (2012). Learning to label aerial images from noisy data. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 203–210, USA. Omnipress.

[54] Mosinska, A., Marquez-Neila, P., Kozinski, M., and Fua, P. (2018). Beyond the pixel-wise loss for topology-aware delineation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[55] Papandreou, G., Chen, L., Murphy, K., and Yuille, A. L. (2015). Weakly- and semi-supervised learning of a DCNN for semantic image segmentation. *CoRR*, abs/1502.02734.

[56] Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640.

[57] Ren, S., He, K., Girshick, R. B., and Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497.

[58] Richards, J. A. (2013). In *Remote Sensing Digital Image Analysis*.

[59] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597.

[60] Ruder, S. (2017). An overview of multi-task learning in deep neural networks.

[61] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[62] Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks.

[63] Seltzer, M. and Droppo, J. (2013). Multi-task learning in deep neural networks for improved phoneme recognition. pages 6965–6969.

[64] Shelhamer, E., Long, J., and Darrell, T. (2016). Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1–1.

[65] Shit, S., Paetzold, J. C., Sekuboyina, A., Zhylka, A., Ezhov, I., Unger, A., Pluim, J. P. W., Tetteh, G., and Menze, B. H. (2020). cldice – a topology-preserving loss function for tubular structure segmentation.

[66] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.

[67] Singh, S., Batra, A., Pang, G., Torresani, L., Basu, S., Paluri, M., and Jawahar, C. (2018). Self-supervised feature learning for semantic segmentation of overhead imagery.

[68] Song, M. and Civco, D. (2004). Road extraction using svm and image segmentation. *Photogrammetric Engineering and Remote Sensing*, 70:1365–1371.

[69] Sun, T., Chen, Z., Yang, W., and Wang, Y. (2018). Stacked u-nets with multi-output for road extraction. pages 187–1874.

[70] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[71] Takikawa, T., Acuna, D., Jampani, V., and Fidler, S. (2019). Gated-scnn: Gated shape cnns for semantic segmentation. *CoRR*, abs/1907.05740.

[72] Ting, K. M. (2010). *Confusion Matrix*, pages 209–209. Springer US, Boston, MA.

[73] Ventura, C., Pont-Tuset, J., Caelles, S., Maninis, K.-K., and Gool, L. V. (2017). Iterative deep learning for network topology extraction.

[74] Vosselman, G. and de Knecht, J. (1995). Road tracing by profile matching and kaiman filtering. In Gruen, A., Kuebler, O., and Agouris, P., editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pages 265–274, Basel. Birkhäuser Basel.

[75] Wang, W., Yang, N., Zhang, Y., Wang, F., Cao, T., and Eklund, P. (2016). A review of road extraction from remote sensing images.

[76] Wegner, J. D., Montoya-Zegarra, J. A., and Schindler, K. (2013). A higher-order crf model for road network extraction. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1698–1705.

[77] Wei, Y., Zhang, K., and Ji, S. (2019). Road network extraction from satellite images using cnn based segmentation and tracing. pages 3923–3926.

[78] Wei Liu and Dragomir Anguelov and Dumitru Erhan and Christian Szegedy and Scott Reed, and Cheng-Yang Fu, and Alexander C. Berg (2016). Ssd: Single shot multibox detector.

[79] Wu, Z., Valentini-Botinhao, C., Watts, O., and King, S. (2015). Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4460–4464.

[80] Xu, P. and Poullis, C. (2019). *Delineation of Road Networks Using Deep Residual Neural Networks and Iterative Hough Transform*, pages 32–44.

[81] Xu, Y., Xie, Z., Feng, Y., and Chen, Z. (2018). Road extraction from high-resolution remote sensing imagery using deep learning. *Remote Sensing*, 10(9).

[82] Yang, X., Li, X., Ye, Y., Lau, R., Zhang, X., and Huang, X. (2019). Road detection and centerline extraction via deep recurrent convolutional neural network u-net. *IEEE Transactions on Geoscience and Remote Sensing*, PP:1–12.

[83] Yang, X., Li, X., Ye, Y., Lau, R. Y. K., Zhang, X., and Huang, X. (2019). Road detection and centerline extraction via deep recurrent convolutional neural network u-net. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):7209–7220.

[84] Zhang, L., Bai, M., Liao, R., Urtasun, R., Marcos, D., Tuia, D., and Kellenberger, B. (2018a). Learning deep structured active contours end-to-end. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[85] Zhang, Q. and Couloigner, I. (2006). Benefit of the angular texture signature for the separation of parking lots and roads on high resolution multi-spectral imagery. *Pattern Recognition Letters*, 27:937–946.

[86] Zhang, Z., Liu, Q., and Wang, Y. (2018b). Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753.

[87] Zhao, X., Li, H., Shen, X., Liang, X., and Wu, Y. (2018). A modulation module for multi-task learning with applications in image retrieval. *Lecture Notes in Computer Science*, page 415–432.

[88] Zhou, L., Zhang, C., and Wu, M. (2018). D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 192–1924.

[89] Zhu, Y., Sapra, K., Reda, F. A., Shih, K. J., Newsam, S. D., Tao, A., and Catanzaro, B. (2018). Improving semantic segmentation via video propagation and label relaxation. *CoRR*, abs/1812.01593.

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.