Automatic Detection of Buildings with Enhanced Boundaries by Analyzing the Remotely Sensed Images.

Candidate Number: SGFR3[1]

MSc Machine Learning

Supervised by:
Andreas Kamilaris
Indrajit Kalita
Yipeng Hu

Submission date: 20 September 2022

Abstract

This dissertation aims to develop a general approach that can improve semantic segmentation performance for building extraction, regardless of the desired deep learning architecture or the dataset. Furthermore, we would like to fix the boundary problem that studies often face — the blurry boundary.

Recently, most solutions to improving semantic segmentation function through constructing more complicated architectures, with irregular boundaries commonly corrected by post-processing. However, continuously increasing the complexity of the models will not be a long-term solution. Focusing on post-processing instead of the raw results from the neural networks breaks the initial intentions of using neural networks as an end-to-end workflow.

The key contribution of this dissertation is to add building borders as an additional class while training. This not only improves the overall result by outperforming state-of-the-art models, but also fixes boundary issues. Moreover, we utilise the border feature in our results and develop an algorithm that further cleans up the results and boosts its metrics. The selected building dataset is the INRIA dataset, applied alongside a custom dataset built with satellite images from Cyprus. The baseline semantic segmentation model we used for most experiments is resnet34.

The code for this dissertation is available in the following repository:

https://github.com/Ed-Cheng/Building_Extraction_Enhanced_Boundary

Table of Contents

1 Introduction

1.1 Motivation

Visual contact is the most fundamental way people receive information from the outside world before initiating any interaction. Prior to taking action, they survey their surroundings and process the information before making decisions. Thus, for machines to interact with humans flawlessly, computer vision is the ideal initial approach for machines to receive input. However, computer vision is a heavily resource-dependent task, as images and videos are the most complicated data types to process for machines, a fact which has made computer-vision-related research unapproachable in the past.

With the advancements in computing power resulting from improved hardware, deep learning theories that had previously existed only as a point of discussion can finally be implemented [1]. Deep learning stands out from all the other analytic methods as it is capable of feature extraction. Methods that involve human intervention either in feature extraction or analysis are prone to error, with defects arising from the difficulty of handling large scale data [2]. However, better computational power is not sufficient alone for the successes that deep learning has achieved over the past decade. The construction of large-scale datasets is also significant [3]. With these two major improvements, a growing interest in interpreting aerial and satellite images has been observed by researchers.

Aerial and satellite images are crucial to help us understand our position in the universe. This is the first time we can see something the eye cannot see and visualise relationships across vastly different scales. It helps humans have a different perspective on the world concerning each individual. At present, this technology is applied to studies in multiple fields such as geography, meteorology, and anthropology. Aerial and satellite image information has also constructed the foundation of landscape design, urban planning, and disaster prevention and control [4].

Although the future of automatic building extraction seems promising according to its various applications, there are still some challenges. Compared to other deep learning topics in computer vision, aerial and satellite images generally have lower resolution. The lighting conditions and building shapes also vary depending on geographical region, season, the sensitivity of equipment available, and the special resolution. Therefore, a method that could improve the performance of building extraction while remaining independent from the quality

of the aerial and satellite images quality would be a solid leap in the field. In summary, the primary motivation of this dissertation is to find other optimisation methods without merely adjusting pre-trained models.

1.2 Objectives

In the past, building extraction was mainly performed using mathematical calculations based on prior knowledge and observations. Features such as building shapes [5], colours [6], shadows [6], and edges [7] were used to perform building extraction. However, these empirically selected features were far from practical, as there were no guidelines on deciding them, meaning that these features might vary across different datasets. There were no explicit rules for extracting features in these domains to perform logical analysis. Consequently, the direction of individual studies might widely vary, despite the fact that they were attempting to solve the same problem. Recently, the Convolutional Neural Network (CNN) has become one of the most powerful solutions for such topics. CNN has a powerful image feature extraction ability, allowing it to solve most of the clustering or classifying problems if the data could be transformed into an image format. A variant of CNN called the Fully Convolutional Networks (FCN) is a commonly used network for pixel-wise image labelling tasks [8]. Many building extraction models have been developed based on FCN, such as the U-net. While many aspects have been thoroughly studied including different combinations of internal layers and various encoding-decoding structures, several pieces could still be improved. Hence, there are three objectives to this dissertation.

1.2.1 Main Objective

Since building extraction is a pixel-wise prediction task and each pixel could be seen as an individual, there is no guarantee that nearby pixels are predicted as the same group. As a result, the edges of the predictions are usually rough, an uncommon occurrence for buildings. The aim is to generate building-like straight edges in our final predictions during training or post-processing.

1.2.2 Sub-objectives

The main factor that is lowering the performance of building extraction is generally misclassified areas: for example, predicting highways as buildings. We believe that being able to detect and correct these misclassified areas would highly improve the performance. Therefore, eliminating misclassified buildings is also one of our goals.

Moreover, an exemplary methodology should work across an extensive range of datasets, not just those cleaned and modified by experts. Instead of only using public datasets, we also create our own datasets based on the satellite images of Cyprus. Our goal is to develop methods that function not only for those clean datasets examined by the public but also for the datasets built by any individual. This strengthens the reliability of our methods and ensures that the methods can be applied in situations with limited data.

1.3 Overview of the Dissertation

The following is the structure of this dissertation. Section 2 presents the overall background and literature related to this topic, including the structure of various neural network models and challenges in current research. Section 3 explores different datasets available to the public and explains why the INRIA dataset is selected. Section 4 lists our methods' details, including pre-processing and post-processing, the initial methods, and the final reliable methods. Section 5 gives the results of our methods and compares our performance with that of other state-of-the-art models. Finally, Section 6 concludes this dissertation and gives some future directions.

2 Background

2.1 Calculation-based Building Extraction

Traditionally, researchers had no clear guidance on building extraction, so the only way of analysing the task was by prior experience and educated guesses. The main issue of such a subjective approach was that researchers could hardly reach a consensus on which particular features to extract and analyse. Moreover, building a universal dataset was impossible as there was no standard for annotating the images. Some of the traditional methods are discussed in the following section.

The literal meaning of building extraction was to separate a building from its background, which could be done by simply performing edge detection [9]. Edge detection is a technique for picking up the outlines of objects in images to separate different regions from one another. However, this method's performance is highly restricted by the quality of its input, as any element blocking the building would distort the outline.

Another similar method is evaluating the texture of the rooftops [5]. Being able to distinguish whether the texture of a section in the image belongs to a rooftop or not is also a feasible way of performing building extraction. Nevertheless, this approach also suffers from the same problem as edge detection, in that any obstacle could easily cause an error.

In addition to analysing the buildings directly, studies have also explored the potential of other features such as the shadows of a building [10]. Others combined several features and methods together to pursue better results [6]. Due to the subjectivity of selected features and parameters [9], progress has been limited—new studies could learn little from previous works, as no universal building extraction method or standard exists. The main obstacle in the past was that although the task objective was clear, there was no substantial evidence on how to approach this task. The buildings in the images were relatively apparent to human eyes in general, but it was hard to differentiate or specify the reasons or observed features that helped distinguish buildings from their backgrounds. The way humans think and process information is complicated and challenging to quantify as an equation, which is why the Neural Network (NN) is a potentially ideal solution due to its automated feature extraction and data analysis.

2.2 Neural Networks

The Deep Neural Network (DNN) [11] is an algorithm (Fig. 2.1) mimicking human neurons. It combines a significant number of nodes layer-by-layer in attempting to learn like a human, rather than having hard-coded equations and programs. The connection between nodes imitates how humans link different ideas together in their thinking process. Each node contains a weight, and the previous node decides which node should be passed to the next layer [11]. The model finishes training by tuning the weights of its nodes with excessive data until reaching equilibrium. Since the output of a DNN is decided based on multiple layers in the model, DNN is capable of reaching high accuracy even when solving non-linear relationships.

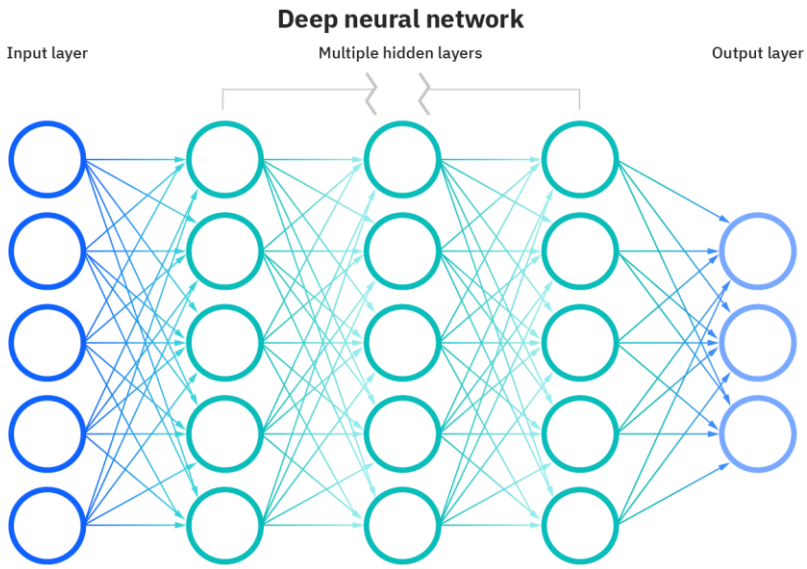

Fig. 2.1 Basic DNN structure [11].

A variant of DNN that excels in image processing is the Convolutional Neural Network (CNN) [9]. The main difference between CNN and DNN is that CNN contains an additional feature learning section. This section decomposes images in attempting to extract as much information as possible before feeding the data into the final classification section (Fig. 2.2). The goal of applying the feature learning section is to maximise the retrieval of critical features for identification, and to reduce the resolution for more straightforward calculations in the next section [12]. Convolutional layers in the feature learning section produce convoluted feature outputs after a series of computational processes. Early convolutional layers pick up low-level features (e.g. colours, outlines, edges), while further layers collect high-level features (e.g. geographic structures).

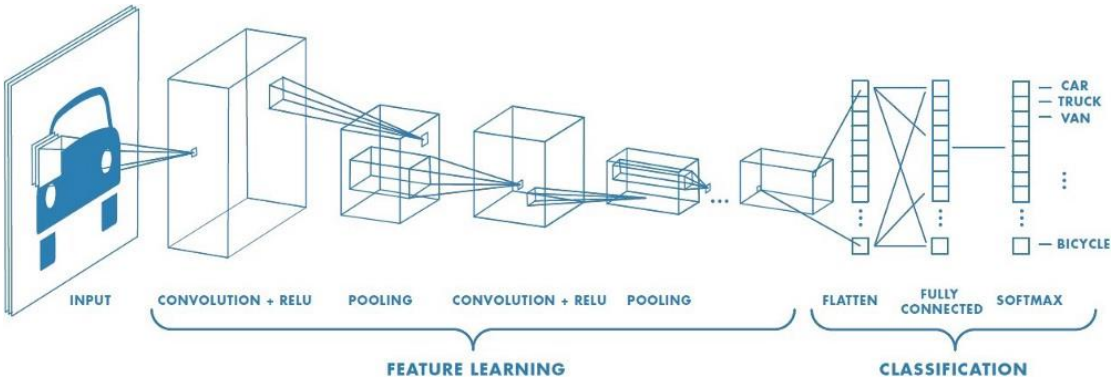

Fig. 2.2 Basic CNN structure [12].

After the convolutional layers come the pooling layers, which further decrease the resolution of the data. Even though the information is compressed, dominant features are amplified while frequent-appearing-insignificant features are abolished [12]. This mechanism is ideal for classification as the model is capable of filtering out the differences among various data and putting them into separate categories. However, the goal of building extraction, as a semantic segmentation task, is to allocate categories to segments, or pixels [11], in an image. Therefore, this goal contradicts how CNN decomposes images and down-samples them to look for patterns in specific categories. One possible solution is to carry out a patch-based CNN [13] that essentially iterates through all the sections of an image, but this will obviously suffer from the need for very high computational power.

2.3 Fully Convolutional Network (FCN) and its Variants

Soon a novel paradigm called Fully Convolutional Network (FCN) was introduced which was capable of executing pixel-wise classification [8]. As a consequence of its high performance, FCN inspired an explosive number of new state-of-the-art building extraction models [14]. The critical difference between CNN and FCN was that the fully connected layers in CNN were replaced with fully convolutional layers (Fig. 2.3). By doing so, FCN was capable of performing pixel-wise classification. The advantage of this structure was that the powerful classifying ability of CNN was kept, but instead of giving one classification score per image, we can now convolute the score with a spatial feature map, so a classification score was given to each pixel. Furthermore, the introduction of skip architecture skipped some of the layers in down-sampling and connected them directly to the up-sampling section. This allowed FCN to combine information from deep and shallow layers before the final segmentation [14], preventing the extracted features from being too abstract for up-sampling.

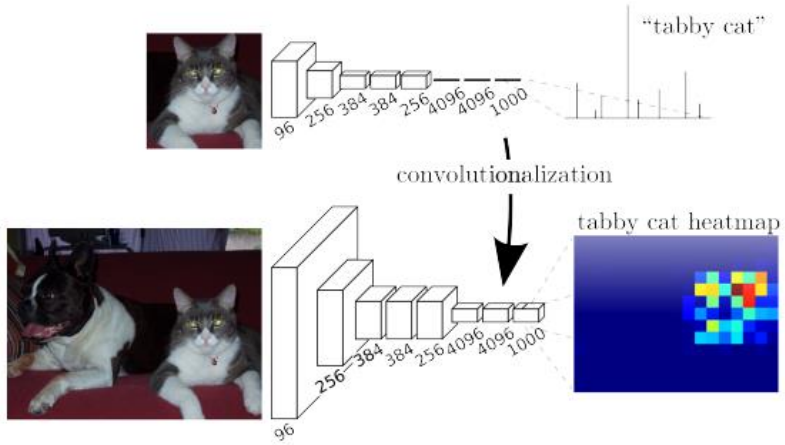

Fig. 2.3 The difference between fully connected layers and fully convolutional layers [8].

U-net is a model based on the FCN. It implements an encoder-decoder architecture that captures context with shrinking paths and accurately performs pixel-wise classification by symmetric paths [8]. In other words, the encoder layers are connected with the corresponding decoder layers (Fig. 2.4). U-net was invented to solve a critical problem in biomedical image processing, the lack of accurate data, which is also applicable for building extraction. Furthermore, one of the most critical features of U-net is that it has a lot of connections in the up-sampling section, which continues propagating information while increasing the resolution [13]. The shape of the architecture will then look like a U, as Fig. 2.4 depicts—thus, the origin of the model's name, U-net.

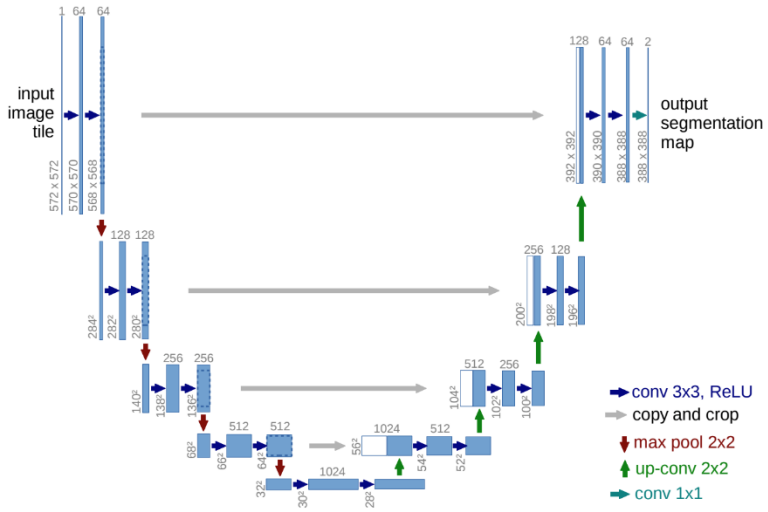

Fig. 2.4 A typical U-net structure [15].

Another similar variant, SegNet, also utilises the encoder-decoder structure. SegNet was a model built on a previous state-of-the-art structure, VGG-16 [16]. The 13 convolutional

layers in the encoder section of SegNet were inspired by the according layers of VGG-16, a robust network designed for object classification. Next, these encoder layers are linked to 13 corresponding decoder layers [17], like the U-net. Essentially, SegNet extracts the most useful parts of VGG-16 and reconstructs the layers based on U-net, creating a powerful model while compressing the complexity.

In addition to tuning the structure of the U-net, there was also a study published which alternatively attempted to improve FCN with recurrent networks [18]. Since the goal of the network was to extract features of an image by hierarchy, the structure of the network looked like a pyramid (Fig. 2.5), deemed Feature Pyramid Networks (FPN). These extractions were then linked together by lateral connections [19]. Nevertheless, the features from the early layers became less noticeable because of such a structure.

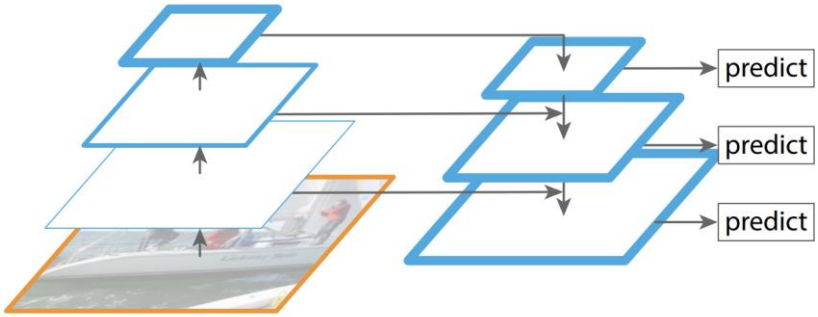

Fig. 2.5 A typical FCN structure [19].

In addition to modifying the network structure from end to end, multi-task learning is also used as a common method of improving segmentation predictions. For these models, different learning goals are fed into the same model using multiple heads [20]. As they are being trained on the same model, the different tasks usually relate to each other at the pixel level. For instance, Benjamin [21] built a model that fused the building masks and the corresponding boundary distance information into a single loss function to improve performance.

2.4 Challenges in Building Extraction

Despite the fact that there have been various studies related to building extraction, there are still some significant issues that could be improved upon. One of the main problems is the misclassification of the boundaries [22], which mostly occurs when variations of lighting conditions and physical objects interfere with the silhouette of the buildings. Another problem

is that the generalisation of each segmentation model is usually relatively poor, meaning that an individual model requires specialised training to suit the particular usage case [22]. This dissertation will focus on the former issue.

To overcome poor performance in boundary identification, several possible solutions have been made available from previous studies. Marmanis [23] tried to perform building extraction while doing edge detection separately. SegNet was applied as the primary building extraction model and an edge detection network was added on top. This resulted in a bulky model which was very difficult to train. Cheng [24] experimented with a multi-task model that included edge detection during training. This shrank the model size but required extensive human modification to the model, as a specialised loss function was needed. Kang [25] designed an algorithm that regularised the building edges solely in the post-processing stage. The results worked fine when the predictions of the building were only slightly off. However, post-processing could not correct the building boundaries that were poorly predicted, meaning that there were only improvements in results that were already performing well, yet offered little help to those performing poorly. Generally speaking, post-processing could be an additional step to further improve performance. However, the appropriateness of using it as the only step to solve the problem needs to be reconsidered.

After extensive literature research, it was found that past papers primarily focused on extending or improving existing models. Accordingly, we thus introduce the idea that there could be other ways of improving the performance of building extraction in lieu of the currently available range of merging and tuning models.

3 Datasets

3.1 Overview of Datasets

There are a great deal of publicly available satellite image datasets on buildings from a variety of sources with a wide range of quality and sizes. The Massachusetts buildings dataset [26] is one of the earliest well-known datasets of building data, collected from the Boston area. However, the resolution of the images is relatively low due to early technology disadvantages, with numerous annotation errors. The SpaceNet challenge dataset [27] is a large dataset consisting of buildings from five different cities. Yet, as Muying [28] points out, the annotation accuracy is relatively low compared to the other datasets mentioned in this section. The OpenAI dataset [29] consisting of buildings in Tanzania was assembled for a building extraction competition in 2018. Although this dataset was created with a very high resolution (0.07 m), there are nonetheless a significant number of faulty annotations.

We have chosen to use the relatively more prominent and more acute INRIA dataset in conjunction with a self-annotated Cyprus dataset to ensure our methods would function for both verified data and more personal data with potential human error.

3.2 INRIA Dataset

The INRIA dataset is a well-constructed data set which features a leader board on its own official website [30]. The dataset is quite large and diverse, as it organises images from certain US and Austrian areas with different resolutions, but all the images were eventually standardised to 0.3 m [30]. It contains 180 images, each with a dimension of 5000 × 5000 pixels, evenly collected from five different cities: Austin, Chicago, Kitsap, Vienna and West Tyrol. For the purpose of fair comparison across different studies, it is suggested to take the first five images of each city as the testing set and train the model on the remaining 31 images. The clear instructions and the allocation of train sets and test sets make the INRIA dataset the perfect candidate for building extraction.

3.3 Cyprus Dataset

This is a self-annotated dataset that we used for the final examination of our experiment. The images were collected in 2012 with a resolution of 0.5 m. The dimension of each image is 256 × 256. As mentioned before, our goal is to develop a general method that could improve the overall performance of building extraction, regardless of the dataset or the model's architecture.

Therefore, we created this small dataset that contains 366 images only as the final experiment to test the consistency and reliability of our method.

4 Methods

4.1 Overview of Methods and Plans

Our final goal is to develop a method that improves the performance of building extraction regardless of the data or the model. Therefore, we follow the procedure below:

- Develop experiments with the INRIA dataset, focused specifically on the city of Austin
- Benchmark our method with previous studies by performing experiments on all the INRIA datasets, including the five cities of Austin, Chicago, Kitsap, Vienna and West Tyrol
- Finalise our experiment by examining the results on our own Cyprus dataset

Our methods include the following:

1. Two-step training with secondary labelled data
2. Enhanced boundary data augmentation
3. Post-processing, removing misclassified buildings
4. Scratch training with no pre-trained model

Before examining how our method generalises across different datasets, we need a baseline to work with. Thus, we have picked the most reliable data from the INRIA dataset as our baseline data. After studying different sources [4] [20] [21] [32], we found that Austin images consistently outperformed those of other cities, hence we decided to take the images of this city as our baseline data while developing our experiments. Furthermore, to test the generalisation ability of our methods, we picked the best workflow and applied it to the remaining four cities in the INRIA dataset for comparison with previous studies. Finally, we ran segmentation on the Cyprus dataset with and without the workflow we developed and compared the differences.

As shown in the bullet points above, we mainly contributed four methods and listed them in this dissertation. The training in the first three methods is based on the pre-trained model, resnet34 [33], and the last method aims to train the model from scratch. Each method was inspired by the previous one. We ran through various experiments and carefully inspected the results to look for further improvements, then collated meaningful outcomes for this report. Since pre-trained models, especially resnet34, have been proven to converge quickly on

segmentation tasks, the training epochs were set to a fixed value across all developing experiments for fairness and considering time costs. As these benchmarking experiments aim to compete with other peer results, the epochs of such experiments are set to a higher value and the best model was autosaved during training.

4.2 Pre-processing of the Data

The primary data we used for training during development were the Austin images in the INRIA dataset. Depending on the objective of the experiment, the 5000 × 5000 images were split into smaller patches with overlapping. We also reserved 20% of the training set as the validating set. When the raw data were divided into smaller images, there might have been situations in which the whole image contained no buildings at all, leaving the entire image as a background. Different ratios of these "blank" parts were removed depending on how much data were required for the specific experiment.

In data augmentation, we decided to rotate the images if the existing data were insufficient. Since the INRIA dataset size was sufficient for model training, the only dataset to which we applied data augmentation was the Cyprus dataset. Each image was randomly rotated between 0 – 360 degrees during data augmentation.

4.3 Two-step Training with Secondary Labelled Data

When running a binary semantic segmentation model, there were four outcomes regarding correctness. We defined correctly classified buildings as True Positive (TP), correctly classified backgrounds as True Negative (TN), incorrectly classified buildings as False Positive (FP), and incorrectly classified backgrounds as False Negative (FN) (as shown in Fig 4.1.). The areas belonging to the same outcome were considered "similar to each other", according to the model. Thus, we came up with an original idea that could potentially boost accuracy.

Initially, we had two labels: the background and the building. After training the model once, we performed this classification on the training set to create a new dataset with our four possible labels: TP, TN, FP, and FN. By combining these results differently, we were able to generate two new datasets. One dataset consisted of all four labels, training the network explicitly on the misclassified areas. From this, the model should learn to identify those FP and FN areas which we could remove or merge back into the TP and TN results during post-processing. However, forcing the network to distinguish between TP and FN areas might

confuse the model as they both belong to the building class, and any misclassification on those two labels will directly reduce the IoU.

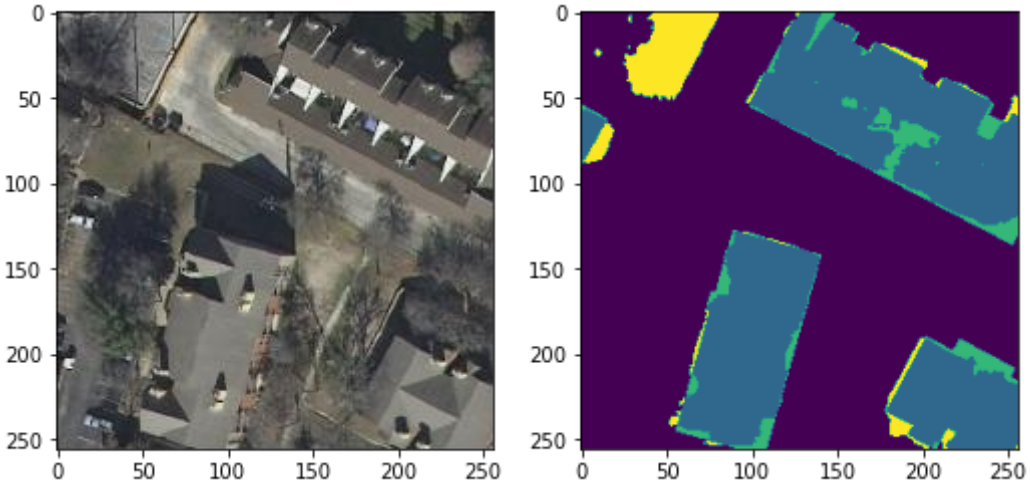

Fig 4.1 An example segmentation results. The right shows the real satellite image. The left is a sample prediction: blue pixels are TP, purple pixels are TN, green pixels are FP, and yellow pixels are FN.

On the other hand, the FP areas mean that the model is weaker in classifying this specific region. By manually examining the corresponding sections of the FP areas, we find that there are primarily obstacles such as trees and shadows. Thus, we decided to create a second dataset of only three labels: TP, TN, and FP. By doing so, we isolated these FP sections as individual labels for a higher chance of improving the overall performance. For clarification, we call the first dataset the four-label model and the second dataset the three-label model.

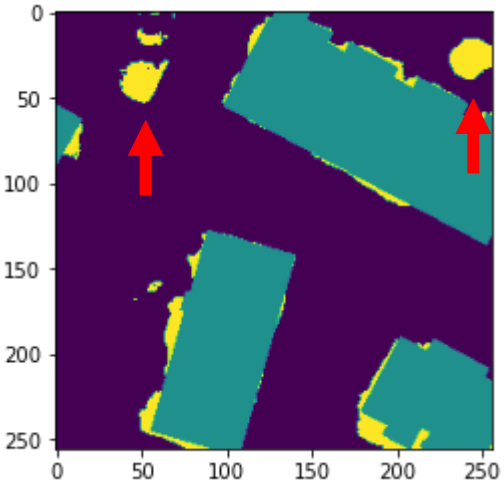

Fig 4.2 An example of 3-label secondary labelled data. Green pixels are labelled as buildings, purple pixels are labelled as background, and yellow pixels are labelled as FP. The two red arrows depict the FP areas totally disconnected from any building.

4.4 Enhanced Boundary Data Augmentation

Inspired by the previous method and discovering that the FP results along the boundary of buildings have similar and repetitive patterns, we applied a thick border around the building as an additional label for training. We adjusted the ground truth building masks by extracting the buildings' outlines and applying borders with the desired width onto the outlines. In doing so, there are two potential advantages. Intuitively, if the model successfully classifies the border, the number of FP areas (Fig 4.2) will be greatly reduced, as an additional label, the border, now surrounds the buildings. Moreover, the border could work as a generaliser that assists the model when classifying buildings. Taking Fig 4.2 as an example, the misclassified areas indicated by the two arrows are not caused by the fluctuating errors around the correct building predictions, instead, they are pure misclassifications. The problem is that the model had only two categories to classify, and no additional information when the decisions were unclear, thus resulting in errors. Meanwhile, if the border label were available, the network might pick up the information that there is no potential border in the surrounding area, lowering the possibility of predicting "building".

Fig 4.3 An example of a training satellite image and its corresponding enhanced data mask.

From Fig 4.3 we can see that the image is covered with trees and shadows everywhere but mostly around the buildings. The additional border label consists mainly of this information, meaning that the model is essentially learning to classify trees and shadows around buildings, which is a positive addition to the model. Another significant factor to consider is the width of the border. Borders that are too thick might include too much information and weaken the effect we are looking for, while borders that are too thin might instead cause disruption to the model. To sum up, this method trains the model to classify between buildings, the areas around

buildings, and the background.

4.5 Post-processing, Removing Misclassified Buildings

Post-processing in previous studies mainly focused on smoothing out the edges of the predictions, as the only information they had was the prediction of buildings and the background. However, including the border as another label allows us to identify whether the predictions of the buildings are reliable or not.

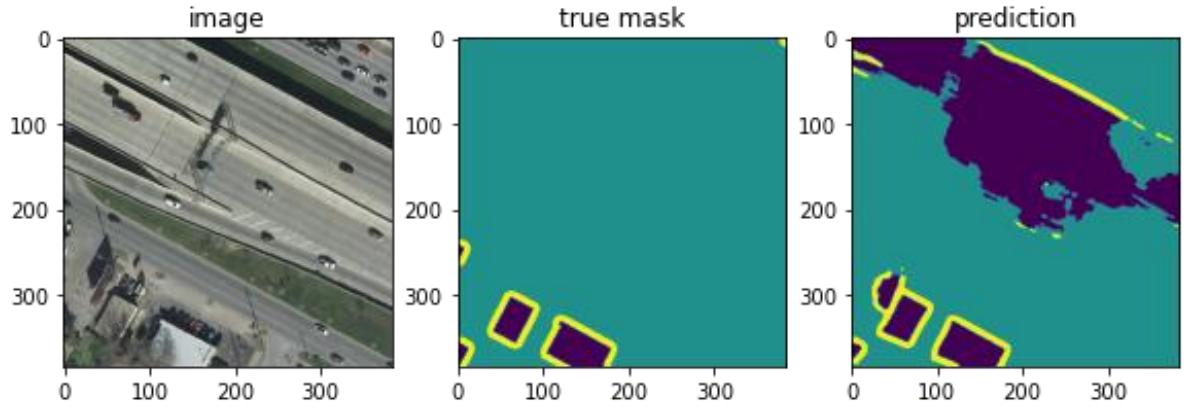

Fig 4.4 An example of building extraction that could be easily corrected by our method.

Take Fig 4.4 as an example. The model misclassified the highway as a vast building, but there is no complete border around the misclassified area. Therefore, we can rely on the predicted borders to determine whether the building predictions should be removed or not in post-processing.

4.5.1False Positive Removal via Expanding Borders

Similar to Section 4.4, additional borders were added around the predicted buildings. By subtracting the additional borders from the predicted borders, the remaining elements on the images will be the borders surrounding the misclassified buildings, which we call removal-borders. Next, we dilated the removal-borders to overlap with the misclassified buildings. Finally, the overlapped areas were removed, meaning that the outer areas of the misclassified buildings were eliminated. By iterating the process several times, all of the misclassified buildings will be entirely removed.

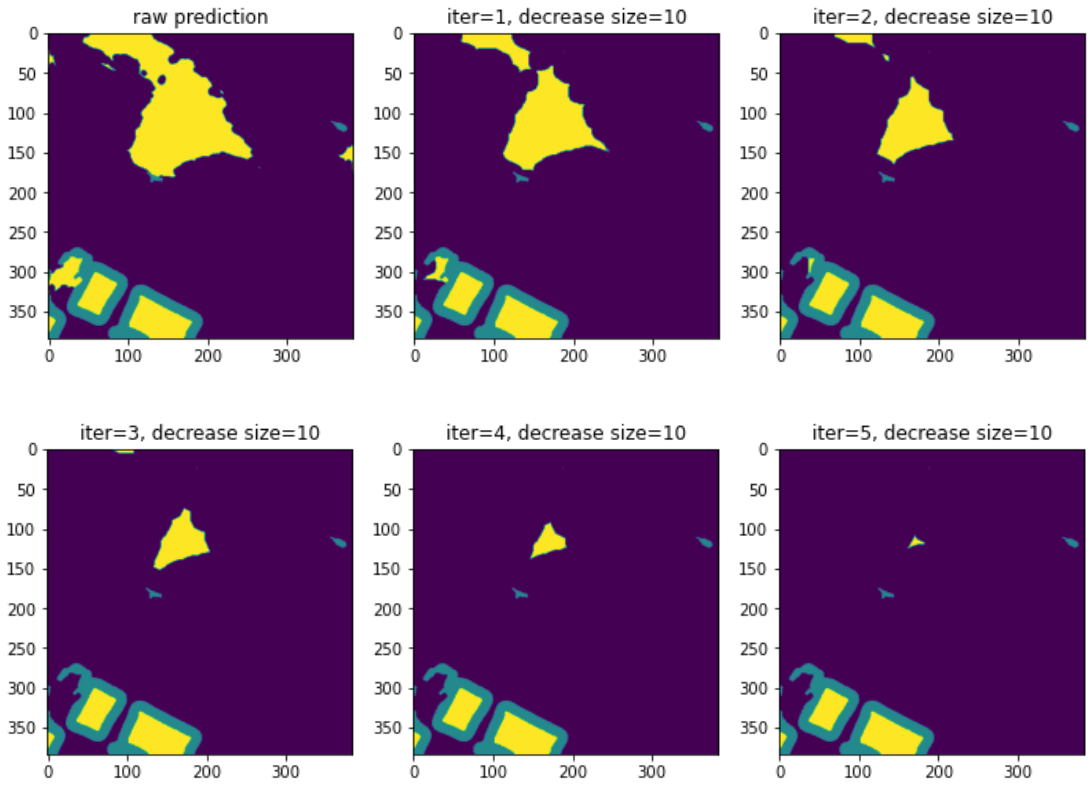

Fig 4.5 Removing incorrect building predictions over five iterations via expanding borders. The yellow pixels are buildings, the green pixels are borders, and the purple pixels are the background.

As shown in Fig 4.5, as more iterations are implemented, more falsely classified building areas are removed. Any building prediction that is not within a closed border will be removed with sufficient iterations. Additionally, the maximum width that can be removed per iteration is the predicted border width of the model. Otherwise, the correctly predicted buildings inside the border might also be eroded. Thus, this method's post-processing results heavily rely on the prediction's quality, as any break in the predicted border will cause a correctly classified building to be removed.

4.5.2 False Positive Removal via Kernels

Inspired by the median blur algorithm [34], kernels are used to scan through the images to perform incorrect building prediction removals. When different sizes of kernels scan through the images, they calculate the ratio of background pixels, border pixels, and building pixels. If there are no border pixels, but the image is full of background and building pixels, there is a high chance that the building pixels in the current kernel will be incorrectly predicted.

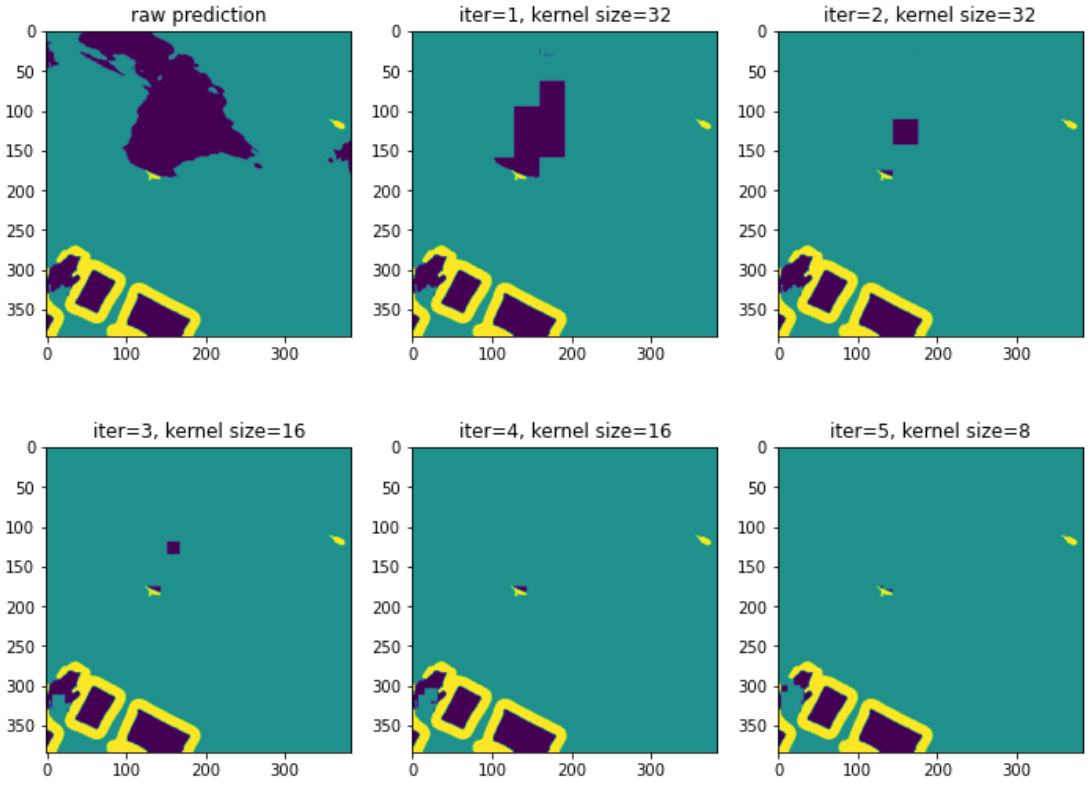

Fig 4.6 Removing incorrect building predictions over five iterations via kernels. The purple pixels are buildings, the yellow pixels are borders, and the green pixels are the background.

As shown in Fig 4.6, square areas were removed according to their kernel size. Unlike the previous method, predicted buildings inside an opened border might not be removed depending on the size of the kernel, which makes this method more adaptable. The maximum removal area per iteration is also not limited by any factor. Although this method has a higher tolerance to the predicting the quality of a border, it is not fully independent. If the border opening is larger than the kernel size, the kernels will still enter the area surrounded by the borders and clean everything. Thus, the performance of this method is proportional to the quality of the border prediction.

4.6 Scratch Training with No Pre-Trained Model

In addition to utilising a pre-trained model, we also built a U-net model from scratch [35] to examine if our methods could improve the performance of building extraction to compete with other state-of-the-art models. Initially, the data loading pipeline, the parameters, and the setups were identical to what we trained with resnet34. Yet, we found that training building extraction from scratch is difficult to converge, as the imbalanced building–background pixels constantly cause the model to predict only backgrounds.

Our solution to this issue is to change the optimiser from Adam to SGD. Adam is a commonly used optimiser, outperforming the SGD optimiser when it came to finalising the model, according to our experiments. However, its fast-converging nature became an issue for our purposes. As soon as it discovered the fact that predicting only backgrounds gives a relatively strong IoU, the Adam optimiser slowed down exploring other possibilities no matter how we tuned the learning rate. On the other hand, while SGD is generally not so popular as it tends to explore more possibilities before finally converging on the data, this feature was exactly what we were looking for in the initial stages.

4.7 Additional Parameter Tuning

Other parameters could also affect the performance of building segmentation, so we verified two additional variables: the input image sizes, and the generalisation ability of blank images. In previous studies, the sizes of the input images have rarely been discussed. It is commonly mentioned in these papers without further explanation or verification. Larger input sizes require heavier memory usage during training, so it would be meaningless to use large input sizes if this factor did not affect results. Another interesting factor is blank images—in other words, those images with only backgrounds. It is possible that blank images could help generalise the model and improve the prediction of background areas. However, it could also be the other way around. If the inclusion of blank images does not improve the final result, removing them in the pre-processing stage could save much computing time.

5 Results and Analysis

5.1 Overview of Results

The parameters were set as the following unless otherwise informed. The training images were removed if they contained less than 20% buildings, according to the pixel counts in the masks. This served to balance training time and performance during developing, with the optimal percentage of removal being experimented upon in a later stage; the training epochs were set to 30, and we evaluated the best model between these 30 epochs. The best model was defined by comparing the validation IoU score. Early stopping was set to 10 epochs, yet this condition was never reached; the input image sizes were set to 384 × 384 pixels. To ensure a fair comparison between all the methods, the classwise building IoU score is the only metric we consider, and the final score for each method is obtained by attaining the average score from three training sessions. Note that the best overall IoU scores for the INRIA Austin dataset from previous studies are 76.76 [21], 78.76 [32], 80.15 [4], and 87.62 [36], which we will take into consideration during our evaluation.

5.2 Metrics

We chose to record the accuracy and the Intersection over Union (IoU, also called the Jaccard index) as the metrics of our experiment, since they are the two most common measurements for segmentation tasks [31].

Accuracy is a standard metric for classifying problems, but it could be misleading when facing pixel-wise classification. The background areas for building extraction data are commonly significantly larger than the building areas; thus, labels are usually heavily unbalanced, which potentially leads to high accuracy even if we misclassify all the buildings. The accuracy is included in this study solely for the purpose of comparing it with other research papers.

The other metric, IoU, is the ratio of the area we classified correctly and the valid area of the ground truth. Assuming that A is the area of the ground truth and B is the area of the prediction, we can get the IoU as follows:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{5.1}$$

5.3 Two-step Training with Secondary Labelled Data

There are two approaches in this section. One trained the model with TP, TN, FP, and FN labels, and another trained the model with TP, TN, and FP as mentioned in Section 4.3. Also, a model train from raw data is provided to serve as a baseline.

Table 5.1 Comparison of the results between secondary-labelled models and a baseline model.

Experiment	Baseline model	Three-label model	Four-label model
Attempt 1, classwise IoU score	74.71	74.80	73.77
Attempt 2, classwise IoU score	75.01	74.95	72.55
Attempt 3, classwise IoU score	74.81	75.23	73.11
Mean classwise IoU score	74.84	74.99	73.14
The scale of change comparing baseline	0.00%	+0.20%	-2.27%

As seen in the results shown in Table 5.1, the four labels are not complementary but lowered the overall score with an average of 2.27% decrease in performance compared to the baseline model. For the three-label model, the inclusion of the FP areas in training helps the model remove some of the previously misclassified areas. The reason for this is that a portion of the FP areas belongs to similar items—for example trees, sport courts, and vehicles. As such, separating them out as an additional label leads to the aforementioned improvement. However, the improvement is insignificant compared to the baseline model, as it only increased by 0.2%, which might even be a statistical error caused by the low sampling numbers available.

5.4 Enhanced Boundary Data Augmentation with Detailed Comparisons

Various parameters are tuned in this section. In order to find the optimal settings to benchmark against other studies, we examined the width of the additional borders, the input data sizes, and the removal threshold of blank input images.

From Table 5.2 we can see that different border widths have an obvious impact. The best border width we found for the INRIA Austin dataset was roughly 15 pixels. However, we believe that according to the building composition and resolution of different datasets, this value should change.

We also examined the effect of input image sizes. Larger input sizes were found to give better performance; a possible explanation for this is that the larger the input images are, the fewer edge cases we will face. Splitting images into small pieces means that buildings might be cut into fractions, reducing the possibility of them being detected by the model.

Table 5.2 Comparison of the results between different input sizes and border widths.

Model	1	2	3	4
Border width (pixel)	0	15	15	25
Input size (pixel^2)	384	384	256	384
Mean classwise IoU score (%)	74.86	76.50	76.22	75.37
Mean classwise IoU score after cleaning (%)	N/A	77.09	76.74	76.18

In terms of the generalisation ability of the blank images, we conclude that including blank images has a positive influence on the performance with the cost of training time. The percentage of the removal threshold stands for the minimum ratio of existing buildings in an image. If the ratio of the building pixels is lower than the threshold, the image is removed from the training set. Hence, a 1% removal threshold is used when running the benchmark to balance the performance and the training time.

Table 5.3 Comparison of the results between input removal thresholds.

Model	5	6	7
Input removal threshold (%)	20%	1%	0%
Border width (pixel)	15	15	15
Training time (in ratio)	1.00	1.20	1.32
Mean classwise IoU score (%)	76.50	79.05	79.35
Mean classwise IoU score after cleaning (%)	77.09	78.85	78.93

Finally, we observe an unexpected outcome for the cleaning in post-processing. As mentioned before, the algorithms we developed more or less rely on the predicted border. When the model is not fully trained, a fair number of predicted buildings are not surrounded by a closed border, including TP and FP results. Usually, this happens more often on FP than TP, meaning that even if our algorithm removes all the predicted buildings that are not surrounded by the predicted border, the final IoU score will increase. However, when the model is fully trained, most of the predicted buildings are surrounded by borders, regardless of the genuineness of the prediction.

Therefore, borders have different meanings in different models. In an under-trained model, the borders appear when there is a high possibility of being a true border, while it is likely to ignore potential borders with a low possibility of being correct. On the other hand, a fully trained model has learned that buildings should be surrounded by borders, thus the possibility of being a true border becomes a less significant consideration, and as such tends to

draw borders around everything that it sees as a building. Consequently, further cleaning of the prediction in the post-process might not work when the model is fully trained with many epochs.

Observing the results, we verify the reasons as to why adding borders enhances performance. The additional border serves as a regulator that fixes unclear situations, as depicted in Fig 5.1. In certain situations, buildings and backgrounds might be similar, and being able to detect the border allows the model to make more reasonable decisions. Moreover, we mentioned the boundary issues in Section 2.4; the messy boundaries are tidied up and straightened as shown in Fig 5.2.

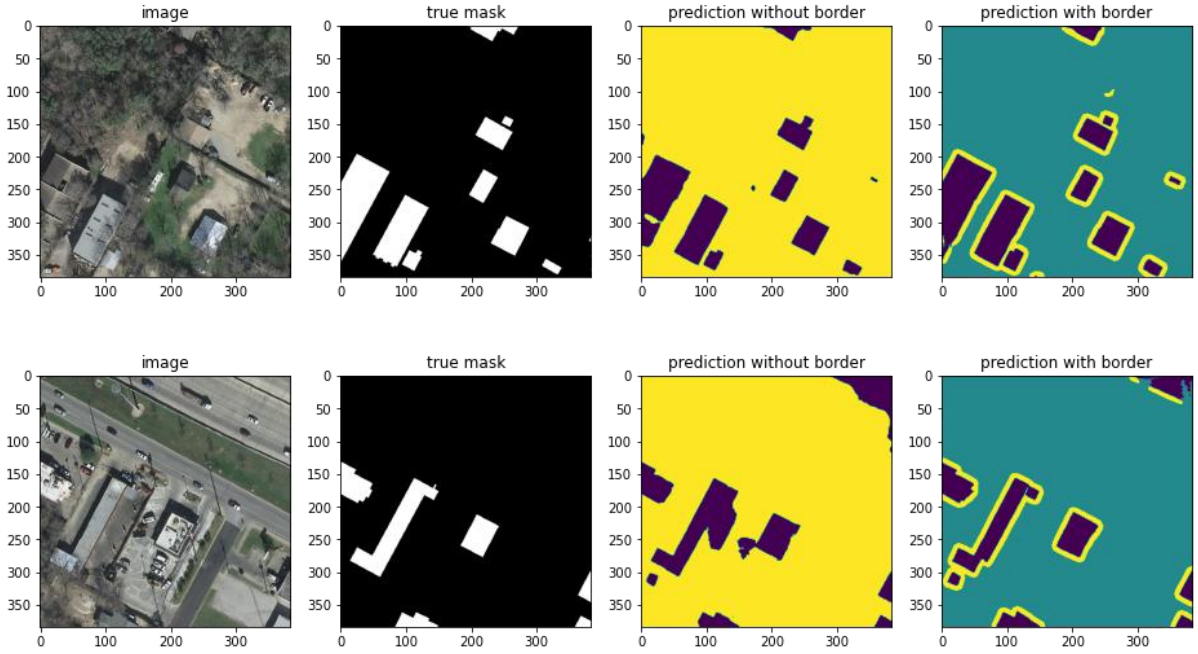

Fig 5.1 Results showing that additional borders enhanced prediction performance.

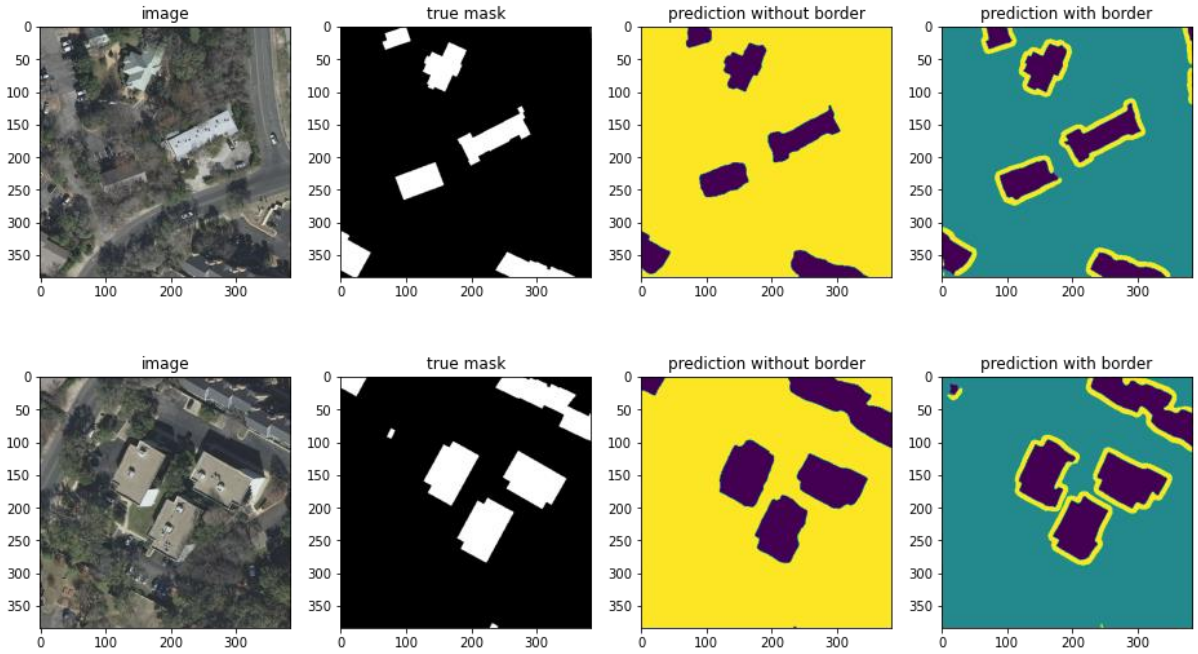

Fig 5.2 Results showing that additional borders give sharper boundaries.

Finally, we compare some of the good and poor results and find some patterns (Fig 5.3 and Fig 5.4). For images that have clear and dark shadows, the results are generally better. However, if the image of a tall building is taken from an angle, the building walls will blend in with the shadows and become difficult to classify, as the middle row of Fig 5.4 shows. Also, larger vehicles or textures that stand out from the background could confuse the model, especially if there is shadow present. Moreover, these misclassified areas are usually surrounded by borders as the model has strong belief that they are buildings, which raises the difficulty of removal during post-processing. A special type of building that we face is massive edifices and construction projects. Both good results and poor results occur when predicting these large buildings, depending on the colours and textures of the rooftops. Lighter-coloured rooftops tend to lead to higher IoU scores, as this is the usual rooftop option in our dataset (Fig 5.3), while darker-coloured rooftops have opposite outcomes (Fig 5.4).

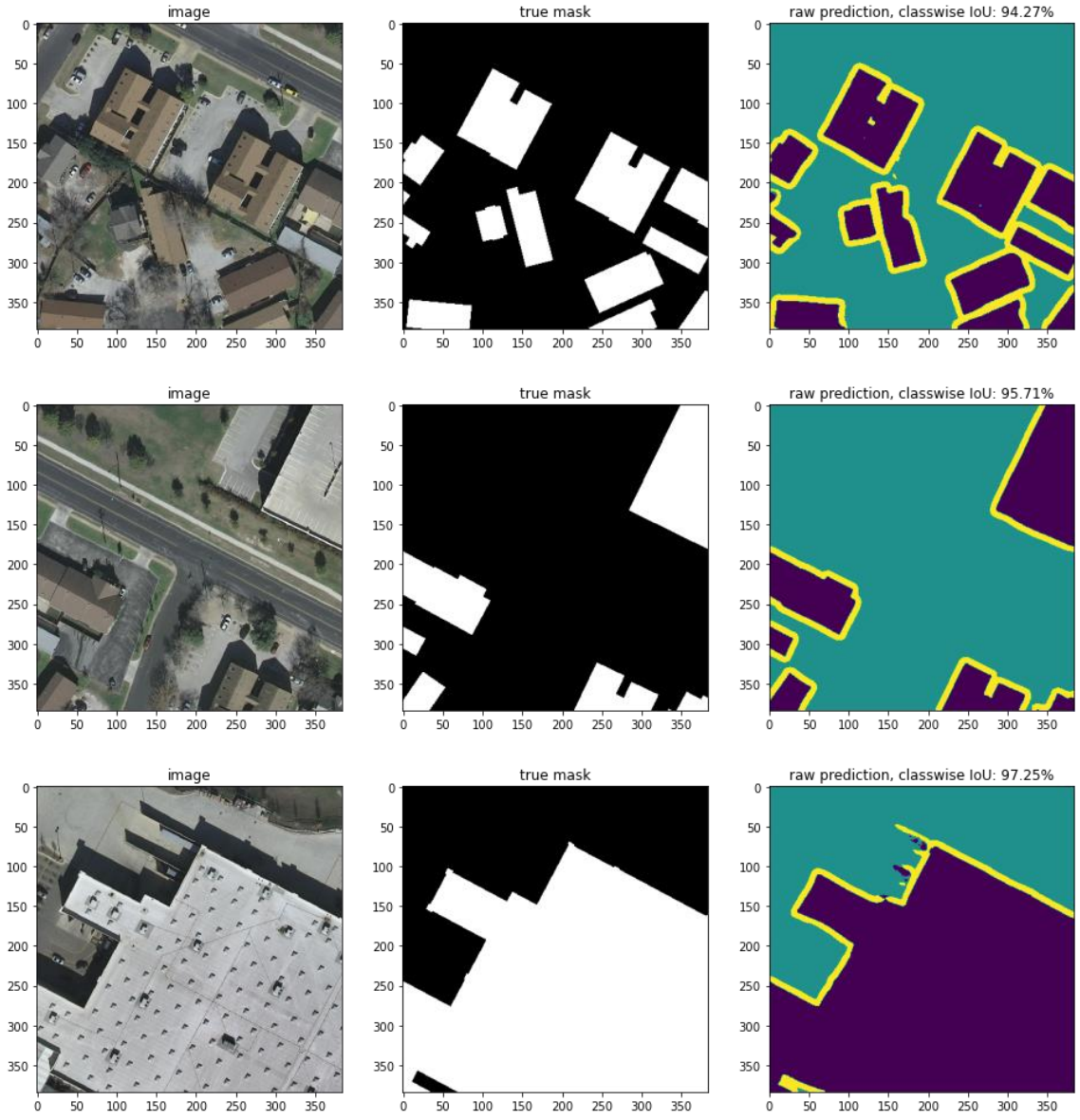

Fig 5.3 Well-classified examples with classwise IoU of 94.42% (up), 95.71% (middle), and 97.25% (bottom).

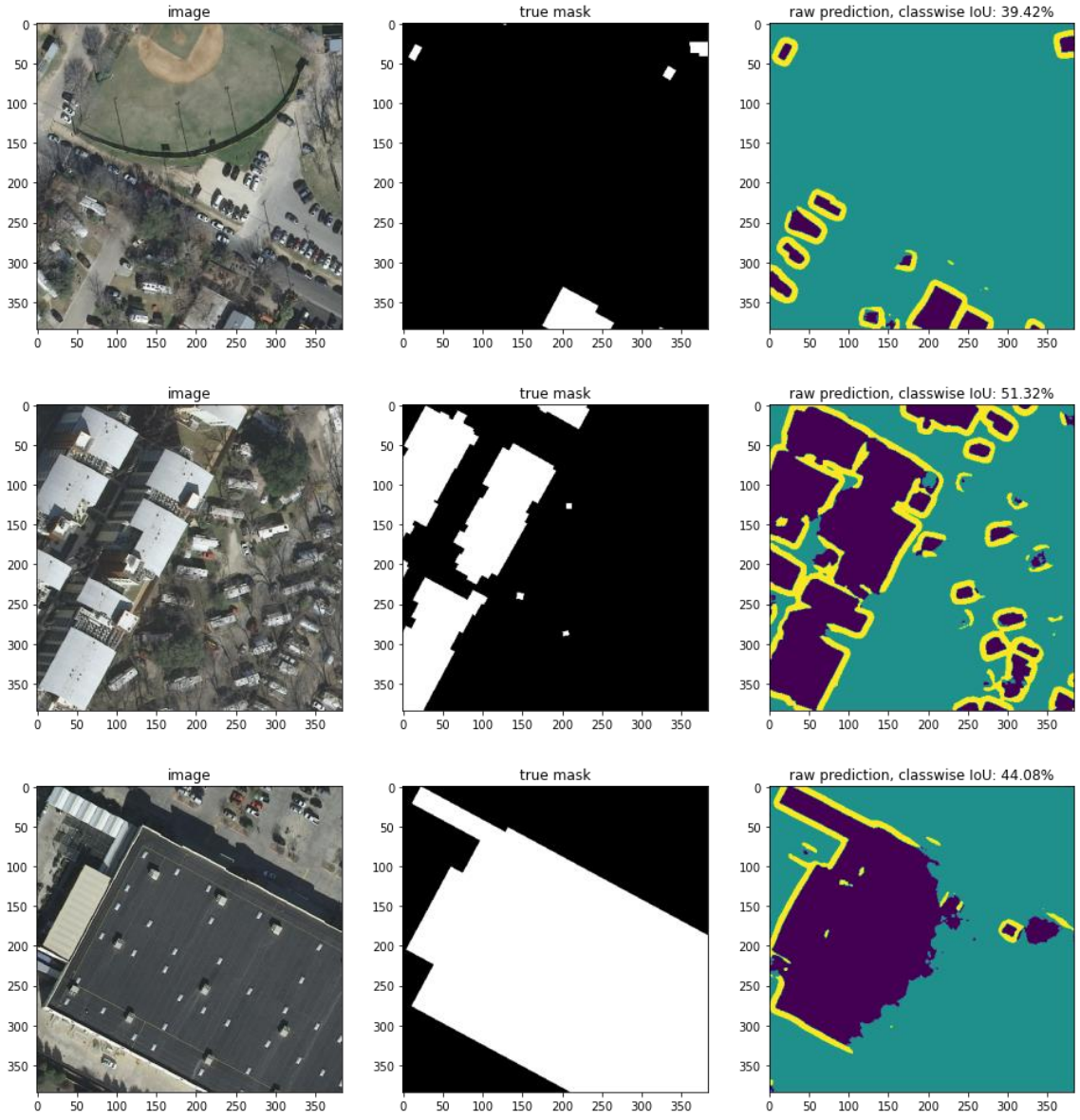

Fig 5.4 Misclassified examples with classwise IoU of 39.42% (up), 51.32% (middle), and 44.08% (bottom).

5.5 Scratch Training with No Pre-trained Model

The usage of an Adam optimiser and SGD optimiser was discussed previously, so we will be focusing on the results only in this section. Since the objective of the SGD model is to generate an initial model that at least starts producing predictions rather than backgrounds, the model only requires to be lightly trained. The direct result that comes from the SGD model is shown in Fig 5.5, which is wildly inaccurate. However, after continuing training with the Adam optimiser, the predictions become more promising, although the performance still cannot compete with pre-trained models.

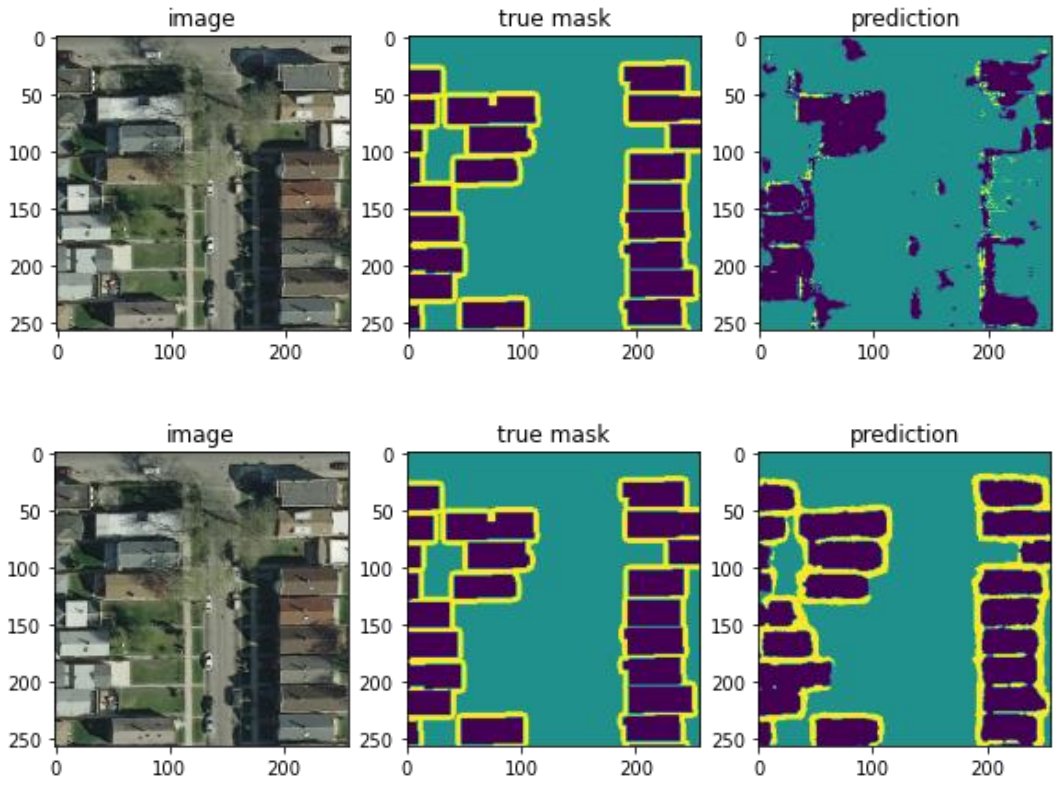

Fig 5.5 The upper result is generated from a lightly trained model using an SGD optimiser (model-SGD). The lower result is generated from a model that continued training from model-SGD using Adam optimiser.

5.6 Evaluation of INRIA Dataset

After confirming the parameters, we tested our approach on all the data in the INRIA dataset. The false positives were removed by the method mentioned in 4.5.2 due to their better tolerance of unclosed borders. The borders were set to 15 pixels and the training epochs were set to 100, with 10-epochs early stopping. Table 5.4 records the results and Fig 5.6 shows a successful result of false positive removal from a misclassified highway. The classwise IoU represents the building IoU scores only, the main focus of building extraction. In contrast, the IoU represents the overall IoU score, which is the standard benchmarking metric across different studies.

Table 5.4 The final results of the INRIA dataset.

Score	Austin	Chicago	Kitsap	West Tyrol	Vienna
Classwise IoU (%)	78.97	74.81	54.7	76.83	81.73
Cleaned classwise IoU (%)	79.11	73.93	51.83	75.49	80.12
IoU (%)	87.54	84.06	76.27	87.94	87.27
Accuracy (%)	96.61	94.41	97.91	99.08	94.56

The results of Kitsap are significantly worse than those of the other four cities, and we find that reasonable after carefully inspecting the images and the masks. Kitsap mostly consists of background forests, which is a factor in their low IoU score. Furthermore, many incorrect labels were found in Kitsap, aggravating the outcome. Austin was the only city that benefitted from false positive removal during post-processing cleaning. It was found that Austin consists of a lot of highways which could easily be predicted as buildings, but also effortlessly detected by our algorithm. From Fig 5.6, we can see that most of the mis-predicted highway pixels at the bottom were removed. However, other cities have a different geographical landscape, which might not benefit from the parameters we found in our experiments on Austin.

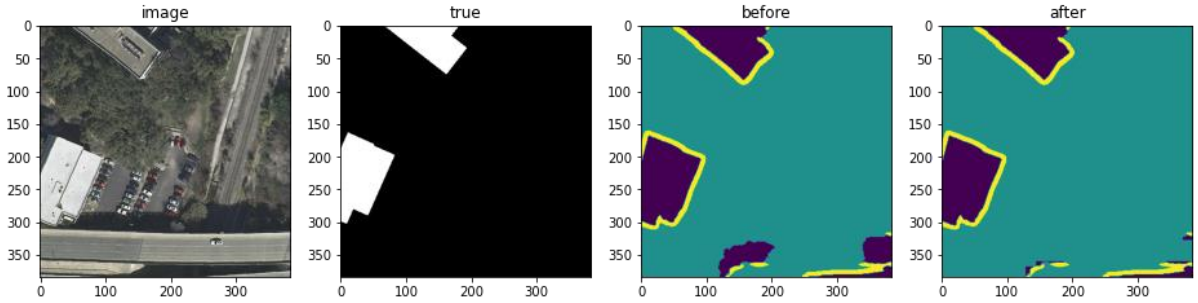

Fig 5.6 Sample false positive removal result of Austin in INRIA data set.

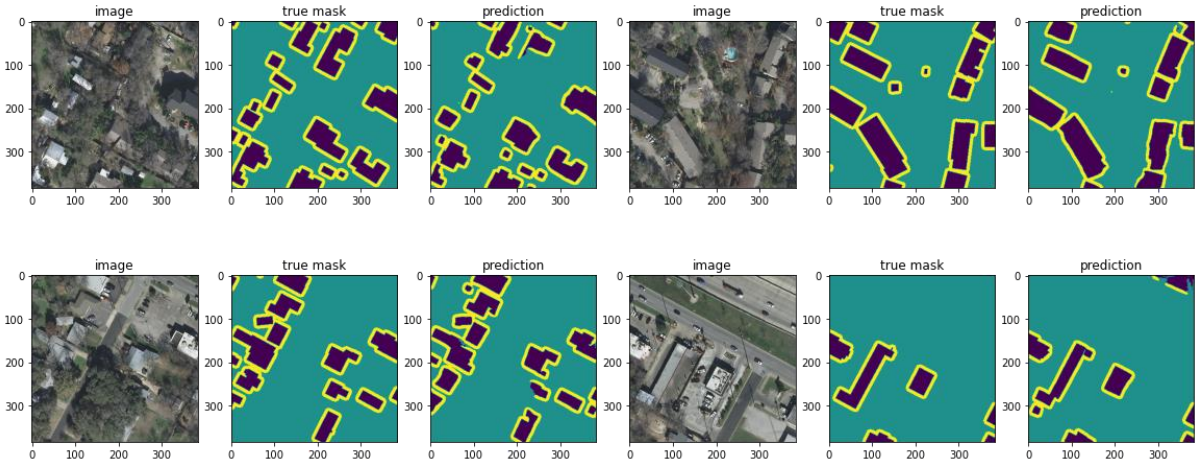

Fig 5.7 Sample prediction results of Austin in INRIA dataset.

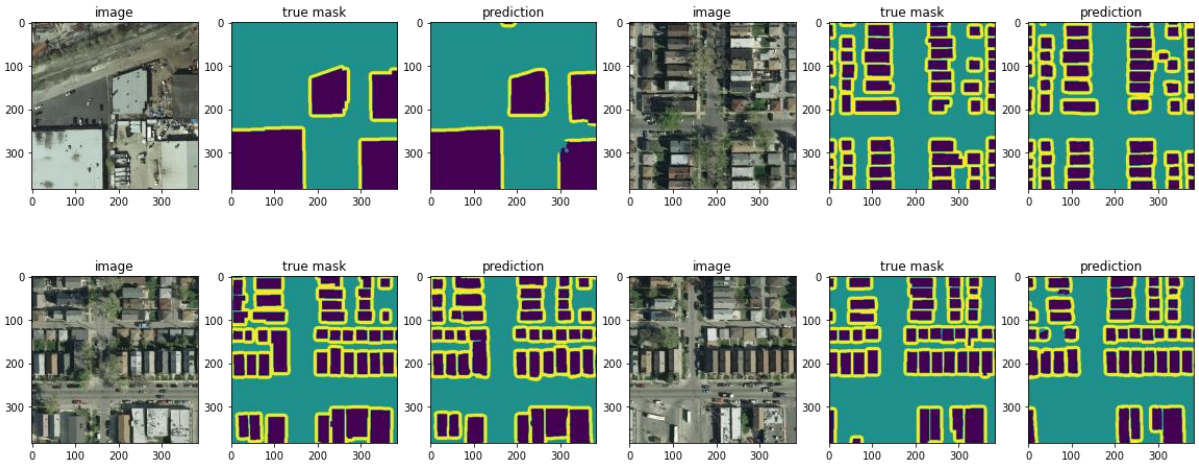

Fig 5.8 Sample prediction results of Chicago in INRIA dataset.

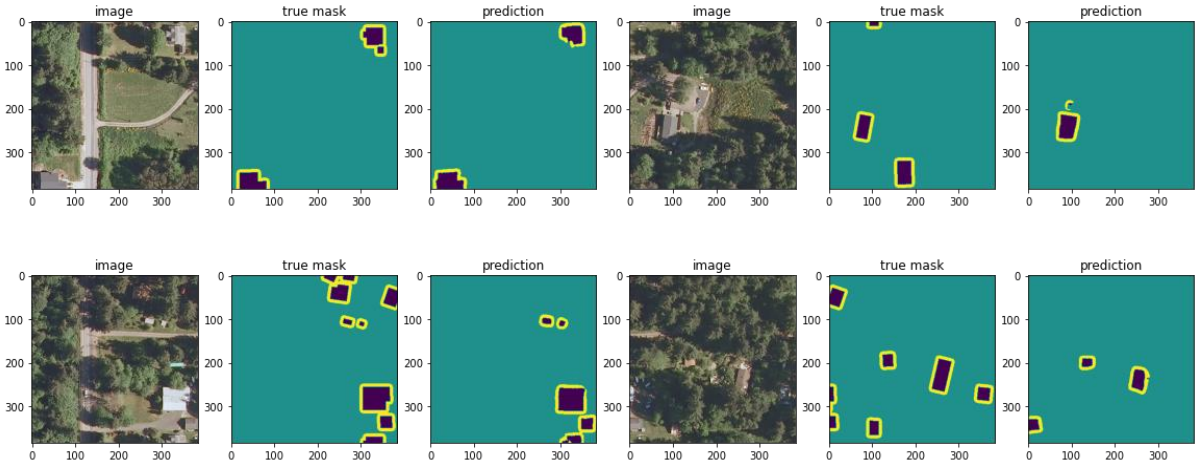

Fig 5.9 Sample prediction results of Kitsap in INRIA dataset.

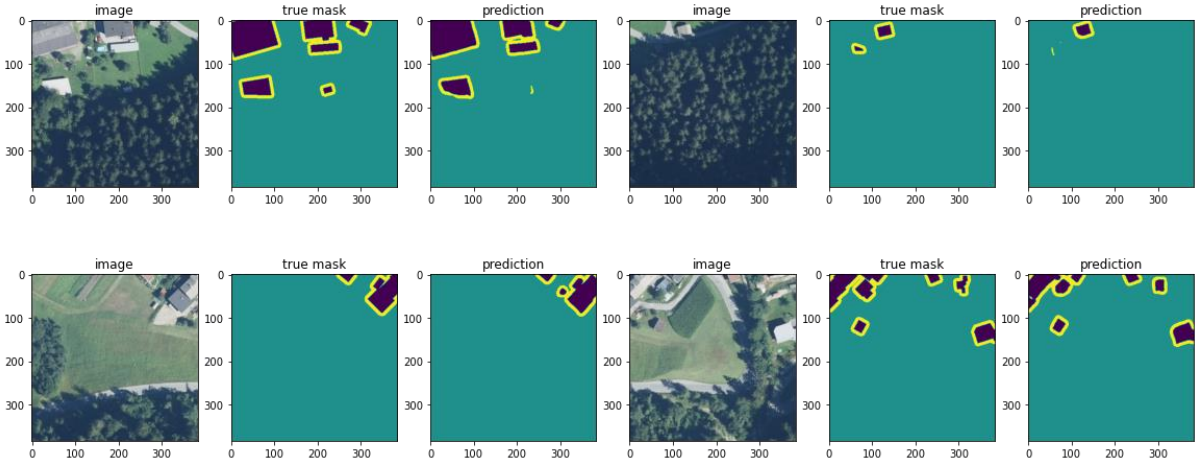

Fig 5.10 Sample prediction results of West Tyrol in INRIA dataset.

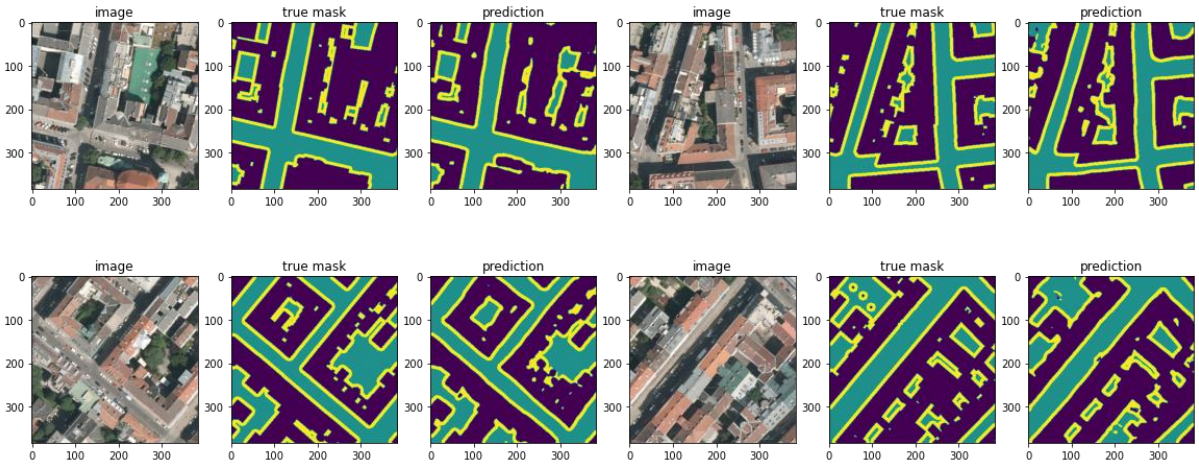

Fig 5.11 Sample prediction results of Vienna in INRIA dataset.

5.7 Benchmark on INRIA Dataset

The benchmark comparison of our model and other models is shown in Table 5.5. The results of the top two performing models for each column are highlighted. Our method outperforms most of the competitors as our IoU scores enter the top two rankings in all cities and even rank the highest in three of them. The fact that our model performs better in most cities compared to other state-of-the-art models implies that our approach is reliable and credible.

Table 5.5 Comparison of different methods. The top two performances of each column are highlighted.

	Austin		Chicago		Kitsap		West Tyrol		Vienna	
Metrics (%)	IOU	Acc	IOU	Acc	IOU	Acc	IOU	Acc	IOU	Acc
Our method	**87.54**	96.61	**84.06**	94.41	**76.27**	97.91	**87.94**	**99.08**	**87.27**	**94.56**
SegNet *[36]* (Multi-Task Loss)	72.43	95.71	77.68	**95.6**	72.28	95.81	64.34	**98.76**	76.15	94.48
2-levels *[36]* U-Nets	77.29	96.69	68.52	92.4	72.84	**99.25**	75.38	98.11	78.72	93.79
CT-UNet *[36]*	**87.62**	**97.28**	**82.87**	**97.31**	**85.28**	96.73	**84.14**	97.96	**86.73**	**96.23**
GMEDN *[37]*	80.53	**97.19**	70.42	92.86	68.47	**99.31**	75.29	98.05	80.72	94.54
FCN *[37]*	76.44	96.56	67.28	91.9	66.05	99.24	71.25	97.44	75.43	93.01

5.8 Performance on Custom Dataset

Using the parameters obtained from previous experiments, we achieved a rough IoU score of 70% on our custom Cyprus dataset. Inspecting the prediction results, we found that some predictions outperformed the manually created masks, meaning that the reduction in IoU score was caused by human error rather than misclassification. This underscores the importance of the dataset, but there is no doubt that similar errors might also occur in large-scale publicly available datasets, as no manual work can be guaranteed to be entirely correct.

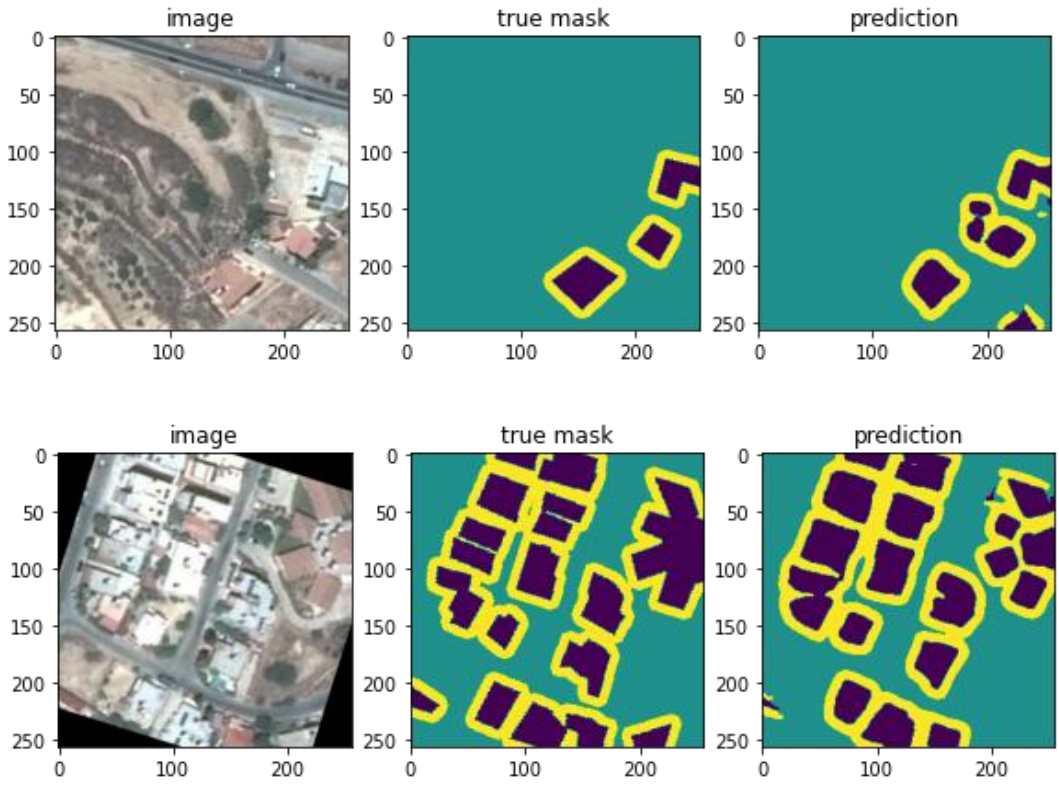

Fig 5.12 Two prediction results of our custom Cyprus dataset. Some predictions seem even more accurate than the human-labelled true mask.

6 Conclusion

This dissertation sought to develop a generic approach that could improve building extraction performance regardless of the model selection or data usage. A thorough literature review is presented and various neural network models are examined. Different datasets are discussed while their strengths and drawbacks are also analysed. We have selected the most reasonable dataset and one of the most reliable pre-trained models from a wide range of options to ensure that the experiments are convincing.

The main contribution of this dissertation is that a novel approach to improving building extraction is presented. We discovered that adding an additional class, the building border, can improve prediction accuracy and the boundaries' regularisation. The mechanism of this method is similar to multi-task learning, as introducing auxiliary tasks regularises the main task, improving the final results. Furthermore, the advantage of this approach over multi-task learning is flexibility. Multi-task learning requires a fixed model and specific use cases, with the need for customised training loss for different models. This is where our method stands out, as it could be applied in any model. In terms of regularising the building boundaries, our approach is significantly more straightforward than attention-based networks and does not require post-processing as [25] did; nevertheless, the quality of our regularised boundaries is comparable to theirs. Finally, we also develop a post-processing approach that utilises the fact that there are borders in our predictions to apply post-cleaning on incorrectly predicted results.

While the approach we presented is promising, several issues could still be improved upon in the future. As borders are an essential factor in such an approach, we have not yet discovered the relation between the width of the border with the data. The value we found was performed by trial and error, which might not be the optimal value for other cases. Being able to quickly determine the width for the added border will be a big leap for this approach. In addition, we believe that we have not fully utilised the existence of the borders. As the major feature of this approach, further post-processing could be carried out based on the additional border information we have. Finally, a minor flaw of the borders is that they might not be closed when initially coming out from the prediction, limiting the possibility of post-processing. Therefore, closing the borders in the predictions might also be a valuable research direction.

7 Bibliography

[1] A. Brock, J. Donahue and K. Simonyan, “Large scale gan training for high fidelity natural,” in *The International Conference on Learning Representations*, New Orleans, 2018.

[2] E. Kavlakoglu, “AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What’s the Difference?,” IBM, 27 May 2020. [Online]. Available: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks. [Accessed 10 September 2022].

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, in *IEEE conference on computer vision and pattern recognition*, 2009.

[4] C. Sebastian, R. Imbriaco, E. Bondarev and P. H. d. With, “Contextual pyramid attention network for building segmentation in aerial imagery.,” *arXiv preprint arXiv:2004.07018,* 2020.

[5] M. Awrangjeb, C. Zhang and C. Fraser, “Improved building detection using texture information.,” *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.,* vol. 38, pp. 143-148, 2011.

[6] B. Sirmacek and C. Unsalan, “Building detection from aerial images using invariant color features and shadow information.,” in *23rd International Symposium on Computer and Information Sciences*, Istanbul, 2008.

[7] Y. Li and H. Wu, “Adaptive building edge detection by combining LiDAR data and aerial images.,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 37 (Part B1), pp. 197-202, 2008.

[8] J. Long, E. Shelhamer and T. Darrell, “Fully convolutional networks for semantic segmentation,” *In Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 3431-3440, 2015.

[9] K. Chen, Z. Zou and Z. Shi, “Building Extraction from Remote Sensing Images with Sparse Token Transformers,” *Remote Sensing,* vol. 13, no. 21, 2021.

[10] D. Chen, S. Shang and C. Wu, “Shadow-Based Building Detection and Segmentation in High-Resolution Remote Sensing Image.,” *J. Multimed,* no. 9, p. 181–188, 2014.

[11] “Neural Networks,” IBM, 17 August 2020. [Online]. Available: https://www.ibm.com/cloud/learn/neural-networks. [Accessed 20 April 2022].

[12] S. Saha, “A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way,” Medium, , 16 December 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed 20 June 2022].

[13] M. L. f. A. I. Labeling, “V. Mnih,” University of Toronto, Toronto, 2013.

[14] N. Audebert, A. Boulch, B. L. Saux and S. L. , “Distance transform regression for spatially-aware deep semantic segmentation,” *Computer Vision and Image Understanding,* vol. 189, 2019.

[15] O. Ronneberger, P. Fischer and a. T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *In International Conference on Medical image computing and computer-assisted intervention,* pp. 234-241, 2015.

[16] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014.

[17] V. Badrinarayanan, A. Kendall and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE transactions on pattern analysis and machine intelligence,* vol. 39, no. 12, pp. 2481-2495, 2017.

[18] L. Mou and X. Zhu, “RiFCN: Recurrent network in fully convolutional network for semantic segmentation of high resolution remote sensing images,” 2018.

[19] T.-Y. Lin, P. Dollar, R. Girshick, B. H. Kaiming He and S. Belongie, “Feature Pyramid Networks for Object Detection,” *the IEEE conference on computer vision and pattern recognition,* p. 2117–2125, 2017.

[20] A. Marcu, D. Costea, E. Slusanschi and M. Leordeanu, “A multi-stage multi-task neural network for aerial scene interpretation and geolocalisation,” 2018.

[21] B. Bischke, P. Helber, J. Folz, D. Borth and A. Dengel, “Multi-Task Learning for Segmentation of Building Footprints with Deep Neural Networks,” 18 September 2017. [Online]. Available: arXiv:1709.05932v1. [Accessed 15 June 2022].

[22] S. Chattopadhyay and A. C. Kak, “Uncertainty, Edge, and Reverse-Attention Guided Generative Adversarial Network for Automatic Building Detection in Remotely Sensed Images,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 15, pp. 3146-3167, 2022.

[23] D. Marmanis, K. Schindler, J. Wegner, S. Galliani, M. Datcu and U. Stilla, “Classification with an edge: Improving semantic image segmentation with boundary detection,” *ISPRS Journal of Photogrammetry and Remote Sensing,* no. 135, pp. 158-172, 2018.

[24] D. Cheng, G. Meng, S. Xiang and C. Pan, “FusionNet: Edge aware deep convolutional networks for semantic segmentation of remote sensing harbor images,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 10, no. 12, pp. 5769-5783, 2017.

[25] K. Zhao, J. Kang, J. Jung and G. Sohn, “Building extraction from satellite images using mask R-CNN with building boundary regularisation,” *In Proceedings of the IEEE conference on computer vision and pattern recognition workshops,* pp. 247-251, 2018.

[26] V. Mnih, “Machine learning for aerial image labeling,” University of Toronto (Canada), 2013.

[27] SpaceNet on Amazon Web Services (AWS), “Datasets,” The SpaceNet Catalog, 1 October 2018. [Online]. Available: https://spacenet.ai/datasets/. [Accessed 30 May 2022].

[28] M. Luo, S. Ji and S. Wei, “A diverse large-scale building dataset and a novel plug-and-play domain generalisation method for building extraction,” 2022.

[29] OpenAI, “OpenAI dataset,” 2018. [Online]. Available: https://competitions.codalab.org/competitions/20100.

[30] E. Maggiori, Y. Tarabalka, G. Charpiat and P. Alliez, “Can Semantic Labeling Methods Generalise to Any City? The Inria Aerial Image Labeling Benchmark,” in *IEEE International Symposium on Geoscience and Remote Sensing (IGARSS)*, Fort Worth, United States, 2017.

[31] V. Iglovikov and A. Shvets, “TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation,” 2018.

[32] J. Hu, L. Li, Y. Lin, F. Wu and J. Zhao, “A comparison and strategy of semantic segmentation on remote sensing images,” *The International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery,* pp. 21-29, 2019.

[33] P. Iakubovskii, “Segmentation Models,” GitHub repository, 2019. [Online]. Available: https://github.com/qubvel/segmentation_models.

[34] doxygen, “Smoothing Images,” doxygen, [Online]. Available: https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html. [Accessed 31 July 2022].

[35] DigitalSreeni, “228 - Semantic segmentation of aerial (satellite) imagery using U-net,” Youtube, 28 July 2021. [Online]. Available: https://youtu.be/jvZm8REF2KY. [Accessed 20 August 2022].

[36] H. Ye, S. Liu, K. Jin and H. Cheng, “CT-UNet: An Improved Neural Network Based on U-Net for Building Segmentation in Remote Sensing Images,” *2020 25th International Conference on Pattern Recognition (ICPR),,* pp. 166-172, 2021.

[37] J. Ma, L. Wu, X. Tang, F. Liu, X. Zhang and L. Jiao, “Building extraction of aerial images by a global and multi-scale encoder-decoder network,” *Remote Sensing,* vol. 12, no. 15, p. 2350, 2020.