UNCERTAINTY, EDGE, AND REVERSE-ATTENTION GUIDED GENERATIVE ADVERSARIAL NETWORK FOR AUTOMATIC BUILDING DETECTION IN REMOTELY SENSED IMAGES

by

Somrita Chattopadhyay

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Electrical and Computer Engineering

West Lafayette, Indiana

May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Avinash C Kak, Chair

School of Electrical and Computer Engineering

Dr. Hong Tan

School of Electrical and Computer Engineering

Dr. Yung-Hsiang Lu

School of Electrical and Computer Engineering

Dr. Tanmay Prakash

Google Inc., Boston

Approved by:

Dr. Dimitri Peroulis

To Baba and Ma

"*It was easy to capture the dreams I ran after*

Because you believed in me" – Gene Watson

ACKNOWLEDGMENTS

"*We must find time to stop and thank the people*
who make a difference in our lives."

– John F. Kennedy

I am grateful to my professors, colleagues, family, and friends for their guidance, encouragement and support throughout my Ph.D. journey.

First and foremost, I would like to express my deepest gratitude to my amazing advisor Professor Avinash Kak for providing me with the opportunity to conduct research with him and be a part of his RVL (Robot Vision Lab) family. His excellent guidance and constant support have been instrumental towards my success. His valuable inputs throughout these years have not only helped me to grow as a researcher, but have also contributed immensely towards my personal growth. Most importantly, I am grateful beyond measure for his trust and belief in me for all these years. I never could have gotten this far, with as much ease, if it wasn't for him.

I would like extend my sincere thanks to the members of my advisory committee — Professor Hong Tan, Professor Yung-Hsiang Lu and Dr. Tanmay Prakash. Their insightful comments and constructive feedback have helped me in conducting meaningful research and writing my thesis. I would also like to acknowledge and thank Dr. David Hughes and Dr. Byung Hoon Park for being my mentors during my stay at the Oak Ridge National Laboratory as well as for providing financial support towards my Ph.D. program during the last 2 years. I have learnt a lot from them and have grown as a researcher under their guidance.

I consider myself extremely fortunate to have crossed paths with some extremely talented and passionate researchers at the RVL. I am specially thankful to Dr. Noha Elfiky (for being an awesome mentor in the first year of my Ph.D. program), Dr. Tommy Chang (for being the most kindhearted and helpful person I have ever met in my life), Dr. Bharath Comandur (for taking us on a ride into the world of deep learning) and Dr. Tanmay Prakash (for his profound research insights). I will always fondly remember the times spent in the lab

with Ankit, Fangda, Constantine, Rohan, Nader and Michael. A lion's share of my research insights can be attributed to their deep understanding about various topics in Computer Vision and Deep Learning. More importantly, I am grateful to them because they were always willing to help and share their knowledge with me.

My life at Purdue would have been incomplete without *my family away from home*. I made some lifelong friends during this phase of life. I feel that I wouldn't have survived the rocky roads of Ph.D. without my best friends, Krishnakali and Satarupa. They were with me in all ups and downs — cheering me up and supporting me through the good and bad times. I would always remember my eccentric friend, Annwesha, as the epitome of strength and ambition. I would like to thank Arindam, Prabudhya, Amartya, Indrani, Sanchari, Sreya, Debadrita, Srishti, Sreyansh, and Sayantan for providing me with some of the best memories of my life.

I would also take this opportunity to pay respect to my late Dida (grandmother) and late uncle (Chhoto) and aunt (Krishna). These are the people who extended their unconditional support towards me in the form of love, care, courage and motivation throughout my academic years.

And finally, I would like to thank and express my love to the closest people in my life — Prasanta Kumar Chattopadhyay (Baba/father), Dr. Ratna Chattopadhyay (Ma/mother) and Dr. Arnob Ghosh (husband). No amount of words are enough to express my gratitude towards them. But I want to tell them that — YOU mean the WORLD to me!

Everything I have and everything I am, I owe it all to my parents. They are my two lifelines — the pillars of my success. Their unparalleled love, care and support have helped me in shaping my life with passion and positivity. Their unwaning belief in me has made me believe that I can do anything and everything in life. I will be forever thankful to Baba and Ma for all the sacrifices they have made to see me as the person I am today. Without them this journey would not have been feasible.

Last but not the least, I am grateful to my husband, Arnob, for holding my hand and making me feel secured in the darkest hours of my life. Since the day we met till today, he has been a constant source of strength and inspiration. His unconditional support, boundless love and endless patience have kept me calm as well as motivated in the last few years of

my Ph.D. His technical insights have been immensely helpful in formulating my dissertation research. I cannot imagine being able to complete this journey without him. He has been and will always be my inspiration for his passion towards research, never-ending perseverance and never-losing-go-getter attitude.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Despite recent advances in deep-learning based semantic segmentation, automatic building detection from remotely sensed imagery is still a challenging problem owing to large variability in the appearance of buildings across the globe. The errors occur mostly around the boundaries of the building footprints, in shadow areas, and when detecting buildings whose exterior surfaces have reflectivity properties that are very similar to those of the surrounding regions. To overcome these problems, we propose a generative adversarial network based segmentation framework with **uncertainty attention unit** and **refinement module** embedded in the generator. The refinement module, composed of edge and reverse attention units, is designed to refine the predicted building map. The edge attention enhances the boundary features to estimate building boundaries with greater precision, and the reverse attention allows the network to explore the features missing in the previously estimated regions. The uncertainty attention unit assists the network in resolving uncertainties in classification. As a measure of the power of our approach, as of January 5, 2022, it ranks at the second place on DeepGlobe's public leaderboard despite the fact that main focus of our approach — refinement of the building edges — does not align exactly with the metrics used for leaderboard rankings. Our overall F1-score on DeepGlobe's challenging dataset is 0.745. We also report improvements on the previous-best results for the challenging INRIA Validation Dataset for which our network achieves an overall IoU of 81.28% and an overall accuracy of 97.03%. Along the same lines, for the official INRIA Test Dataset, our network scores 77.86% and 96.41% in overall IoU and accuracy. We have also improved upon the previous best results on two other datasets: For the WHU Building Dataset, our network achieves 92.27% IoU, 96.73% precision, 95.24% recall and 95.98% F1-score. And, finally, for the Massachusetts Buildings Dataset, our network achieves 96.19% relaxed IoU score and 98.03% relaxed F1-score over the previous best scores of 91.55% and 96.78% respectively, and in terms of non-relaxed F1 and IoU scores, our network outperforms the previous best scores by 2.77% and 3.89% respectively.

1. INTRODUCTION

1.1 Motivation

The current decade has witnessed a growing interest in processing remotely sensed imagery at a large scale, often the entire earth at once, for the purpose of extracting meaningful information related to the earth's resources and environment. The plethora of information that is obtained from the numerous high-resolution drones and satellite images are being used for continuous 24/7 surveillance of the globe and for monitoring various applications such as image fusion, change detection and land cover classification. Interpretation and analysis of remotely sensed data are essential for availing the wealth of useful information it has to offer, and involves the identification of various targets in an image and extracting their semantic information from the image such as the location of objects like roads and buildings, changes in land-cover, population estimates, climate modeling, etc.

The work described in this dissertation specifically focuses on the task of semantic labeling of building footprints from high-resolution satellite and aerial images. Semantic labeling of building footprints refers to the task of assigning every pixel in an image to building or non-building class. Semantic segmentation of buildings from remote sensing images is of profound importance to a myriad of applications including but not limited to disaster risk management, map revision, urban planning, autonomous navigation, crop and forest management. Much of this semantic labeling work, however, is still performed by human experts. One of the key objectives in this dissertation is to mitigate any human labor for building segmentation task. To this end, we aim to develop an automatic building footprint extraction framework. We would also like to highlight the fact that the novel concepts proposed by us in this dissertation for improving the current state-of-the-art building detection algorithms can be applied to the detection of many other objects as well.

While a great deal of progress has already been made in the automatic detection of building footprints in aerial and satellite imagery, several challenges still remain. Most of these can be attributed to the high variability in how the buildings show up in such images in different parts of the world — man-made structures like buildings are often built in different materials and with different structures, leading to an incredible diversity of colors, sizes,

Figure 1.1. Illustration of some issues prevalent in current state-of-the-art building segmentation approaches. (a) Diversity in building appearance across the globe. (b) Similar spectral signatures of background and foreground pixels. (c) Errors mostly occur near the boundaries of buildings. (d) Errors due to occlusion from high vegetation. (e) Errors due to the presence of shadows.

shapes, and textures. Problems are also caused by the fact that the reflectivity signatures of several types of building materials are close to those for the materials that are commonly used for the construction of roads and parking lots. Moreover, difficulties in automatic building detection also arise by the effect of shadows on the sensed data, and by the presence of occlusions caused by nearby tall structures and high vegetation. An illustration of these issues can be found in Figure 1.1.

Traditionally, the algorithms for discriminating between the buildings and the background have relied heavily on the domain knowledge for the purpose of extracting handcrafted spatial or spectral features such as texture, shapes, color, etc. However, such manually specified features are not always robust to illumination changes, shadows and occlusions — especially considering the possible wide variations in building shapes, sizes, and the locale-based density of the buildings. Fortunately, with the advent of deep learning, there is no longer a need for such handcrafted features. While the results obtained with the deep learning based methods [1]–[4] are indeed impressive in relation to what could be done before with the traditional methods, several challenges remain.

With regard to the performance of the deep-learning based methods for building detection, the commonly used metrics used for evaluating the algorithms only ensure that the bulk of the building footprints is extracted. The metrics do not enforce the requirement of contiguity of the pixels that belong to the same building [1], [5]–[9]. This has led some researchers to formulate post-processing steps like the Conditional Random Fields (CRFs) [10], [11] during inference for invoking spatial contiguity in the output label maps.

Even more importantly, the semantic-segmentation metrics for identifying the buildings are silent about the quality of the boundaries of the pixel blobs [5], [7], [12]–[15]. Since the number of pixels at the perimeter of a convex shape is roughly proportional to the square-root of the pixels in the interior, incorrectly labeling even a tiny fraction of the overall building pixels may correspond to an exaggerated effect on the quality of the boundary.

These problems related to enforcing the spatial contiguity constraint and to ensuring the quality of the building boundaries only become worse in the presence of confounding factors such as shadows, the similarity between the reflectivity properties of the building exteriors and their surroundings, etc.

In this dissertation, we focus on developing an automatic building detection framework which can solve the above challenges i.e. extract accurate boundaries of the buildings, and can be generalized for building segmentation task to any city across across the globe. We aim to develop an architecture that performs well on both satellite as well as aerial images.

1.2 Our Method

To solve the challenges mentioned in Section 1.1, we propose a new generative adversarial network (GAN) [16] for automatically segmenting building footprints from high-resolution remotely sensed images. We adopt an adversarial training strategy to enforce long-range spatial label contiguity, *without adding any complexity to the trained model during inference*. In our adversarial network, the discriminator is designed to correctly distinguish between the predicted labels and the ground-truth labels and is trained by optimizing a multi-scale L_1 loss [17]. The generator, an encoder-decoder framework with embedded *uncertainty attention*

and *refinement modules*, is trained to predict one-channel binary maps with pixel-wise labels for building and non-building classes.

Our network incorporates several novel ideas, such as the *Uncertainty Attention Unit* that is introduced at each data abstraction level between the concatenation of the encoder feature map with the decoder feature map. This unit focuses on those feature regions where the network has not shown confidence during its previous predictions. This is likely to happen at the building boundaries, in shadow areas, and in those regions of an image where the building pixels look very similar to the background pixels.

Another novel aspect of our network is the *Refinement Module* that consists of a *Reverse Attention Unit* and an *Edge Attention Unit*. This module is introduced after each stage in the decoder to gradually refine the prediction maps. Starting with the bottleneck layer of the encoder-decoder network and using an Atrous Spatial Pyramid Pooling (ASPP) [7] layer, the network first predicts a coarse prediction map that is rich in semantic information but lacks fine detail. The coarse prediction map is then gradually refined by adding residual predictions obtained from the two attention units in each stage of decoding. The Edge Attention Unit enhances the boundary features, and, thus, helps the network to learn precise boundaries of the buildings. Specifically, the edge attention tries to improve the corrupted semantic boundary at the previous layer's building prediction using the new spatial information available at the current layer. And the Reverse Attention Unit allows the network to explore the regions that were previously classified as non-building, which enables the network to discover the missing building pixels in the previously estimated results. The idea of the reverse attention is to reconsider the predictions coming out of a lower-indexed layer in the decoder in light of the spatial detail available at the current layer.

In addition to the adversarial loss, we also use deep supervision in our architecture for efficient back propagation of the gradients through the deep network structure. By deep supervision, we refer to the losses computed for each intermediate prediction map. These intermediate losses are added to the final layer's loss. To stabilize the training of our GAN and boost the performance of our generator, we compute weighted dice loss and shape loss for the final prediction map as well as for each intermediate prediction map.

In the experimental results that we will report in this dissertation, the reader will see significant performance improvements over the previous-best results for four different datasets, two of which are known to be challenging (DeepGlobe [18] and INRIA [19]), and two others that are older but very well known in semantic segmentation research (WHU [20] and the Massachusetts Buildings Dataset [5]). In addition to generating accurate semantic labels of building footprints from aerial imagery, our results on the INRIA Aerial Image Labeling Dataset [19] verifies the generalization capability of our proposed network — in this dataset, the cities included in the test subset are different from those of the training subset. Impressive performance of our algorithm on the INRIA dataset validates that our proposed network once trained on a certain dataset, is capable of generalizing to other areas of the earth. Moreover, for the Deepglobe Building Detection Dataset, as of January 5, 2022, our method ranks at the second place on DeepGlobe's public leaderboard despite the fact that main focus of our approach — refinement of the building edges — does not align exactly with the metrics used for leaderboard rankings. These results demonstrate the superiority of our proposed framework on diverse datasets including aerial and satellite images.

1.3 Primary Contributions

Towards solving the above mentioned issues in the current state-of-the-art building footprint extraction algorithms from remotely sensed images, we put forth the following contributions:

1. We propose a novel *Refinement Module* and embed the module in the fully convolutional encoder-decoder generator network of our GAN framework. The *Refinement Module* consists of a *Reverse Attention Unit* and an *Edge Attention Unit*.

    (a) The *Edge Attention Unit* is designed to amplify the boundary features, and, thus, helps the network to learn precise boundaries of the buildings.

    (b) The *Reverse Attention Unit* allows the network to explore the regions that were previously classified as non-building and enables the network to discover the missing building pixels in the previously estimated results.

We introduce the Refinement Module after each stage in the decoder for refining the 'intermediate prediction maps' gradually by recovering the fine details lost during encoding. We explain what we mean by 'intermediate prediction maps' in the next paragraph.

Starting with the bottleneck layer of the encoder-decoder network, the encoded features extracted from the Atrous Spatial Pyramid Pooling (ASPP) layer predict the top-most prediction map that is at low resolution but rich in semantic information. The decoder starts with this coarse prediction map and looks back at it in the next layer of the decoder where additional image detail is available for improving the prediction probabilities that were put out by ASPP and for improving the edge detail associated with the predictions. The former is accomplished by the Reverse Attention Unit and the latter by Edge Attention Unit. The refined prediction maps that we obtain at each level of decoding are referred to as the 'intermediate prediction maps'.

Specifically, this module learns residual predictions after every stage of decoding and gradually refine the prediction map estimated in the previous stage until the final prediction map is obtained. What's important here is the fact that the Refinement Module focuses on those regions of an image where the accuracy of semantic segmentation is likely to be poor — in the vicinity of building boundaries, shadow and occluded areas. Details of the Refinement Module is provided in Section 5.2.1 of Chapter 5.

2. We propose a novel *Uncertainty Attention Unit* and add it into the generator of our GAN-based framework. This unit assists the network in resolving uncertainties in classification.

   A classical encoder-decoder network does not provide for feature selection when fusing together the encoder and decoder features through the skip connections. Over-segmentation may occur in the final output due to indiscriminately fusing the low-level features from the encoder with the high-level features in the decoder.

   To mitigate against over-segmentation, we introduce this uncertainty attention unit in every encoder-to-decoder skip connection. The purpose of this attention unit is to

mediate the level of inclusion for the encoder-generated low-level features when they are copied over to the decoder side. More specifically, this unit uses the low-level detail made available by the encoder only in those regions of a prediction map where the degree of uncertainty exceeds a threshold. We use pixel-wise entropy as a measure of this uncertainty.

We emphasize on the fact that the Uncertainty Attention Unit focuses on those feature regions where the network has not shown confidence during its previous predictions — that is likely to happen at the boundaries of the building shapes, in shadow areas, and in those regions of an image where the building pixel signatures are too close to the background pixel signatures.

3. We introduce an Atrous Spatial Pyramid Pooling (ASPP) layer just after the bottleneck of our encoder-decoder segmentation framework. In literature, ASPP has been used to capture the global contextual information so that we can get more accurate pixel-wise predictions. However, to the best of our knowledge, ASPP has not been applied in the building segmentation context.

   In the context of detecting buildings from remotely sensed images, ASPP proves to be very useful. In the same overhead imagery, there can be very large building footprints; while some of the building footprints can be extremely small. Atrous convolutions are suitable for segmenting these unevenly distributed targets because atrous convolutions involve extracting features at multiple scales by exploiting different dilation rates.

   Our ASPP layer consists of a 1×1 Conv layer, three 3×3 Conv layers with dilation rates of 2, 4, and 6, and a global context layer incorporating average pooling and bilinear interpolation. The resulting feature maps from the five layers of ASPP are concatenated and passed through another 3×3 Conv layer, where they form the output of the ASPP layer that is fed directly into the decoder.

4. We introduce deep supervision in our architecture for efficient back propagation of the gradients through the deep network structure. As mentioned briefly earlier, we produce prediction maps at each level of decoding and refine these intermediate prediction maps

hierarchically in a top-down fashion to produce the final prediction map. By deep supervision, we refer to the losses computed for each of these intermediate prediction maps. These losses are added to the final layer's loss. Deep supervision allows for more direct backpropagation of loss to the hidden layers of the network and guides the intermediate prediction maps to become more directly predictive of the final labels.

In an encoder-decoder framework, concatenating shallow encoder features with deep decoder features can adversely affect the predictions if the semantic gap between the features is large. And, it stands to reason that introducing uncertainty attention prior to concatenation has the possibility of amplifying this problem by injecting "noisy" encoder features in those regions of a building prediction map where the probabilities are low. Deep supervision guards against such corruption of the prediction maps by forcing the intermediate feature maps to be discriminative at all levels of the decoder.

5. Our proposed method for building segmentation achieves significant improvement over the previous-best results for 3 publicly available datasets for detecting building footprints in high altitude aerial images — the challenging INRIA Aerial Image Labeling Dataset [19], the Massachusetts Buildings (MB) Dataset [5], and WHU Building Dataset [20]. These datasets cover different regions of interest across the world and include diverse building characteristics.

   For the challenging INRIA Aerial Image Labeling Validation Dataset, our network achieves an overall IoU of 81.28% and an overall accuracy of 97.03%. Along the same lines, for the official INRIA Test Dataset, our network scores 77.86% and 96.41% in overall IoU and accuracy. Our performance on this dataset also demonstrates that our proposed network can be generalized to detect buildings in different cities across the world without being directly trained on each of them.

   For the WHU Building Dataset, our network achieves 92.27% IoU, 96.73% precision, 95.24% recall and 95.98% F1-score. And, for the Massachusetts Buildings Dataset, our network achieves 96.19% relaxed IoU score and 98.03% relaxed F1-score over the previous best scores of 91.55% and 96.78% respectively, and in terms of non-relaxed

F1 and IoU scores, our network outperforms the previous best scores by 2.77% and 3.89% respectively.

6. We show that our proposed segmentation technique performs equally well on satellite images. To the best of our knowledge, this is the first work that performs well on both aerial as well as satellite images. Our overall F1-score on the challenging DeepGlobe Building Detection Dataset [18], [21] is 0.745.

   The power of our approach is best illustrated by its ranking at number 2 in the "DeepGlobe Building Extraction Challenge" at the following website:[1]

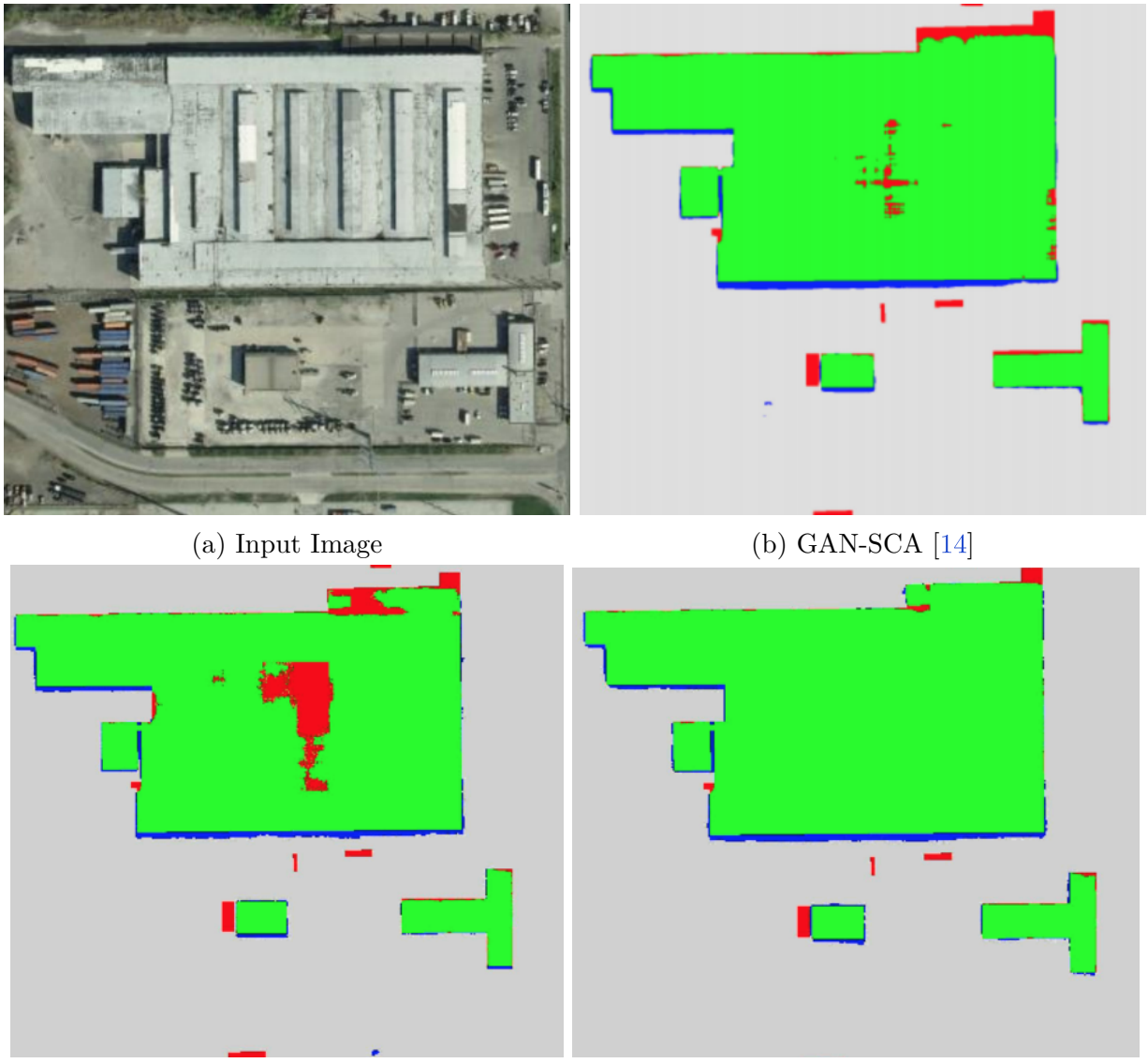
   https://competitions.codalab.org/competitions/18544#results

While our performance numbers presented in the Results section speak for themselves, we provide a visual example of the improvements in the quality of the building prediction maps produced by our framework. Figure 1 shows a typical example.

1.4 Organization of the Dissertation

This dissertation is organized as follows. In Chapter 2, we first review popular state-of-the-art semantic segmentation algorithms. Then we discuss the best performing algorithms for detecting building footprints in remotely sensed images and elaborate on the distinctive features of our proposed algorithm in relation to those works. As mentioned previously in the Introduction, our primary contributions involve intelligent use of attention mechanism for solving the issues in current state-of-the-art building segmentation algorithms. Chapter 3 discusses the evolution of attention mechanism in deep learning, and explains how we use attention in the context of building segmentation. Chapter 4 prepares the base for our semantic segmentation framework. This chapter makes the reader familiar with the

[1]↑Our entry is under the username 'chattops' with the upload date November 30, 2021. As mentioned earlier in the Introduction, the metrics used in all such competitions only measure the extent of the bulk extraction of the pixels corresponding to the building footprints. *In other words, these metrics do not directly address the main focus of our dissertation, which is on improving the boundaries of the extracted shapes and the contiguity of the pixel blobs that are recognized as the building pixels.* Nonetheless, it is noteworthy that improving the boundary and the pixel contiguity properties also improves the traditional metrics for building segmentation.



(a) Input Image

(b) GAN-SCA [14]

(c) Our baseline network with no attention units

(d) Our network with attention units

Figure 1.2. Comparing segmentation results using our approach and another state-of-the-art approach (GAN-SCA) on an image patch over Chicago from the INRIA Dataset. Green: True positives ; Blue: False Positives; Red: False negatives, Grey: True negatives.

necessary deep learning terms and concepts that we have used to build our building segmentation framework. Chapter 5 gives a detailed description of our network architecture and its various components. We explain our training strategy and the loss functions used in Chapter 6. In Chapter 7, we describe in details the four publicly available datasets – Massachusetts Buildings (MB) Dataset [5], INRIA Aerial Image Labeling Dataset [19], WHU Building Dataset [20] and DeepGlobe Building Detection Dataset [18], [21] – on which we have shown our experimental evaluations. Chapter 8 describes our experimental setup. We also report the details of our data pre-processing and post-processing strategies in this chapter. Subsequently, extensive quantitative and qualitative evaluations of our proposed method are presented in Chapter 9. We conduct a detailed discussion about our results and present an ablation study involving various components of our network in Chapter 10. Concluding remarks and possible future directions are discussed in Chapter 11.

The work presented in this dissertation has been submitted for publication in the *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. The online version of the paper can be found in [22].

2. STATE-OF-THE-ART SEMANTIC SEGMENTATION NETWORKS AND AUTOMATICALLY SEGMENTING BUILDING FOOTPRINTS FROM REMOTELY SENSED IMAGES

The topics covered in this chapter gives a general overview of the works done in the field of semantic segmentation and focuses on the approaches that are most relevant to our current work i.e. detecting and segmenting building footprints from remotely sensed imagery. In Section 2.1, I highlighted the popular semantic segmentation algorithms, and in Section 2.2, I highlighted the popular building segmentation framework on the remotely sensed images. Finally, I highlight the main contributions of my proposed framework compared to the state-of-the-art approaches Section 2.2.3.

2.1 An Overview of Popular Semantic Segmentation Algorithms

Semantic Segmentation is the process of associating every pixel of an image with a class label. This type of segmentation treats multiple objects of the same class as a single entity. Such semantically segmented images are useful for a variety of applications, such as face detection, medical image analysis, and video surveillance. In this Section, we explore some popular methods to perform semantic segmentation using classical as well as deep learning based approaches.

2.1.1 Classical Methods for Semantic Segmentation

In literature, numerous algorithms have been developed for image segmentation. The earliest methods of semantic segmentation include thresholding [23], region growing and split-merge [24], [25], clustering [26]–[28], watersheds [29], [30] and edge-extraction [31]–[33] based approaches.

Thresholding based methods are the simplest form of image segmentation algorithms where image pixels are labeled into different classes based on their intensity values. For a binary segmentation task, this is achieved by selecting a threshold, and setting the pixel

values lower than this threshold to 0 and the higher ones to 1. The most popular way of choosing the threshold is the Otsu method [23] — an automatic threshold deciding method based on the image histogram.

Region-based segmentation algorithms look for similarities between neighboring pixels and group them under a common class. Region-growing [25] and split-merge [24] techniques are the two most used region-based segmentation methods. In region-growing methods, segmentation procedure starts with some initial seed pixels and the algorithm works by detecting if immediate boundaries of the seed pixels are similar or dissimilar. The immediate boundaries are then chosen as seeds and the process repeats till a complete segmented image is obtained.

The split-merge method is the opposite of Region growing. It starts with the whole image as 1 region and splits it into sub-regions. Whenever a region in a sub-region is non-homogeneous, it is again divided. If neighboring sub-regions are homogeneous, they are merged. This continues until all the regions are homogeneous. Watershed segmentation [29], [30] is another region-based method that uses morphological operations to perform segmentation. It starts with selection of seed points inside each object present in an image, including the background. Then these regions are grown using a morphological watershed transformation [29].

Typically, in edge-based methods [31]–[33], segmentation is performed by filling up the holes in an edge map of an image using Morphology. Clustering based methods [26]–[28] have been very successful in image segmentation. Clustering refers to the process of grouping data according to their similarities and obtaining different clusters. The most popular method in this category is the k-means clustering [26] which is an unsupervised clustering algorithm. The K-means algorithm starts with 'k' randomly selected centroids, the initial points 'k' clusters, and then iteratively optimizes the positions of the centroids to generate the final clusters.

The more advanced classical semantic segmentation techniques include active contours [34], graph cuts [35], [36], conditional and Markov random fields [37], and sparsity based [38] approaches.

Active contour models [34], also known as 'snakes' were introduced to identify uneven shapes in images. In this segmentation technique, energy forces and constraints are used to dissociate certain pixel from an image for further processing. Active contours can generate smooth closed contours in images. In Graph-cut methods [35], [36], an image is first transformed into a graph where each pixel is connected to its neighbor, the source and the sink. As the name suggests, a cut of the graph divides an image into foreground and background pixels. Edge weights decide whether a pixel is likely to have the same label as its neighbors.

Markov and Conditional Random Fields [10], [37] are also used widely in image segmentation task. They are a class of statistical modelling methods used for structured prediction. The strength of such an approach over the former approaches is that they consider relation between pixels prior to making predictions.

Though all the above methods were successful in image segmentation, the main problem with all these approaches was that they relied heavily on the prior knowledge for the purpose of extracting hand-crafted spatial or spectral features such as texture, shapes, color, etc. However, such manually specified features are not always invariant to illumination changes. Moreover, these features are not robust to the presence of shadows and occlusion. Thus, the classical segmentation algorithms often failed when dealing with complex scenes.

Fortunately, with the advent of deep learning, there is no longer a need for such hand-crafted features. Deep neural networks have shown remarkable performance for semantic segmentation task, and have achieved state-of-the-art results on popular benchmarks— leading to a paradigm shift in the field of image segmentation.

2.1.2 Deep Learning Based Methods for Semantic Segmentation

The past decade of research in image segmentation methods has witnessed the deep learning based approaches [7], [12], [39]–[49] outperforming the classical approaches [50], [51] that relied heavily on domain knowledge to extract hand-crafted spatial or spectral features like edges, texture, shape, etc. for image segmentation. In this Section, we review some of the popular categories of deep learning based semantic segmentation methods.

Long et al. was the first to use deep learning for semantic segmentation, using Fully Convolutional networks (FCN) [52]. FCNs are formed of only convolutional layers where features are extracted by convolving a weighted kernel. In a FCN-based segmentation framework, the input image is first downsampled to an encoded representation through some convolutional layers. Then, the encoder output is upsampled using interpolation or deconvolution to produce a segmentation map of same resolution as the input image. In order to generate accurate segmentation maps, these approaches use long skip connections to fuse semantic information from the deep layers with fine details from the shallow layers. Though FCN-based frameworks have been very successful in producing detailed segmentation maps, they suffer from checkerboard artifacts arising from deconvolution operations. Moreover, the boundaries of segmented objects are not precise due to lossy encoding process.

To address the poor localization property of FCNs, Conditional Random Fields (CRF) and Markov Random Fields (MRF) were used as a post-processing steps [53], [54]. CRFs and MRFs are probabilistic framework for labeling and segmenting structured. They try to model the dependency between pixels, e.g., neighboring pixels are likely to have the same label, similar color pixels are more likely to be from the same class, things looking like boats are probably going to appear near things looking like water, etc. Typically, the final layers of CNNs were combined with these graphical models to refine the results through iterations and improve the accuracy of the final prediction map.

Another popular approach which is used for segmentation is an encoder-decoder framework [39], [41], [55]. This kind of architecture became popular with the introduction of the SegNet paper by Badrinarayanan et al. [55]. Segnet proposed an encoder that combines convolutional layers and downsampling layers to squeeze information into a bottleneck. The decoder uses a deconvolutional network which reconstructs the bottleneck output to generate a map of pixel-wise class probabilities. The most popular encoder-decoder network for semantic segmentation is the UNet [39]. UNet is introduced to segment biological microscopy images. It has a contracting part for capturing details and a symmetric expanding path for enabling precise localization. UNet uses skip connections from the convolutions blocks on the encoding side to the transposed-convolution blocks on the decoding side at the same

level. This particular way of using skip connections allows for better gradient flow through the network guiding it to learn multi-scale information.

Using multi-scale information has been another attractive approach for segmentation tasks. Some deep learning models [44], [56], [57] explicitly utilized multi-scale information for improving the state-of-the art of semantic segmentation. The most popular work in this category is the Feature Pyramid Network (FPN). FPN constructs pyramid of features. The shallow and deep features are concatenated from a bottom-up pathway, a top-down pathway and lateral connections. Outputs are generated in each stage by applying 3×3 convolution on the concatenated features. The predictions are obtained at each stage of the top-down pathway.

Another popular category of deep segmentation models is the Dilated Convolutional Model and DeepLab Family [7], [49], [58], [59]. The core concepts of this category are dilated convolution and Atrous Spatial Pyramid Pooling (ASPP). Dilated convolution is a technique that expands the convolutional kernel by inserting holes between its consecutive elements to cover larger area of the input image, thus, increasing the receptive field of the kernel. ASPP is a technique of resampling a given feature layer at multiple dilation rates in parallel prior to convolution. This is equivalent to probing the input image with multiple filters with complementary field of view. This helps the network to detect targets of different sizes efficiently by capturing multi-scale contextual information. The DeepLab family (DeepLabV1, DeepLabV2 and DeepLab V3) proposed the concepts of dilated convolution and ASPP unit to resolve the issue of loss in resolution in a network caused by pooling and striding operations.

Recurrent Neural Network Based Models [60]–[62] have also been very successful in semantic segmentation. These methods focus on learning long-range and short-range relationships between the image pixels to improve the performance of segmentation. The first RNN-based segmentation model was the ReSeg by Visin et al. in [60]. ReSeg uses VGG16 network [63], followed by ReNet layers [64], to extract generic local features from the input image. These features are then upsampled to obtain the final segmentation map. The ReSeg paper uses Gated Recurrent Units [65] for their good balance of memory usage and compu-

tational power. Some other papers in this category [61], [62] use Long Short-Term Memory (lstm) network [66] as the recurrent unit.

Following the huge success of attention mechanism in the field on Natural Language Processing, attention based models [48], [67]–[71] are also becoming predominant in the field of semantic segmentation. Attention can help a model in learning importance of features at different positions and scales. Researchers are combining attention mechanisms and spatial pyramids to extract dense features for pixel classification [70]. Reverse attention technique [71] is used to capture the background features, thus, guiding the network to be aware of more discriminative features. Recently, self-attention mechanism is also used for semantic segmentation task [48] where the authors appended channel and position attention modules on top of dilated FCN to learn the semantic dependencies along channel and spatial axes.

Generative adversarial models [16] proposed by Goodfellow et al. in 2014 have found their applications in all fields of computer vision, including image segmentation [17], [72]–[74]. Typically, in GAN based segmentation, the generator is the segmentation network which outputs the segmentation map corresponding to an input image, and the disciminator tries to classify the generated maps as fake and the ground-truth label maps as real. Research has shown that such as adversarial training setup can improve the performance of segmentation.

The FCNs along with Active Contour Models (ACMs) have recently gained interest [75], [76]. Initially, ACMs [34] were mostly used as a post-processing step [76] i.e. refining the results of a fully convolutional network. However, recent approaches are proposing new loss functions [75] based on the global energy formulation principle of ACMs [34].

It is evident that deep learning has made semantic segmentation approaches extremely powerful and has paved the way for their easy adoption in real-world applications. However, these algorithms still suffer huge memory and time requirement during training. Extensive research is ongoing to handle these issues.

2.2 Semantic Segmentation of Building Footprints in Remotely Sensed Imagery

Inspired by the success of the deep learning based methods in all areas of image processing and computer vision, more recently the researchers have focused on developing neural

network based frameworks for detecting building footprints from high-resolution remotely sensed images [1], [4], [77]–[88]. In this Section, we discuss the state-of-the-art building footprint extraction algorithms.

Initially, deep learning based approaches employed patch-based segmentation approaches for building detection. Mnih was the first to use a CNN to carry out patch-based segmentation in aerial images [5] and refined the segmentation results by using Conditional Random Fields (CRFs) [10] as post-processing process. Saito et al. in [6] also used a patch based CNN for road and building detection from aerial images, and outperformed Mnih's model on the Massachusetts Dataset.

However, the patch-based methods suffered from limited receptive field and large computational overhead, and required post-processing steps [10] to refine the segmentation results. Thus, the patch-based methods were soon surpassed by pixel-based methods [8], [9]. Maggiori et al. in [9] applied hierarchical fully convolutional network (FCNs) to perform pixel-wise prediction of building footprints. Khalel et al. in [8] proposed an architecture consisting of stacked U-Nets to perform pixel-to-pixel prediction of buildings in aerial images. The stacked U-Nets are designed end-to-end such that each U-Net improves on the results of the previous one. However, these approaches do not fully utilize the structural and contextual information of the ground objects that can help to distinguish the buildings from their complex heterogeneous backgrounds.

In order to alleviate the shortcomings listed above, researchers are exploring diverse techniques to detect building footprints overhead imagery. The shortcomings of the current state-of-the-art in deep learning based methods are being addressed by several ongoing research efforts [2], [20], [77], [82], [89]–[93]. To deal with the problem of huge variation in building sizes occurring in aerial images, Hamaguchi et al. [2] proposed a multi-task model that is an ensemble of multiple building detectors, each dedicated to a specific size building. In addition, the model implicitly utilizes context information by simultaneously training road extraction task along with building detection task. Most deep learning approaches suffer from high computational cost and humongous training time. These problems becomes more pronounced when handling large-scale high-resolution remotely sensed images. Lin et al. proposed a light-weight neural network named ESFNet [77] that can be trained in less

time, without high computational cost and large memory needs. ESFNet employs separable factorized residual blocks and utilizes the dilated convolutions to preserve slight accuracy loss with low computational cost and memory consumption.

Researchers have used multi-task learning to solve the challenges of semantic segmentation of buildings in remote sensed images. Marcu et al. in [78] proposed a cascaded multi-task multi-stage neural network framework where in the first stage, the network performs pixel-wise semantic segmentation, and in the second stage, the network learns to perform precise geo-localisation of the prediction obtained in the first stage. In [90], Bischke et al. introduced an uncertainty weighted multi-task loss based on the distance transform to preserve the building boundaries in semantic segmentation predictions.

On the other hand, the works reported in [82], [89]–[91], [94]–[96] deal with the preservation of the sharpness of the building boundaries. Liao et al. in [82] proposed a boundary-preserving building detection framework where both the contours and structures of the buildings are learned jointly. Specifically, they designed a structural feature constraint module to combine the structure and contour information of the buildings with multi-scale semantic features to extract accurate building boundaries. In [89], Hu et al. proposed a fully convolutional network for on-orbit semantic segmentation, and named it light-weight edge enhanced network (LEN). Zhao et al. in [91] added a building boundary regularization component to the Mask R-CNN framework for generating regularized polygons that are essential for many cartographic applications. Their proposed boundary regularized Mask R-CNN [45] achieved good results on the DeepGlobe Building Detection Dataset [18]. Another building segmentation approach focused towards preserving the details of the building boundaries is proposed by Zu et al. in [95]. They named the network – edge-detail-network or E-D-Net. The E-D-Net consists of 2 sub-networks — (a) an edge information generation network (E-Net) which focuses on capturing the edge details of the buildings, and (b) a detail recovery network (D-Net) that refines the output from E-Net to produce a the final segmentation map with higher quality. The authors also proposed a novel fusion strategy to combine the outputs of the two networks (i.e. the edge information and fine details) in a weighted manner. Recently, Jung et al. proposed a novel method to enhance the boundaries of building segmentation masks. They adopted the holistically-nested edge detection (HED) network

[97] to extract the edge features of the detected buildings. These edge features are then fed to a novel boundary enhancement (BE) module. Inside the BE module, two parallel sub-networks extract building boundary mask and segmentation mask. These masks are combined based on shared mutual information between the sub-networks. In [94], He et al. proposes a novel FCN-based architecture to accurately extract buildings from remotely sensed images. The central idea of the paper is to train a boundary learning network in parallel with the segmentation network and then fuse information from both these networks using a novel spatial variation fusion (SVF) module. Additionally, the authors use separable convolutions with large kernels to increase the receptive fields and reduce computational load. Furthermore, a convolutional block attention module (CBAM) [98] is used to improve the performance of the segmentation network.

Event though a large section of work has focused on building segmentation from remote sensed images, their performance suffered when the backgorund becomes more complex such as there are shadow, occlusion, presence of nearby buildings, or images with poor resolution. Recently, authors in [15], [20] attempted to detect buildings even when only a part of a building is visible. Ji et al. [20] proposed a Siamese U-Net with shared weights in two branches that aggregates context from multiple scales. Their model combines the segmentation maps of two different resolutions and produce scale-invariant predictions. Wang et al. proposed a neural network architecture for semantic segmentation where they used an U-shaped encoder-decoder architecture as the backbone and then, embeds an asymmetric pyramid non-local block between the backbone and the final classifier to capture global contextual information from high-resolution aerial images [15].

A variety of approaches have focused on incorporating contextual information that can provide critical cues for identifying buildings from the background even when a part of the building is visible due to the presence of obstacles such as shadow, cars and trees. In order to leverage on large-scale contextual information and extract critical cues for identifying building pixels in the presence of complex background and occlusion, researchers, recently, have proposed methods to capture local and long-range spatial dependencies among the ground entities in the aerial scene [92], [93]. In the field of land cover mapping from remote sensing images, Mou et al. [92] proposed a relation-augmented semantic segmentation network that

enables spatial and channel relational reasoning and learns long-range spatial relationships between any two spatial positions or features maps. Long-range spatial relationship is needed to leverage on comprehensive contextual information captured from the whole image. Zhang et al. proposed a dual-stream network (DS-Net) [93] consisting of a local and global branch that adaptively captures local and long-range information for the accurate mapping of building rooftops in VHR RS images. A spatial long-range dependency module is introduced in the global branch to capture the long-range dependencies between the ground entities in the aerial scene.

Several researchers are also using transformers [99], attention modules [15], [100]–[103] and multi-scale information [4], [11], [83], [85], [104], [105] for the purpose on detecting building footprints on the remotely sensed images. Details of the attention-based building segmentation approaches are provided in Section 2.2.2. Here we discuss some of the building segmentation approaches that utilize multi-scale information to improve performance.

The current fully convolutional networks (FCNs) often face difficulty in understanding whether the contrasting features are coming from different parts of the same building or from a building and its surroundings. Ran et al. proposed a building multi-feature fusion refined network (BMFR-Net) [83] in an attempt to overcome the previously mentioned limitation of FCNs. BMFR-Net consists of a continuous atrous convolution pyramid (CACP) module and a multi-scale output fusion constraint (MOFC) structure. The CACP module uses parallel continuous small-scale atrous convolution to enhance the continuity between local information and minimize the information loss in multi-scale feature extraction and fusion. The MOFC structure enhances the ability of the network to aggregate multi-scale semantic information.

In [85], Ma et al. introduced a new building detection approach named global multi-scale encoder-decoder network (GMEDN). The network consists of a local and global encoder and a distilling decoder. The distilling decoder is introduced to learn multi-scale information. The local and global encoder is designed to learn representative building features from aerial images. First, a VGG16 network [63] extracts local features from the input image and then, a non-local block is used to capture global information from the local feature maps. The

local and global information are fused to facilitate the segmentation of buildings with diverse shapes.

A remotely sensed image may contain buildings of different scales. It is difficult to extract multi-scale features using a single receptive field. To resolve this issue, Zhu et al. in [11] proposed a multi-scale semantic segmentation network (D-LinkNet) that is embedded within a multiscale-aware and segmentation-prior conditional random field (MSCRF) framework. The D-LinkNet with the help of multiple parallel dilated convolution modules integrates multiscale contextual information while preserving the building details during downsampling and the MSCRF framework help in obtaining precise boundaries by maintaining the continuity inside the buildings.

Although the state-of-the-art methods have been very successful in detecting building footprints accurately in remotely sensed images. Most of the frameworks have high computational cost and need a long time to train. Liu et al. in [4] proposed the ARC-Net to reduce the computation complexity and the model size. ARC-Net consists of residual blocks with asymmetric convolution that employ depth-wise separable convolution and asymmetric convolution with residual connections to reduce the computation load. The authors also use dilated convolutions and multi-scale pyramid pooling to enlarge the field of view of the network.

In [104], Wei et al. presented a FCN-based automatic building footprint extraction framework and a novel empirical polygon regularization technique to convert the predicted building segmentation maps to structured individual building polygons. The network uses convolutional feature pyramids to accumulate features from multiple scales. The predicted building maps are vectorized to irregular boundaries and then, polygonized using the Douglas–Peucker algorithm [106] before feeding them to the proposed polygon regularization algorithm.

Another paper that exploits multi-scale contexts for semantic understanding is [105] by Liu et al. In this paper, the authors introduce a novel spatial residual inception (SRI) module to capture and aggregate multi-scale contextual information. The SRI module can detect large buildings accurately and completely while retaining global morphological characteristics

and local details. The authors also use depth-wise separable convolutions and convolution factorization to shrink the size of the model.

Recently, multi-view satellite images [107], [108] are being used to improve semantic segmentation buildings in remotely sensed images. In Chapter 1 – the Introduction – of this dissertation, we mentioned that building detection from remotely sensed suffer from a variety of challenges. The authors in [107], [108] believed that a single view taken from directly overhead does not always have enough information to resolve the difficulties faced by the researchers while detecting buildings in remote sensing images. Moreover, many real-world scenarios demand the use of multi-view images, e.g., in scenarios such as natural disasters the first images are often from off-nadir views. Thus, Weir et al. in [104] presents an open source Multi-View Overhead Imagery dataset to address those problems. The dataset consists 62,000 overhead images — 2222 geographically unique image chips, each with 27 unique looks from a broad range of viewing angles ($-32.5°$ to 54.0) — collected over Atlanta, Georgia USA and the surrounding areas. In [107], Comandur et al. deals with multi-view satellite images and carries out a semantic segmentation of points on the ground. Therefore, ortho-rectification is fundamental to their work. Specifically, they present a novel multi-view CNN framework and a novel multi-view loss to combine information from multiple overlapping satellite images to semantically segment buildings and roads across large geographic regions. The authors also demonstrate the robustness of their approach to noisy training labels derived from OpenStreetMaps (OSM).

2.2.1 GAN Based Building Segmentation

GANs [16] are also gaining popularity in solving semantic segmentation problems. In GAN-based approaches for building detection [13], [14], [109], [110], the generator is basically a segmentation network that aims to produce building label maps that cannot be distinguished from the ground-truth ones by the discriminator. By training the segmentation and the discriminator networks alternatively, the likelihood associated with the joint distribution of all the labels that are possible at the different pixel locations can be max-

imized as a whole, which amounts to enforcing long-range spatial dependency among the labels.

In [13], Sebastian et al. illustrated how the use of adversarial learning can improve the already available best performing image segmentation models [7], [44] as GANs can enforce spatial label contiguity to refine the segmentation results without any time consumption during the inference.

Along roughly the same lines, Li et al. [109] adopted an adversarial training strategy to detect buildings in remote sensing images. In their network, the generator produces pixelwise image classification map using a fully convolutional DenseNet model, whereas the discriminator uses a simple autoencoder network to enforce forms of high-order structural features learned from ground-truth label map.

In [110], the authors used a SegNet model with Bi-directional Convolutional LSTM (BConvLSTM) as the generator network. The BConvLSTM module was added to the expansive part of the SegNet model to mix encoded features with higher resolution and local information and decoded features with more semantic information, which eliminate the noises and improve the performance of the model in building detection under complicated backgrounds.

The work presented in this dissertation comes closest to the approach adopted in [14] in which the authors have proposed a GAN with spatial and channel attention mechanisms to detect buildings in high-resolution aerial imagery. In this contribution, the spatial and the channel attention mechanisms are embedded in the segmentation architecture to selectively enhance important features on the basis of their spatial information in the different channels. In contrast with [14], our framework focuses the attention units where they are needed the most — these would be the pixels where the predictions are being made with low probabilities or low confidence.

2.2.2 Attention Based Building Segmentation

After the tremendous success of Transformers and attention mechanisms in the field of Natural Language Processing [111], [112], these concepts have started impacting the Com-

puter Vision community [113], [114] as well. Recently, researchers are using Transformers [99] and attention mechanisms [15], [100]–[103] for automatically extracting building footprints from high-resolution remotely sensed images. In this Section, we discuss some of the best performing building segmentation methods that apply the concept of attention.

In [99], Chen et al. proposed a novel building extraction framework that combines self-attention and reconstruction-bias modules in a U-Net-like architecture. The self-attention module is embedded in the encoder to focus on salient regions. Additionally, the proposed network uses a transformer module to learn the channel weights. While decoding, the network uses large kernels at multiple scale to enlarge the receptive fields and improve reconstruction ability.

Wang et al. in [15] presented the ENRU-Net for extracting both small and large building footprints efficiently from high-resolution aerial imagery. The main contribution of this paper is a novel non-local block named asymmetric pyramid non-local block (APNB). APNB is introduced between the ResNet-50 [115] backbone and the final classifier to capture global contextual information with the help of self-attention mechanism.

Though CNNs are widely applied for semantic segmentation of remote sensing imagery, they often fail to capture global contextual information that are essential for semantic understanding of these images. Sebastian et al. in [100] proposed a novel self-attention based contextual pyramid attention (CPA) module to capture multi-scale long-range spatial dependencies that are needed for segmenting buildings of different sizes accurately. CPA comprises of attentive multi-scale pathways. Each pathway utilizes non-local information to handle buildings of varying sizes and at the end, information from all the pathways are fused in a weighted manner. The authors also use a channel-wise attention unit to learn inter-dependencies across channels.

In [101], Zhang et al. proposed a novel end-to-end attention based building detection model – the DeepAttentionUnet. This paper also focuses on extracting buildings of different sizes and shapes accurately. Attention mechanism helps in enhancing the representative features of the buildings while suppressing the unimportant areas of the input images. The attention mechanism proposed by Ozan et al. in [116] is used in this paper since it is robust and can be easily integrated into other networks without much computational overhead.

Zhou et al. proposed the Pyramid Self-Attention Network (PISANet) [102] which can model both long-range as well as short-range dependencies. PISANet comprises of a backbone network and a pyramid self-attention module. The backbone network extract the local feature maps and aims to learn short-range spatial relations. Whereas, the pyramid self-attention module tries to model the long-range dependencies by extracting global features.

Guo et al. [103] proposed a novel deep-supervision based fully convolutional encoder-decoder network to extract building footprints from high-resolution remote sensing images. Deep supervision is introduced on top of a lightweight encoder to learn representative deep features from buildings of different scales. Deep supervision enables direct back-propagation of loss throughout the network, thus, allowing for multi-scale supervision. A scale attention module (SAM) is introduced to aggregate those multi-scale features and compute global-local attention of varying scales.

2.2.3 My Contribution

Despite the successes of the previous contributions mentioned in this section, the predicted building label maps are still found lacking with regard to the overall quality of building segmentation. At the pixel level, we still have misclassifications at a higher rate at those locations where the classification accuracy is most important — at and in the vicinity of the boundaries of the buildings and where there are shadows and obscurations. Furthermore, the methods that have been proposed to date tend to be locale specific. That is, they do not generalize straightforwardly to the different geographies around the world without further training.

In this research, we aim to overcome these shortcomings with the help of the refinement and the uncertainty modules that we embed in the segmentation network of our adversarial framework. The refinement module, composed of edge and reverse attention units, is designed to refine the predicted building map. The edge attention enhances the boundary features to estimate building boundaries with greater precision, and the reverse attention allows the network to explore the features missing in the previously estimated regions. The uncertainty attention unit assists the network in resolving uncertainties in classification.

We show empirically that our model *outperforms* the state-of-the-art models on well-known publicly available datasets [5], [18]–[21]. We empirically depict how each component (adversarial training, reverse attention, uncertainty attention) adds value to the performance using ablation study.

3. ATTENTION IN DEEP NEURAL NETWORKS

In this dissertation, we present a novel *attention enhanced GAN framework* for detecting buildings automatically in remotely sensed images. Specifically, our primary contributions involve proposing novel *attention units* to resolve the issues present in the state-of-the-art building segmentation approaches. As 'attention' forms the core of this dissertation, in this chapter we provide a background about the evolution and applications of 'attention mechanism' in the field of deep learning. Subsequently, I describe how 'attention' is used in the Computer Vision applications. Finally, I describe how I have applied 'attention' mechanism in my framework.

The concept of attention is of significance to different scientific disciplines. For decades, attention has permeated most areas of research such as neuroscience, psychology, and philosophy. Recent years have seen the ascent of attention in deep learning community — specifically, how attention is revolutionizing the domains of natural language processing and computer vision. In the context of deep learning, attention mechanism tries to mimic cognitive attention — the ability to focus selectively on discrete aspect of information relevant to a specific task, while ignoring other perceptible information.

When processing a complex scene or understanding an entire sentence, humans do not analyze the scene or the sentence in its entirety. Instead, humans tend to focus on the relevant parts of the scene or the sentence which would help them in faster analysis and comprehension of the scene or the sentence. In human beings, attention is a core property for all mental processes, from sudden reactive responses to complex mental processes like emotions, planning and reasoning. Environment is constantly providing us with unlimited supply of perceptual information which is much more than what we can process effectively. Given limited ability of our brain to process this endless amount of information, attention mechanism select, modulate, and focus on the most relevant information in a situation. Inspired by this capability of human brain, attention mechanism has become a hot research topic in the deep learning field.

Although the recent popularity of attention is often attributed to the field of natural language processing [111], the idea of mimicking human attention first originated in the field

of computer vision [117], [118]. The idea was to develop a model that would only focus on specific regions of images instead of the entire images. This would lead to reduction in the computational complexity of image processing while improving performance. Today attention mechanisms have a wide range of applications: Natural Language Generation [111], (machine translation, chatbots and multimedia description), Sentiment analysis [119], [120], Recommender Systems (user profiling for e-commerce), Speech Processing [121], [122], and Computer Vision (Image Captioning, Image Generation, Video Captioning) [114], [123].

Attention units can used as add-on components in neural networks and are easily trained in conjunction with a base model, such as a recurrent neural networks [124] or a CNN using standard backpropagation [111]. These attention units can model complicated interpretations into neural networks, thus improving their overall performance. The popularity of attention mechanisms peaked after the introduction of the Transformer model in [112]. In NLP, attention was originally introduced to replace the RNNs [125] which were difficult to be parallelized. Transformers demonstrates that the attention mechanism is sufficient to build a state-of-the-art model. This means that the drawbacks associated with RNNs can be eliminated. Recently, Vision Transformers [113] are introduced for image processing [126] and video processing [127] tasks.

In the upcoming sections of this chapter, we describe how attention is being used in the NLP community and in the Computer Vision community. I briefly explain the landmarks contributions involving attention in these communities with main focus on Visual Attention Mechanisms (i.e. attention in Computer Vision). Finally, I discuss my contribution in this space, and how I use attention to resolve the issues present in state-of-the-art building footprint detection approaches.

3.1 Attention in Natural Language Processing

It is evident from what we discussed earlier in this chapter, in the context of deep learning, 'attention' refers to a mechanism by which a network can weigh features according to the level of their importance to a task, and use this weighting to accomplish the task. Recent

years have seen soaring popularity of attention mechanism in the field of Natural Language Processing (NLP).

An early application of attention in NLP was in machine translation [111] where the aim was to translate a sentence in a source language (e.g., Spanish) to an output sentence in a target language (e.g., French). Attention was introduced to guide the decoder to utilize the most relevant parts (i.e. the vectors with the highest weights) of the input sequence in a flexible manner and improve the performance of the encoder-decoder model.

3.1.1 The Shortcomings of RNNs

Before the introduction of attention, RNN-based methods were used for machine translation. In such methods, an entire input sequence is read in and compressed into a fixed-length vector 'z' (shown in Figure 3.1). 'z' needs to capture all the information about the input sequence. RNN-based architectures work very well, especially with LSTM and GRU components; however, only for very small sequences.

In this approach, the decoder always have limited access to the information provided by the input. This becomes a huge problem while dealing with long and/or complex sequences. A long complex sentence encoded into a fixed-length vector 'z' inevitably leads to information loss. Moreover, RNNs tend to forget information from timesteps that are far behind. Thus, 'z' cannot encode information from all the input time-steps. Furthermore, the stacked RNN layers usually suffer from the vanishing gradient problem. All these shortcomings of the RNNs results in inaccurate language translation.

3.1.2 Attention to Rescue

To resolve this issue, Bahdanau et al. in [111] proposed a soft alignment model called soft attention mechanism for neural machine translation. The attention mechanism partially fixes the problem associated with fixed-length encoding vector by allowing the machine to look over all the information of the original sequence instead of just the last one, and then generating a proper word according to its context.

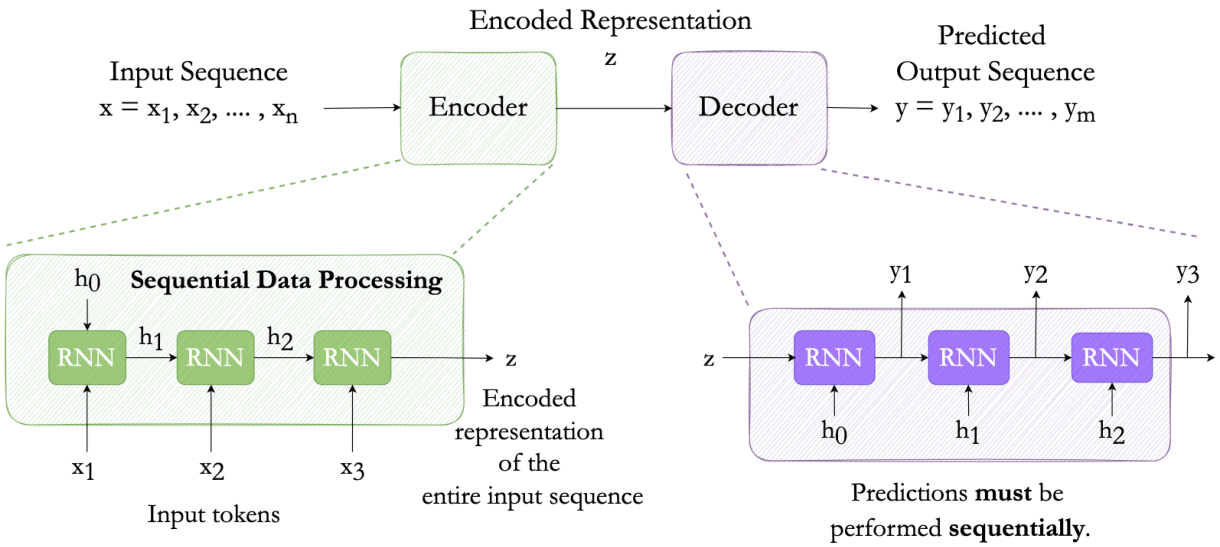


Figure 3.1. Illustration of how machine translation (seq2seq) used to work before attention. Stacked layers of RNNs are used inside encoder-decoder architecture to treat the sequences *sequentially*. The encoder processes the elements (token) of the input sequence and produces one compact fixed-length context vector 'z' for the entire input sequence. The decoder generates the output sequence from 'z'. h_{i}, represents the hidden state vectors where $\mathrm{i} \in \{1, ...n\}$.

In Figure 3.2, we show how attention is used in [111] to address the problem of RNNs for machine translation task. Attention (α_{ij}) is added to the previously explained encoder-decoder RNN (please refer to Figure 3.1). α_{ij} can be expressed as:

$$\alpha_{\mathrm{ij}} = \frac{\exp(\mathrm{e}_{\mathrm{ij}})}{\Sigma_{k=1}^{T_x} \exp(\mathrm{e}_{\mathrm{i}k})} \tag{3.1}$$

where

$$\mathrm{e}_{\mathrm{ij}} = attention_{net}(y_{\mathrm{i}-1}, h_{\mathrm{j}}). \tag{3.2}$$

e_{ij} is a score defined between the $\mathrm{i}-1$-th state of the decoder and the j-th hidden state of the encoder. Note that h_{j} represents the hidden states of the encoder. $y_{\mathrm{i}-1}$ is the $\mathrm{i}-1$-th hidden state of the decoder. The state $y_{\mathrm{i}-1}$ is used to predict the i-th output for $\mathrm{i} = 1, \ldots, N$. The attention tries to put weight between every hidden state of the encoder and the current hidden state of the decoder. This captures how "aligned" the previous state of the decoder

and the encoded information are. $attention_{net}$ is the 'Attention Neural Network' shown in Figure 3.2. T_x is the number of total states of the encoder.

Basically these α's are the attention weights which are computed using softmax function obtained from the e_{ij} scores. The weights – specifically, the scores e_{ij} – are learned dynamically from the training data using a neural network. Finally, dynamic context vectors (z_i) in each prediction step are calculated as:

$$z_i = \Sigma_{j=1}^{T_x} \alpha_{ij} h_j \tag{3.3}$$

3.1.3 Transformers

As explained in Section 3.1.1, RNNs work sequentially. Thus, to compute the n-th word in a sentence, the network needs to remember and wait for the results from $n-1$ steps. Thus, the task can not be parallelized. Moreover, RNNs require huge number of computations and resources.

In 2017, Vaswani et al. [112] proposed *Transformers* to eliminate the problems associated with RNNs. The Transformers striped RNNs from the picture and introduced the concept of *self-attention*. They still use an encoder-decoder architecture; but now the encoding component is a stack of encoders, each having the same internal structure. The decoding component also follows a similar structure.

The encoding component of the model proposed in the original paper [112] has 6 self-attention layers and a feed-forward network. The decoder shares the same structure with a additional layer called the Encoder-Decoder attention. The additional attention layer is used to create a bridge between the encoder and the decoder.

This new structure is easily parallelizable. The calculations require less resources. Moreover, the Transformer model can extract temporal dependencies from an entire sequence for any finite length of the sequence without increasing the computational burden.

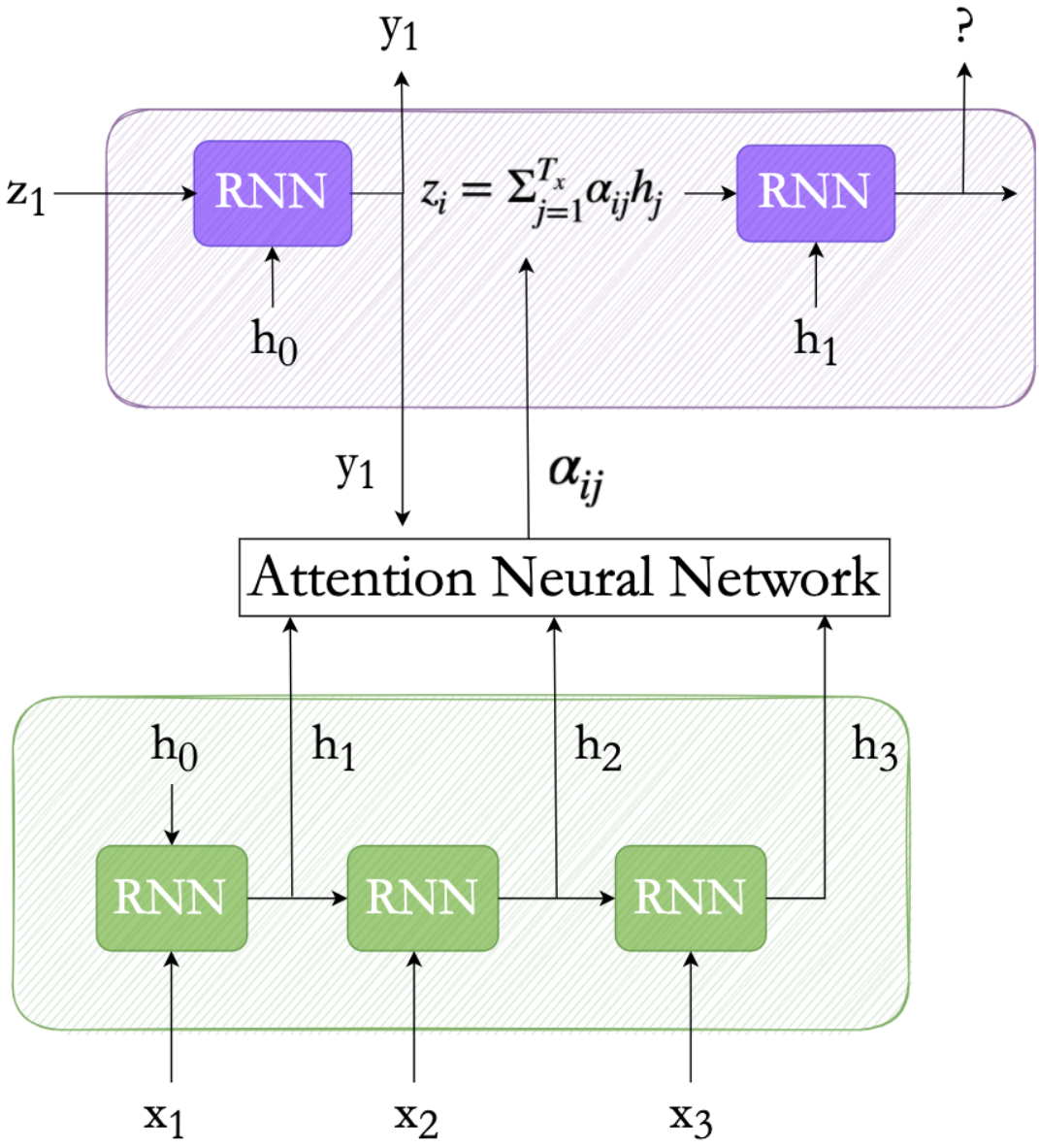


Figure 3.2. Illustration of how attention is used by Bahdanau et al. in [111] for machine translation task (seq2seq). α_{ij}-s are the attention weights for the encoder hidden states which decide the importance of different parts of the input sequence for accomplishing a given task.

3.1.3.1 Self-Attention

With the advent of *Transformers* [112], self-attention has become the most popular type of attention for different machine learning and deep learning tasks.

Before attention mechanism was used to connect the encoder and the decoder. As shown in Figure 3.2, attention is computed between input and output sentences. In Transformers, attention mechanism is used to compute the dependency between words of the same sentence i.e. attention is calculated between a sentence and itself. This mechanism is referred to as Self-attention — the mechanism of relating different positions of a sequence to compute the encoding vector representation of the same sequence. The intuition is simple– it tries to learn the context of each word depending on the composition of the sentence. The success of the Transformer model has led to a trend of replacing RNNs with attention-based networks.

Let us explain with an example the concept of self-attention in NLP. Say we have the following sequence: *school of fish*. The word 'school' has multiple meanings. If we do not see the word 'fish', we would not understand the context in which the word 'school' is being used in the given sequence. This is the idea behind self-attention. It tries to associate context with each word of a sequence guiding the network to understand exactly what is meant by that word in the sequence.

3.1.3.2 Multi-Head Attention

Another interesting idea that the Transformer paper [112] introduced is the concept of Multi-Head Attention. This refers to the process where the self-attention mechanism is applied several times in parallel to add dimensions to the self-attention mechanism in order to retrieve more meaning.

The high-level structure of such an attention module is as follows: there is a Query vector and a Key-Value Pair. The attention function is defined as a mapping from the Query (Q) and a set of Key-Value pairs (K,V) to an Output, where Q, K, V and Output are all vectors. The attention mechanism is applied as follows:

1. Compute the similarity between Q and K using one of the following ways: (a) Calculate Cosine Similarity between K and Q, (b) Find Scaled Dot Product between K and Q, and (c) Use a neural network to estimate the similarity.

   Scaled Dot Product is most commonly used to compute similarity between Q and K.

2. Apply Softmax Function to normalize the similarity scores. The normalized similarity scores become the attention weights.

3. Using the attention weights, calculate the weighted sum of the Values. The weight assigned to each Value is computed by a compatibility function of the Q with the corresponding K"

Mathematically, the attention computation process is expressed as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}) \tag{3.4}$$

where K^T is the transpose of the matrix K. d_k is the size of the query.

It is to be noted that the concepts of Q, K, V originated in search and retrieval systems. Let us explain with an explain what is meant by Query, Keys and Values in such context — When we search a music video on YouTube, the text in the search bar becomes our Query. Some example of Keys can be the song title, description, singer, lyricist, etc. The best-matched videos would be the Values.

Q, K and V serve as the inputs of the Multi-Headed Attention module in a Transformer. At the high-level, this Multi-Headed (MH) Attention module can be viewed as:

$$MH(Q, K, V) = [head_1, head_2, ..., head_h]W_0 \tag{3.5}$$

where $head_\text{i}$ = Attention$(QW_\text{i}^Q, KW_\text{i}^K, VW_\text{i}^V)$. W_i is the i-th learnable parameter matrix.

As shown in Figure 3.3, independent attention outputs are obtained from the 'Scaled Dot Product Attention' unit. These outputs are then concatenated and passed through a linear layer to generate an encoded representation of expected dimension. The multiple attention heads shown in the 'Scaled Dot Product Attention' unit guides the network to attend to different parts of the sequence differently. This is achieved by dividing the features into multiple heads, each head focusing on a subset of features.

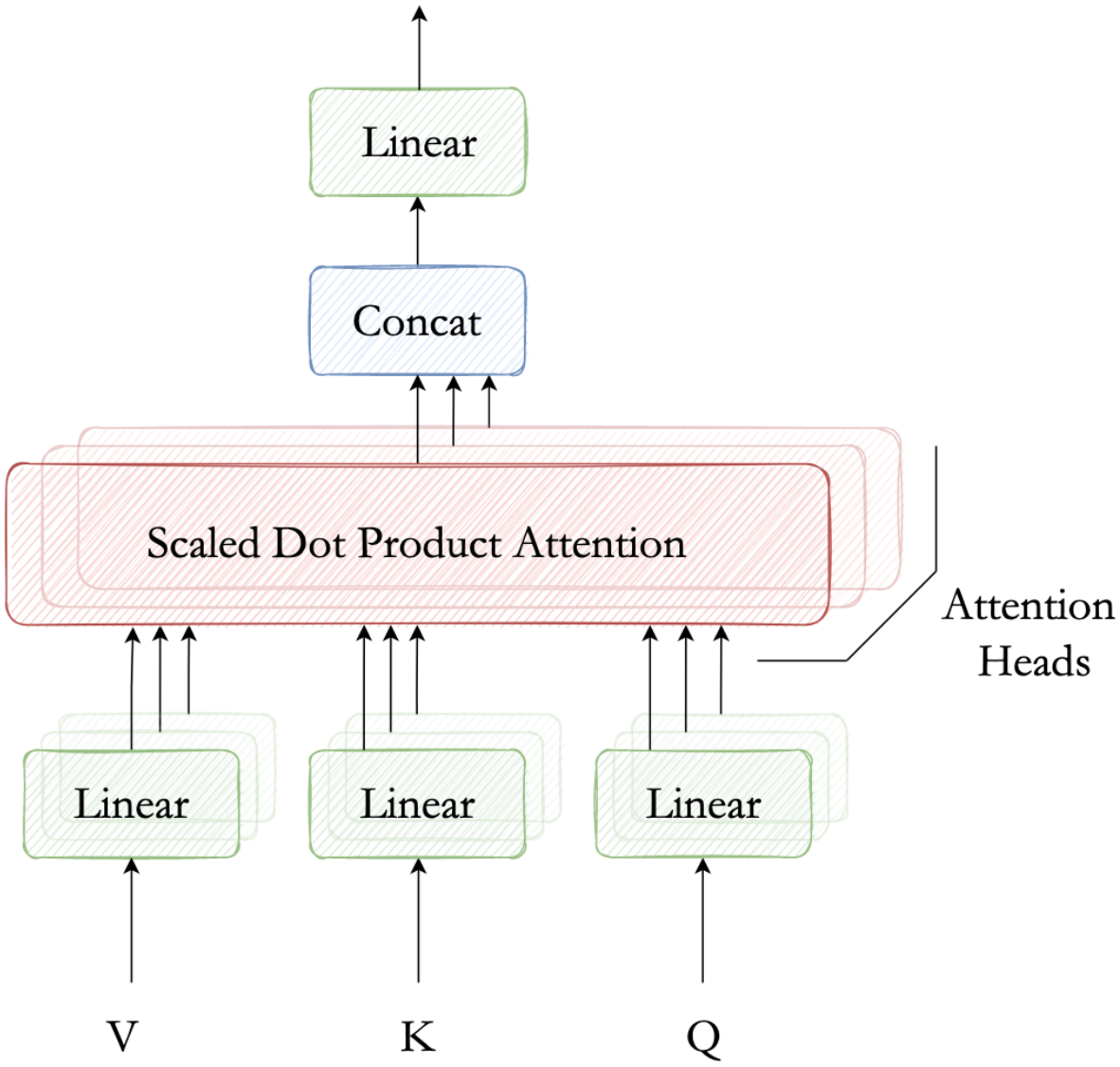


Figure 3.3. Illustration of Multi-head Attention.

3.2 Attention in Computer Vision — Visual Attention Mechanisms

We already explained the pioneering works on attention in the field of NLP. In this section, we provide a review of the visual attention mechanisms. Figure 3.4 illustrates the structure of a basic visual attention module. First, I explain the intuitions behind why 'attention' can be useful and can be applied in the computer vision applications. I also draw similarities and differences with the attention mechanism used in NLP.

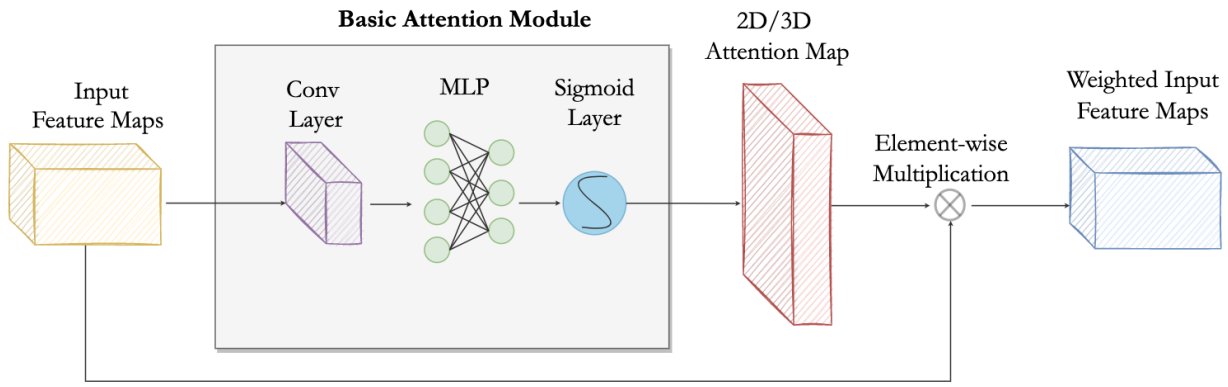


Figure 3.4. Illustration of a basic attention module used in computer vision. The attention module consists of a simple convolutional layer, a multi-layer perceptron and a Sigmoid activation layer. The input to the attention module is a $C \times H \times W$ feature map. The output is a $1 \times H \times W$ (2D) or $C \times H \times W$ (3D) attention map. This attention map is then multiplied element-wise with the input feature map to get a more refined and highlighted feature map.

Focusing on the road while driving, looking at the nearby vehicles while crossing roads, glancing at the food on your plate before taking a bite, looking at the person with whom you are talking — Visual Attention is a constant and subconscious part of our everyday life. However, if we want a neural network to exhibit similar attention property as humans, we need to figure out an explicit way to incorporate that. The network needs to 'learn' the most relevant parts of a visual scene and filter out the irrelevant parts.

In NLP, attention mechanism helped in learning a flexible dependency mechanism to figure out which elements of the input sequence are most important in generating an accurate output sequence. This needed learning dependency across temporal domain. However, in computer vision, in most the cases, dependency needs to be learned along the spatial domain. The same feature may be present in multiple regions of a scene. The target object might have occluded parts throughout the image. Semantic cues from the entire image would improve the classification or segmentation of that object in the image in such scenarios. Attention mechanism can help in learning spatial dependencies beyond the receptive field of a convolutional filter, allowing the neural networks to build wider intuition and provide maximum performance.

In computer vision, attention mechanisms are broadly classified into 3 categories – (a) Soft Attention, (b) Hard Attention, and (c) Self-attention. In the next few subsections, I will discuss these categories briefly. Figure 3.5 shows an intuitive example of soft and hard attention mechanism.

Figure 3.5. Illustration of soft attention and hard attention in visual attention mechanism. In this case, we want to attend to objects resembling chocolate cake. An intuitive explanation of soft attention can be related to blurry vision of an entire scene with more focus on certain areas, in the case, the cake. Hard attention is like binocular vision where we look at only a part of the scene that is most relevant to us, again in this case, the cake.

3.2.1 Soft Attention

In soft attention, each input element is assigned a weight (probability) between 0 and 1. The process is deterministic and differentiable because the attention weights are calculated using smoothly varying softmax function. However, one of the drawbacks of the approaches with soft attention usually suffer from high computational cost. Nevertheless, soft-attention is widely applied.

Soft attention was first introduced in [114] where the task at hand was image caption generation. Attention mechanism helped in improving the task by allowing the user to understand what and where the model is focusing on. In the paper, based on a set of

features ($\mathbf{a} = \{a_1, ..., a_L\}, a_\mathrm{i} \in \mathbb{R}^D$) extracted from the input image, a model is trained to produce a caption by generating one word at each time step. The decoder is a long short-term memory network (LSTM) [66]. Based on the features $\mathbf{a}$ of the input image and the previous hidden states h_{t-1}, a context vector z_t is generated at each time step t using the proposed attention mechanism. The attention weight $\alpha_{t,\mathrm{i}}$ for the feature vector a_i at the time step t is defined as the relative importance of location i to the next word. Mathematically, $\alpha_{t,\mathrm{i}}$ is calculated as:

$$\alpha_{t,\mathrm{i}} = \frac{\exp(\mathrm{e}_{t,\mathrm{i}})}{\Sigma_{k=1}^{L} \exp(\mathrm{e}_{t,k})} \tag{3.6}$$

where

$$\mathrm{e}_{t,\mathrm{i}} = f_{att}(a_\mathrm{i}, h_{t-1}) \tag{3.7}$$

f_{att} is the attention function computed by a multi-layer perceptron on the previous hidden state. Finally, the context vector z_t is obtained from the attention weights as follows:

$$z_t = \Sigma_{\mathrm{i}=1}^{L} \alpha_{t,\mathrm{i}}\, a_\mathrm{i} \tag{3.8}$$

With the help of attention mechanism, the 'image caption generation' paper [114] achieved state-of-the-art results on 3 challenging datasets. These impressive results encouraged researchers to use visual attention for many other applications. Since then soft attention mechanisms have evolved into different categories. Different models are created to pay attention to diverse feature domains. In the next few sections, I discuss those categories briefly with typical examples.

3.2.1.1 Spatial Attention

Spatial attention mechanism refers to an adaptive spatial region selection technique that decides *where* to pay attention. The spatial attention approaches try to weight spatial features based on their relevance to a given task, and use these weights to select or generate the important spatial positions [118], [128], [129].

Extensive research has been conducted on spatial attention mechanisms. Some pioneering works in this area are RAM [118], STN [130], GENet [131] and Non-Local[128].

RAM [118] came out from Google's DeepMind team. The authors consider that vision is a sequential task where parts of an image are glimpsed in sequence – similar to moving over an image using a sliding window protocol. Thus, they adopt a recurrent approach to predict relevant positions in an image and train the model in an end-to-end manner using reinforcement learning.

STN or Spatial Transformer Networks [130] selects important regions by learning affine transformations explicitly. CNNs are inherently translation invariant. This enables them to correctly recognize targets in an image during inference, even when that target was not observed at the same location in the image during training. However, CNNs do not provide scale invariance and rotation invariance. STNs are sub-networks that can be inserted into a CNN architecture to obtain spatial invariance with respect to translation, scaling, rotation and warping.

GENet [131] uses an implicit calibration function in the spatial domain to capture global contextual information and predict soft attention masks for relevant positions. This paper combined gathering and excitation operations where at first, the network tries to capture local as well as global contextual information by accumulating features from large neighborhoods of the input image. Then the network produces an attention map of the same resolution as the input feature map, using interpolation. Each position in the input feature map is then weighed by multiplying it element-wise with the attention map.

Non-local neural networks [128] were proposed for video understanding and object detection task. This approach uses self-attention to learn non-local dependencies.

Further details of these methods are out of scope for this dissertation. Please refer to the cited papers for more details.

3.2.1.2 Channel Attention

Channel attention refers to generating dynamic attention masks across across the channel domain to select the most informative channel.

A CNN usually takes input an image with n-channels ($n = 3$ in RGB images). As the image passes through different convolutional layers, new channels are generated containing

different feature maps representing different information. A network with channel attention adaptively recalibrates the weights of different channels — higher weights are given to the channels containing more relevant information for a given task.

Hu et al. [132] first proposed the concept of channel attention. They introduced the novel concept of a squeeze-and-excitation (SE) block that is capable of capturing channel dependencies as well as global contextual information.

Inside a SE block, there is a squeeze module that is responsible for capturing the global context via global average pooling. And the excitation module works towards learning the channel relationships and generates an attention vector by using fully-connected layers and non-linear activation layers.

SE blocks have become very popular in attention-based image processing tasks after their success on several benchmark models. Other popular works in this category are GSop-Net [133], FCANet [134], SRM [135], GCT [136], etc.

3.2.1.3 Mixed Attention

Mixed attention technique combines multiple attention mechanisms into one framework in an attempt to achieve performance improvements over methods that use one type of attention mechanism. Woo et al. [98] first proposed a mixed-attention approach – Convolutional Block Attention Module (CBAM). This module is designed to focus on meaningful features along the spatial and channel axes, thus, learning both *where to look* and *what to look*. This module is specially designed for feed-forward CNNs and can be inserted at every convolutional layer.

Inside a CBAM, there are 2 sub-networks – the Channel Attention Module and the Spatial Attention Module – placed parallelly or sequential.

The spatial attention module (SAM) aims to find 'where' is the information. It performs Max Pooling and Average Pooling along channel dimension and concatenates the outputs of the pooling layers to obtain an informative feature vector. This feature vector is fed to a convolutional layer, a Batch Normalization layer, an optional ReLU layer and a Sigmoid Activation layer in sequence to obtain the spatial attention map.

Given an input image and a specific task to accomplish, the Channel Attention Module (CAM) tries to find out 'what' is relevant. CAM also consists of a Max pooling layer and an Average pooling layer. Average Pooling is used to aggregate spatial information; whereas Max Pooling gathers important cues about discriminative features. The feature vectors obtained from the pooling layers are fed to a fully connected layer, followed by a ReLU activation layer to obtain the channel attention map. It is empirically observed that better performance is achieved when the SAM and the CAM are in sequence with channel-first order.

Around the same time when CBAM was introduced, Park et al. proposed the Bottleneck Attention Module (BAM). In BAM, channel and spatial attention masks generated parallelly. The channel and spatial masks are then added to obtain the final attention map. Whereas, CBAM achieved better results when a sequential approach was used. Moreover, CBAM used Global Average Pooling in conjunction with Max Pooling and Average Pooling; whereas, BAM used only Global Average Pooling. Another difference between these two approaches is that BAM incorporated dilated convolutions to increase the receptive field; Whereas, CBAM relied on larger kernel sizes and regular convolutions to increase the field of view.

The success of CBAM encouraged researchers to explore different mixed-attention mechanisms. Some other popular papers in this category proposed 3D attention masks [137], [138] with channel, height and width attention as dimensions.

3.2.1.4 Temporal Attention

Temporal attention mechanism refers to a dynamic time selection technique that decides *when* to pay attention. The attention weights are adjusted based on the samples in sliding time windows. Samples in different windows have different contributions and are weighed accordingly. This kind of attention technique is usually applied in video processing.

Previously, RNN and temporal pooling based approaches were used for temporal relation modeling in videos. However, these methods suffered in terms of long-term temporal modeling. Li et al. [139] addressed this issue by introducing a global local temporal representation learning approach which can capture multi-scale temporal cues in video sequences. In this

paper, dilated temporal pyramids are used for learning local temporal context, and temporal self-attention is used to capture global context. Dilated convolutions are used along temporal domain to capture multi-scale temporal information.

To further improve the performance of temporal attention mechanism and capture temporal dependencies efficiently, Liu et al. in [140] proposed a temporal adaptive module that uses adaptive convolutional filters instead of self attention to capture global context. This paper also showed improvement over [139] in terms of time complexity.

3.2.2 Hard Attention

In a network with hard attention, only the part of the input that the network considers relevant is retained, and the rest of the input is discarded i.e., the attention weights assigned are either 0 or 1, as shown in Figure 3.5. This makes the process non-differentiable. Reinforcement learning techniques are necessary to train such models. Similar to soft attention, hard attention can be applied to both temporal and spatial context. In temporal context, based on the current step information, the network decides where to attend in the next step. However, for such an approach, ground-truth is not available. Thus, hard-attention type mechanisms are represented by stochastic processes. The computation cost for hard attention models are less compared to the soft attention models.

Hard attention mechanism was also proposed in [114]. In hard attention, $\alpha_{t,\mathrm{i}}$ (shown in Equation 3.6) can be interpreted as the probability of location i to be the right location to focus on. The context vector z_t for hard attention is expressed as:

$$z_t = \Sigma_{\mathrm{i}=1}^{L} \, s_{t,\mathrm{i}} \, \alpha_{\mathrm{i}} \tag{3.9}$$

where $\alpha_{t,\mathrm{i}}$ is treated as the parameters of a multinoulli distribution and z_t as a random variable:

$$p(s_{t,\mathrm{i}} = 1 \mid \mathbf{a}, h_{t-1}) = \alpha_{t,\mathrm{i}} \tag{3.10}$$

3.2.3 Self-Attention

The concept of Self-Attention has shown immense promise in the field of NLP [111], [112], [141], [142]. Recently, self-attention has gained popularity in the computer vision community as well [113], [128], [143], [144].

CNNs have limited ability to capture global information due to the inherently narrow receptive fields [145], [146] of the convolutional filters. To increase the field of view, self-attention was introduced in computer vision by Wang et al. [128]. Self-attention can be thought of as a spatial attention mechanism to capture global contextual information.

As explained in Section 3.1.3.1, self-attention mechanism tries to learn the interdependence between the input elements of a given task. In a self-attentive network, the input elements interact with each other and decide what they should pay more attention to. The biggest advantage of such an approach is that it is easily parallelizable. A self-attention layer can compute attention weights with all the same input elements using simple and easily parallelizable matrix calculations. However, self-attention mechanisms suffer from quadratic computational complexity. A lot of researchers [67], [68], [147], [148] are focusing on reducing the computation complexity associated with self-attention.

3.2.3.1 Vision Transformers

Currently, the transformers [112] are the most popular attention-based networks. Transformers are pure attention-based networks that eliminated the RNN-based attention frameworks with the help of self-attention mechanism and multi-head attention module. They have had great success in NLP [112], [141], [149], [150]. Recent times have witnessed superior performance of Transformer models in computer vision [143], [144].

Motivated by the success of Transformers in computer vision, Dosovitskiy et al. [113] proposed the first pure transformer-like architecture for image classification task and named it 'Vision Transformer (ViT)'.

ViT employs a Transformer-like architecture for image classification. An image is split into fixed-size patches. First, each patch is linearly projected to generate flattened patches. Then, position embeddings are added to obtain position-aware encoded representation.

Along with the position embeddings, learnable class embeddings are added as "classification tokens" for performing the classification task. Finally, these position embeddings are fed to the encoder of a standard Transformer.

Vision transformers showed the pure attention-based architectures can perform better than CNNs especially for large datasets [151], [152]. Following ViT, several other papers based on transformer-like architectures [153]–[160] have shown excellent results for a variety of computer vision tasks including object detection, object classification, semantic segmentation, action recognition and self-supervised learning.

3.3 How We Use Attention

Non-uniformity in building appearances in different parts of the world, the presence of shadows in remotely sensed images and the presence of occlusions caused by nearby tall structures and high vegetation make distinction of building pixels from complex background a challenging task. Moreover, difficulties also arise from the fact that in many cases, various objects (such as roads, parking lots and building roofs) that are present in aerial and satellite images have very similar appearances and very small inter-class differences. Moreover, building footprints may appear in variety of shapes and sizes; can be present in sparse remote locations or can appear in densely populated localities.

With the advent of deep neural methods and high-end computational resources, researchers have already achieved remarkable success in the area of semantic segmentation of building footprints from remotely sensed imagery [14], [77]. However, owing to the challenges mentioned in the last paragraph, most of the state-of-the-art building detection approaches still face difficulty in predicting accurate building boundaries. These algorithms often get confused if the building regions bear strong resemblance with the background, and also perform poorly in presence of shadow and occlusion. Moreover, the networks, showing impressive results over one geographical region, often fail to succeed in different regions across the globe if they are not specifically trained to detect buildings in those regions.

In this dissertation, our focus is on resolving the above mentioned issues i.e. guiding our network towards extracting accurate boundaries of the buildings, resolving mis-classifications

in the predicted segmentation maps, and improving the generalization capability of our network for any city across across the globe (without training the network on each city explicitly). Further, we aim to develop an architecture that performs well on both satellite as well as aerial images.

To this end, we have proposed an *attention-enhanced* generative adversarial network for building footprint detection in remotely sensed images. The core idea is to use different types of attention mechanisms intelligently to mitigate the problems present in the state-of-the-art building segmentation methods. Specifically we propose the following 3 attention units:

1. **Uncertainty Attention Unit**: This unit highlights the feature of those regions where the network has not shown confidence during its previous prediction. These are mostly features around the building boundaries, the shadow areas and the areas where the foreground and background pixels have similar reflectivity properties.

2. **Reverse Attention Unit**: This unit allows the network to explore features which have been predicted as non-building class, thus, enabling the network to discover the missing building parts in the previously estimated result.

3. **Edge Attention Unit**: This unit enhances the boundaries of the buildings, thus, helping the network to learn precise crisp boundaries of the buildings

As all the proposed attention units work towards attending different spatial features, they fall under the category of *soft spatial attention mechanism*. As opposed to standard spatial attention mechanisms where the focus is on learning 'where' the relevant information for a given task are present, our attention modules already know 'where' they need to steer their attention to. They only attend to the *concern areas*. By 'concern areas', we refer to the areas where the network finds challenging to come with an accurate prediction — the areas near building boundaries, the areas where predictions are being made with low probabilities, and the areas where predictions have low confidence. Our attention units enhance the features in those regions so that more loss can propagate in those regions forcing the model to reconsider the predictions of those regions in light of newly available information.

The readers may wonder why anyone would use the attention units proposed by us in the era of transformers. Though transformers have been super successful in replacing CNNs for computer vision tasks, it is to be noted that there is a serious constraint in applying them — the need for extremely large datasets. Transformers need expensive pre-training on large datasets. Whereas, CNNs can be trained with reasonably small amount of data [161]. In our case, super large datasets with accurate labels for building segmentation are not easily available.

4. BACKGROUND

The purpose of this chapter is to act as a primer that the reader can use to become familiar with the important concepts and techniques of deep learning that have served as essential parts for the method proposed in this dissertation. In Section 4.1, we describer the encoder-decoder architecture. In Section 4.2, we describe the GAN architecture. Subsequently, in Section 4.3, we describe Atrous Spatial Pyramid pooling.

4.1 Encoder-Decoder Architecture for Image Segmentation

In deep neural networks, the shallow layers extract lower-level features, while the deeper layers learn higher-level specialised features. Thus, deeper neural networks with increasing number of feature maps are constructed to learn expressive features for image representation. These deep networks come with increased computational burden which be alleviated by periodically down-sampling the feature maps through pooling or strided convolution for a task like image classification or object detection because the goal for such tasks is to identify the target and not its location in the image. However, an image segmentation model needs to produce a segmentation map of same resolution as that of the input image. A naive approach like stacking a number of convolutional layers with 'same' padding to keep the resolution fixed and producing a final label map would be computationally expensive.

A popular approach for image segmentation models is to follow an encoder/decoder structure. It provides a way of creating full-resolution segmentation predictions without the need of preserving full-resolution feature maps throughout the network. In this kind of architecture, lower-level feature maps which are highly efficient at discriminating between different target classes are first learned by down-sampling the spatial resolution of the input image. The lower resolution feature representation is then upsampled to produce a full-resolution segmentation map. Unpooling, bilinear/bicubic interpolation and transpose convolutions are some commonly used approaches for upsampling the low-resolution feature maps.

Fully convolutional encoder-decoder networks for image segmentation provide superior semantic segmentation where powerful, pre-trained image classification networks (eg. AlexNet, VGG19) are used as the encoder (i.e. the feature extraction) module of the network and a de-

coder module with transpose convolutional layers is added after the encoder to upsample the coarse feature maps into a full-resolution segmentation map. However, the main drawback of such an architecture is that the decoder struggles to produce fine-grained full-resolution segmentation maps from the low-resolution feature maps.

A good semantic segmentation model needs access to local information as well as global information — local information to correctly recognize the objects present in an input image and global information to know their locations in the image. This can be achieved by the combining fine details extracted from the shallow layers of the network with the semantic information obtained from the layers closer to the output. In literature, this is done by upsampling the encoded representation in stages to produce the decoder features, adding "skip connections" from the the encoder layers, and summing the encoder and the decoder feature maps. The skip connections help the network in recovering fine-grained details of the input images that are needed to reconstruct segmentation maps with precise shapes and accurate boundaries of the objects.

This architecture is further improved by Ronneberger et al. [39] when they proposed the UNet architecture. The UNet "consists of a contracting path to capture context and a symmetric expanding path that enables precise localization."

In this dissertation, the baseline architecture of our segmentation network (i.e. the generator network of our GAN framework) resembles a fully convolutional encoder-decoder architecture like UNet [39]. In Section 4.1.1, we provide a brief idea about our base encoder-decoder architecture.

4.1.1 Our Base Encoder-Decoder Architecture

The baseline generator of our proposed GAN framework is a fully convolutional encoder-decoder network. The input to the network is a 3-channel remotely sensed image and the output is a 1-channel prediction map in which each pixel value indicates that pixel's probability of belonging to the building class.

The *encoder* has four strided convolutional (Conv) layers with 7×7 kernel for the first two layers and 5×5 kernel for the next two. In each layer, the number of channels in the

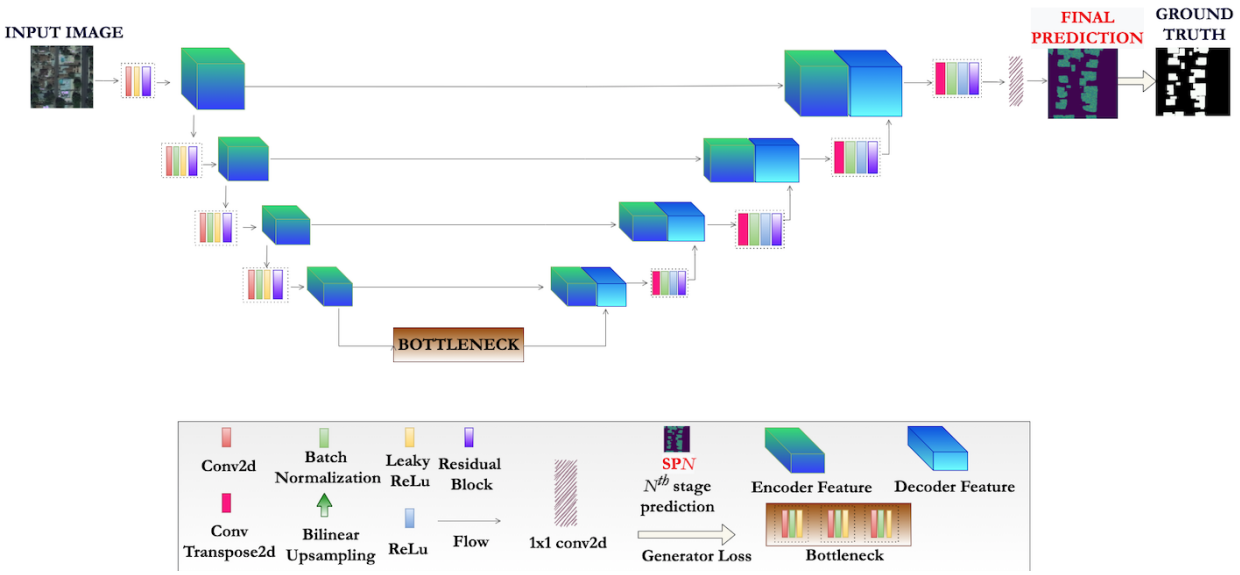


Figure 4.1. The encoder-decoder architecture of our baseline segmentation framework. Both the encoder and the decoder have 4 strided convolutional (Conv) blocks. Each Conv block has a stride of 2, and consists of a Conv2d layer, a batch normalization layer, and a Leaky ReLU layer. Each Conv block is followed by a residual block. The decoder is similar to the encoder except the following — kernel sizes are larger, and Leaky ReLU is replaced by standard ReLU. Between the encoder and decoder is the bottleneck layer that consists of 3 3×3 Conv blocks. Batch normalization is used after each convolutional layer except the first layer of the encoder. Skip connections are added to concatenate the corresponding layers of the encoder and the decoder.

feature maps is doubled and the number of feature maps is halved till the resolution of the feature maps becomes $\frac{1}{16} - th$ the spatial resolution of the input images.

The *decoder* is symmetric to the encoder. We use kernels with larger receptive fields (7 for the first 2 layers, 9 and 11 for the last 2 layers) to enlarge the representational scope of each pixel. We experiment with both *transpose convolution* as well as *bilinear interpolation followed by regular convolutional layers* for upsampling the incoming feature map while halving the number of feature channels. We observe that transpose convolution (i.e. deconvolution) creates some form of checkerboard artifacts due to uneven overlap when the kernel size is not divisible by the stride. This issue is resolved in bilinear interpolation based upsampling.

The Bottleneck of our generator network consists of 3 Conv layers consisting of a 3×3 Conv2d, batch normalization and leaky ReLU with a leak slope of 0.2.

Residual blocks are added after every down-sampling and upsampling layer. Each residual block consists of a 1×1 Conv2d, followed by a 3×3 Conv2d and then another 1×1 Conv2d.

Batch normalization is used after each convolutional layer except the first layer of the encoder. After each batch normalization layer, Leaky ReLU with a leak slope of 0.2 is used in the down-sampling blocks, and a regular ReLU for the upsampling layers.

Skip connections are used in a similar fashion as that of the U-Net [39] to concatenate the corresponding layers of the encoder and the decoder.

4.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs) are introduced by Ian Goodfellow et al. [16] in 2014 and considered as **"the most interesting idea in the last 10 years in ML"** by Yann LeCun. GANs are about creation, painting like Van Gogh or composing like Mozart. GANs have several applications: GANs can convert zebras to horses. GANs can convert a black & white image into colour. Given a satellite image, GANs can label roads and buildings in the image. Given the image of an angry person, GANs can estimate how the person will look when he is happy. It is evident from the above examples that potential of GANs is huge.

The term 'generative' in Generative Adversarial Networks describes a class of statistical models that can generate new data instances. Specifically, given a set of data instances X and a set of labels Y, generative models can capture the joint probability $p(X, Y)$, or just $p(X)$ if there are no labels. Hence, the generative model can learn to generate data with the similar distribution as $p(\cdot)$ with which the original data is being generated. This goes in sync with the previously mentioned application of GANs. The main focus for GANs is to generate data from scratch — it has the potential to learn and mimic any distribution of data.

Unlike conventional neural networks, GANs adopt a game-theoretic approach. The network learns to generate data from a training distribution via a 2-player minimax game.

The two entities of a GAN are the *Generator* and the *Discriminator*. These two are the adversaries in a GAN framework — constantly competing with each other throughout the training process — the generator tries to fool the discriminator, while the discriminator tries not to be fooled. The generator learns to generate realistic images. The discriminator learns to distinguish the generated (i.e. fake) data from the real data. Initially, the generator produces garbage and thus, can be easily identified by the discriminator as the fake data. But as training progresses, the generator starts producing data which is very similar to the real data. Finally upon successful training, the generator produces such realistic data that the discriminator can not anymore tell the difference between real data and fake data, and starts classifying generated data as real. This process is illustrated in Figure 4.2.

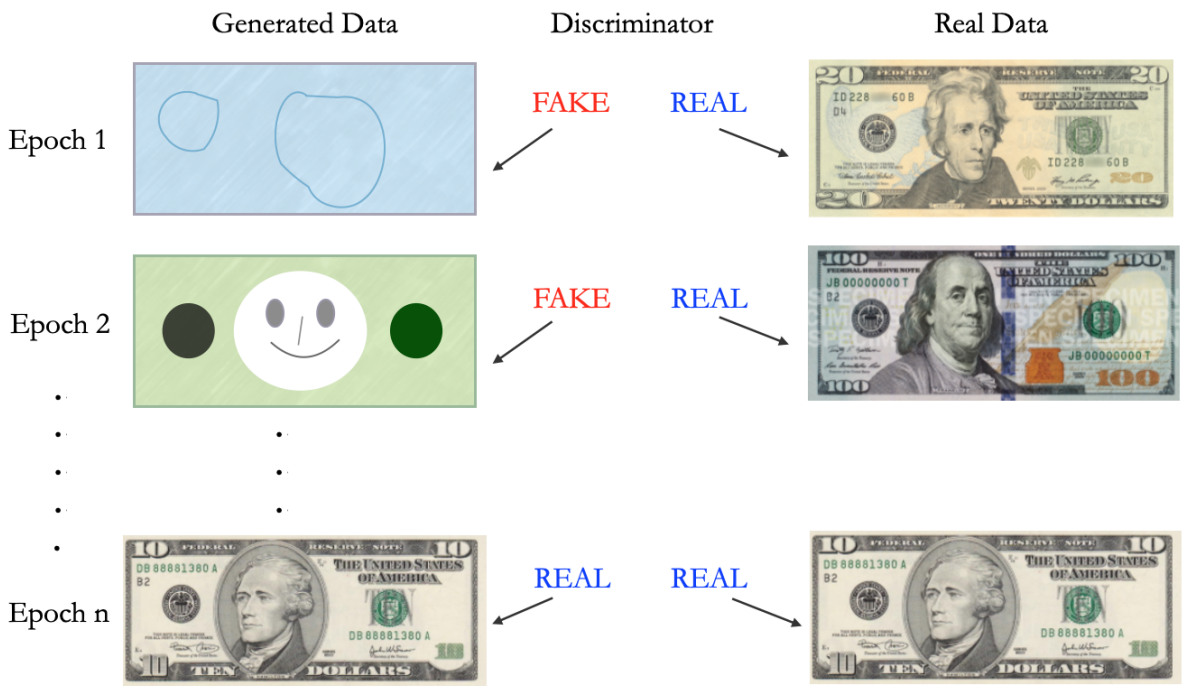


Figure 4.2. Illustration of the GAN process. When training begins, the generator produces garbage data, that can be easily identified as fake by the discriminator. As training progresses, the generator improves. Upon successful training, the generator produces such realistic images that the discriminator starts to classify fake data as real.

4.2.1 GAN Training

The input to the generator G of a GAN is noise z sampled from a normal or uniform distribution. With z as input, the generator is trained to produce an image $x = G(z)$. Basically, z represents the latent features (such as, color, edge, texture, etc.) of the generated image x. The semantic meaning of z is not user-controlled; instead the semantics are learned automatically by the GAN during the training process. However, G alone will just create random noise. The discriminator network D guides G to produce the expected image. Figure 4.3 is an illustration of a GAN pipeline.

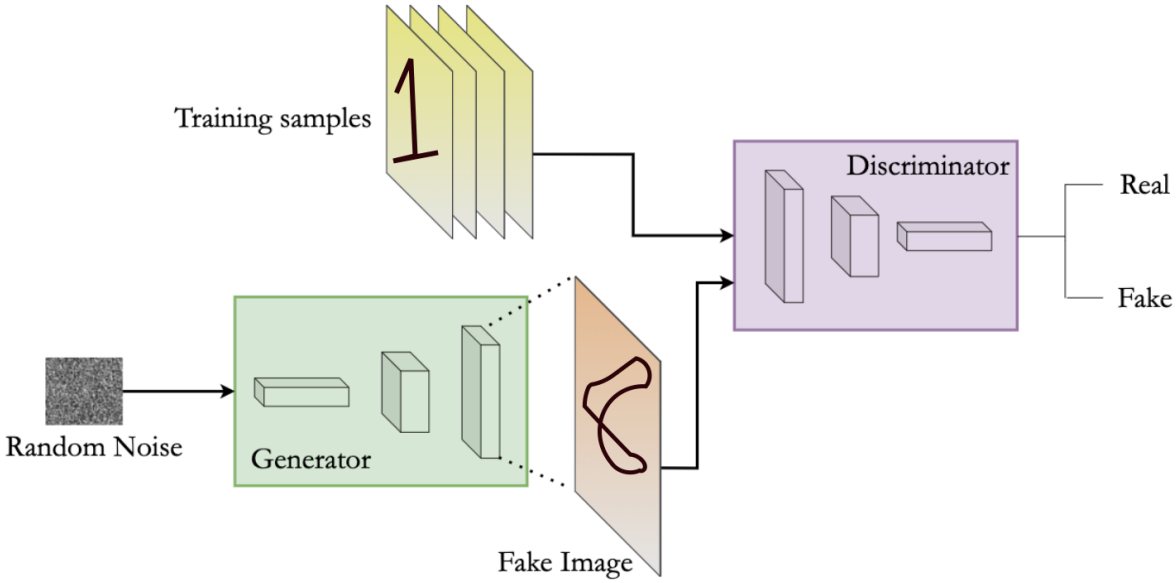


Figure 4.3. Architecture of a Generative Adversarial Network. A GAN has 2 entities competing against each other — a generator network (green box) and a discriminator network (purple box). The generator learns to generate realistic images. The discriminator learns to distinguish the generated (i.e. fake) data from the real data (i.e. training sample).

D is simply a classifier, and is trained with real and generated images to learn the features of the real images. Specifically, D looks at the real images from the training distribution and the images generated by G separately, and tries to understand if the image it's seeing is a real image from the training sample or a fake image generated by G.

The output of the discriminator: $D(x)$ is the probability that the input x is real. D is trained such that for the (real) images from the training sample, $D(x) = 1$. If the input to D is from the generated sample (i.e. fake image), we want $D(x) = 0$. This process guides the D network to identify features of the real images.

On the other hand, G is expected to generate images such that $D(x) = 1$ for all those x generated by the generator. Thus, G is trained by backpropagating this value all the way to G. This way the generator is trained to create images that the discriminator thinks are real.

G and D are trained alternatively, pitted against each other trying to improve themselves. First D is trained keeping the generator parameters fixed. Gradually, D learns to recognize the flaws of G. Similarly, during the training phase of G, the weights of D are kept constant. Otherwise the generator would be trying to hit a moving target and might never converge. Eventually, the GAN training converges with G producing realistic images. As G improves with training, performance of D gets worse as D can't easily tell the difference between real and fake. If G produces perfectly real-looking images, then the discriminator has a 50% accuracy — basically, the decision of D becomes synonymous with the outcome of a coin flip.

4.2.2 GAN Loss

In this section, we briefly talk about the loss functions proposed in the original GAN paper [16] to train the generator and the discriminator.

As mentioned earlier in Section 4.2, GAN adopts a game-theoretic approach. Thus, GAN loss is a min-max function.

Recall that D outputs a value $D(x)$ indicating the probability of x being real. The goal of D is to maximize this probability if x is real i.e. recognizing real images as real, and minimizing $D(x)$ if x is fake i.e. identifying generated images as fake. Say, the true label p for real images is 1 and for generated images, the label is -1. Then, the loss function $V(D)$ for the discriminator D can be expressed as:

$$\max_D V(D) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{4.1}$$

where, the first term on the left hand side of the equation aims at recognizing real images better; whereas the second term guides the network to identify generated images. G(z) represents the generated image from input white noise z. p_z is the distribution of the latent space (either uniform distribution or the normal distribution). Note that if $D(x) = 1$ when the image is real, and $D(G(z)) = 0$, for the generated image, the maximum is achieved which is also coincides with the optimal performance of the generator.

On the other hand, the objective function of G guides the model to generate images which can fool the discriminator. Thus, the job of the generator is to set $D(G(z) = 1$, hence, the discriminator would classify the generated image as the real one. The loss function $V(G)$ for the generator G is expressed as:

$$\min_G V(G) = \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{4.2}$$

It is evident that as $D(G(z))$ increases the value decreases and ultimately it is minimized when $D(G(z)) = 1$.

G tries to minimize V while D tries t maximize V. The loss functions (V(G) and V(D)) are learned jointly by alternating between gradient ascent and gradient descent. Thus, the overall GAN loss function is defined as:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{4.3}$$

As mentioned previously, D and G are trained alternatively, starting with the training of D for one iteration and keeping the model parameters of G fixed. Then for the next iteration, G is trained while keeping the parameters of D constant. The training continues until the model converges and the generator produces real-looking images.

Often vanishing gradient problem is encountered for the generator objective function. In the initial stages of training, D is likely to perform better than G. This causes $-\log(1 - D(G(z))) \to 0$. Thus $V(G) \to 0$ resulting in extremely slow gradient descent optimization.

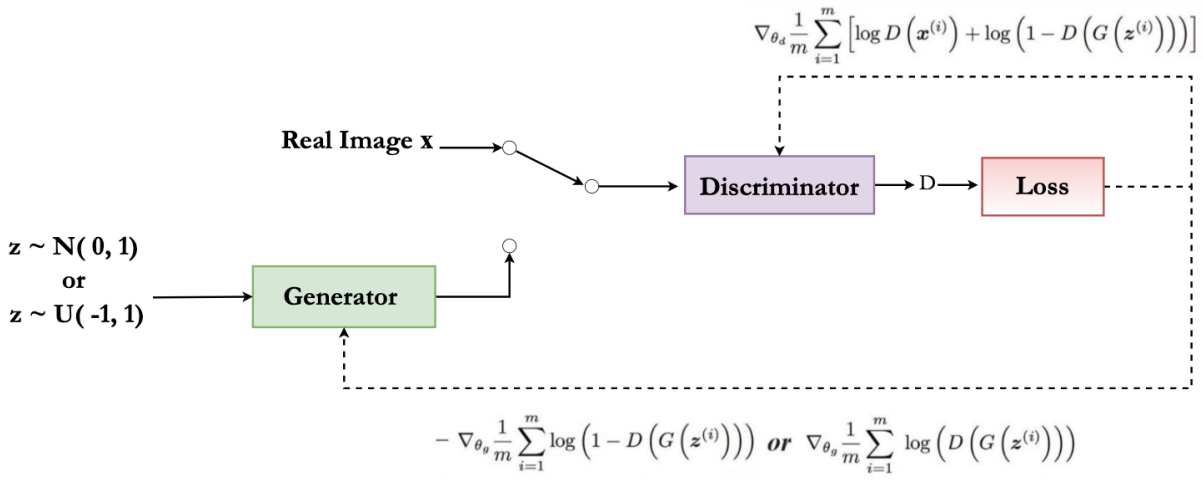


Figure 4.4. Illustration of data flow and backpropagation through the generator and the discriminator of a Generative Adversarial Network. m denotes the mini-batch size. θ_d and θ_g represent the model parameters of the discriminator and the generator respectively.

To improve the performance of G, an alternate cost function for better backpropagation of gradients to G is provided:

$$\max_G V(G) = \mathbb{E}_{z \sim p_z(z)}[\log D(G(z))] \tag{4.4}$$

Instead of minimizing the likelihood of discriminator being correct, the likelihood of discriminator being wrong is maximized. Figure 4.4 shows an illustration of gradient flow through a GAN.

An important point to keep in mind is that GAN training is very unstable and often does not converge because two adversarial networks are trying to learn from a single backpropagation. Thus, right choice of objective functions can make a big difference. In our framework, we use an adversarial L_1-loss which is suitable for building segmentation network which we describe in Chapter 6.

4.3 Atrous Spatial Pyramid Pooling

We use Atrous Spatial Pyramid Pooling (ASPP) in the segmentation network of our proposed GAN framework for automatically detecting buildings in remotely sensed images. In this section, we provide detailed background about ASPP.

ASPP was first introduced in the DeepLabV2 paper [7]. ASPP is basically the *atrous* version of *spatial pyramid pooling proposed* in SPPNet [162]. In Sections 4.3.1 and 4.3.2, we discuss about what is meant by 'atrous' and explain the concept of 'spatial pyramid pooling'.

In literature, ASPP has been used to extract multi-scale contextual information from images. In DeepLabV2, atrous convolutions with four different dilation rates are applied parallelly on the last feature map extracted from the backbone network and concatenated together to handle segmenting the object at different scales at much improved accuracy. In DeepLabV3 [7], this technique is further improved by applying image-level features — global context information is captured by applying global average pooling on the last feature map of the backbone. After applying all the operations parallelly, the results of each operation along the channel is concatenated and 1 x 1 convolution is applied to get the output. The ASPP module used in DeepLabV3 is shown in Figure 4.5.

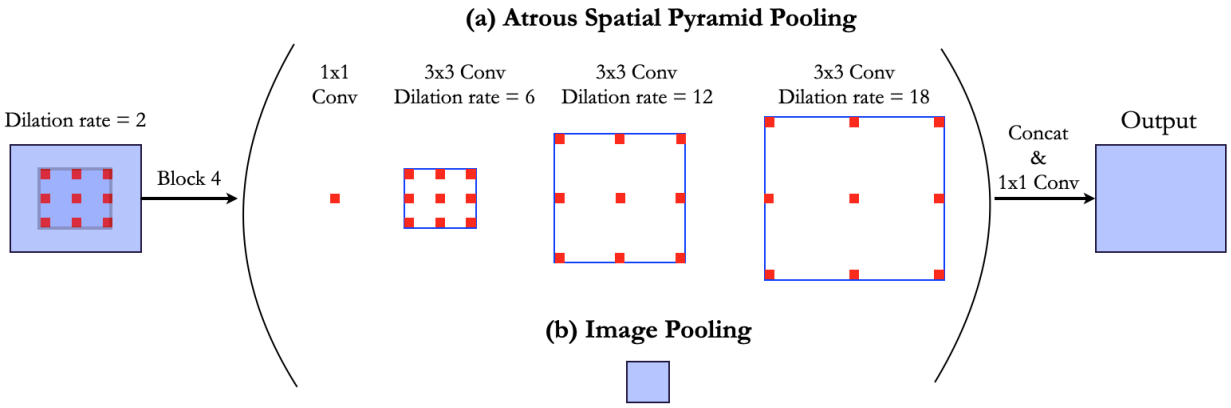


Figure 4.5. The architecture of ASPP module used in DeepLabV3. The module consists of (a) atrous convolutions and (b) image pooling. The final output is obtained by a convolution layer after concatenation of feature maps.

As objects of the same class can appear in different scales in an image, the ASPP module helps to account for different object scales, thus, improving overall accuracy of segmentation.

4.3.1 Atrous Convolution

Atrous Convolution is introduced in DeepLab [7] to control the effective field-of-view of the convolution. This technique enlarges the field of view of convolutional filters without changing the number of parameters, and finds the best trade-off between accurate localization (small field-of-view) and context assimilation (large field-of-view).

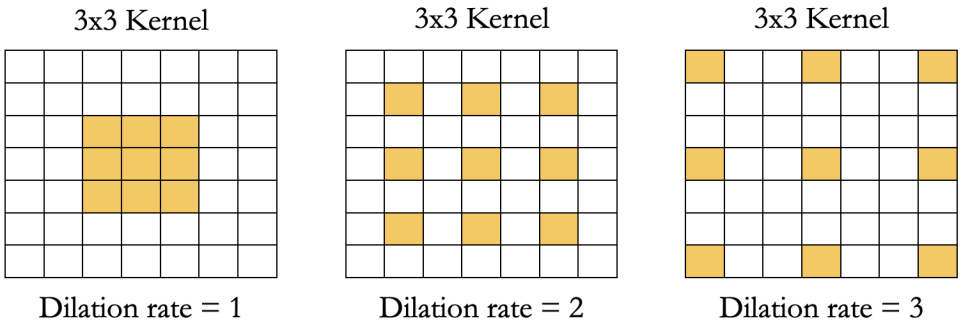


Figure 4.6. Atrous Convolution with different dilation rates.

The term 'atrous' means 'hole' in French. In atrous convolution, the kernel is upsampled by inserting zeros between two successive elements of the kernel along each spatial dimension. E.g., if 'r' is the dilation or atrous rate, $(r-1)$ zeros are inserted between the successive elements of the filter. This is equivalent to creating $r-1$ holes between two consecutive filter values in each spatial dimension, as shown in Figure 4.6. Hence, the method is named 'atrous convolution'. If $r = 1$, this becomes standard convolution. When $r > 1$, it is the atrous convolution which is the stride to sample the input sample during convolution.

Atrous convolution can be mathematically expressed as:

$$y[\mathrm{i}] = \Sigma_{k=1}^{K} x[\mathrm{i} + r.k]\, w[k] \tag{4.5}$$

where, for each location i on the output y and a filter w, atrous convolution is applied over the input feature map x where the dilation rate r corresponds to the stride with which we sample the input signal.

With a distinct dilation rate r, the filter will have a different field of view. Thus, features from multi-scale targets can be captured by changing the dilation rate in different layers without reducing the size of feature maps. This replaces standard strided convolution or pooling operation in deep neural networks. Thus, this method is also popularly known as 'Dilated Convolution.'

4.3.2 Spatial Pyramid Pooling

Spatial Pyramid Pooling (SPP) is a technique that facilitates efficient handling of multi-scale images, especially while dealing with machine learning algorithms to address problems like classification. Specifically, SPP layer is a pooling layer that helps get rid of the fixed-size constraint in convolutional neural networks (CNN).

Usually CNNs are used for classification tasks where fully connected layers usually follow the convolutional feature extraction layers. During feature extraction, convolution operations are performed in a sliding window fashion. This operation is capable of accepting varied size inputs and produces varied size outputs. However, the fully connected layers following the CNN can only accept fixed-size inputs. This makes a CNN incapable of accepting varied size inputs. The transition from the convolution layers to the fully-connected layers imposes the size restriction. Thus, to fit the size requirements of the network, the images need to be reshaped into some specific dimension before feeding them into the CNN. This shortcoming of CNNs results in image warping and reduced resolution.

SPP avoids the need for cropping or warping at the beginning of a CNN by adding a new layer between the convolutional layers and the fully-connected layers. The purpose of the SPP layer is to map a variable size input down to a fixed size output. The SPP layer is added on top of the last convolutional layer in a CNN, and this layer pools the features and produces fixed-length outputs. These fixed-length outputs are then passed into the fully-connected layers.

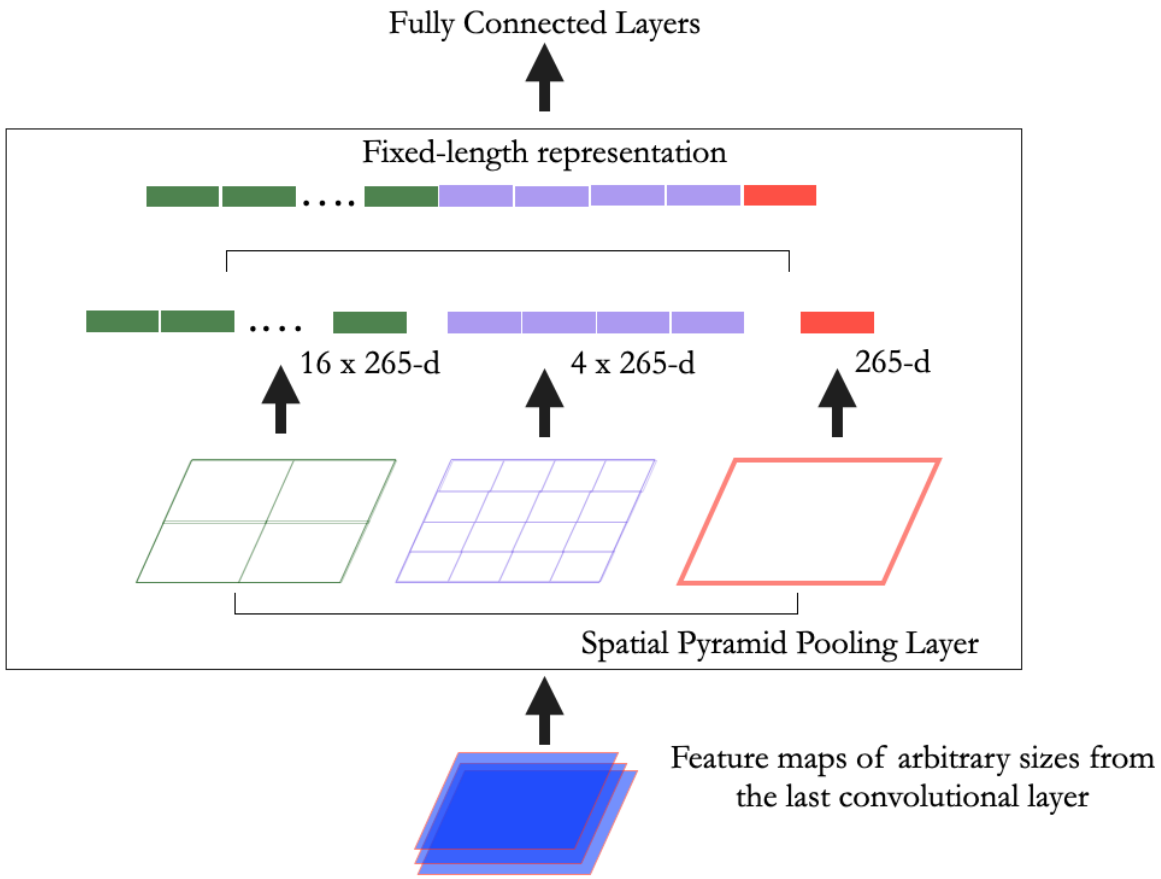


Figure 4.7. The structure of Spatial Pyramid Pooling layer.

SPP maintains spatial information by diving the feature maps of the last convolutional layer into a number of local spatial bins. The bin sizes are made proportional to the image size, so that the number of bins is fixed regardless of the image size. Bins are formed at different levels of granularity. E.g., in Figure 4.7, one layer of 16 bins divides the image into a 4×4 grid, another layer of 4 bins divides the image into a 2×2 grid, and a final layer pools the entire image. In the SPPNet paper [162], max pooling is used to pool the responses of each filter in each spatial bin.

SPP allows arbitrary aspect ratios as well as arbitrary scales. When the input image is at different scales, the network will extract features at different scales. The coarsest pyramid level consists of a single bin that covers the entire image. This performs a "global pooling" operation.

5. ATTENTION GUIDED GENERATIVE ADVERSARIAL NETWORK FOR BUILDING FOOTPRINT EXTRACTION FROM REMOTELY SENSED IMAGERY

In this chapter, we describe our proposed attention-enhanced generative adversarial network for detecting building footprints in remotely sensed images.

The framework is composed of two parts: an attention-enhanced segmentation network ($\mathcal{S}$) and a critic network ($\mathcal{C}$). Our segmentation network, attention units and critic network are described in details in Sections 5.1, 5.2 and 5.3 respectively.

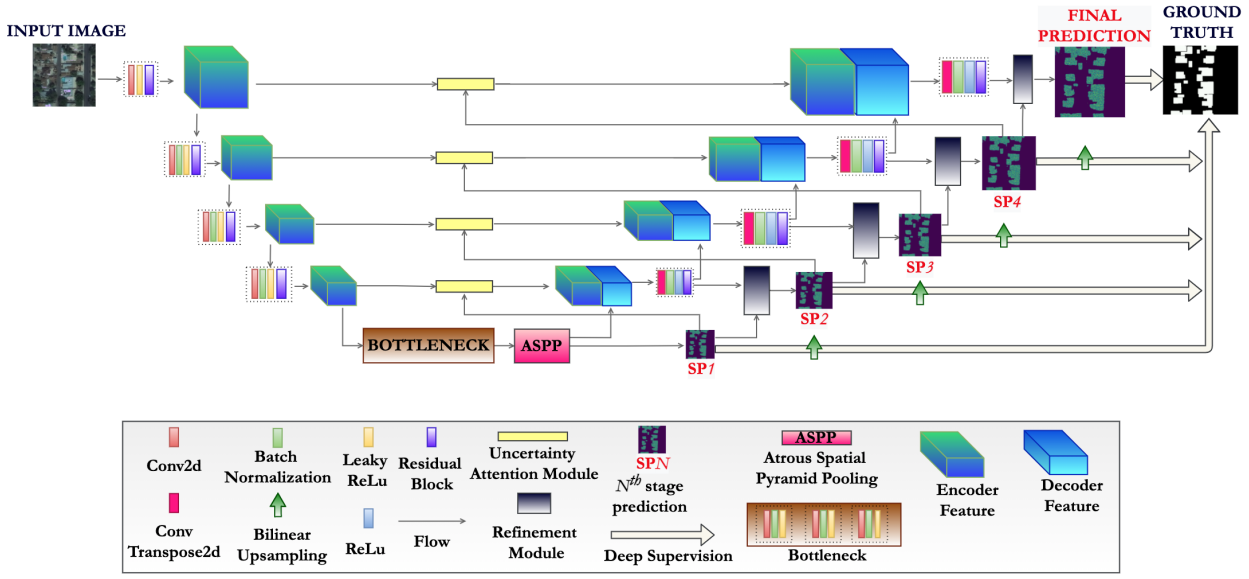


Figure 5.1. The architecture of our proposed segmentation network of our GAN framework. The encoder has 4 strided convolutional layers. At the bottleneck, the feature maps are at 1/16 spatial resolution of input. The decoder is symmetric to the encoder. But larger receptive fields are used to increase scope of each pixel. Residual blocks are added after every downsampling and upsampling layer. ASPP layer is added just after the bottleneck to capture global contextual information. Batch normalization is used after each convolutional layer except the first layer of the encoder. After each batch norm layer, Leaky ReLU is used for the downsampling blocks, and regular ReLU for the upsampling layers. Skip connections via Uncertainty Attention Units are used to concatenate the corresponding encoder and decoder features. Intermediate prediction maps are produced after each stage of decoding. Refinement Module is introduced after each stage in the decoder to gradually refine the intermediate prediction maps.

5.1 Segmentation Network

Our segmentation framework ($\mathcal{S}$), illustrated in Figure 5.1, is a fully convolutional encoder-decoder network that takes in a 3-channel remotely sensed image and generates a 1-channel prediction map in which each pixel value indicates that pixel's probability of belonging to the building class.

$\mathcal{S}$ uses four strided convolutional (Conv) layers for *encoding* the input images. The kernel size is set to 7 for the first two layers and 5 for the next two. The stride is set to 2 in all the layers. The output of the encoder is a feature map at 1/16-th the spatial resolution of the input images. The number of channels goes up by a factor of 2 in each layer.

The feature maps thus produced at the bottleneck layer of the network are processed by an ASPP module [7] to capture the global contextual information for more accurate pixel-wise predictions. The ASPP module consists of a 1×1 Conv layer, three 3×3 Conv layers with dilation rates of 2, 4, and 6, and a global context layer incorporating average pooling and bilinear interpolation. The resulting feature maps from the five layers of ASPP are concatenated and passed through another 3×3 Conv layer, where they form the output of the ASPP module that is fed directly into the decoder. In addition to that, we pass the feature maps from the ASPP module through a 1×1 Conv layer to produce the top-most prediction map that is low in resolution but rich in semantic information.

The *decoder* uses kernels with increasingly larger receptive fields (7,9 and 11) in order to enlarge the representational scope of each pixel. Each layer of the decoder uses a transpose convolution (ConvTranspose2d) to up-sample the incoming feature map while halving the number of feature channels.

Residual blocks are added after every downsampling and upsampling layer. Each residual block consists of a 1×1 Conv, followed by a 3×3 Conv and then another 1×1 Conv. Skip connections are used in a similar fashion as that of the U-Net [39] to concatenate the corresponding layers of the encoder and the decoder. As shown by the yellow boxes in Figure 5.1, an *Uncertainty Attention Module* is used for this concatenation at each abstraction level in network. This allows the network to focus on the features in those regions where

the network has not shown confidence in the predictions made at the lower abstraction level. Detailed description of this module is presented in Section 5.2.2.

Batch normalization is used after each convolutional layer except the first layer of the encoder. After each batch normalization, Leaky ReLU with a leak slope of 0.2 is used in all downsampling blocks, and a regular ReLU has been used for all the upsampling layers.

We also apply a *Refinement Module* consisting of a *Reverse Attention Unit* and an *Edge Attention Unit* in each stage of the decoder. This module is used to learn residual predictions after every stage of decoding and gradually refine the prediction map estimated in the previous stage until the final prediction map is obtained. Details of this module are provided in Section 5.2.1.

5.2 Attention in Segmentation Network

5.2.1 Refinement Module

In general, given a deep network for image segmentation, the high-level feature maps extracted in layers closer to the final output will contain accurate localization information about the objects in the image, but will be lacking in fine detail regarding those objects. On the other hand, the layers closer to the input will be rich in fine detail but with unreliable estimates of where exactly the object is located. The purpose of the Refinement Module is to fuse the fine detail from the lower-indexed layers with the spatial features in the higher-indexed layers with the expectation that such a fusion would lead to a segmentation mask that is rich in fine details and that, at the same time, exhibits high accuracy with regard to object localization.

Such a fusion in our framework is carried out by the *Refinement Module* that is used in each stage of the decoder for refining the prediction map gradually by recovering the fine details lost during encoding. This module does its work through two attention units: *Reverse Attention Unit* (RAU) and *Edge Attention Unit* (EAU). Through residual learning, both these units seek to improve the quality of the predictions made in the previous decoder level on the basis of the finer image detail captured during the current decoder level. What's important here is the fact that both these actions are meant to be carried out in those regions

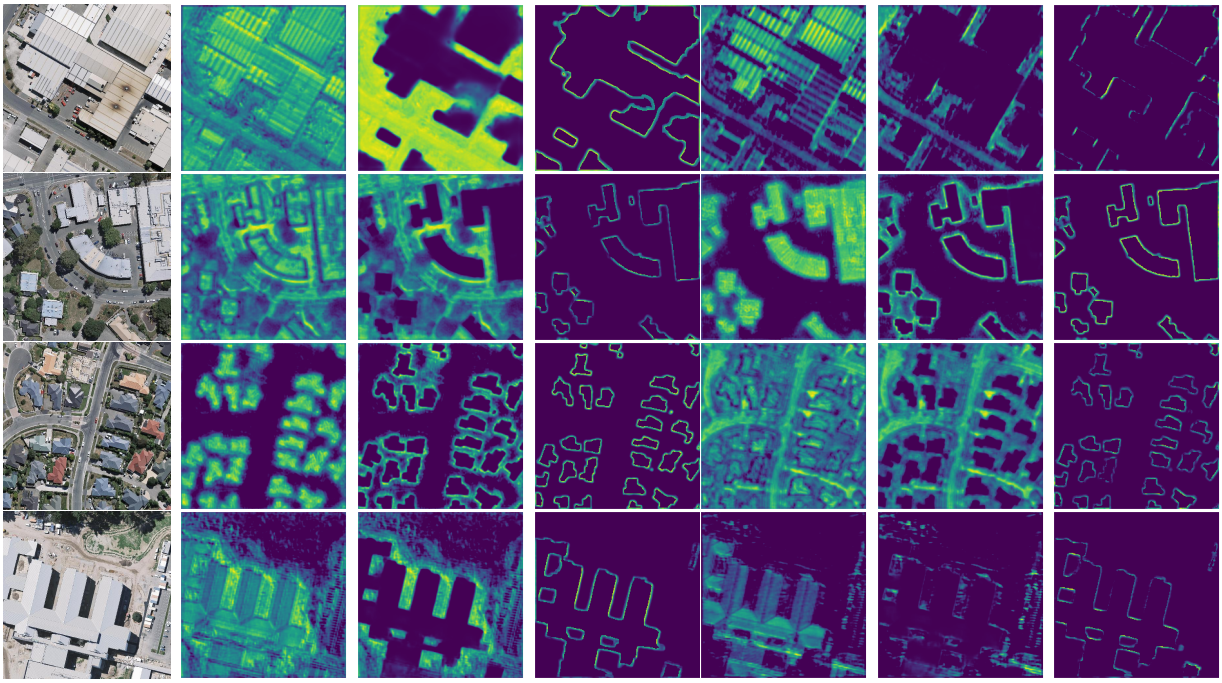


Figure 5.2. Visualization of the decoder feature maps before and after applying reverse and edge attention. Both the attention units focus on areas in the vicinity of building boundaries and in shadow and occluded areas. Column 1: Input image. Columns 2, 5: Decoded Convolutional Features *without* any attention. Columns 3, 6: Decoded Convolutional Features *with* Reverse Attention. Columns 4, 7: Decoded Convolutional Features *with* Edge Attention.

of an image where the accuracy of semantic segmentation is likely to be poor — e.g. in the vicinity of building boundaries, as can be seen in Figure 5.2.

For example, starting with the bottleneck, the encoded features extracted from the ASPP module predict the top-most prediction map that is at low resolution but rich in semantic information. The decoder starts with this coarse prediction map and looks back at it in the next layer of the decoder where additional image detail is available for improving the prediction probabilities that were put out by ASPP and for improving the edge detail associated with the predictions. The former is accomplished by RAU and the latter by EAU. While similar techniques have been used in the past to improve the output of semantic segmentation [163], [164] and object detection [165], we believe that ours is the first contribution that incorporates these ideas for a reliable extraction of building footprints in aerial and satellite imagery.

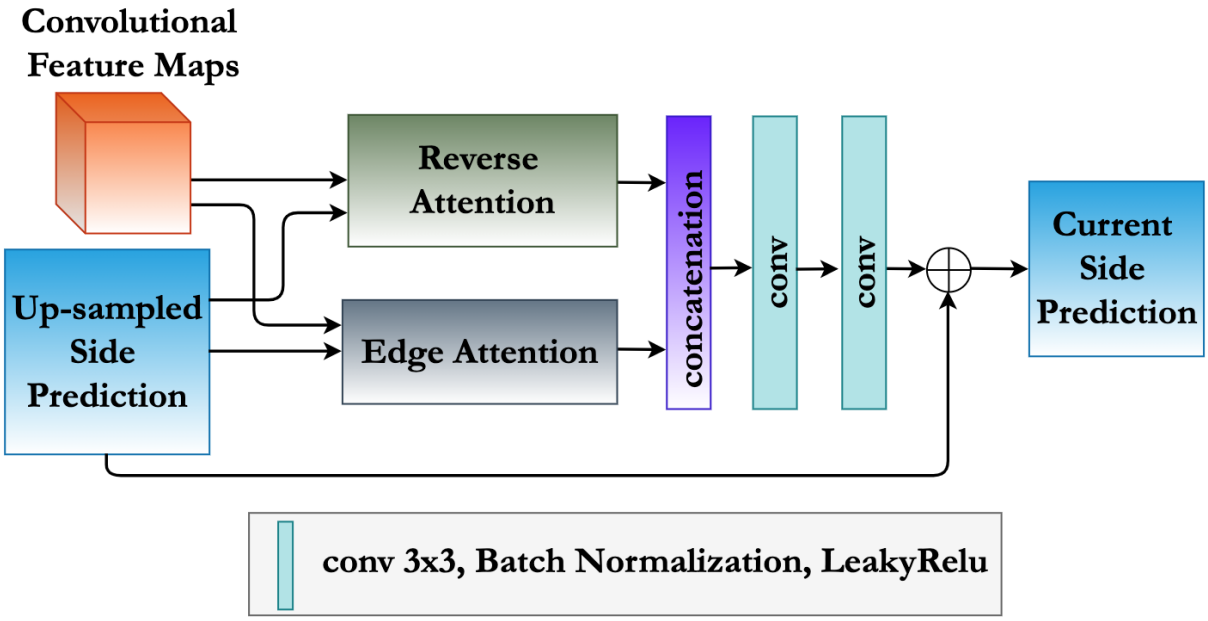


Figure 5.3. Block diagram of our proposed Refinement Module (RM). At the n^{th} layer, the RM takes 2 inputs — (1) the upsampled building prediction map, $U(P_{n-1})$, obtained at the $(n-1)^{th}$ decoder layer, and (2) the concatenated encoder-decoder convolutional feature maps, F_n, after they have been processed by the decoder logic in the n^{th} layer. These inputs are first fed to the reverse attention unit and the edge attention unit in parallel. Then they are passed through two sequential 3×3 Conv blocks, and the output is element-wise added to $U(P_{n-1})$ to generate the predicted building map, P_n, of the n^{th} layer. $\oplus$ denotes element-wise addition.

As shown in Figure 5.3, the Refinement Module concatenates the feature maps that are produced by RAU and EAU. The concatenated feature maps are then passed through two 3×3 Conv layers, and the output of the Refinement Module is then added to the upsampled upper-layer prediction to obtain a finer lower-level prediction, as shown in the figure. The circle with a plus sign inside it in the figure means an element-wise addition of the two inputs. Details regarding the two attention units are presented in the next two subsections.

5.2.1.1 Reverse Attention

The idea of reverse attention is to reconsider the predictions coming out of a lower-indexed layer in the decoder in light of the spatial details available at the current layer. This

amounts to a backward look in the decoder chain and justifies the name of this attention unit.

Figure 5.4 illustrates how the reverse attention mechanism works. The RAU takes two inputs: (1) the upsampled version of the building prediction map produced by the previous decoder layer; and (2) the finer detailed Conv features copied over from the encoder side after they have been processed by the decoder logic in the current layer. As should be evident from the data flow arrows in Figure 5.1, the Reverse Attention Unit (RAU) guides the network to use the fine detail in the current layer of the decoder and reevaluate the building predictions coming out of the lower layer. We refer to these reassessed predictions as *Reverse Attention Map*. At the n^{th} layer, the Reverse Attention Map is generated as follows:

$$A_R^n = 1 - Sigmoid(U(P_{n-1})) \tag{5.1}$$

where P_{n-1} is the building prediction map produced by the $(n-1)^{th}$ layer and $U(P_{n-1})$ is its upsampled version that can be understood directly in the n^{th} layer.

There is a very important reason for the subtraction in the equation shown above: As one would expect, the building detection probabilities are poor near the building edges and that's exactly where we want to direct RAU's firepower, hence the reversal of the probabilities in the equation shown above. As it turns out, this is another reason for "Reversal" in the name of this attention unit.

We now define a *Reverse–Weighted Feature Map*, F_R^n, for the n^{th} layer:

$$F_R^n = A_R^n \otimes F_n \tag{5.2}$$

where the symbol $\otimes$ denotes element-wise multiplication, and F_n represents the convolutional feature maps of the n^{th} layer.

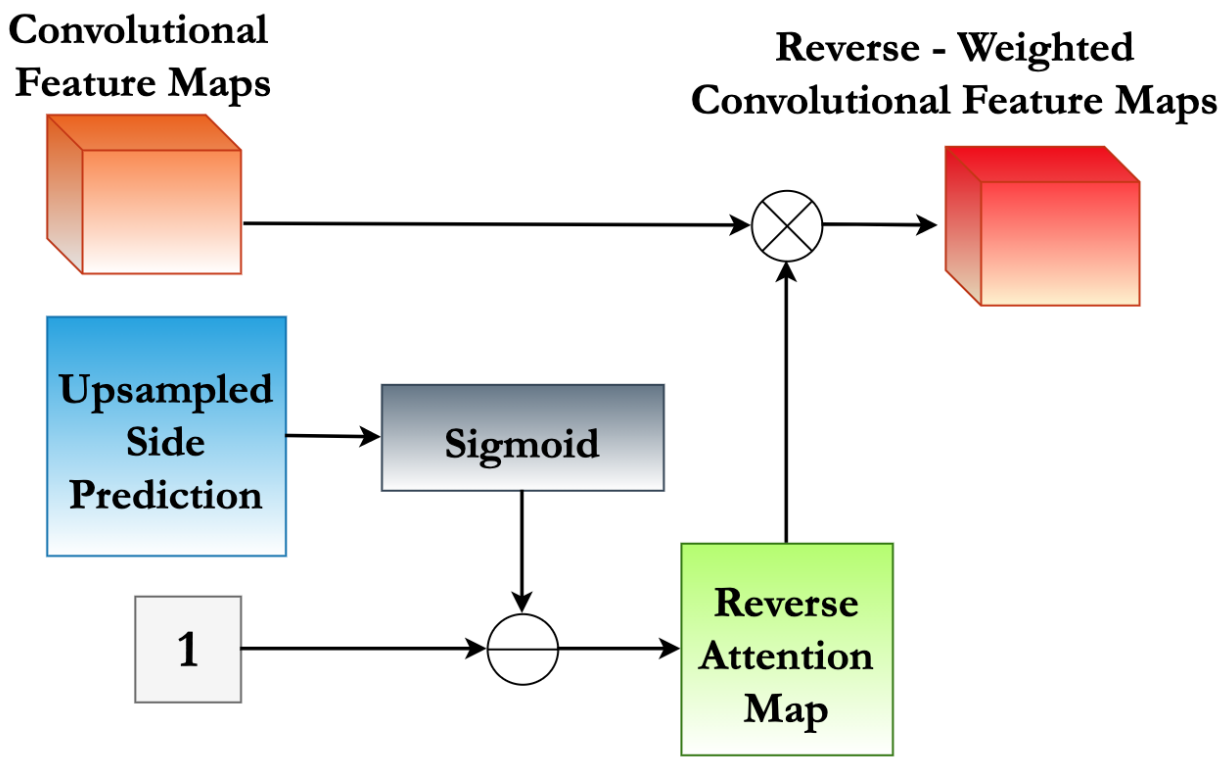


Figure 5.4. Block Diagram of our proposed Reverse Attention Unit (RAU). At the n^{th} layer, the RAU takes 2 inputs — (1) the upsampled building prediction map, $U(P_{n-1})$, obtained at the $(n-1)^{th}$ decoder layer, and (2) the concatenated encoder-decoder convolutional feature maps, F_n, after they have been processed by the decoder logic in the n^{th} layer. $U(P_{n-1})$ is first passed through a Sigmoid activation layer to obtain a probability map. A reverse attention map, A_R^n, is obtained by subtracting the elements of the probability map from an all-one map of same resolution. A_R^n is element-wise multiplied with F_n to obtain the *Reverse–Weighted Feature Map*, F_R^n, of the n^{th} layer. $\otimes$ and $\ominus$ denote element-wise multiplication and subtraction respectively.

5.2.1.2 Edge Attention

The purpose of the edge attention is to improve the quality of the boundary edges of the building predictions made by the previous layer of the decoder using the additional image detail available in the current layer.

Essential to the logic of what improves the boundary edges is the notion of contour extraction. At each layer on the decoder side, we want to extract the contours in the fine detail provided by the encoder side in order to improve the edges in the building prediction

map yielded by the lower layer. Note that there is a significant difference between just detecting the edge pixels and identifying the contours. Whereas the former could yield just a disconnected set of pixels on the object edges, the latter is more likely to yield a set of connected boundary points — even when using just contour fragment (as opposed to, say, closed contours). On account of the need to make these calculations GPU compatible, at the moment the notion of contour extraction is carried out by applying the Sobel edge detector [166] to a building prediction map followed by a p-pixel dilation of the edge pixels identified in order to connect what would otherwise be disconnected pixels.

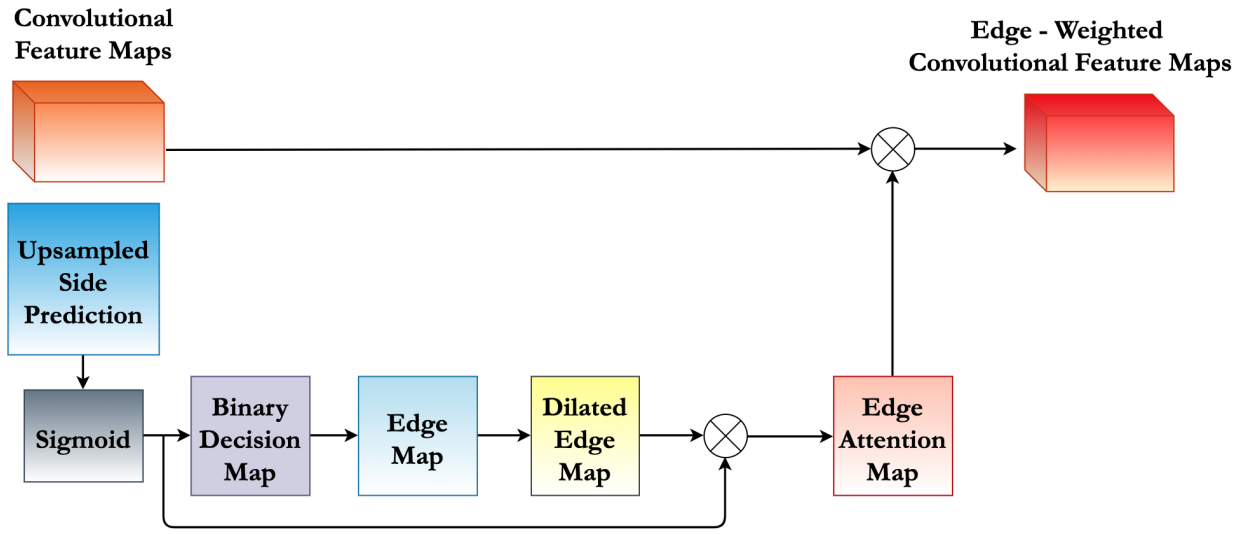


Figure 5.5. Block Diagram of our proposed Edge Attention Unit (EAU). At the n^{th} layer, the EAU takes 2 inputs — (1) the upsampled building prediction map, $U(P_{n-1})$, obtained at the $(n-1)^{th}$ decoder layer, and (2) the concatenated encoder-decoder convolutional feature maps, F_n, after they have been processed by the decoder logic in the n^{th} layer. $U(P_{n-1})$ is first passed through a Sigmoid activation layer to obtain a probability map, $pmap_n$. A binary decision map, B_E^n, is generated by thresholding $pmap_n$. The Sobel edge detector is applied on B_E^n, followed by a dilation operator to get a dilated edge map, D_E^n. D_E^n is then element-wise multiplied with $pmap_n$ to produce the edge attention map, A_E^n. A_E^n is element-wise multiplied with F_n to obtain the *Edge–Weighted Feature Map*, F_E^n, of the n^{th} layer. $\otimes$ denotes element-wise multiplication.

As shown in Figure 5.5, the Edge Attention Unit (EAU) takes two inputs: 1) the upsampled version of the building prediction map produced by the previous decoder layer; and 2) the finer detailed convolutional features copied over from the encoder side after they have

been processed by the decoder logic in the current layer. The output of EAU consists of an *edge-weighted feature map*. If n denotes the index for the current layer in the decoder, the building prediction map produced by the previous layer, denoted P_{n-1}, is first upsampled using bilinear interpolation to get $U(P_{n-1})$, which is then used to generate a *binary decision map*, B_E^n, for the current layer as follows:

$$B_E^n = \begin{cases} 1 & \text{if } Sigmoid(U(P_{n-1})) \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

Subsequently, the Sobel edge detector is applied to the binary decision map in order to detect edge fragments in the predicted binary map. As shown in Figure 5.5, the next step is to dilate the edge fragments produced by Sobel so that they become p-pixels wide. The edge dilation step connects what could otherwise be disjoint edge fragments. Typically, we dilate the edge pixels by a kernel of size 7×7 to get a *dilated edge map*, D_E^n, which leads to the *edge attention map* as defined by:

$$A_E^n = Sigmoid(U(P_{n-1})) \otimes D_E^n \tag{5.4}$$

The edge attention map could be thought of as a boundary confidence map. This confidence map is then multiplied with the n^{th} layer feature map to obtain the edge-weighted features, F_E^n as shown below:

$$F_E^n = A_E^n \otimes F_n \tag{5.5}$$

where F_n is the n^{th} layer feature map.

5.2.2 Uncertainty Attention

In general, a classical encoder-decoder network does not provide for feature selection when fusing together the high-level features going through decoder with the low-level features being copied over from the encoder side through the skip connections. A manifestation of this phenomenon is over-segmentation in the final output of the network that is caused by

indiscriminately fusing the low-level features from the encoder with the high-level features in the decoder.

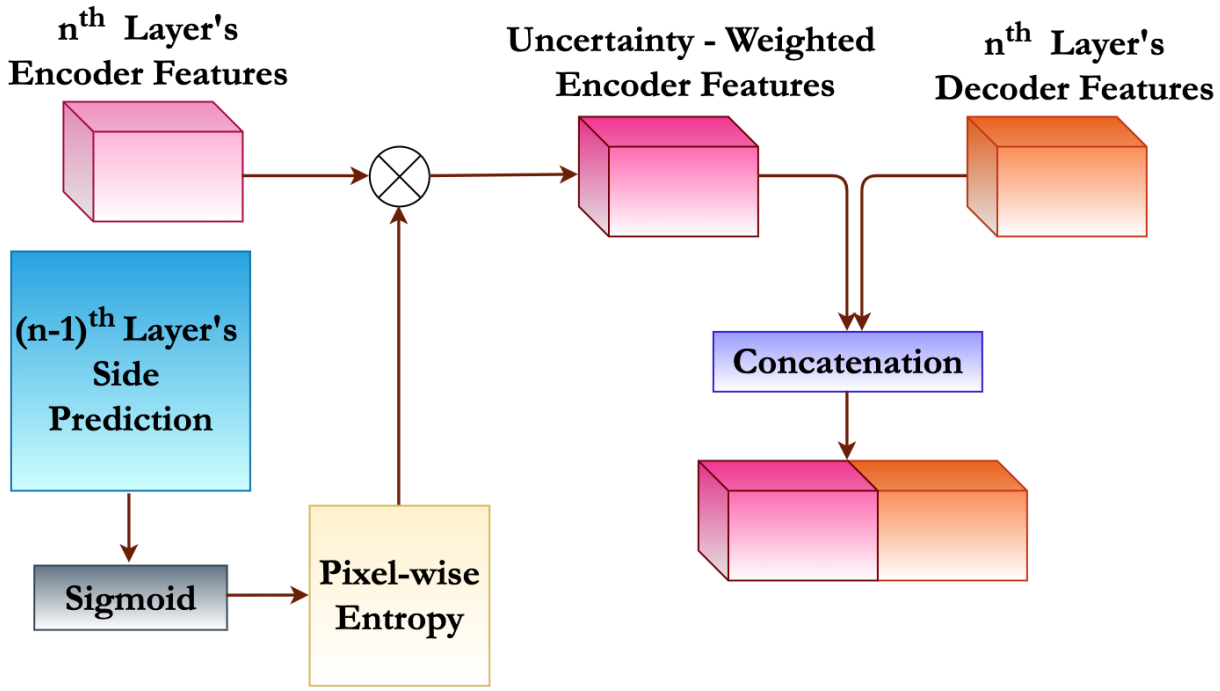


Figure 5.6. Block Diagram of our proposed Uncertainty Attention Module (UAM). At the n^{th} layer, the UAM takes 2 inputs — (1) the upsampled building prediction map, $U(P_{n-1})$, obtained at the $(n-1)^{th}$ decoder layer, and (2) the encoder features, F_n^{enc}, of the n^{th} layer. $U(P_{n-1})$ is first passed through a Sigmoid activation layer to obtain a probability map, p. A pixel-wise entropy map, E, is computed from p. E becomes our uncertainty attention map. E is element-wise multiplied with F_n^{enc} to obtain the *Uncertainty–Weighted Encoder Features* of the n^{th} layer. $\otimes$ denotes element-wise multiplication.

To mitigate such over-segmentation, we introduce an *Uncertainty Attention Module* in every encoder-to-decoder skip connection, as shown by the yellow boxes in the middle of the 'U' in Figure 5.1. The purpose of these attention units is to mediate the level of inclusion for the encoder-generated low-level features when they are copied over to the decoder side. More specifically, we want the Uncertainty Attention Module to use the low-level detail made available by the encoder only in those regions of a prediction map where the degree of uncertainty exceeds a threshold. Experience with such architectures tells us that we can

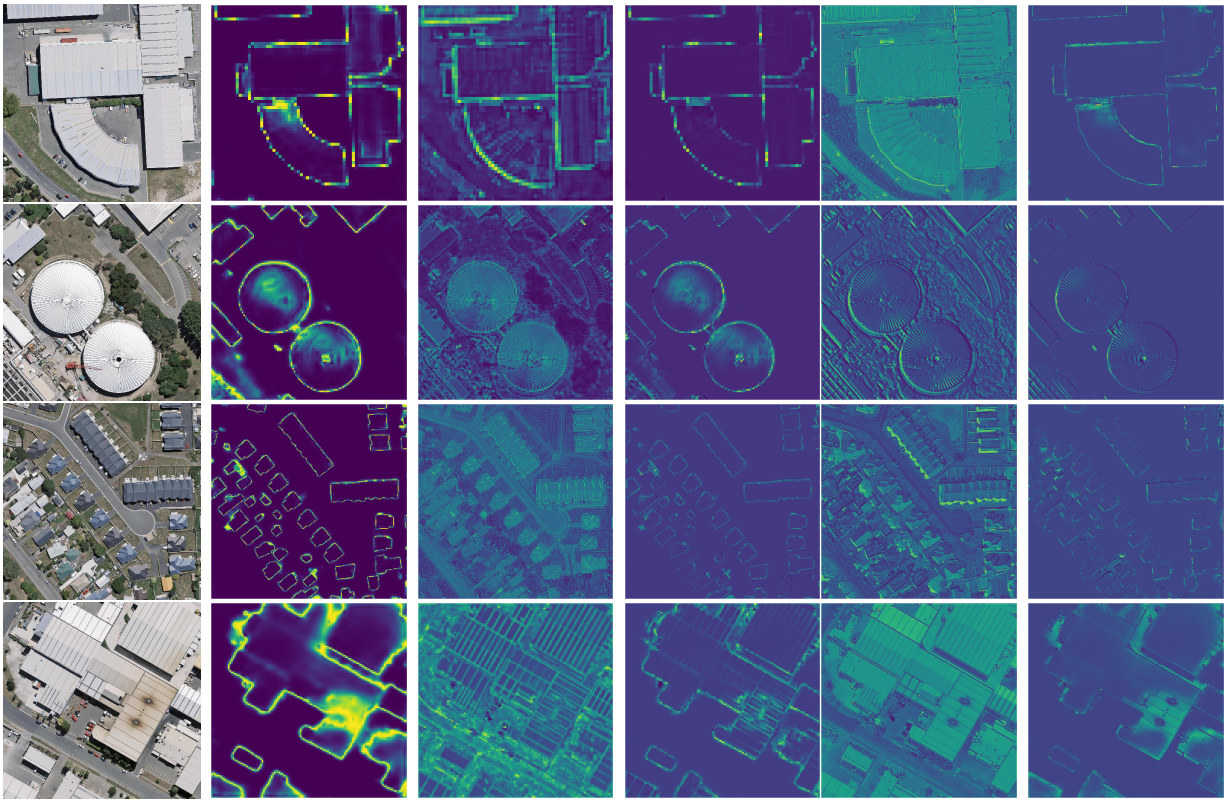


Figure 5.7. Visualization of the encoder feature maps before and after applying uncertainty attention. The uncertainty attention unit focuses on regions in the vicinity of the building boundaries, in the shadow and occluded areas, and in those regions of an image where the building pixel signatures are too close to the background pixel signatures. Column 1: Input image. Column 2: Uncertainty Attention Map. Columns 3, 5: Encoder Features *without* Uncertainty Attention. Columns 4, 6: Corresponding Encoder Features *with* Uncertainty Attention.

expect the uncertainty to be relatively large in the vicinity of the object boundaries in the input images, as can be seen in Figure 5.7.

That raises the question of how to measure the degree of uncertainty associated with the predictions on the decoder side. As it turns out, that's an easy thing to do by measuring the entropy associated with the building predictions in the different levels of decoder. We compute pixel-wise entropy in a prediction map to produce the uncertainty attention map at each level of our network as follows:

$$E(\mathrm{i}) = -p_{\mathrm{i}} \log(p_{\mathrm{i}}) - (1 - p_{\mathrm{i}}) \log(1 - p_{\mathrm{i}}) \tag{5.6}$$

where p_{i} denotes the probability of the i^{th} pixel belonging to the *building class*. This uncertainty attention map is then element-wise multiplied with the low-level feature maps in that specific layer to create an uncertainty–weighted low-level feature map, as shown in Figure 5.6.

Recent research [167] has shown that concatenating shallow encoder features with deep decoder features can adversely affect the predictions if the semantic gap between the features is large. And, it stands to reason that introducing uncertainty attention prior to concatenation has the possibility of amplifying this problem by injecting "noisy" encoder features in those regions of a building prediction map where the probabilities are low. We guard against such corruption of the prediction maps by using deep supervision (shown by thick arrows in Figure 5.1) that forces the intermediate feature maps to be discriminative at all levels of the decoder. Deep supervision [97], [168]–[170] allows for more direct backpropagation of loss to the hidden layers of the network.

5.3 Critic Network

We now present the details regarding the critic network ($\mathcal{C}$) in our framework. The network for $\mathcal{C}$ is essentially the same as the encoder in $\mathcal{S}$ *minus the residual blocks*. Our experiments have shown that adding the residual blocks in $\mathcal{C}$ increase the parameter space of the model without any significant improvement in the performance of the critic.

$\mathcal{C}$ is supplied with two inputs: (a) 3-channel remotely sensed images masked by the corresponding ground-truth building labels; and (b) 3-channel remotely sensed images masked by the building labels generated by $\mathcal{S}$. These masks (predicted and the ground-truth) are created by element-wise multiplication of the one-channel label maps with the original RGB images, as shown in Figure 5.8. $\mathcal{C}$ extracts features from the predicted mask as well as the ground-truth mask at multiple scales, reshapes these multi-scale features into one-dimensional vectors and concatenate them together. Finally, $\mathcal{C}$ seeks to maximize the difference between the vectors created from the true instances and the predicted instances.

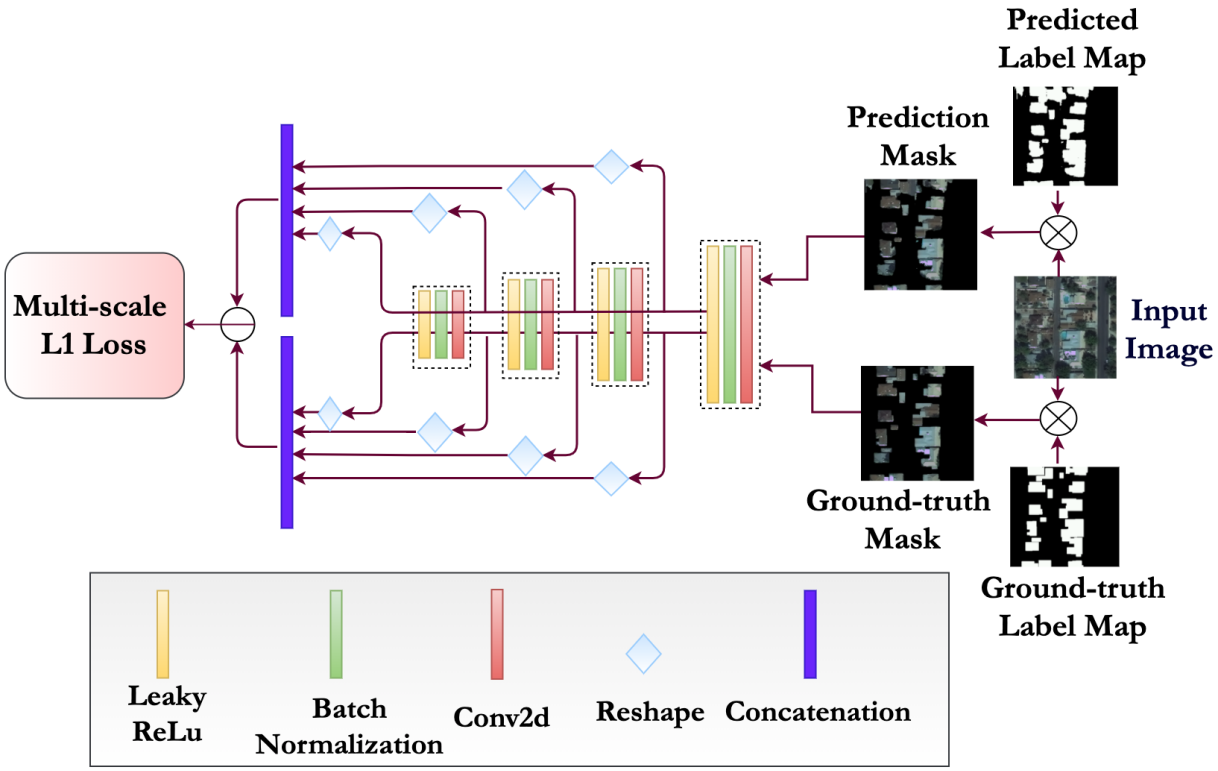


Figure 5.8. The architecture of our Critic framework. The Critic network has 4 strided convolutional layers with kernel size 7 for the first 2 layers and kernel size 5 for the next 2 layers. Each convolutional layer is followed by a batch norm layer and a leaky ReLU layer. The input masks of the critic — the prediction mask and the ground-truth mask — are calculated by pixel-wise multiplication of multiple channels of the input image with the corresponding predicted label map and the corresponding ground-truth label map. Features from the predicted mask and the ground-truth mask are extracted from multiple layers of the Critic. These multi-scale features are then reshaped into one-dimensional vectors and concatenated together. The multi-scale L_1 loss is computed by taking the absolute difference between the vectors created from the true instances and the predicted instances.

6. TRAINING STRATEGY AND LOSS FUNCTIONS

In the previous chapter, we provided the details of our proposed segmentation framework. In this chapter, in Section 6.1 we first describe the strategy we adopted for training our proposed model. Then in Section 6.2, we explain in depth the loss functions used to train our model.

6.1 Training Strategy

We train our proposed segmentation model for detecting building pixels in remotely sensed images in an *adversarial fashion*. We adopt a generative adversarial framework (GAN) as described in Section 4.2. The generator of our proposed GAN framework is basically an attention-enhanced segmentation network ($\mathcal{S}$) whose aim is to predict an accurate label map for the buildings present in the input remotely sensed image such that the adversarial component of our framework cannot distinguish between the predicted map and the ground-truth map. Whereas the adversarial component of our framework i.e. the discriminator, which acts as a critic ($\mathcal{C}$) in our case, aims to discriminate the predicted label maps from the ground-truth label maps.

$\mathcal{S}$ and $\mathcal{C}$ in our proposed architecture are trained *alternatively* by optimizing a *multi-scale L1 loss* [17]. First, we train $\mathcal{C}$ keeping the parameters of $\mathcal{S}$ fixed and try to minimize the negative of L_1 loss. Next, we keep the parameters of $\mathcal{C}$ fixed and train $\mathcal{S}$ minimizing the same L_1 loss. Thus, we can say that the training of $\mathcal{S}$ and $\mathcal{C}$ resembles like playing a min-max game — while $\mathcal{S}$ tries to minimize the multi-scale feature loss, $\mathcal{C}$ aims to maximize it. It is observed that as training progresses, performance of both $\mathcal{S}$ and $\mathcal{C}$ improves, and eventually, $\mathcal{S}$ starts producing predicted label maps that are very close to the ground-truth label maps. The multi-scale feature loss used to optimize our training is the adversarial loss of our framework. It is calculated using the hierarchical features extracted from the multiple layers of $\mathcal{C}$. This loss, proposed by Xue et al. in [17], enables the network to capture the long- and short-range spatial relations between the pixels. Additional details of this multi-scale L_1 loss is provided in Section 6.2.1.

The reason for adopting an adversarial training strategy is by training the network in this manner, the joint distribution of all label variables at each pixel location can be assessed as a whole and forms of high-order consistency can be enforced that neither cannot be enforced by using pair-wise terms, nor can be measured by a per-pixel cross-entropy loss.

Moreover, as we calculate the multi-scale L_1 loss in our proposed GAN architecture by taking the absolute difference between the features maps of generated and ground-truth masked label maps that are extracted at *multiple scales* from the critic, this loss can learn global and local features from multiple layers of the critic. This enables our network to capture the long- and short- range spatial relations between the pixels of an image. Thus, our final model enforces long as well as short-range spatial label contiguity to refine the segmentation results without any time consumption during the inference.

In the context of this dissertation, aggregation of spatial contextual information of the ground objects is essential for the purpose of making full use of the spatial information in the very-high-resolution remotely sensed images. Long-range spatial relationship is needed to leverage on global contextual information captured from the whole image; whereas, short-range spatial relationship is needed to capture the local contexts. We have mentioned throughout this dissertation that one of the main challenges is detecting buildings automatically from remotely sensed images arise from the fact that in many cases, various objects (such as roads, parking lots and building roofs) that are present in aerial and satellite images look very similar and have *very small inter-class differences*. This happens because the reflectivity signatures of several types of building materials are close to what gets used for the construction of roads and parking lots. We also mention that the state-of-the-art algorithms for automatic building detection also face difficulty because of the high variability in the appearance of buildings across the globe — man-made structures like buildings are often built in different materials and with different structures, leading to an incredible diversity of colors, sizes, shapes, and textures. This gives rise to *very high intra-class variation*. Both long-range as well as short-range spatial relationships are crucial to resolve the confusions occurring due to this high intra-class and small inter-class differences.

To this end, we would also like to mention that in addition to the multi-scale adversarial loss, we incorporate extra supervision in the form of weighted dice and shape losses. This

extra supervision is applied to stabilize the training of $\mathcal{S}$ and boost its performance. Details of these losses are provided in Section 6.2.2.

6.2 Training Losses

In this section, we explain in the details the loss functions used to train our proposed model. In Section 6.2.1 we explain the adversarial loss of our GAN framework. Then in Section we provide detailed description of the weighted dice loss and Hausdorff loss that we have used to stabilize the training of our generator network.

6.2.1 Adversarial Loss: Multi-scale L_1 Loss

As mentioned earlier in Section 6.1, the adversarial loss function of our proposed GAN framework is the multi-scale L_1 loss. This loss is calculated using features extracted from multiple layers of $\mathcal{C}$.

Specifically, we calculate the multi-scale loss as follows. $\mathcal{C}$ is supplied with two inputs: (a) 3-channel remotely sensed images masked by the corresponding ground-truth building labels; and (b) 3-channel remotely sensed images masked by the building labels generated by $\mathcal{S}$. These masks (predicted and the ground-truth) are created by element-wise multiplication of the one-channel label maps with the original RGB images. $\mathcal{C}$ extracts features from the predicted mask as well as the ground-truth mask at multiple scales, reshapes these multi-scale features into one-dimensional vectors and concatenate them together. Finally, the multi-scale feature loss is computed by taking the absolute difference between the vectors created from the true instances and the predicted instances.

Utilization of multi-scale features in computation of the adversarial loss function forces $\mathcal{S}$ and $\mathcal{C}$ to learn both global as well as local features of the input image, hence, capturing long- and short-range spatial relationships between pixels.

Our adversarial loss function L_1 is mathematically expressed as:

$$L_1 = \frac{1}{N}\sum_{\mathrm{i}=1}^{N} l_{mae}(f_C(x_\mathrm{i} \circ S(x_\mathrm{i})), f_C(x_\mathrm{i} \circ y_\mathrm{i})) \tag{6.1}$$

where N is the batch size and x_{i} is the i^{th} image in a batch. The notation $S(x_{\mathrm{i}})$ stands for the output label map of $\mathcal{S}$, and y_{i} is the corresponding ground-truth label map. The notation $x_{\mathrm{i}} \circ S(x_{\mathrm{i}})$ stands for the original input sample masked by predicted map and $x_{\mathrm{i}} \circ y_{\mathrm{i}}$ is the input image masked by the ground-truth label map. The notation $f_C(x)$ stands for the features extracted from the image x in multiple layers of $\mathcal{C}$ and l_{mae} stands for the Mean Absolute Error (MAE) defined as:

$$l_{mae}(f_C(x), f_C(y)) = \frac{1}{L} \sum_{k=1}^{L} \|(f_C^k(x) - f_C^k(y))\|_1 \tag{6.2}$$

where $f_C^k(x)$ is the feature map extracted from the image x at the k^{th} layer of $\mathcal{C}$, the subscript mae stands for "mean absolute error", 'L' is the number of layers in $\mathcal{C}$, and $\|.\|_1$ represents $\ell 1$ norm.

6.2.2 Joint Dice and Shape Loss

The overall loss function of our framework also includes weighted dice and shape losses for stabilizing the training of $\mathcal{S}$ and for boosting its performance. It is observed that only using adversarial loss leads to unstable training of the GAN.

The *dice* part of the loss, shown below in Eq. (6.5), optimizes the dice similarity coefficient (DSC) and the *shape* part of the same, shown in Eq. (6.14), minimizes the Hausdorff Distance (HD) [171] between the ground-truth and prediction. Detailed description of these two losses are provided in the upcoming subsections.

6.2.2.1 Weighted Dice Loss

The Dice coefficient (DC) is a widely used metric in computer vision community to gauge the similarity between two images.

In Figure 6.1, we show two sets — 'red set' and 'blue set'. If the sets 'red' and 'blue' overlap perfectly, the Dice coefficient achieves its maximum value of 1; otherwise, the coefficient decreases and becomes 0 if the sets are non-overlapping. Thus, the range of DC is between 0 and 1. Larger the value of the DC, better it is.

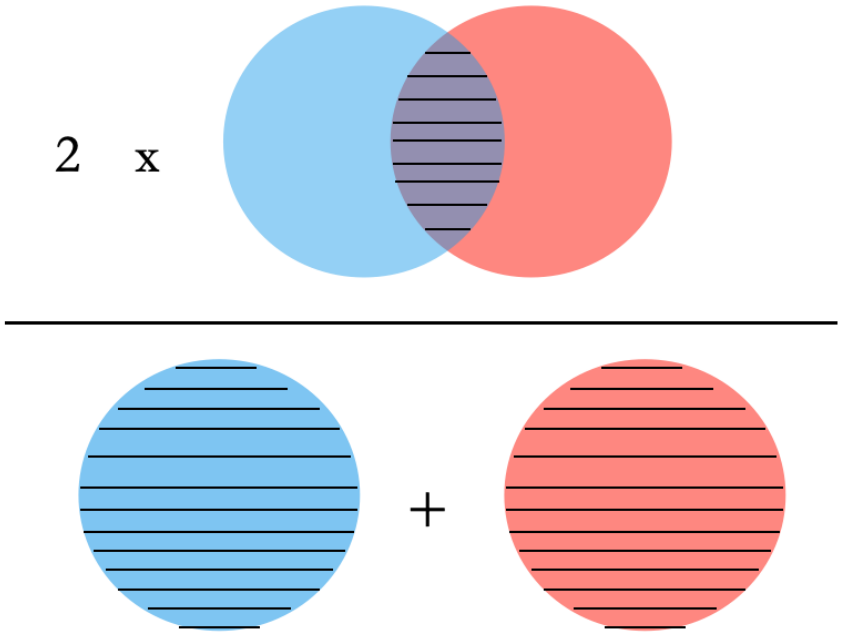


Figure 6.1. Illustration of Dice coefficient (DC) from the perspective of set theory, in which the DC is a measure of overlap between two sets. The areas marked with horizontal lines represents the areas used in computation of the DC.

In semantic segmentation tasks, the ground truth pixel-wise labels and predicted pixel-wise labels can be viewed as two sets (like the 'red' and 'blue' sets in Figure 6.1). By leveraging Dice loss, we can train the two sets to overlap gradually. As shown in Figure 6.1, the denominator considers the total number of boundary pixels at global scale, while the numerator considers the overlap between the two sets at local scale. Therefore, Dice loss considers the loss information both locally and globally, which is critical for high accuracy.

The Dice coefficient (DC) can be mathematically expressed as:

$$DC = \frac{2 * \sum_{\mathrm{i}}^{N} p_{\mathrm{i}} g_{\mathrm{i}}}{\sum_{\mathrm{i}}^{N} p_{\mathrm{i}}^2 + \sum_{\mathrm{i}}^{N} g_{\mathrm{i}}^2} \tag{6.3}$$

where, p_{i} and g_{i} represent pairs of corresponding pixel values of prediction and ground truth, respectively. In an image segmentation task with 2 classes — the target class and the background class, the values of p_{i} and g_{i} are either 0 or 1, representing whether the pixel

belongs to the target class (value of 1) or to the background class (value of 0). Therefore, the denominator is the sum of total pixels of both prediction and ground truth that belong to the target class, and the numerator is the sum of correctly predicted target pixels because the sum increments only when p_{i} and g_{i} match (both of value 1).

In a segmentation problem, the goal is to minimize the the dice loss because that would imply maximizing overlap between the ground-truth label maps and the predicted label maps. Thus, we can write the dice loss as :

$$Dice\ \ Loss = 1 - DC \tag{6.4}$$

In the context of segmenting buildings from remotely sensed images, We observe that the datasets on which we perform experimental evaluations come with a disproportionately large number of true negatives for the background images. So, in this dissertation we use an weighted version of the dice loss. Here is the formula that we used for the dice loss:

$$L_{dice} = 1 - \left[\alpha_1 \frac{2 * \sum_{\mathrm{i}}^{N} p_{\mathrm{i}} g_{\mathrm{i}}}{\sum_{\mathrm{i}}^{N} p_{\mathrm{i}}^2 + \sum_{\mathrm{i}}^{N} g_{\mathrm{i}}^2} \quad + \quad \alpha_2 \frac{2 * \sum_{\mathrm{i}}^{N} (1 - p_{\mathrm{i}})(1 - g_{\mathrm{i}})}{\sum_{\mathrm{i}}^{N} (1 - p_{\mathrm{i}})^2 + \sum_{\mathrm{i}}^{N} (1 - g_{\mathrm{i}})^2}\right] \tag{6.5}$$

where $\alpha_1 + \alpha_2 = 1$. α_1, $\alpha_2 \geq 0$. $p_{\mathrm{i}}, g_{\mathrm{i}}$ represent, respectively, the i^{th} pixel of the ground-truth and the prediction map. This way, in addition to the contribution from the positive samples, we also ensure contribution from the negative samples. This becomes particularly useful if an entire sample is composed of only foreground or only background class. In our experiments, we set $\alpha_1 = 0.8$.

6.2.2.2 Hausdorff Loss

Regarding the shape loss, it helps the system keep a check on the shape similarity between the ground-truth and predicted building labels by minimizing the HD distance between them. Hausdorff Distance loss aims to estimate HD from the CNN output probability so as to learn to reduce HD directly. Specifically, HD can be estimated by the distance transform of ground-truth and segmentation. We first mathematically describe the Hausdorff distance.

Hausdorff distance is defined for distance between two point sets X and Y as

$$h(X,Y) = \max_{x \in X} \min_{y \in Y} \|x - y\|_2 \tag{6.6}$$

By interchanging max and min we obtain

$$h(Y,X) = \max_{y \in Y} \min_{x \in X} \|x - y\|_2 \tag{6.7}$$

In general, $h(X,Y) \neq h(Y,X)$. Hausdorff distance is defined as

$$HD(X,Y) = \max(h(X,Y), h(Y,X)) \tag{6.8}$$

Note that if X and Y are identical, then, $HD(X,Y) = 0$. Otherwise, $HD(X,Y)$ is positive.

In semantic segmentation, HD is computed between the boundaries of the estimated and the ground-truth pixels. We now mathematically define the Hausdorff distance for semantic segmentation.

First, recall that that $p_{\mathrm{i}} \in \{0,1\}$ and $g_{\mathrm{i}} \in [0,1]$ denote the ground truth probability map, and the predicted probability map for the i-th pixel. We denote $\tilde{g}_{\mathrm{i}} \in \{0,1\}$ as the predicted binary label for the i-th pixel. Now, we define the Hausdorff distance between the δp and δg where δp and δq are the boundary of the ground-truth and the boundary of the estimated binary pixel-map respectively.

First, we estimate the HD distance based on the distance-transformation. Specifically, for a 2D binary image $X[\mathrm{i},\mathrm{j}] = 0$ denotes background and $X[\mathrm{i},\mathrm{j}] = 1$ denotes the foreground. Then, the distance transformation for a pixel (i,j) as

$$D_X(\mathrm{i},\mathrm{j}) = \min_{k,l;X[k,l]=1} d([\mathrm{i},\mathrm{j}],[k,l]) \tag{6.9}$$

where

$$d([\mathrm{i},\mathrm{j}],[k,l]) = \sqrt{(k-\mathrm{i})^2 + (\mathrm{j}-l)^2} \tag{6.10}$$

Note that if $X[\mathrm{i},\mathrm{j}] = 1$, then $D_X(\mathrm{i},\mathrm{j}) = 0$ in (6.9). On the other hand if $X[\mathrm{i},\mathrm{j}] = 0$, then let $(k,l) = \arg\min_{(k',l'),X[k',l']=1} d([\mathrm{i},\mathrm{j}],[k',l'])$. (k,l) belongs to the boundary of X and is closest to (i,j). Hence, such a distance transformation can be used to measure the distance from the boundary. Thus, we simply denote d_p (d_q, resp.) as the matrix consisting of the element $D_p(\mathrm{i},\mathrm{j})$ ($D_q(\mathrm{i},\mathrm{j})$, resp.).

Let $p\Delta\tilde{g} = (p \setminus \tilde{g}) \cup (\tilde{g} \setminus p)$. Note that the above expression is equal to $\|p - \tilde{g}\|_1$. See Fig. 6.2 for the illustration of $p\Delta\tilde{g}$. Then, we approximate the Hausdorff distance between the points at the boundary of δp from the boundary δg as

$$hd_{DT}(\delta p, \delta g) = \max_{\Omega}((p\Delta\tilde{g}) \cdot d_p) \tag{6.11}$$

where Ω is total 2D image space or the total number of pixels. See Fig. 6.2 for illustration of $hd_{DT}(\delta p, \delta g)$.

Similarly, the Hausdorff distance from the points at the boundary of δg from the boundary δp is defined as

$$hd_{DT}(\delta g, \delta p) = \max_{\Omega}((p\Delta\tilde{g}) \cdot d_q) \tag{6.12}$$

See Fig. 6.2 for illustration of $hd_{DT}(\delta g, \delta p)$.

Similar to Hausdorff distance between two sets X and Y as in (6.8), we define the Hausdorff distance between two 2D segments (boundaries) as

$$HD_{DT}(\delta g, \delta p) = \max\{hd_{DT}(\delta p, \delta g), hd_{DT}(\delta g, \delta p)\} \tag{6.13}$$

Note that the above expression is not differentiable. Further, it is computationally expensive to obtain $hd_{DT}(\delta p, \delta g)$ and $hd_{DT}(\delta g, \delta p)$. Thus, we approximate such a Hausdorff distance using a smoothed expression which we describe next.

We approximate (6.13) using a shape loss. We compute the average shape loss as follows

-

$$L_{HD} = \frac{1}{N}\sum_{\mathrm{i}=1}^{N}\left[(p_{\mathrm{i}} - g_{\mathrm{i}})^2(d_{p_{\mathrm{i}}}^2 + d_{g_{\mathrm{i}}}^2)\right] \tag{6.14}$$

where d_{p_i} and d_{g_i} are the taxicab (i.e. $\ell - 1$) distance transforms of the ground-truth and predicted label maps. Thus, d_{p_i} is basically $DT_p(i)$ in (6.9) for the ground-truth label map p. N is the number of pixels. Note that we have used g instead of $\tilde{g}$, thus, we use the predicted probability map rather than the predicted binary labels. Thus, we eliminate the errors due to the thresholds. Further, we operate on the continuous domain rather than the discrete domain. Finally, we have used L_2 norm instead of L_1 norm between p and g for smoothness.

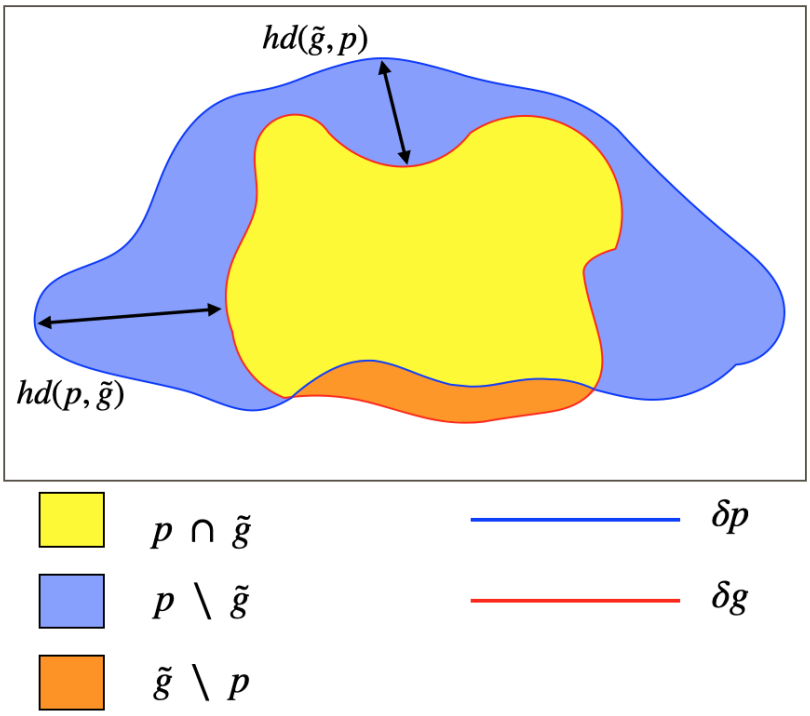


Figure 6.2. Illustration of Hausdorff Distance (hd) for semantic segmentation. Here p is ground-truth binary label map, $\tilde{g}$ is the predicted binary label map obtained from the predicted probability map g. δp and δg are boundaries of the ground-truth foreground and the predicted foreground. $hd(\tilde{g}, p)$ and $hd(p, \tilde{g})$ are described in (6.11) and (6.12) respectively.

7. DATASETS AND EVALUATION METRICS

This chapter gives a detailed description of the datasets that we have used to conduct the qualitative and quantitative evaluations of our proposed framework.

To show the power of our proposed research, we show results on four publicly available datasets – Massachusetts Buildings (MB) Dataset [5], INRIA Aerial Image Labeling Dataset [19], WHU Building Dataset [20] and DeepGlobe Building Detection Dataset [18], [21]. These datasets cover different regions of interest across the world and include diverse building characteristics. We have used different evaluation metrics for different datasets in order to carry out a fair comparison with the other state-of-the-art methods which we have also detailed in this chapter.

7.1 Massachusetts Buildings Dataset

The Massachusetts Buildings (MB) Dataset [5] consists of 151 high-resolution aerial images of urban and suburban areas around Boston. Each image is 1500×1500 pixels and covers an area of $2250 \times 2250 m^2$. The dataset is randomly divided into training (137 tiles), validation (4 tiles), and testing (10 tiles) subsets.

Performance Metric: We now elaborate on the metrics that we have used for comparisons. For the Massachusetts Buildings Dataset, we report **relaxed as well as non-relaxed (i.e. regular) versions of F1-score and IoU score**. We use the **relaxed version of precision, recall, and F1-score** to calculate the precision-recall breakeven point as in [5]. A relaxation factor of ρ was introduced to consider a building prediction correct if it falls within a radius of ρ pixels of any ground-truth building pixel. This relaxation factor is used to provide a realistic performance measure because the building masks in the Massachusetts Buildings Dataset are not perfectly aligned to the actual buildings in the images. The formula for the F1-measure is:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{7.1}$$

where

$$precision = \frac{tp}{tp + fp} \tag{7.2}$$

$$recall = \frac{tp}{tp + fn} \tag{7.3}$$

The relaxed version of precision denotes the fraction of predicted building pixels that are within a radius of ρ pixels of a ground-truth building pixel, and the relaxed version of recall represents the fraction of the ground-truth building pixels that are within a radius of ρ pixels of a predicted building pixel. To conduct a fair comparison with previous research [8], [14], we set $\rho = 3$.

7.2 INRIA Aerial Image Labeling Dataset

This dataset [19] features aerial orthorectified color imagery having a spatial resolution of 0.3m with a coverage of $810km^2$ and contains publicly available ground-truth labels for the building footprints in the training and validation subsets. The images range from densely populated areas like San Francisco to sparsely populated areas in the alpine regions of Austria. Thus, the dataset represents highly contrasting terrains and landforms. *Moreover, the population centers in the training subset are different from those in the testing subset, which makes the dataset very appropriate for assessing a network's generalization capability.*

The training set contains 180 color image tiles of size 5000×5000, covering a surface of $1500 \times 1500m^2$ each (at a 0.30m resolution). There are 36 tiles for each of the following regions: Austin, Chicago, Kitsap County, Western Tyrol and Vienna. Each tile has a corresponding one-channel label image indicating buildings (255) and the not-building class. The test set also contains 180 tiles but from different areas: Bellingham (WA), Bloomington (IN), Innsbruck, San Francisco and Eastern Tyrol.

The performance Measures: The performance measures used for this dataset are: **(a) Intersection over Union (IoU)**: number of pixels labeled as building in both the prediction and the ground truth, divided by the number of pixels labeled as pixel in the

prediction or the ground-truth, and, **(b) Accuracy (acc)**: percentage of correctly classified pixels. The metrics are defined as:

$$IoU = \frac{tp}{tp + fp + fn} \tag{7.4}$$

$$acc = \frac{tp + tn}{tp + tn + fp + fn} \tag{7.5}$$

where tp, tn, fp and fn represent the true positives, true negatives, false positives and false negatives respectively.

7.3 WHU Aerial Building Dataset

The WHU Aerial Buiding Dataset [20] covers an area of 450 km^2 around Christchurch, New Zealand (Figure 7.1) and consists more than 187,000 buildings. The original dataset having a ground resolution of 0.075m comes from the New Zealand Land Information services website. Ji et al. [20] has downsampled the images to 0.3m resolution and cropped them into 8189 non-overlapping tiles with 512 × 512 pixels. The dataset is divided into three parts — 4,736 tiles (130,500 buildings) for training, 1,036 tiles (14,500 buildings) for validation and 2,416 tiles (42,000 buildings) for testing. In this dissertation, we have used the following metrics for evaluating the performance of our proposed method on this dataset – IoU (Eq.7.4), Precision (Eq. 7.2), Recall (Eq. 7.3) and F1-score (Eq. 7.1).

7.4 DeepGlobe Building Dataset

The DeepGlobe Building Dataset [18] uses the SpaceNet Building Detection Dataset [21] (Challenge 2 of the SpaceNet Series). This dataset has been used for the DeepGlobe 2018 Satellite Image Understanding Challenge organised as a part of CVPR 2018 Workshops.

The DeepGlobe Dataset for building detection consists of Digital Globe's WorldView-3 satellite images with 30 cm resolution. The dataset covers 4 different areas of interest (AOIs) with very different landscapes – Vegas, Paris, Shanghai and Khartoum. The training set has 3851 images for Vegas, 1148 images for Paris, 4582 images for Shanghai and 1012

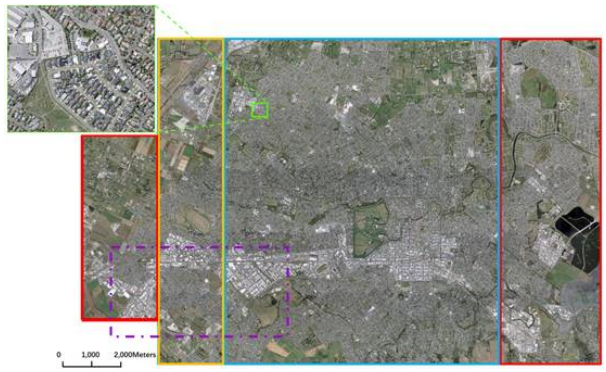


Figure 7.1. The WHU Aerial Building Dataset in Christchurch, New Zealand. The boxes in blue, yellow and red represent the areas used for creating the training, validation and test sets, respectively.

images for Khartoum. In the test set, there are 1282, 381, 1528 and 336 images for Vegas, Paris, Shanghai and Khartoum respectively. Each image is of size 650×650 pixels and covers 200×200 m^2 area on the ground. Each region consists of high-resolution RGB, panchromatic, and 8-channel lower resolution multi-spectral images. In our experiments, we use pansharpened RGB images. Each image comes with its corresponding geojson file with list of polygons as building instances.

The dataset provides its own evaluation tool to compute F1-score as a performance measure. The F1-score is based on individual building object prediction. Each proposed building is a geospatially defined polygon label representing the footprint of the building. The proposed footprint is considered a "true positive" if the intersection over union (IoU) between the proposed and the ground-truth label is at least 0.5. For each labeled polygon, there can at most one "true positive". The number of true positives and false positives are counted for all the test images, and the F1-score is computed from this aggregated count.

8. EXPERIMENTAL SETTINGS AND DATA PREPARATION

This chapter provides a detailed description of our experimental setup. We also discuss how we prepare the datasets during training and testing of our framework.

Our entire segmentation pipeline involves the following steps – image preparation, training our GAN based segmentation model using the training and validation datasets, and, finally applying our trained model to predict building masks for the test images. In this dissertation report, we have shown results on 4 different datasets. Due to the diverse characteristics of the datasets and for performing a fair comparison of our algorithm with other state-of-art methods on those datasets, we preprocess our data differently for each dataset.

In this chapter, we first describe our experimental setup in Section 8.1. Then, we give detailed explanation of the data processing strategies that we use for each dataset during training and inference. We explain our data augmentation strategies in Section 8.2. Then in Section 8.3, we explain in details how we create the training, testing and validation subsets for each of our datasets. In Section 8.4, we discuss how we extract small patches from each image in our datasets and we also explain how we finally fuse the predictions of individual patches to form the integral prediction for the final whole image. Finally, we explain the post-processing strategies adopted by us in Section 8.5.

8.1 Experimental Setup

We have trained our network on four Nvidia GeForce GTX 1080 Ti (11GB) GPUs with images of size 400×400 and batch size of 32. We used the Adam stochastic optimizer with an initial learning rate of 0.0005 and a momentum of 0.9. A poly-iter learning rate [172] with a *power* of 0.9 was used for 200 epochs. The poly-iter learning rate is calculated as -

$$lr = lr_0 * \left(1 - \frac{\mathrm{i}}{T_\mathrm{i}}\right)^{power} \tag{8.1}$$

where lr is the learning rate in the i^{th} iteration, lr_0 is the initial learning rate and T_i is the total number of iterations. Note that when $power = 1$, the learning rate decreases linearly with the number of iterations. As *power* increases, the learning rate decreases at a faster

rate. Since we want that the learning rate to be high at least during initial stages, we set $power = 0.9$. To avoid over-fitting, an L_2 regularization was applied with a weight decay of 0.0002.

8.2 Data Augmentation

During training and inference, we carry out different data augmentation strategies on all four datasets. During training, we perform the following data augmentations – random horizontal flips, random vertical flips, random rotations, and color jitter.

To improve predictive performance of our algorithm, we apply a data augmentation technique during inference – popularly known as Test Time Augmentation (TTA). Specifically, it creates multiple augmented copies of each image in the test set, the model then makes a prediction for each; subsequently, it returns an ensemble of those predictions. We perform 5 different transformations on each test image – flipping the image horizontally and vertically, and rotating the image by 90°, 180° and 270°. This means we obtain 6 predictions for each image patch. We align these 6 predictions by applying appropriate inverse transformation, and produce the final prediction for each patch by averaging these predictions.

8.3 Creating Training, Test and Validation Datsets

The WHU and Massachusetts datasets provide training, validation and testing subsets.

The DeepGlobe dataset provides training and test subsets. We randomly divide the training set into 80/20 ratio with 80% images in the training dataset and 20% images in the validation dataset. This 80/20 subsets are formed such that the ratios of number of images in each of the 4 AOIs is maintained in the training and validation sets.

For the INRIA dataset, we take a different approach for creating the training, validation and test subsets. This dataset also provides training and testing subsets; however, the regions covered in the training and testing subsets are different. The regions in the training subset includes Austin, Chicago, Kitsap, Vienna and West Tyrol; whereas, the test subset consists of image patches from Bellingham, Bloomington, Innsbruck, San Francisco and East Tyrol. It is evident that this dataset is created with the purpose of investigating how transferable

models trained on one set of cities to another set of cities are; to fulfill the same purpose and make our model generalizable to any city in the world, we adopt a k-fold validation technique for training our model, and accordingly, we generate our train, test and validation subsets.

Following the suggestion of the authors of the INRIA dataset paper [19], we create a dataset of 25 images by taking out the first five tiles of each city from the training set (e.g., Austin1-5). In the original dataset paper [19], these 25 images serve as the validation dataset. So, throughout this dissertation, we have referred to these 25 images as *INRIA Validation Dataset*. However, most of the state-of-the-art papers have regarded these 25 images as the testing subset and shown inference results on these images. In our work, we report the performance of our algorithm on the INRIA Validation Dataset (Table 9.5) as well as on the actual test dataset (Table 9.6).

The rest of the training data now consists of a total of 155 images with 31 images from each region. We split these images into 5 folds, one for each region. We train an ensemble of 5 models - each model being trained on 4 regions and validated on the 5^{th} region. Finally, we use an ensemble of 5 models to do prediction on the test images in the INRIA dataset. We compute the integral prediction for an input patch by averaging predictions for each of the models in the ensemble.

8.4 Patch Extraction and Prediction Fusion

During training, we use image patches of size 400×400. For the INRIA Aerial image Labeling Dataset and the Massachusetts Buildings Dataset, the images provided in the datasets are huge – 5000×5000 for the INRIA dataset and 1500×1500 for the Massachusetts dataset. To fit into the GPU memory, we extract a series of patches, of size 400×400, from the original RGB input images and the corresponding ground-truth label maps. The patches are extracted with 30% overlap so that different parts of the images are seen in multiple patches in different locations. The size of the images in the DeepGlobe dataset is 650×650 and that in the WHU dataset is 512×512. So instead of creating overlapping patches, for

these two datasets, we randomly crop patches of size 400×400 as a part of the dynamic data augmentation process.

During inference, memory constraint of a 1080Ti GPU limits the maximum image size to be processed by our algorithm to 2000×2000. We could process whole images from the WHU, Massachusetts and DeepGlobe datasets in one pass. However, to evaluate the performance of our algorithm on the INRIA dataset, we extract patches of size 2000×2000 with 50% overlap, perform segmentation on individual patches and merge the predictions of individual patches into an integral prediction for the whole image. Weighted averaging is applied to merge the predictions in overlapping areas.

8.5 Post-processing

Once we have a prediction map for a whole test image, we binarize it to obtain our final building mask. The optimal threshold for binarization is determined by evaluating the respective metrics on the validation images of a specific dataset.

9. RESULTS

In this chapter we describe the quantitative and qualitative results of several experiments that we have conducted to verify the effectiveness of our proposed framework. We report detailed comparison of our proposed algorithm with the state-of-the-art building segmentation approaches, and show how our method overcome the shortcomings of the current best-scoring approaches.

The chapter has four sections covering the experiments that we have conducted on the four different datasets. We start with reporting our experiments on the Massachusetts Buildings (MB) Dataset [5] in Section 9.1. Section 9.2 gives a detailed account of our algorithm's performance on the INRIA Aerial Image Labeling Dataset [19]. Subsequently in Section 9.3, we explain the results obtained on the WHU Building Dataset [20]. Finally, Section 9.4 describes the results obtained on the DeepGlobe Building Detection Dataset [18], [21] using our attention enhanced GAN-based segmentation framework.

9.1 Quantitative Evaluation on the Massachusetts Buildings Dataset

Table 9.1 presents a relaxed F1-Score (as discussed in Section 7.1) based comparison between the different frameworks on the Massachusetts Buildings Dataset. Our network without TTA achieves a 0.53% performance improvement over the previous best performance [109] which uses a significantly deeper neural network of 158 layers. The non-TTA version of our algorithm outperforms the shallower version of their network (56 layers) by 0.92% in terms of relaxed F1-score. With TTA, we outperform the previous best model by 1.29%.

Table 9.2 demonstrates that our proposed method outperforms other state-of-the-art approaches by at least 2.77% and 3.89% in terms of non-relaxed F1 and IoU scores respectively. Figure 9.1 presents our semantic segmentation result on 1500×1500 test image patches from the Massachusetts Buildings Dataset. It is evident from the figure that our proposed architecture's performance is very close to the optimal result. The relaxed F1 score achieved is 98.03 which is almost close to the perfection. Hence, it shows the efficacy of our approach.

In Table 9.3, we report the relaxed F1 as well as relaxed IoU scores for our framework and compare the performance of the framework with some benchmark image segmentation

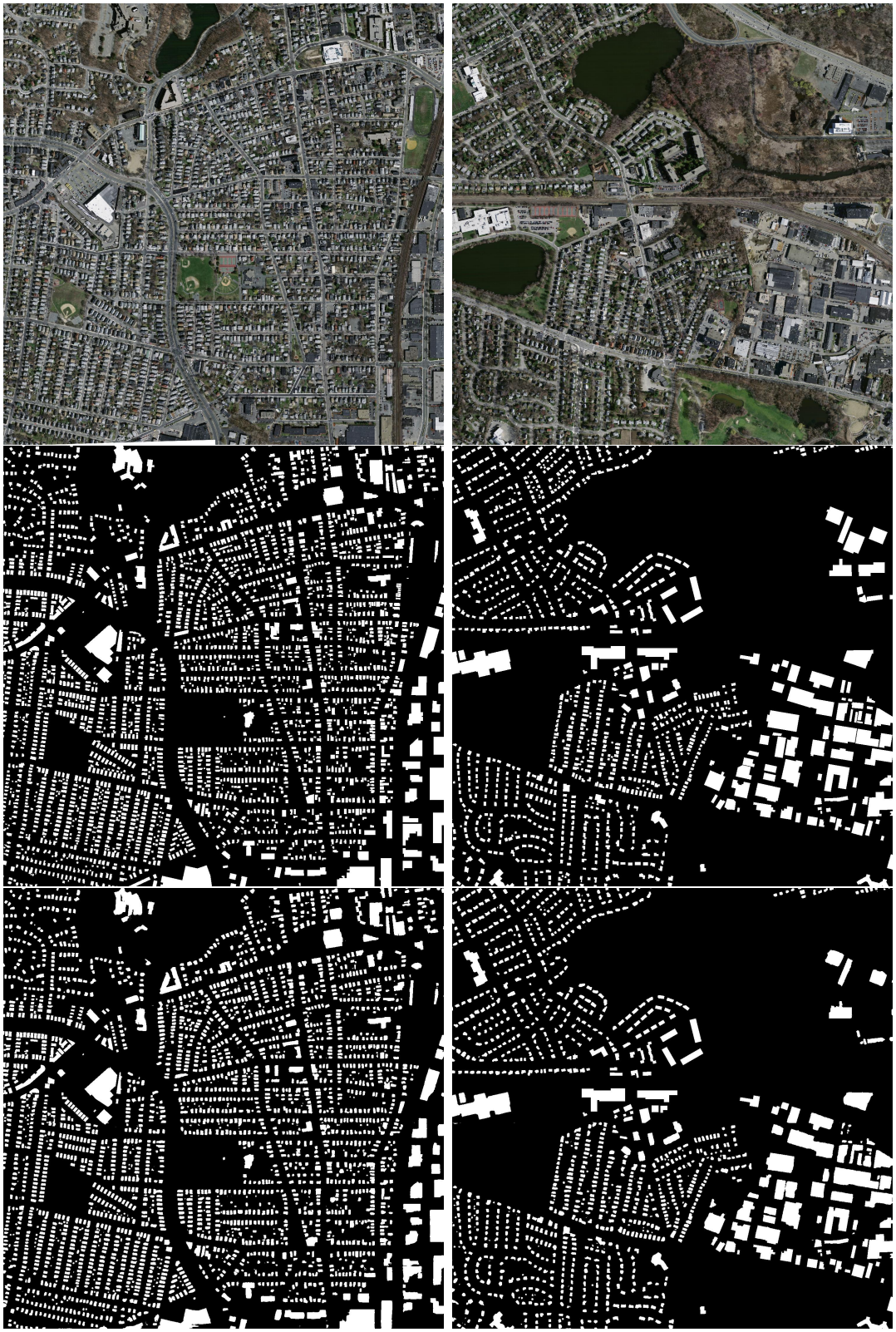


Figure 9.1. Illustration of our qualitative results on the Massachusetts Buildings Dataset. Row 1: Input image. Row 2: Ground-truth Label Map. Row 3: Predicted Label Map.

Table 9.1. Relaxed F1-scores of different deep learning based networks on the Massachusetts Buildings Dataset. TTA: Test Time Augmentation. The best results are highlighted in bold.

Method	Relaxed F1
Mnih & Hinton [5]	92.11
Saito et al. [6]	92.30
DeepLab v3+ [13]	92.65
Khalel et al. [8]	96.33
MSMT-Stage-1 [78]	96.04
GAN-SCA [14]	96.36
Building-A-Nets (56 layers) [109]	96.40
Zhang et al. [93]	96.72
Building-A-Nets (158 layers) [109]	96.78
Our Method (no TTA)	97.29
Our Method + TTA	**98.03**

approaches when adversarial loss is added to them [13]. Rows 5 and 6 show the performance of our vanilla generator (no attention) and our attention-enhanced generator (with attention) networks. It is clear that the addition of adversarial loss consistently offers better performance across all the metrics, and our attention-guided adversarial model performs best among all the adversarial networks as well. Thus, it shows that using attention mechanism with the adversarial losses, the performance can be ramped up.

Table 9.2. Regular F1 and IoU scores for the state-of-the-art networks on the Massachusetts Buildings Dataset. TTA: Test Time Augmentation. The best results are highlighted in bold.

Method	F1	IoU
DRNet [80]	79.50	66.0
GMEDN [85]	-	70.39
SRI-Net [105]	83.58	71.8
ENRU-Net [15]	84.41	73.02
MSCRF [11]	84.75	71.19
Chen et al. [99]	84.72	73.49
DS-Net2 [103]	84.91	73.79
DS-Net [84]	-	74.43
BMFR-Net [83]	85.14	74.12
BRRNet [81]	85.36	74.46
Liao et al. [82]	85.39	74.51
Zhang et al. [93]	85.49	-
Our Method (no TTA)	86.98	76.97
Our Method + TTA	**87.86**	**77.41**

9.2 Quantitative Evaluation on the INRIA Aerial Image Labeling Dataset

As mentioned in Section 8.3, we adopt a k-fold validation strategy for training our network on the INRIA Dataset. In our experiments, k = 5. In Table 9.4, we report the training as well as the validation IoU and accuracy scores of these 5 models. We also report the overall performance of each model on the INRIA Validation Dataset. When we train the model using the datasets from the cities Austin, Kitsap, W.Tyrol, and Vienna, the IoU and the accuracy scored on the INRIA validation datasets are the highest. In this fold, the trained model is validated on Chicago. Intuitively, Chicago dataset has dense buildings, hence, the

Table 9.3. Comparison of benchmark image segmentation models with adversarial loss on the Massachusetts Buildings Dataset. *adv* represents adversarial loss. The scores of our method reflect the results of our algorithm using TTA. The best results are highlighted in bold.

Method	Relaxed F1	Relaxed IoU
PSPNet	89.52	81.2
PSPNet + *adv*	91.17	83.78
FC-DenseNet	94.33	89.27
FC-DenseNet + *adv*	95.59	91.55
Our vanilla Generator	94.11	91.64
Our proposed Generator ($\mathcal{S}$)	96.82	94.79
Our Method ($\mathcal{S}$ + $\mathcal{C}$)	**98.03**	**96.19**

Table 9.4. Comparison of different models in our ensemble of k-fold training on the training and validation subsets of the INRIA Aerial Image Labeling Dataset. Val.: Validation. Acc.: Accuracy

Model #	Train Cities	Train IoU	Train Acc.	Val. City	Val. IoU	Val. Acc.	INRIA Val. IoU	INRIA Val. Acc.
1	Austin, Chicago, Kitsap, W. Tyrol	80.26	96.01	Vienna	78.24	94.13	79.47	96.54
2	Austin, Chicago, Kitsap, Vienna	81.86	96.74	W. Tyrol	79.32	98.29	79.15	97.23
3	Austin, Chicago, W. Tyrol, Vienna	82.93	94.11	Kitsap	70.26	99.22	81.74	97.14
4	Austin, Kitsap, W. Tyrol, Vienna	82.26	95.03	Chicago	72.63	92.46	82.97	95.38
5	Chicago, Kitsap, W. Tyrol, Vienna	79.66	95.29	Austin	80.29	96.78	77.45	96.37

detection of buildings is much difficult which hinders the generalization capability. For more discussion on this, please see the next section. However, the scores do not differ much across different folds.

In Table 9.5, we compare the result of our framework with some of the state-of-the-art approaches on the INRIA Validation Dataset. Specifically, we report the IoU and accuracy scores for the different methods. Since the dataset comes with a disproportionately large number of true negatives for the background images, the accuracy numbers achieved with

Table 9.5. Comparison of the performance of our proposed algorithm with the state-of-the-art networks on the INRIA Validation Dataset. The best results are highlighted in bold. TTA: Test Time Augmentation.

Method	Evaluation Metrics	Austin	Chicago	Kitsap	W. Tyrol	Vienna	Overall
FCN (baseline) [19]	IoU	47.66	53.62	33.70	46.86	60.60	53.82
	Accuracy	92.22	88.59	98.58	95.83	88.72	92.79
MLP (baseline) [19]	IoU	61.20	61.30	51.50	57.95	72.13	64.67
	Accuracy	94.20	90.43	98.92	96.66	91.87	94.42
Mask R-CNN [14]	IoU	65.63	48.07	54.38	70.84	64.40	59.53
	Accuracy	94.09	85.56	97.32	98.14	87.40	92.49
MSMT-Stage-1 [78]	IoU	75.39	67.93	66.35	74.07	77.12	73.31
	Accuracy	95.99	92.02	99.24	97.78	92.49	96.06
SegNet+Multi-Task Loss [90]	IoU	72.43	77.68	72.28	64.34	76.15	74.49
	Accuracy	95.71	95.60	95.81	98.76	94.48	96.07
2-levels U-Nets [8]	IoU	77.29	68.52	72.84	75.38	78.72	74.55
	Accuracy	96.69	92.40	99.25	98.11	93.79	96.05
U-Net [14]	IoU	79.95	70.18	68.56	76.29	79.92	76.16
	Accuracy	97.10	92.67	99.31	98.15	94.25	96.31
GMEDN [85]	IoU	80.53	70.42	68.47	75.29	80.72	76.69
	Accuracy	97.19	92.86	99.30	98.05	94.54	96.43
GAN-SCA [14]	IoU	81.01	71.73	68.54	78.62	81.62	77.75
	Accuracy	97.26	93.32	99.30	98.32	94.84	96.61
SEResNeXt101-FPN-CPA [100]	IoU	80.15	69.54	70.36	80.83	81.43	77.29
	Accuracy	97.18	92.78	99.32	98.46	94.67	96.48
Building-A-Nets [109]	IoU	80.14	**79.31**	72.77	74.55	75.71	78.73
	Accuracy	96.91	**97.06**	96.99	93.52	**98.09**	96.71
Our Method (no TTA)	IoU	82.97	75.77	72.96	84.68	82.78	80.24
	Accuracy	97.67	94.45	99.19	98.82	94.91	96.89
Our Method + TTA	IoU	**83.78**	76.39	**73.25**	**85.72**	**83.19**	**81.28**
	Accuracy	**97.75**	94.83	**99.37**	**98.91**	95.09	**97.03**

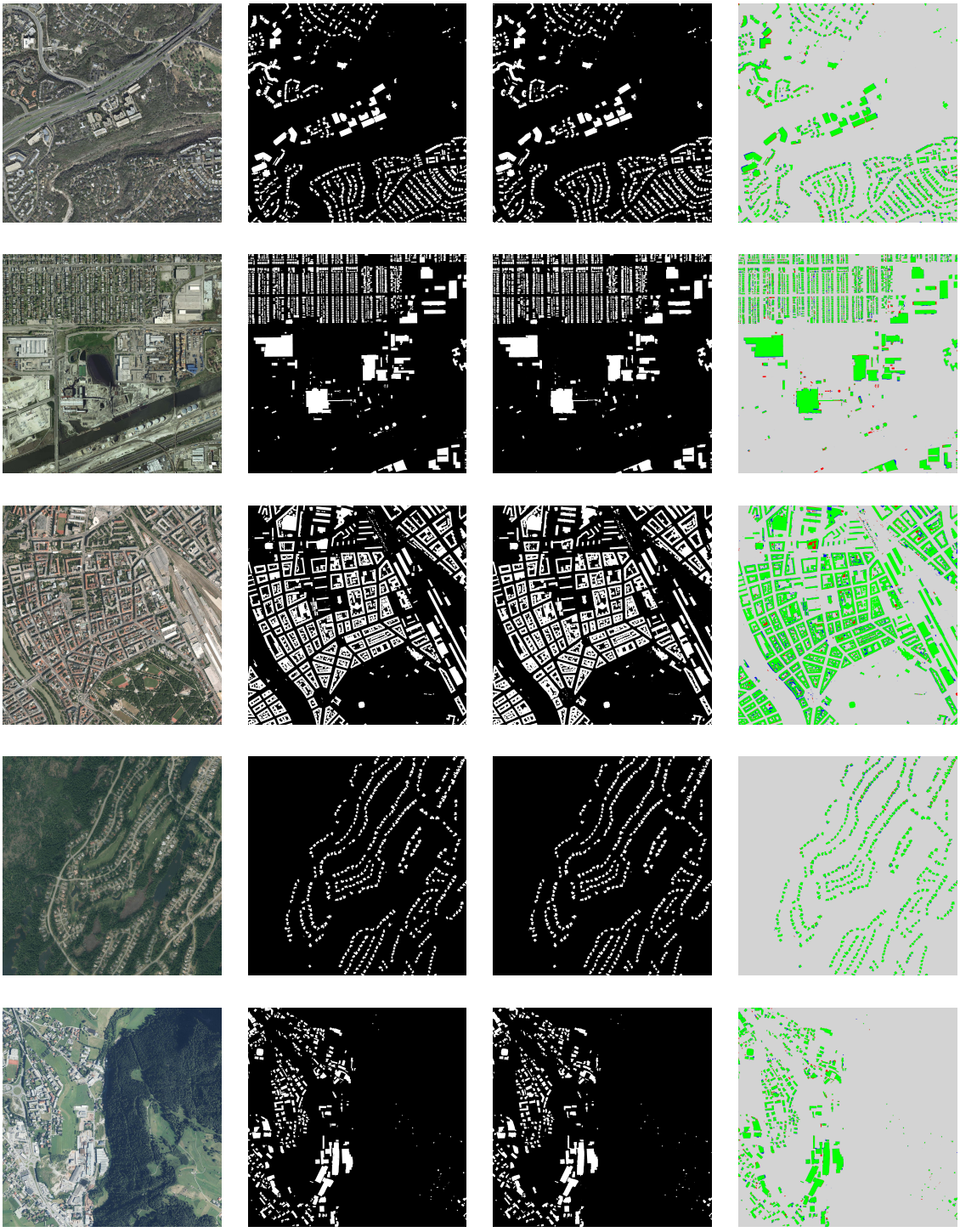


Figure 9.2. Illustration of our qualitative results on the INRIA Aerial Image Labeling Validation Dataset. Rows 1, 2, 3, 4 and 5 show results on image patches over Austin, Chicago, Vienna, Kitsap and West Tyrol respectively. Column 1: Input Image. Column 2: Ground-truth Label Map. Column 3: Predicted Label Map. Column 4: Green: True Positives; Blue: False Positives; Red: False Negatives; Grey: True Negatives.

Table 9.6. Comparison of our framework with other state-of-the-art approaches on the test set of the INRIA Aerial Image Labeling Dataset. The best results are highlighted in bold.

Method	Evaluation Metrics	Bellingham	Bloomington	Innsbruck	San Francisco	East Tyrol	Overall
Building-A-Nets [109]	IoU	65.50	66.6	72.59	76.14	71.86	72.36
	Accuracy	96.39	96.85	96.73	91.96	97.48	95.88
U-Net-ResNet101 [173]	IoU	69.75	72.04	74.64	74.55	77.40	73.91
	Accuracy	96.77	97.13	96.83	91.14	97.92	95.96
Zorzi et al. [174]	IoU	70.36	73.01	73.34	75.88	76.15	74.40
	Accuracy	96.99	97.36	96.77	91.55	97.84	96.10
DS-Net [84]	IoU	71.74	70.55	75.44	77.26	78.54	75.52
	Accuracy	97.22	97.27	97.11	92.47	98.10	96.43
Zhang et al. [93]	IoU	72.25	72.49	75.21	77.70	78.06	75.94
	Accuracy	97.25	97.41	97.07	92.54	98.04	96.46
Milosavljevic et al. [175]	IoU	73.90	72.97	77.31	76.46	80.41	76.27
	Accuracy	97.35	97.39	97.32	92.01	98.23	96.46
E-D-Net [95]	IoU	73.12	75.58	77.66	79.81	80.61	78.08
	Accuracy	97.22	97.64	97.31	93.26	98.25	96.73
ICT-Net [3]	IoU	**74.63**	**80.80**	**79.50**	**81.85**	**81.71**	**80.32**
	Accuracy	**97.47**	**98.18**	**97.58**	**94.08**	**98.39**	**97.14**
Our Method	IoU	74.41	77.29	76.93	76.82	80.11	77.86
	Accuracy	97.03	97.64	96.70	90.49	98.16	96.41

this dataset are generally high, as can be seen by the entries for accuracy in Tables 9.4-9.6. On the other hand, since the IoU metric takes into account both the false alarms and missing detections, we believe that that is a better metric of performance on this dataset.

For the individual cities, as shown in Table 9.5, we have highlighted the highest valued entries for each of the two evaluation metrics. Our network achieves performance improvement of at least 3.42%, 0.56%, 6.05% and 1.92% over Austin, Kitsap, W. Tyrol and Vienna

respectively. Our network also gives better accuracy for Austin, Kitsap and W. Tyrol. For Chicago, though our IoU and accuracy are smaller than [109] by 3.82% and 2.35% respectively, overall our algorithm outperforms [109] as well as other state-of-the-art methods by at least 3.24% and 0.33% in terms of IoU and accuracy respectively. The potential reason behind the slight deterioration in performance is discussed in next section.

These results show that our network gives consistently good performance over all the cities in the INRIA Validation Dataset, while also yielding the best performance for a subset of the cities. Figures 9.2 and 9.3 illustrate some of our building segmentation results on the INRIA Validation and Test Dataset.

In Table 9.6, we compare the performance of our framework with some other state-of-the-art methods on the official INRIA Test Dataset. Though we do not achieve best scores on this subset, our performance is pretty competitive with the state-of-the art methods. Most of the state-of-the-art methods that perform better than us on the INRIA Test Dataset either use pretrained feature extraction networks [63], [151] as backbones or are significantly deeper than our proposed network. Hence, the comparison is not fair, yet, our algorithm achieves competitive result. Apart from dataset on San Francisco, our model achieves second-best result, further, the difference with the best result is not much. The potential reason behind the slight deterioration of the performance on the San Francisco dataset has been addressed in the next section. This shows effective generalization capability of our network. Notice the drop in both the accuracy and IoU values when applying the trained network to a set of different geographic areas. This is to be expected, since each city has some unique specifics.

9.3 Quantitative Evaluation on the WHU Building Dataset

In Table 9.7, we report the IoU, precision, recall and F1-scores obtained using our proposed algorithm on the WHU test dataset and compare these scores with some of the best performing state-of-the-art building segmentation approaches.

Our method without TTA achieves 91.68% IoU, 96.41% precision, 94.92% recall and 95.66% F1 score, respectively. As can be seen from Table 9.7, our proposed approach without

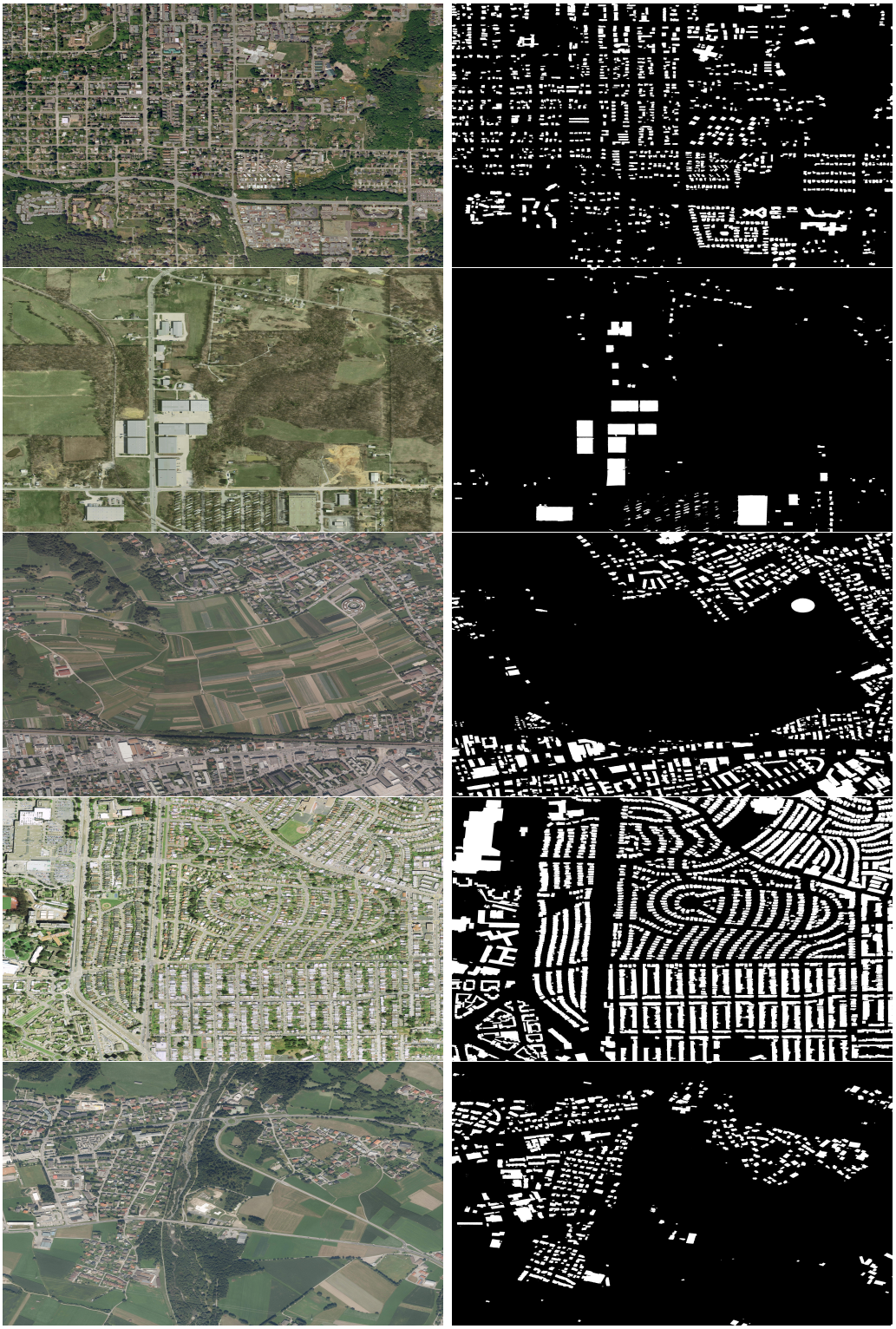


Figure 9.3. Illustration of our qualitative results on the INRIA Aerial Image Labeling Test Dataset. Rows 1, 2, 3, 4 and 5 show results on image patches over Bellingham, Bloomington, Innsbruck, San Francisco and East Tyrol respectively. Column 1: Input Image. Column 2: Predicted Label Map.

Table 9.7. IoU, Precision, Recall and F1-scores for the state-of-the-art networks on the WHU Building Dataset. The best results are highlighted in bold. TTA: Test Time Augmentation.

Method	IoU	Precision	Recall	F1
BRRNet [80], [81]	85.9	93.5	91.3	92.4
DRNet [80]	86.0	92.7	92.2	92.5
RefineNet [46], [94]	86.9	93.7	92.3	93.0
PISANet [102]	87.97	94.20	92.94	93.55
SiU-Net [20]	88.4	93.8	93.9	93.8
SRI-Net [105]	89.09	95.21	93.28	94.23
BMFR-Net [83]	89.32	94.31	94.42	94.36
Chen et al. [99]	89.39	93.25	95.56	94.4
Res-U-Net [176]	89.46	94.29	94.53	94.43
HRLinkNetv2 [86]	89.53	94.56	94.40	94.48
DeepLab v3 + [103]	89.61	94.68	92.36	94.52
DE-Net [177]	90.12	95.00	94.60	94.80
DS-Net2 [103]	90.4	94.85	95.06	94.96
He et al. [94]	90.5	95.1	94.9	95.0
MA-FCN [104]	90.7	95.2	95.1	95.15
ARC-Net [4]	91.8	96.4	95.1	95.70
Our Method (no TTA)	91.68	96.41	94.92	95.66
Our Method + TTA	**92.27**	**96.73**	**95.24**	**95.98**

TTA performs very similar to the previous best scoring algorithm – ARC-Net [4] by Liu et al.

As can be seen from Table 9.7, with TTA our proposed method outperforms the previous best scoring algorithm (ARC-Net [4]) by 0.51%, 0.34%, 0.15% and 0.29% in IoU, precision, recall and F1-score respectively. Figure 9.4 illustrates some qualitative results of our algorithm on the WHU dataset. The last column in the figure shows the high degree of completeness (i.e. high number of true positives and true negatives, very few false positives and false negatives) in our segmentation results. Hence, it shows that our algorithm consistently outperforms the existing state-of-the-art results in all the metrics.

9.4 Quantitative Evaluation on the DeepGlobe Building Dataset

Table 9.8 illustrates the quantitative performance of our proposed algorithm on the DeepGlobe Building Dataset. Our algorithm without TTA achieves F1-scores of 0.895, 0.780, 0.679 and 0.607 over Vegas, Paris, Shanghai and Khartoum respectively; on applying TTA, the F1-scores improves to 0.896, 0.785, 0.687 and 0.613 over Vegas, Paris, Shanghai and Khartoum respectively. Note that the variance of the performance across different cities. It stems from the quality of datasets for different cities. We discuss this issue in the next section.

We outperform the previous best (published) F1-scores obtained by TernausNetV2 [1] by 0.56%, 0.51%, 1.03% and 1.65% over Vegas, Paris, Shanghai and Khartoum respectively. Overall, our algorithm outperforms the popular TernausNetV2 network by 0.81%.

The power of our approach is best illustrated by its ranking at number 2 on the overall scenario in the "DeepGlobe Building Extraction Challenge" at the following website:

https://competitions.codalab.org/competitions/18544#results

Our entry is under the username 'chattops' with the upload date November 30, 2021. The metrics used in all such competitions only measure the extent of the bulk extraction of the pixels corresponding to the building footprints. These metrics do not directly address the main focus of this dissertation, which is on improving the boundaries of the extracted

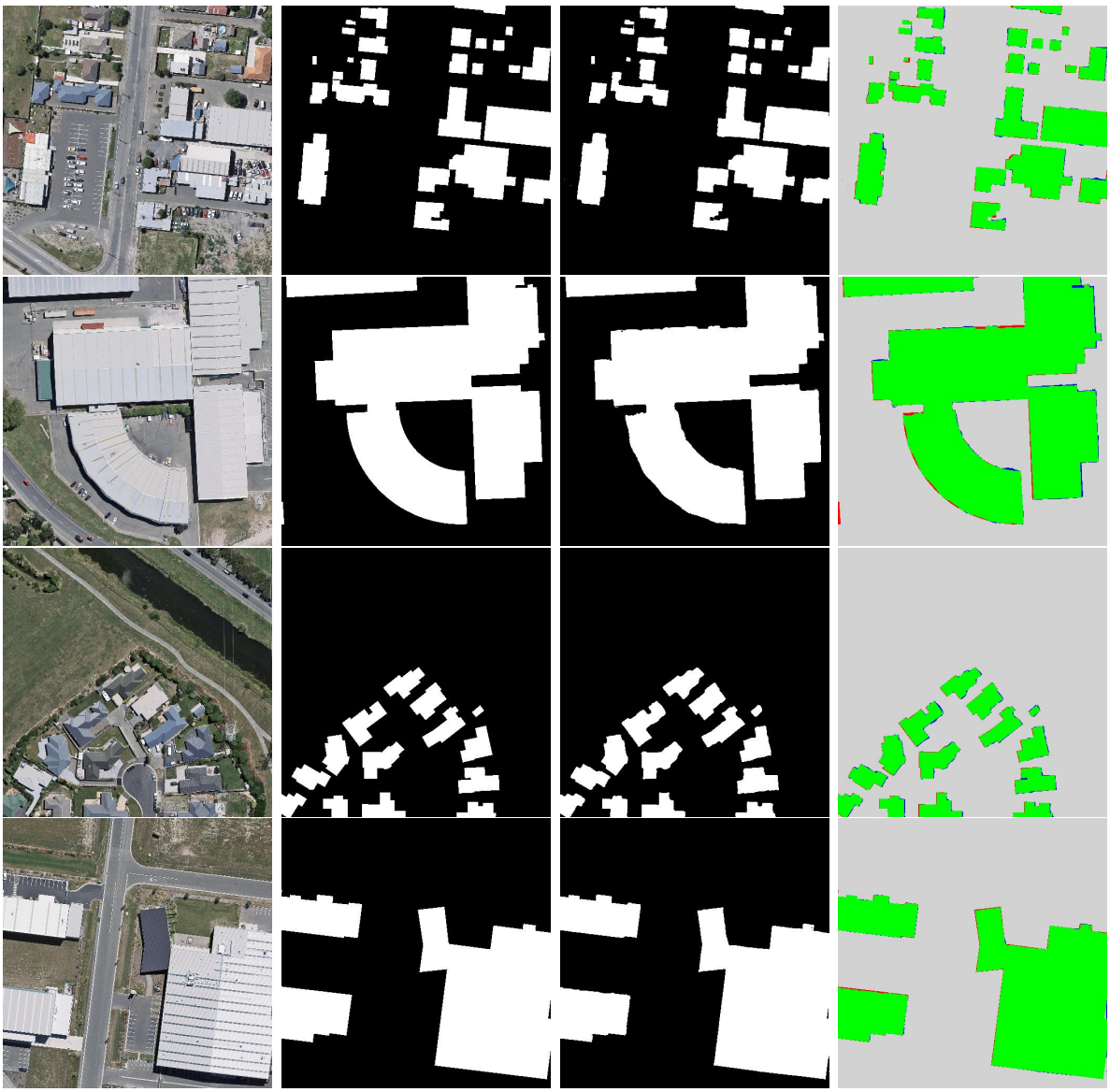


Figure 9.4. Illustration of our qualitative results on the WHU Building Dataset. Column 1: Input image. Column 2: Ground-truth Label Map. Column 3: Predicted Label Map. Column 4: Green: True Positives; Blue: False Positives; Red: False Negatives; Grey: True Negatives.

Table 9.8. F1-scores for the state-of-the-art networks on the test subset of DeepGlobe Building Dataset. The best results are highlighted in bold. **Leading the DeepGlobe 2018 public leaderboard. Citation is unknown. TTA: Test Time Augmentation.

Method	Vegas	Paris	Shanghai	Khartoum	Overall
Li et al. [88]	0.886	0.749	0.618	0.554	0.701
Golovanov et al. [87]	-	-	-	-	0.707
Zhao et al. [91]	0.879	0.753	0.642	0.568	0.713
Hamaguchi et al [2]	-	-	-	-	0.726
TernausNetV2 [1]	0.891	0.781	0.680	0.603	0.739
Ali_DI_Deep_Learning**	-	-	-	-	**0.749**
Our Method (no TTA)	0.895	0.780	0.679	0.607	0.740
Our Method + TTA	**0.896**	**0.785**	**0.687**	**0.613**	0.745

shapes and the contiguity of the pixel blobs that are recognized as the building pixels. Nonetheless, it is noteworthy that improving the boundary and the pixel contiguity properties also improves the traditional metrics for building segmentation. Here, we would like to point out that we do not find any report or published based on the the best performing algorithm (Ali_DI_Deep_Learning). Hence, we could not compare its performance on individual cities. Our proposed architecture outperforms all the published algorithms both in the overall results and for individual city dataset as well.

We emphasize the fact that most of the state-of-the-art methods reported in Table 9.8 use multi-spectral information; whereas our algorithm uses only RGB images for building footprint extraction. We believe incorporating additional spectral information would further improve our algorithm's segmentation performance.

In addition to the state-of-the-art methods reported in Table 9.8, several other papers [93], [96], [178] have shown experimental results on the DeepGlobe Building Dataset. However, they have either chosen their own set of *test* images or have reported pixel-wise performance scores. In this research, we report only those works which have reported object-wise performance scores on the test dataset provided by the original DeepGlobe 2018 Competi-

tion organizers during the development phase. In summary, our proposed architecture has better generalization capability and outperforms existing published results without using customization on the datasets.

Figures 9.5 and 9.6 depict the performance of our proposed architecture. From Figure 9.5, it is evident that the datasets comprises of diverse set of buildings. Specially, the dataset corresponding to Khartoum, it is very difficult to identify buildings even in the naked eyes. The buildings are located in a dense scenario in Paris. Still, our proposed architecture performs reasonably well across the datasets.

Figure 9.6 depicts the figures corresponding to the validation dataset for DeepGlope dataset. Our proposed architecture performs perfectly on the Vegas dataset. The number of false negatives and false positives are little bit higher for the Paris dataset since the dataset contains a cluster of small buildings separated by forest or road. Due to dense nature of Shanghai, the false negatives increases for the Shanghai dataset. For the Khartoum dataset, the buildings are hardly detected manually. Further, the ground-truths are also not perfect. Hence, our proposed architecture returns some false negative and false positives. However, as Table 9.8 suggests, our proposed architecture's overall performance is close to the best one.

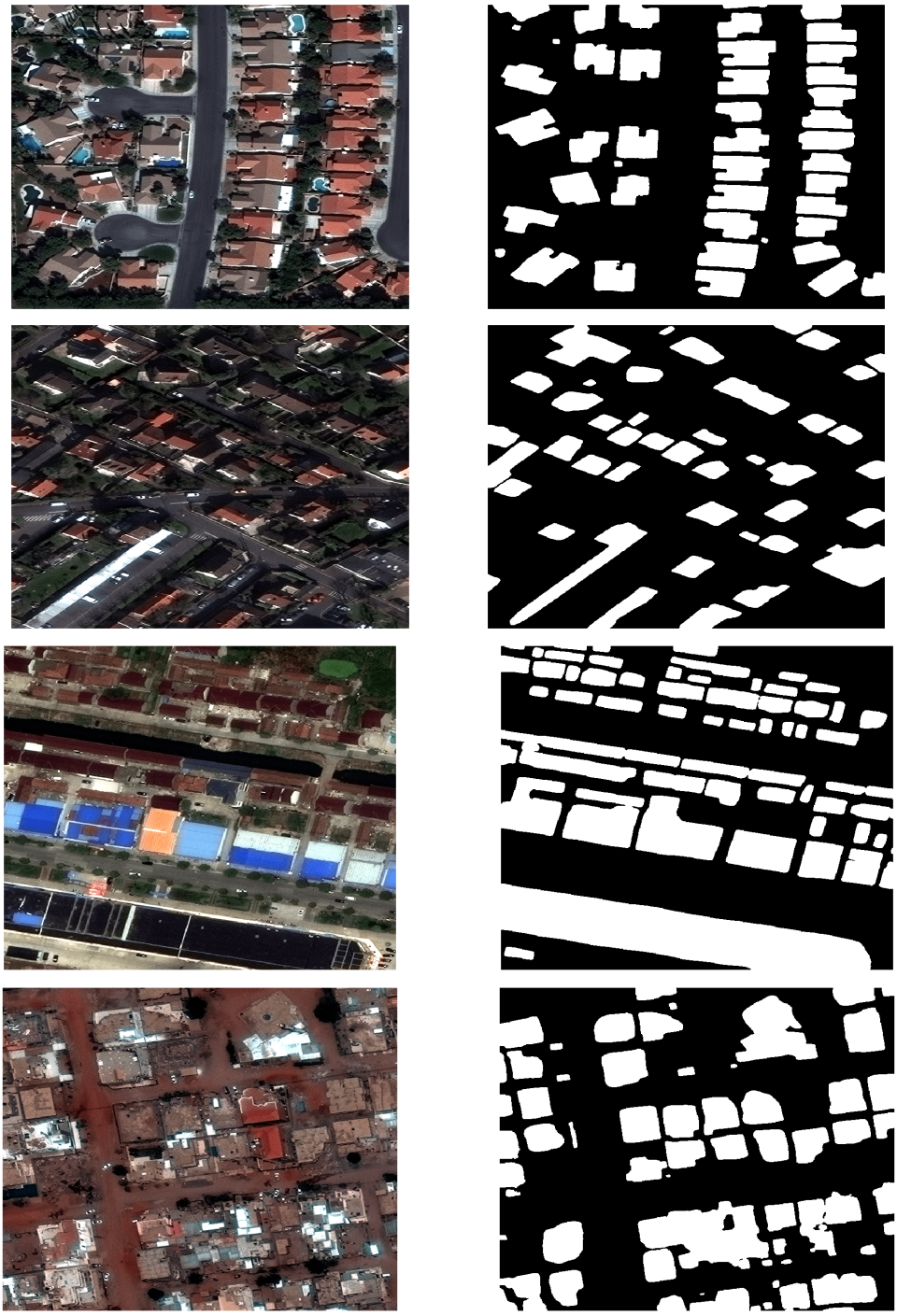


Figure 9.5. Illustration of our qualitative results on the test subset of DeepGlobe Building Dataset. Rows 1, 2, 3 and 4 show results on image patches over Vegas, Paris, Shanghai and Khartoum respectively. Column 1: Input Image. Column 2: Predicted Label Map.

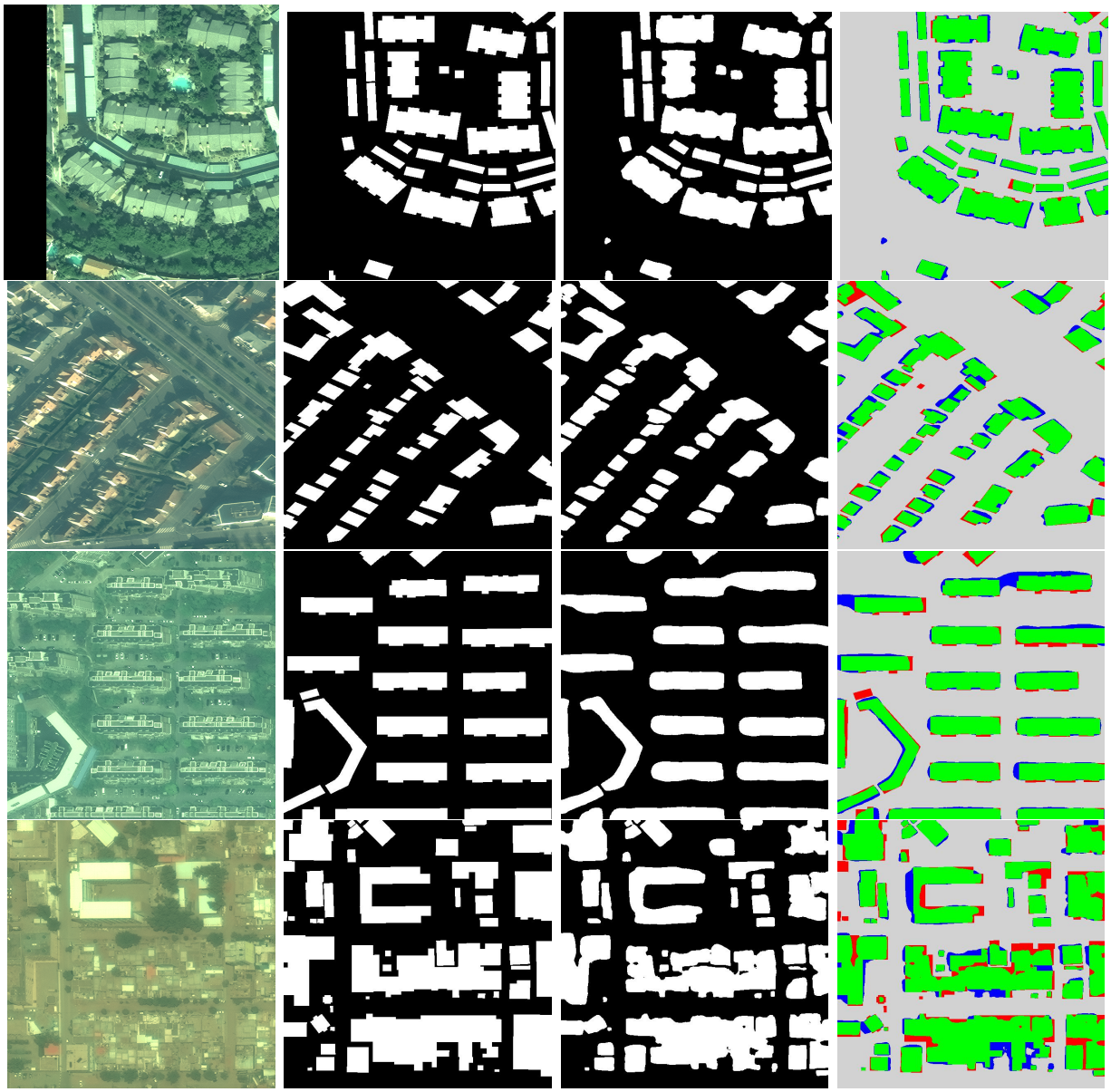


Figure 9.6. Illustration of our qualitative results on the validation subset of DeepGlobe Building Dataset. Rows 1, 2, 3 and 4 show results on image patches over Vegas, Paris, Shanghai and Khartoum respectively. Column 1: Input image. Column 2: Ground-truth Label Map. Column 3: Predicted Label Map. Column 4: Green: True Positives; Blue: False Positives; Red: False Negatives; Grey: True Negatives.

10. DISCUSSION ON THE RESULTS AND AN ABLATION STUDY

The goal of this chapter is to present a comprehensive overview of the performance of our approach over all four datasets that takes into account the characteristics of each. In Section 10.1, a detailed discussion on the performance of our approached is provided. Subsequently, in Section 10.2, we present an ablation study to verify the effectiveness of the modules for the uncertainty attention and refinement, and also of the deep supervision that is used in our network.

10.1 Discussion

10.1.1 Effectiveness of our framework

The results reported in Tables 9.1–10.1 clearly demonstrate the effectiveness of our proposed algorithm in building segmentation from remotely sensed images.

Owing to the Edge Attention Unit and the Hausdorff Loss used in our framework for training, we get accurate building boundaries, as can be seen in Figure 10.2. These two losses guide our framework to identify the boundaries of the buildings in a more rigorous manner.

The Uncertainty Attention Module helps us to achieve high number of true positives and avoid false alarms (See column 4 of Figure 9.2) by giving more attention to the ambiguous regions of an aerial scene. The uncertainty attention module reduces the uncertainty in the decision process, hence, our framework provides more confidence in detecting the building boundaries. Further, the Reverse Attention Unit assists us to identify the missing detection by refining the intermediate label maps in a top-down fashion.

We also observe significant improvement in the predictive performance of our algorithm when TTA is applied. Tables 9.1, 9.2, 9.5, 9.7 and 9.8 report scores for both TTA and non-TTA versions of our algorithm. Tables 9.3, 9.4, 9.6, and 10.1 only report our TTA applied results. The difference is apparent.

10.1.2 Limitation of our framework

We now discuss where our algorithm does not perform the best and the potential reasons behind it. We also draw potential techniques which can be applied to mitigate those limitations.

With regard to the INRIA dataset, it is evident from Table 9.5 that the performance of our algorithm for the Chicago area is not the best. The buildings in Chicago are located very close to one another, and the network finds it difficult to clearly separate the building boundaries of adjacent buildings. We see the same situation in the San Francisco region – buildings in San Francisco area are also densely packed. Obviously, our framework needs further improvements in separating the buildings that are in close proximity to one another. We believe this issue arises as we use a dilation operator in our edge refinement module. We presume that using an accurate contour extraction algorithm should help us in alleviating this problem and it has been left for the future.

In general, ground-truth label inconsistencies in the datasets hinder our training process to some extent, and also impact the overall evaluation scores. Misaligned building masks where building masks are not perfectly aligned with the actual buildings are pretty common in the dataset. In addition, the Massachusetts Buildings Dataset also contains false labels. Some examples of noisy labels in the Massachusetts Dataset can be seen in row 2 of Figure 10.1. Moreover, in some of the images, the buildings encompassing playgrounds or parking lots are labeled as a single building instance without capturing the actual shape of the building (row 1 of Figure 10.1). Nevertheless, our network identifies the building pixels accurately, as illustrated in column 3 of rows 1 and 2 of Figure 10.1. Though our framework accurately identifies the building pixels, the performance metric suffers because of the wrong ground-truth labels.

Similar noisy labels appear in the INRIA Aerial Image Labeling Dataset. Row 3 of Figure 10.1 shows an image patch over Vienna where in the ground-truth, smaller building structures close to one-another are clubbed as a one large building. Still, our network accurately predicts each smaller structure. Kitsap County not only has a very sparse distribution of buildings, but mis-labels are also prevalent in the dataset. This severely impacts

the evaluation scores. Out of 5 images in the validation dataset, 2 of the images have false ground-truth building labels. One such example is shown in row 4 of Figure 10.1. We achieve an IoU of 86.42% as opposed to 73.25% when we leave out those 2 images from the validation set which is significantly better compared to the reported number in Table 9.5. This kind of mis-labels are found through the training subset as well. However, our network is robust to such mis-labels as evident from the qualitative as well as quantitative results.

Our network yields across-the-board superior performance on the WHU Building Dataset. We believe that the main reason for that is the fact that the ground-truth building maps provided in the WHU dataset are more accurate. We should also mention the relatively low complexity of this dataset in relation to the other three datasets that cover more difficult terrains with high buildings, diverse topography, more occlusions and shadows. Hence, it shows that if the ground-truth labels of the INRIA and the Massachusetts datasets were more accurate, the performance of our framework would outperform the existing state-of-the-art methods by a wider margin.

For the DeepGlobe Dataset, our algorithm outperforms the existing published state-of-the-art results for each of the dataset [1]. With TTA, the performance again improves for the DeepGlobe dataset as well. The F1-score is the highest for Vegas followed by the dataset for Paris. The images in the Vegas and Paris subsets are mostly collected from residential regions. Unlike the other two cities in the DeepGlobe dataset, the buildings in Vegas and Paris have more unified architectural style. For Shanghai, our proposed method faced difficulty in correctly extracting buildings with green roofs or buildings that are of extremely small size. In Khartoum, there are many building groups, and it is hard to judge, even by the human eye, whether a group of neighboring buildings should be extracted entirely or separately in many regions. Still, our proposed framework outperforms the existing state-of-the-art methods by a significant margin. Thus, our proposed framework performs well across various datasets and outperforms the plethora of existing algorithms by a significant margin.

[1]↑Though the overall performance is the second-best, we do not find any report or published paper on the best performing algorithm.

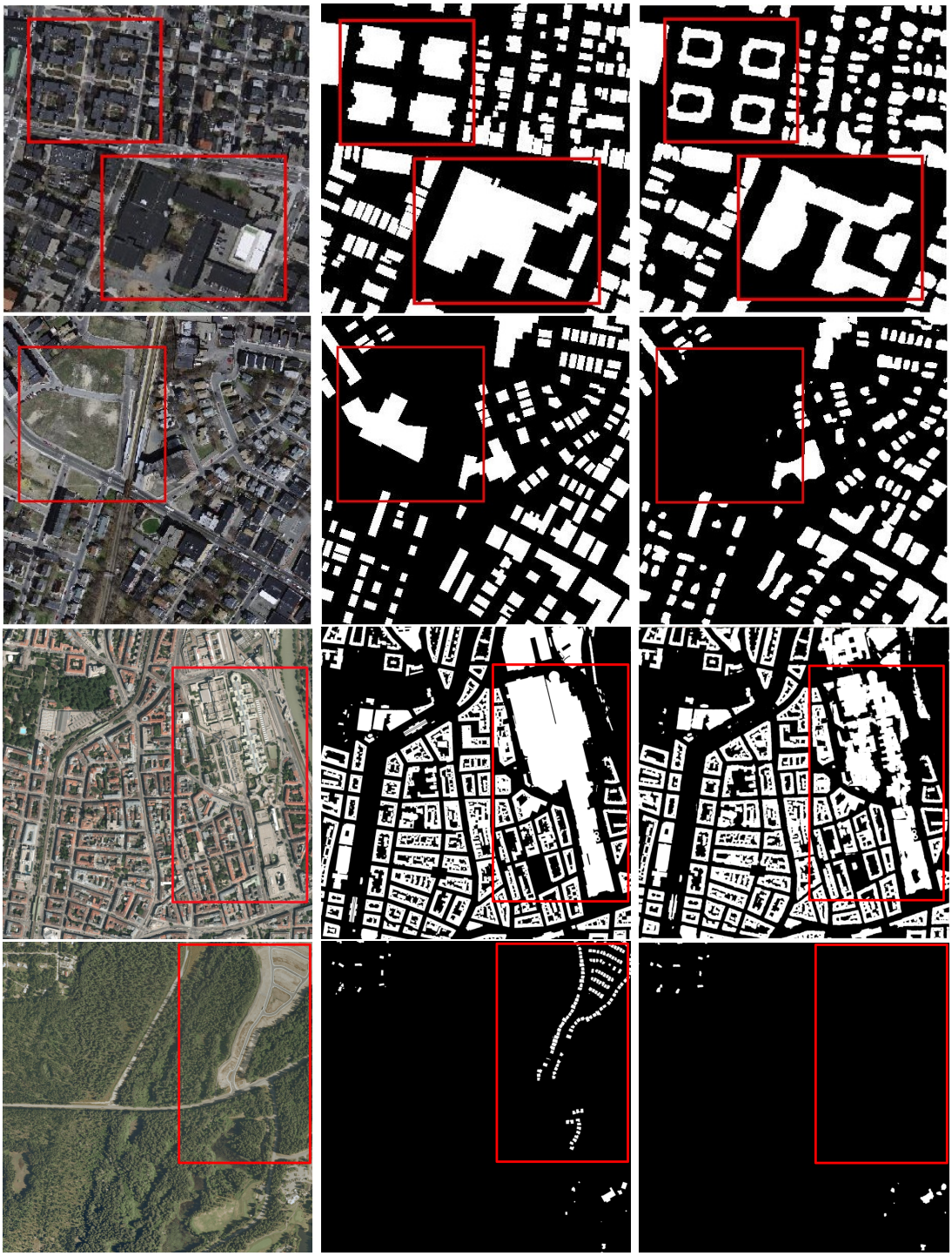


Figure 10.1. Illustration of noisy labels in the Massachusetts Buildings Dataset (rows 1, 2) and the INRIA Aerial Image Labeling Dataset (rows 3, 4). Column 1: Input Image. Column 2: Ground-truth Labels. Column 3: Predicted Labels. The red boxes represent the areas where noisy labels are present in the ground-truth label maps.

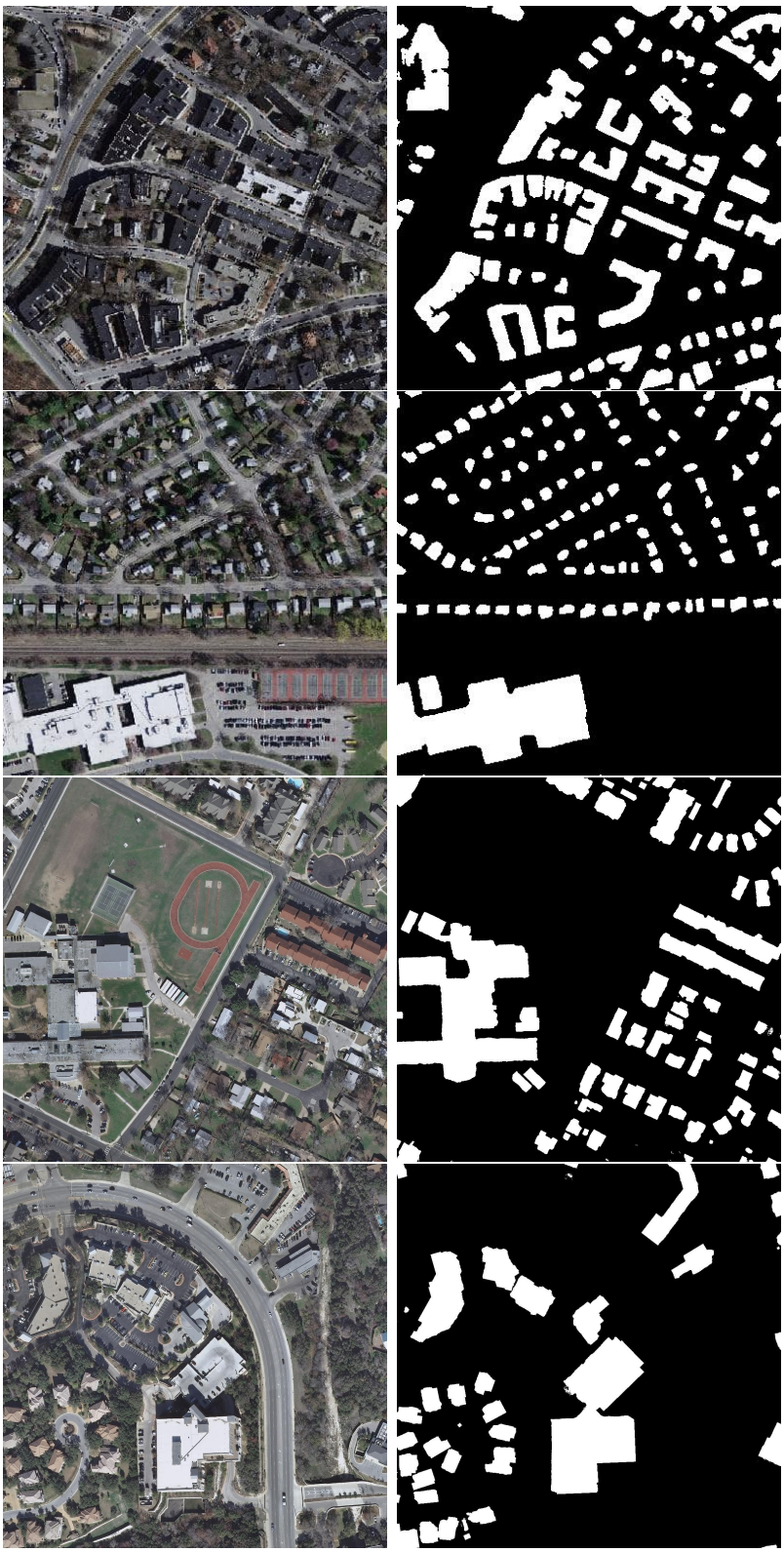


Figure 10.2. Illustration of *crisp* building boundaries obtained using our proposed approach. Column 1: Input Image. Column 2: Predicted Labels.

10.2 Ablation Study

To verify the effectiveness of the Uncertainty Attention Module, the Refinement Module, and of the deep supervision technique we have used, we conducted ablation studies using the INRIA Aerial Validation Dataset. We trained 6 different architectures – (a) the vanilla Generator (VG — no attention, deep supervision or critic) (b) the base GAN architecture (BGA — VG + critic); (c) the base GAN architecture with deep supervision (DS); (d) the base GAN architecture with deep supervision and the Uncertainty Attention Module; (e) the base GAN architecture with deep supervision and the Refinement Module; and, (f) the base GAN architecture with Deep Supervision, the Uncertainty Attention Module and the Refinement Module. All the architectures were trained independently with identical training hyper-parameters. Test Time Augmentation is applied while evaluating the performance of the trained models on the validation images. As mentioned in Section 8.3, for the INRIA dataset, all the experiments are conducted using our k-fold validation strategy.

The mean IoU scores for these 6 models are reported in Table 10.1. On adding the critic, the overall IoU of the Vanilla Generator improves by 0.82%. With deep supervision, we achieve an overall improvement of 2.58% relative to the BGA. The Uncertainty Attention Module and the Refinement Module further improve the mean IoU scores by 1.89% and 1.22% respectively. Finally when we combine all these components, our model outperforms the baseline GAN model by 7.04%. Thus, it shows that each proposed component adds to the performance which enhances the final performance by a significant margin.

Figure 10.3 demonstrates the qualitative performance improvements obtained with the Uncertainty Attention Module and the Refinement Module. In the first row and second column of Figure 10.3, the large building is labeled incorrectly due to the presence of shadow and absence of global context in the base architecture. However, adding the Uncertainty Attention Module improves the segmentation result, as shown in row 1 and column 3 of Figure 10.3. Similar results can be seen in row 2, where the base network can not distinguish between roads and buildings since they are similar in color. On the contrary, the model with the Uncertainty Attention Module accurately identifies the building pixels. Thus, the

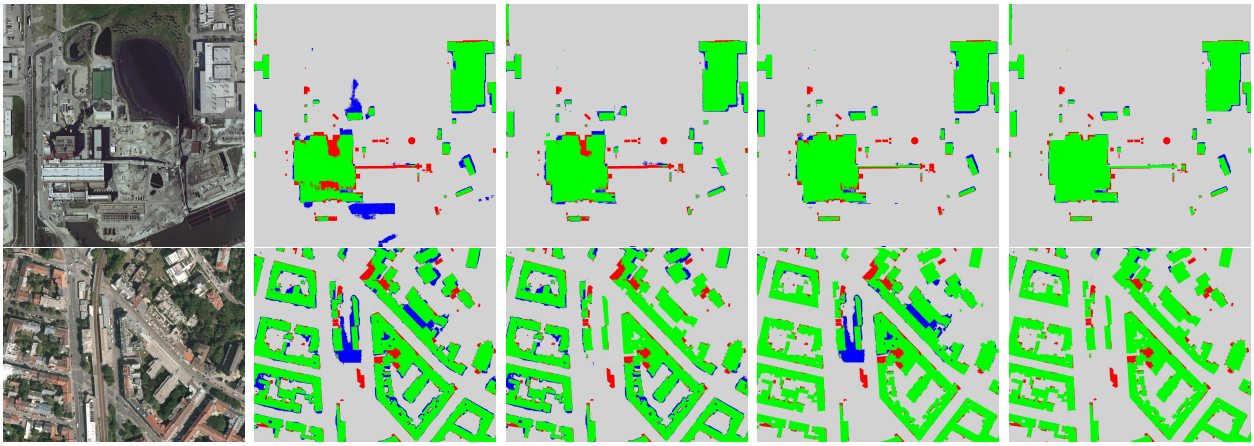


Figure 10.3. Ablation study results on Chicago (row 1) and Vienna (row 2) areas of the INRIA Aerial Image Labeling Dataset. Column 1: Input Image. Column 2: Base GAN Architecture (BGA). Column 3: BGA + Uncertainty Attention Module (UAM). Column 4: BGA + Refinement Module (RM). Column 5: BGA + UAM + RM. All the results are from models trained with deep supervision. Test time augmentation is used for all models. Green: True Positives; Blue: False Positives; Red: False Negatives; Grey: True Negatives.

uncertainty module guides the framework to correctly identify the building boundaries when the uncertainty is large.

Column 4 of Figure 10.3 demonstrates results when we add the Refinement Module to the base GAN architecture. We can observe that the Refinement Module has identified precise building boundaries compared to the base model. When we incorporate both the Uncertainty Attention and the Refinement Modules, we can observe the overall improvement compared to the base module in column 5 of Figure 10.3. It demonstrates how each component adds to the performance which results in significant improvement over the base GAN architecture.

Table 10.1. Mean IoU scores for the ablation studies performed on the INRIA Validation Dataset. C: Critic, DS: Deep Supervision, UAM: Uncertainty Attention Module, RM: Refinement Module.

Method	Austin	Chicago	Kitsap	W. Tyrol	Vienna	Overall
Vanilla Generator (VG)	77.52	69.08	65.69	76.89	79.45	75.31
Base GAN Architecture (BGA) (VG + C)	78.97	70.21	68.07	77.86	79.98	75.93
BGA + DS	80.31	71.77	68.86	79.67	80.18	77.89
BGA + UAM + DS	81.56	73.86	70.64	81.49	81.87	79.36
BGA + RM + DS	80.95	73.12	72.01	82.73	81.13	78.84
BGA + UAM + RM + DS	83.78	76.39	73.25	85.72	83.19	81.28

11. CONCLUSION AND FUTURE WORK

In this chapter, we summarize the contributions of this dissertation and discuss the possible future directions of new research.

11.1 Research Summary

Owing to the remarkable advances in high-resolution Earth Observation, researches have shown immense interest in using this high resolution remotely-sensed information in smart city domain. Automatic building footprint detection from remotely sensed aerial imagery has become one of the most critical and active areas of research. Such research has wide range of applications in urban planning, disaster assessment, green sites development, map revision, population estimation and many more. However, huge non-uniformity in building appearances across the globe, and occurrence of shadows and occlusion due to surrounding tall structures and high vegetation in overhead imagery make distinction of building pixels from complex background a challenging task. Moreover, challenges also arise from the fact that in many cases, various objects (such as roads, parking lots and building roofs) that are present in aerial and satellite images look very similar and have very small inter-class differences, This happens because the reflectivity signatures of several types of building materials are close to what gets used for the construction of roads and parking lots.

Though a lot of work is being conducted to mitigate the effects of the above issues and detect buildings efficiently in remotely sensed images, challenges still persist. Our research in this dissertation is also focused towards resolving the issues present in the current state-of-the-art automatic building segmentation algorithms.

To mitigate the above mentioned issues, we propose a novel **attention-enhanced residual refining generative adversarial network (GAN)** for detecting building footprints automatically in high-resolution aerial and satellite images. We train our network using an adversarial strategy to enforce long-range spatial label contiguity, without any added complexity to the trained model during inference. We have embedded novel attention units in the generator network of our GAN framework to focus and improve predictions in the specific *concern areas* of an image. By 'concern areas', we refer to the areas near building boundaries,

areas with shadows and occlusion, and the regions where our network might get confused while predicting building labels due to to almost identical background and foreground pixels.

The adversarial component of our network, i.e. the critic, is designed to correctly identify between the generator predicted labels and ground-truth labels, and is trained by optimizing a multi-scale L_1 loss [17]. The critic does not directly discriminate by classifying its inputs as real or fake; instead it tries to maximise the distance between the feature maps extracted from the true and generated instances at multiple scales. The multi-scale L_1 loss (the adversarial loss of our GAN framework) is calculated using the hierarchical features extracted from the multiple layers of our critic — specifically, this loss computes the absolute difference between the features maps of generated and real masked building label maps that are extracted from multiple layers (at multiple scales) of the critic. This loss enables our network to capture both long as well short range spatial relations between the pixels of an image.

Our generator architecture is based on the framework of a fully convolutional encoder-decoder network — it takes 3-channel RGB images as inputs and predicts 1-channel binary maps with pixel-wise labels of building and non-building classes for the corresponding input image. As mentioned throughout the dissertation, our main contribution involves the incorporation of novel attention units in the generator of our GAN framework to overcome the shortcomings of the state-of-the-art building segmentation algorithms. Specifically, we introduce the novel concepts of **Uncertainty Attention Module** and the **Refinement Attention Module** which we summarize in the following —

1. The **Uncertainty Attention Unit** is proposed to resolve uncertainties in classification. The unit is introduced at each data abstraction level between the concatenation of the encoder feature map with the decoder feature map. It focuses on those particular feature regions where the network has not shown confidence during its previous prediction –- mostly in the vicinity of the building boundaries, in the shadow and occluded areas, and in those regions of an image where the building pixel signatures are too close to the background pixel signatures.

   The uncertainty attention unit also provides protection against any over-segmentation that may be present in the final output of the network due to indiscriminately fusing

the low-level features from the encoder with the high-level features in the decoder. It mitigates against such over-segmentations by controlling the level of inclusion of the encoder features before they are concatenated with the decoder features.

2. The **Refinement Module** consisting of a **Reverse Attention Unit** and an **Edge Attention unit** is used to learn residual predictions after every stage of decoding and gradually refines the prediction map estimated in the previous stage until the final prediction map is obtained. Specifically, the network first predicts a coarse prediction map from its bottleneck layer. This topmost prediction map that is rich in semantic information but lacking in fine details is then gradually refined by adding residual predictions obtained from the hierarchical residual attention modules.

   (a) The **Edge Attention Unit** is used to improve the building boundaries. The unit enhances to the boundary pixels, thus, helping the network to learn precise crisp boundaries of the buildings.

   This unit guides the network to use the fine detail in the current layer of the decoder and reevaluate the boundaries of the building predictions coming out of the lower layer. This unit first amplifies the features near the boundary of the previously estimated prediction map, and then uses the current layer's information to rectify the corrupted boundary pixels in the previous layer.

   (b) The **Reverse Attention Unit** is used to seek missed detections in the intermediate building prediction maps. This unit allows the network to explore features which have been predicted as non-building class, thus, enabling the network to discover the missing building parts in the previously estimated result.

   Similar to the edge attention unit, the idea of the reverse attention unit is to reconsider the predictions coming out of a lower-indexed layer in the decoder in light of the spatial details available at the current layer; but unlike the edge attention unit, this unit emphasizes on the pixels classified as non-building in the the predictions coming out of a lower-indexed layer in the decoder.

The purpose of the Refinement Module is to fuse the fine detail from the lower-indexed layers with the spatial features in the higher-indexed layers with the expectation that such a fusion would lead to a segmentation mask that is rich in fine details and that, at the same time, exhibits high accuracy with regard to object localization. Through residual learning, both the attention units in the refinement module seek to improve the quality of the predictions made in the previous decoder level on the basis of the finer image detail captured during the current decoder level. Again, the important thing to note here is that both these attention units focus on those regions of an image where the accuracy of semantic segmentation is likely to be poor — e.g. in the vicinity of building boundaries, in shadow and occluded areas, etc.

In our network, we also introduce an Atrous Spatial Pyramid Pooling (ASPP) layer just after the bottleneck of our encoder-decoder segmentation framework. The same remotely sensed image might contain very large as well as extremely small buildings. The ASPP layer can help in accurate segmentation of these unevenly distributed targets by capturing global contextual information from the image.

As mentioned above, we train our network using an adversarial strategy and a multi-scale L_1 loss as adversarial loss to enforce long as well as short range spatial label contiguity. In addition to the adversarial loss, we also use deep supervision in our architecture to guard against corruptions of the predicted building maps due to semantic gap between the encoder and decoder. It is our understanding that our proposed uncertainty attention module can amplify the semantic gap between the encoder and decoder by injecting "noisy" encoder features in those regions of a building prediction map where the probabilities are low. Deep supervision guards against such corruption of the prediction maps by forcing the intermediate feature maps to be discriminative at all levels of the decoder. Furthermore, to stabilize the training of our GAN and boost the performance of our generator, we compute weighted dice loss and shape loss for the final prediction map as well as for each intermediate prediction map.

In this dissertation, we have shown results on four publicly available building footprint detection datasets — Massachusetts Buildings (MB) Dataset [5], INRIA Aerial Image La-

beling Dataset [19], WHU Building Dataset [20] and DeepGlobe Building Detection Dataset [18], [21]. Our experiments show significant performance improvement in semantic segmentation of building footprints over the other state-of-the-art approaches. For the challenging DeepGlobe dataset which consists of Digital Globe's WorldView-3 satellite images with 30 cm resolution we hold the 2^{nd} rank on the DeepGlobe's public leaderboard with an overall F1-score of 0.745. For the validation subset of the INRIA Aerial Image Labeling Dataset, our network achieves an overall IoU of 81.28% and an overall accuracy of 97.03%. And for the official INRIA Test subset, our network scores 77.86% and 96.41% in overall IoU and accuracy. Superior performance on the INRIA test subset also verifies the generalization capability of our proposed algorithm. We have also improved upon the previous best results on two other datasets: For the WHU Building Dataset, our network achieves 92.27% IoU, 96.73% precision, 95.24% recall and 95.98% F1-score. And, finally, for the Massachusetts Buildings Dataset, our network achieves 96.19% relaxed IoU score and 98.03% relaxed F1-score over the previous best scores of 91.55% and 96.78% respectively, and in terms of non-relaxed F1 and IoU scores, our network outperforms the previous best scores by 2.77% and 3.89% respectively.

The results reported in this dissertation demonstrate the effectiveness of our building detection approach even when the buildings are present amidst complex background or are only partly visible due to the presence of shadows. The experimental evaluations also show that the proposed method performs equally well on aerial as well as satellite images which shows the robustness of our proposed algorithm on diverse modalities of remotely sensed imagery.

11.2 Future Scope

- In the future, we plan to investigate how to utilize multi-spectral information for further improvement of our network's capability. Specifically, satellite imagery contains additional channels corresponding to different wavelengths. Approaches that do not use all channels are unable to fully exploit these images for optimal performance. For the DeepGlobe building detection dataset, we have noticed that the previous best

scoring algorithms used multi-spectral information; whereas our algorithm uses RGB images for building footprint extraction. We believe incorporating additional spectral information would further improve our algorithm's segmentation performance.

- In the future, we plan to apply an accurate contour extraction algorithm in our proposed edge attention unit (refer to Section 5.2.1.2 of Chapter 5).

  In Chapter 10, we mentioned that our algorithm finds it difficult to clearly separate the building boundaries of adjacent buildings that are in close proximity to one another. This shortcoming has affected the performance of our algorithm in cities like San Francisco and Chicago where the buildings are densely packed. We believe this issue arises as we use a *dilation operator* in the Edge Attention Unit of our Refinement Module, as explained in the next paragraph.

  The edge attention unit is introduced to improve the boundaries of buildings. Essential to the logic of what improves the boundary edges is the notion of contour extraction. At each layer on the decoder side, we want to extract the contours in the fine detail provided by the encoder side in order to improve the edges in the building prediction map yielded by the lower layer. Note that there is a significant difference between just detecting the edge pixels and identifying the contours. Whereas the former could yield just a disconnected set of pixels on the object edges, the latter is more likely to yield a set of connected boundary points — even when using just contour fragment. On account of the need to make these calculations GPU compatible, at the moment the notion of contour extraction is carried out by applying the Sobel edge detector to a building prediction map followed by a p-pixel dilation of the edge pixels identified in order to connect what would otherwise be disconnected pixels. It stands to reasoning that this dilation operation hinders the performance of the edge attention unit — especially in scenarios where the buildings are in close proximity to one another. Using an accurate contour extraction algorithm should help us in alleviating this problem.

- Our proposed proposed uncertainty attention unit and refinement module can used as add-ons to any segmentation network. It would be interesting to see how our

proposed attention modules would affect the performance of state-of-the art semantic segmentation networks. We would also like to check how our proposed architecture performs in the context of multi-class segmentation.

- This dissertation focuses on extracting buildings from *nadir* images. A future direction of this research can be to investigate how our proposed segmentation architecture performs on *off-nadir* images. Detecting buildings automatically from off-nadir images can be beneficial for many applications — e.g., in disaster management scenarios, most of times, the first post-event imagery is usually captured from a more off-nadir image than is used in standard mapping use cases. The ability to detect buildings from off-nadir imagery will allow for more flexibility in such scenarios.

- Extensive investigations on more diverse datasets (like, roads) have been left for the future. As we do not utilise any domain knowledge or prior constraint (such as, location, shape, etc.) for building class, we conjecture that the network would work well on other objects like as well.

Our proposed algorithm will assist in developing frameworks to solve the above envisioned research problems. Building segmentation is an important topic and our work is definitely not the last work on this topic. We believe that the framework put forth in this work will serve as a stepping stone for the future directions of research in this area.

REFERENCES

[1] V. Iglovikov, S. Seferbekov, A. Buslaev, and A. Shvets, "TernausNetV2: Fully convolutional network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 233–237.

[2] R. Hamaguchi and S. Hikosaka, "Building Detection From Satellite Imagery Using Ensemble of Size-Specific Detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2018.

[3] B. Chatterjee and C. Poullis, "Semantic segmentation from remote sensor data and the exploitation of latent learning for classification of auxiliary tasks," *Computer Vision and Image Understanding*, vol. 210, p. 103 251, 2021.

[4] Y. Liu, J. Zhou, W. Qi, X. Li, L. Gross, Q. Shao, Z. Zhao, L. Ni, X. Fan, and Z. Li, "ARC-Net: An efficient network for building extraction from high-resolution aerial images," *IEEE Access*, vol. 8, pp. 154 997–155 010, 2020.

[5] V. Mnih, "Machine Learning for Aerial Image Labeling," Ph.D. dissertation, University of Toronto, 2013.

[6] S. Saito and Y. Aoki, "Building and road detection from large aerial imagery," in *Image Processing: Machine Vision Applications VIII*, International Society for Optics and Photonics, vol. 9405, 2015, 94050K.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[8] A. Khalel and M. El-Saban, "Automatic pixelwise object labeling for aerial imagery using stacked U-Nets," *arXiv preprint arXiv:1803.04953*, 2018.

[9] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 645–657, 2016.

[10] H. M. Wallach, "Conditional random fields: An introduction," *Technical Reports (CIS)*, p. 22, 2004.

[11] Q. Zhu, Z. Li, Y. Zhang, and Q. Guan, "Building extraction from high spatial resolution remote sensing images via multiscale-aware and segmentation-prior conditional random fields," *Remote Sensing*, vol. 12, no. 23, p. 3983, 2020.

[12] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 475–12 485.

[13] C. Sebastian, R. Imbriaco, E. Bondarev, and P. H. de With, "Adversarial Loss for Semantic Segmentation of Aerial Imagery," *arXiv preprint arXiv:2001.04269*, 2020.

[14] X. Pan, F. Yang, L. Gao, Z. Chen, B. Zhang, H. Fan, and J. Ren, "Building extraction from high-resolution aerial imagery using a generative adversarial network with spatial and channel attention mechanisms," *Remote Sensing*, vol. 11, no. 8, p. 917, 2019.

[15] S. Wang, X. Hou, and X. Zhao, "Automatic Building Extraction From High-Resolution Aerial Imagery via Fully Convolutional Encoder-Decoder Network With Non-Local Block," *IEEE Access*, vol. 8, pp. 7313–7322, 2020.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[17] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang, "Segan: Adversarial network with multi-scale l1 loss for medical image segmentation," *Neuroinformatics*, vol. 16, no. 3-4, pp. 383–392, 2018.

[18] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar, "DeepGlobe 2018: A Challenge to Parse the Earth Through Satellite Images," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2018.

[19] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can Semantic Labeling Methods Generalize to Any City? The INRIA Aerial Image Labeling Benchmark," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE, 2017.

[20] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 1, pp. 574–586, 2018.

[21] A. Van Etten, D. Lindenbaum, and T. M. Bacastow, "SpaceNet: A remote sensing dataset and challenge series," *arXiv preprint arXiv:1807.01232*, 2018.

[22] S. Chattopadhyay and A. C. Kak, "Uncertainty, edge, and reverse-attention guided generative adversarial network for automatic building detection in remotely sensed images," *arXiv preprint arXiv:2112.05335*, 2021.

[23] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[24] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 26, no. 11, pp. 1452–1458, 2004.

[25] S. Angelina, L. P. Suresh, and S. K. Veni, "Image segmentation based on genetic algorithm for region growth and region merging," in *2012 international conference on computing, electronics and electrical technologies (ICCEET)*, IEEE, 2012, pp. 970–974.

[26] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image segmentation using k-means clustering algorithm and subtractive clustering algorithm," *Procedia Computer Science*, vol. 54, pp. 764–771, 2015.

[27] M. Yambal and H. Gupta, "Image segmentation using fuzzy C-means clustering: A survey," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 7, pp. 2927–2929, 2013.

[28] V. K. Dehariya, S. K. Shrivastava, and R. Jain, "Clustering of image data set using k-means and fuzzy k-means algorithms," in *2010 International Conference on Computational Intelligence and Communication Networks*, IEEE, 2010, pp. 386–391.

[29] S. Beucher *et al.*, "The watershed transformation applied to image segmentation," *Scanning microscopy-supplement-*, pp. 299–299, 1992.

[30] L. Najman and M. Schmitt, "Watershed of a continuous function," *Signal Processing*, vol. 38, no. 1, pp. 99–112, 1994.

[31] T. Lindeberg and M.-X. Li, "Segmentation and classification of edges using minimum description length approximation and complementary junction cues," *Computer Vision and Image Understanding*, vol. 67, no. 1, pp. 88–98, 1997.

[32] S. S. Al-Amri, N. Kalyankar, and S. Khamitkar, "Image segmentation by using edge detection," *International journal on computer science and engineering*, vol. 2, no. 3, pp. 804–807, 2010.

[33] R. Muthukrishnan and M. Radha, "Edge detection techniques for image segmentation," *International Journal of Computer Science & Information Technology*, vol. 3, no. 6, p. 259, 2011.

[34] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[35] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[36] M. R. Khokher, A. Ghafoor, and A. M. Siddiqui, "Image segmentation using multilevel graph cuts and graph development using fuzzy rule-based system," *IET image processing*, vol. 7, no. 3, pp. 201–211, 2013.

[37] N. Plath, M. Toussaint, and S. Nakajima, "Multi-class image segmentation using conditional random fields and global classification," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 817–824.

[38] J.-L. Starck, M. Elad, and D. L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE transactions on image processing*, vol. 14, no. 10, pp. 1570–1582, 2005.

[39] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[40] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," *arXiv preprint arXiv:2005.10821*, 2020.

[41] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, Springer, 2020, pp. 173–190.

[42] X. Li, X. Li, L. Zhang, G. Cheng, J. Shi, Z. Lin, S. Tan, and Y. Tong, "Improving semantic segmentation via decoupled body and edge supervision," *arXiv preprint arXiv:2007.10035*, 2020.

[43] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[44] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[45] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[46] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[47] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, "FastFCN: Rethinking dilated convolution in the backbone for semantic segmentation," *arXiv preprint arXiv:1903.11816*, 2019.

[48] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.

[49] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[50] Y.-J. Zhang, "An overview of image and video segmentation in the last 40 years," *Advances in Image and Video Segmentation*, pp. 1–16, 2006.

[51] H. G. Kaganami and Z. Beiji, "Region-based segmentation versus edge detection," in *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, IEEE, 2009, pp. 1217–1221.

[52] 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*, 2015, pp. 3431–3440.

[53] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[54] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1377–1385.

[55] 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.

[56] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[57] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *European conference on computer vision*, Springer, 2016, pp. 519–534.

[58] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[59] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE winter conference on applications of computer vision (WACV)*, IEEE, 2018, pp. 1451–1460.

[60] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville, "Reseg: A recurrent neural network-based model for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 41–48.

[61] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki, "Scene labeling with lstm recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3547–3555.

[62] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with graph lstm," in *European Conference on Computer Vision*, Springer, 2016, pp. 125–143.

[63] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[64] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio, "Renet: A recurrent neural network based alternative to convolutional networks," *arXiv preprint arXiv:1505.00393*, 2015.

[65] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[66] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[67] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9167–9176.

[68] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 603–612.

[69] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3640–3649.

[70] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," *arXiv preprint arXiv:1805.10180*, 2018.

[71] Q. Huang, C. Xia, C. Wu, S. Li, Y. Wang, Y. Song, and C.-C. J. Kuo, "Semantic segmentation with reverse attention," *arXiv preprint arXiv:1707.06426*, 2017.

[72] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," *arXiv preprint arXiv:1611.08408*, 2016.

[73] N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5688–5696.

[74] W.-C. Hung, Y.-H. Tsai, Y.-T. Liou, Y.-Y. Lin, and M.-H. Yang, "Adversarial learning for semi-supervised semantic segmentation," *arXiv preprint arXiv:1802.07934*, 2018.

[75] X. Chen, B. M. Williams, S. R. Vallabhaneni, G. Czanner, R. Williams, and Y. Zheng, "Learning active contour models for medical image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 632–11 640.

[76] C. Rupprecht, E. Huaroc, M. Baust, and N. Navab, "Deep active contours," *arXiv preprint arXiv:1607.05074*, 2016.

[77] J. Lin, W. Jing, H. Song, and G. Chen, "ESFNet: Efficient network for building extraction from high-resolution aerial images," *IEEE Access*, vol. 7, pp. 54 285–54 294, 2019.

[78] A. Marcu, D. Costea, E. Slusanschi, and M. Leordeanu, "A multi-stage multi-task neural network for aerial scene interpretation and geolocalization," *arXiv preprint arXiv:1804.01322*, 2018.

[79] A. E. Marcu and M. Leordeanu, "Object Contra Context: Dual local-global semantic segmentation in aerial images," in *AAAI Workshops*, 2017.

[80] M. Chen, J. Wu, L. Liu, W. Zhao, F. Tian, Q. Shen, B. Zhao, and R. Du, "DR-Net: An improved network for building extraction from high resolution remote sensing image," *Remote Sensing*, vol. 13, no. 2, p. 294, 2021.

[81] Z. Shao, P. Tang, Z. Wang, N. Saleem, S. Yam, and C. Sommai, "BRRNet: A fully convolutional neural network for automatic building extraction from high-resolution remote sensing images," *Remote Sensing*, vol. 12, no. 6, p. 1050, 2020.

[82] C. Liao, H. Hu, H. Li, X. Ge, M. Chen, C. Li, and Q. Zhu, "Joint Learning of Contour and Structure for Boundary-Preserved Building Extraction," *Remote Sensing*, vol. 13, no. 6, p. 1049, 2021.

[83] S. Ran, X. Gao, Y. Yang, S. Li, G. Zhang, and P. Wang, "Building Multi-Feature Fusion Refined Network for Building Extraction from High-Resolution Remote Sensing Images," *Remote Sensing*, vol. 13, no. 14, p. 2794, 2021.

[84] Y. Liao, H. Zhang, G. Yang, and L. Zhang, "Learning Discriminative Global and Local Features for Building Extraction from Aerial Images," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2020, pp. 1821–1824.

[85] 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.

[86] M. Wu, Z. Shu, J. Zhang, and X. Hu, "HRLINKNet: LinkNet with High-Resolution Representation for High-Resolution Satellite Imagery," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, IEEE, 2021, pp. 2504–2507.

[87] S. Golovanov, R. Kurbanov, A. Artamonov, A. Davydow, and S. Nikolenko, "Building detection from satellite imagery using a composite loss function," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 229–232.

[88] W. Li, C. He, J. Fang, and H. Fu, "Semantic segmentation based building extraction method using multi-source GIS map datasets and satellite imagery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 238–241.

[89] J. Hu, L. Li, Y. Lin, F. Wu, and J. Zhao, "Light-Weight Edge Enhanced Network for On-orbit Semantic Segmentation," in *International Conference on Artificial Neural Networks*, Springer, 2019, pp. 321–333.

[90] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, "Multi-task learning for segmentation of building footprints with deep neural networks," in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 1480–1484.

[91] K. Zhao, J. Kang, J. Jung, and G. Sohn, "Building Extraction From Satellite Images Using Mask R-CNN With Building Boundary Regularization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2018.

[92] L. Mou, Y. Hua, and X. X. Zhu, "A relation-augmented fully convolutional network for semantic segmentation in aerial scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 416–12 425.

[93] H. Zhang, Y. Liao, H. Yang, G. Yang, and L. Zhang, "A Local-Global Dual-Stream Network for Building Extraction From Very-High-Resolution Remote Sensing Images," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[94] S. He and W. Jiang, "Boundary-Assisted Learning for Building Extraction from Optical Remote Sensing Imagery," *Remote Sensing*, vol. 13, no. 4, p. 760, 2021.

[95] Y. Zhu, Z. Liang, J. Yan, G. Chen, and X. Wang, "ED-Net: Automatic Building Extraction From High-Resolution Aerial Images With Boundary Information," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4595–4606, 2021.

[96] H. Jung, H.-S. Choi, and M. Kang, "Boundary Enhancement Semantic Segmentation for Building Extraction From Remote Sensed Image," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[97] S. Xie and Z. Tu, "Holistically-Nested Edge Detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[98] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[99] Z. Chen, D. Li, W. Fan, H. Guan, C. Wang, and J. Li, "Self-attention in reconstruction bias U-Net for semantic segmentation of building rooftops in optical remote sensing images," *Remote Sensing*, vol. 13, no. 13, p. 2524, 2021.

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

[101] Z. Zhang, C. Zhang, and W. Li, "Semantic Segmentation of Urban Buildings from VHR Remotely Sensed Imagery Using Attention-Based CNN," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2020, pp. 1833–1836.

[102] D. Zhou, G. Wang, G. He, T. Long, R. Yin, Z. Zhang, S. Chen, and B. Luo, "Robust Building Extraction for High Spatial Resolution Remote Sensing Images with Self-Attention Network," *Sensors*, vol. 20, no. 24, p. 7241, 2020.

[103] H. Guo, X. Su, S. Tang, B. Du, and L. Zhang, "Scale-Robust Deep-Supervision Network for Mapping Building Footprints From High-Resolution Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 10 091–10 100, 2021.

[104] S. Wei, S. Ji, and M. Lu, "Toward automatic building footprint delineation from aerial images using CNN and regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 3, pp. 2178–2189, 2019.

[105] P. Liu, X. Liu, M. Liu, Q. Shi, J. Yang, X. Xu, and Y. Zhang, "Building footprint extraction from high-resolution images via spatial residual inception convolutional neural network," *Remote Sensing*, vol. 11, no. 7, p. 830, 2019.

[106] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[107] B. Comandur and A. C. Kak, "Semantic Labeling of Large-Area Geographic Regions Using Multiview and Multidate Satellite Images and Noisy OSM Training Labels," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4573–4594, 2021.

[108] N. Weir, D. Lindenbaum, A. Bastidas, A. V. Etten, S. McPherson, J. Shermeyer, V. Kumar, and H. Tang, "SpaceNet MVOI: A Multi-view Overhead Imagery Dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.

[109] X. Li, X. Yao, and Y. Fang, "Building-A-Nets: Robust building extraction from high-resolution remote sensing images with adversarial networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 10, pp. 3680–3687, 2018.

[110] A. Abdollahi, B. Pradhan, S. Gite, and A. Alamri, "Building Footprint Extraction from High Resolution Aerial Images Using Generative Adversarial Network (GAN) Architecture," *IEEE Access*, vol. 8, pp. 209 517–209 527, 2020.

[111] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[112] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[113] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[114] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, PMLR, 2015, pp. 2048–2057.

[115] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[116] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[117] H. Larochelle and G. E. Hinton, "Learning to combine foveal glimpses with a third-order boltzmann machine," *Advances in neural information processing systems*, vol. 23, pp. 1243–1251, 2010.

[118] V. Mnih, N. Heess, A. Graves, *et al.*, "Recurrent models of visual attention," in *Advances in neural information processing systems*, 2014, pp. 2204–2212.

[119] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based lstm for aspect-level sentiment classification," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 606–615.

[120] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[121] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2016, pp. 4945–4949.

[122] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *arXiv preprint arXiv:1506.07503*, 2015.

[123] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.

[124] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model.," in *Interspeech*, Makuhari, vol. 2, 2010, pp. 1045–1048.

[125] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[126] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *International Conference on Machine Learning*, PMLR, 2018, pp. 4055–4064.

[127] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8739–8748.

[128] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[129] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[130] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, pp. 2017–2025, 2015.

[131] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," *arXiv preprint arXiv:1810.12348*, 2018.

[132] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[133] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3024–3033.

[134] Z. Qin, P. Zhang, F. Wu, and X. Li, "Fcanet: Frequency channel attention networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 783–792.

[135] H. Lee, H.-E. Kim, and H. Nam, "Srm: A style-based recalibration module for convolutional neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1854–1862.

[136] Z. Yang, L. Zhu, Y. Wu, and Y. Yang, "Gated channel transformation for visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 794–11 803.

[137] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3156–3164.

[138] L. Yang, R.-Y. Zhang, L. Li, and X. Xie, "Simam: A simple, parameter-free attention module for convolutional neural networks," in *International Conference on Machine Learning*, PMLR, 2021, pp. 11 863–11 874.

[139] J. Li, J. Wang, Q. Tian, W. Gao, and S. Zhang, "Global-local temporal representations for video person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3958–3967.

[140] Z. Liu, L. Wang, W. Wu, C. Qian, and T. Lu, "Tam: Temporal adaptive module for video recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 708–13 718.

[141] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[142] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[143] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*, Springer, 2020, pp. 213–229.

[144] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *International Conference on Machine Learning*, PMLR, 2020, pp. 1691–1703.

[145] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *arXiv preprint arXiv:1506.04579*, 2015.

[146] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4353–4361.

[147] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[148] Y. Yuan, X. Chen, X. Chen, and J. Wang, "Segmentation transformer: Object-contextual representations for semantic segmentation," in *European Conference on Computer Vision (ECCV)*, vol. 1, 2021.

[149] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[150] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[151] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[152] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.

[153] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6881–6890.

[154] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," *arXiv preprint arXiv:2102.12122*, 2021.

[155] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," *arXiv preprint arXiv:2101.11986*, 2021.

[156] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 299–12 310.

[157] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, Z. Jiang, Q. Hou, and J. Feng, "Deepvit: Towards deeper vision transformer," *arXiv preprint arXiv:2103.11886*, 2021.

[158] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *arXiv preprint arXiv:2105.15203*, 2021.

[159] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.

[160] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *arXiv preprint arXiv:2103.00112*, 2021.

[161] S. d'Ascoli, H. Touvron, M. Leavitt, A. Morcos, G. Biroli, and L. Sagun, "Convit: Improving vision transformers with soft convolutional inductive biases," *arXiv preprint arXiv:2103.10697*, 2021.

[162] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[163] J.-Y. Sun, S.-W. Kim, S.-W. Lee, Y.-W. Kim, and S.-J. Ko, "Reverse and Boundary Attention Network for Road Segmentation," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[164] J.-Y. Sun, S.-W. Jung, and S.-J. Ko, "Lightweight Prediction and Boundary Attention-Based Semantic Segmentation for Road Scene Understanding," *IEEE Access*, vol. 8, pp. 108 449–108 460, 2020.

[165] S. Chen, X. Tan, B. Wang, and X. Hu, "Reverse attention for salient object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 234–250.

[166] W. Gao, X. Zhang, L. Yang, and H. Liu, "An improved Sobel edge detection," in *2010 3rd International conference on computer science and information technology*, IEEE, vol. 5, 2010, pp. 67–71.

[167] Y. Pang, Y. Li, J. Shen, and L. Shao, "Towards bridging semantic gap to improve semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4230–4239.

[168] L. Wang, C.-Y. Lee, Z. Tu, and S. Lazebnik, "Training deeper convolutional networks with deep supervision," *arXiv preprint arXiv:1505.02496*, 2015.

[169] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-Supervised Nets," in *Artificial intelligence and statistics*, PMLR, 2015, pp. 562–570.

[170] Q. Dou, L. Yu, H. Chen, Y. Jin, X. Yang, J. Qin, and P.-A. Heng, "3D deeply supervised network for automated segmentation of volumetric medical images," *Medical image analysis*, vol. 41, pp. 40–54, 2017.

[171] D. Karimi and S. E. Salcudean, "Reducing the Hausdorff distance in medical image segmentation with convolutional neural networks," *IEEE Transactions on medical imaging*, vol. 39, no. 2, pp. 499–513, 2019.

[172] P. Mishra and K. Sarawadekar, "Polynomial learning rate policy with warm restart for deep neural network," in *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, IEEE, 2019, pp. 2087–2092.

[173] N. Girard, D. Smirnov, J. Solomon, and Y. Tarabalka, "Polygonal Building Extraction by Frame Field Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5891–5900.

[174] S. Zorzi, K. Bittner, and F. Fraundorfer, "Machine-learned regularization and polygonization of building segmentation masks," in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 3098–3105.

[175] A. Milosavljević, "Automated processing of remote sensing imagery using deep semantic segmentation: A building footprint extraction case," *ISPRS International Journal of Geo-Information*, vol. 9, no. 8, p. 486, 2020.

[176] Y. Xu, L. Wu, Z. Xie, and Z. Chen, "Building extraction in very high resolution remote sensing imagery using deep learning and guided filters," *Remote Sensing*, vol. 10, no. 1, p. 144, 2018.

[177] H. Liu, J. Luo, B. Huang, X. Hu, Y. Sun, Y. Yang, N. Xu, and N. Zhou, "DE-Net: Deep encoding network for building extraction from high-resolution remote sensing imagery," *Remote Sensing*, vol. 11, no. 20, p. 2380, 2019.

[178] J. Liu, Z. Wang, and K. Cheng, "An improved algorithm for semantic segmentation of remote sensing images based on DeepLabV3+," in *Proceedings of the 5th International Conference on Communication and Information Processing*, 2019, pp. 124–128.

VITA

Somrita Chattopadhyay was born on May 23, 1987 in the 'City of Joy' Kolkata in West Bengal, India. She joined the Elmore Family School of Electrical and Computer Engineering at Purdue University, West Lafayette, Indiana in August 2015 to pursue PhD degree. Throughout her PhD program, she was a member of the Robot Vision Laboratory led by professor Avinash Kak. She interned at 3M Corporation from May to August in 2017 and Oak Ridge National Laboratory from June to December in 2019. Prior to joining the PhD program at Purdue University, she earned dual Masters of Science (M.S.) degrees in Electrical Engineering and Mathematics from North Carolina State University, Raleigh, NC, USA, in 2014 and 2015 respectively. She received the Bachelor of Technology (B.Tech) degree in Electronics and Communications Engineering from the West Bengal University of Technology, India, in 2009. She did her schooling from Patha Bhavan, one of the top educational institutes in India. Her research interests include Computer Vision, and applying Deep Learning and Machine Learning techniques in the areas of object detection and tracking, style transfer, and remote sensing.