Urban Feature Classification from Remote Sensor Imagery Using Deep Neural Networks

Bodhiswatta Chatterjee

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Computer Science (Computer Science) at

Concordia University

Montréal, Québec, Canada

November 2019

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Bodhiswatta Chatterjee**

Entitled: **Urban Feature Classification from Remote Sensor Imagery Using Deep Neural Networks**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________________ Chair
Dr. Andrew Delong

______________________________ Examiner
Dr. Ching Suen

______________________________ Examiner
Dr. Thomas Fevens

______________________________ Supervisor
Dr. Charalambos Poullis

Approved by ______________________________
Dr. Lata Narayanan, Chair
Department of Computer Science and Software Engineering

September 2019 ______________________________
Dr. Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Urban Feature Classification from Remote Sensor Imagery Using Deep Neural Networks

Bodhiswatta Chatterjee

Convolutional neural networks have been shown to have a very high accuracy when applied to certain visual tasks and in particular semantic segmentation. In this thesis we address the problem of semantic segmentation of buildings from remote sensor imagery. We explore different architectures to semantic segmentation and propose ICT-Net: a novel network with the underlying architecture of a fully convolutional network, infused with feature re-calibrated Dense blocks at each layer. Uniquely, the proposed network (ICT-Net) combines the localization accuracy and use of context of the U-Net network architecture, the compact internal representations and reduced feature redundancy of the Dense blocks, and the dynamic channel-wise feature re-weighting of the Squeeze-and-Excitation(SE) blocks. The proposed network has been tested on two benchmark datasets and is shown to outperform all other state-of-the-art by more than 1.5% on the Jaccard index on INRIA's dataset and 1.8% on the Jaccard index on AIRS dataset.

Furthermore, as the building classification is typically the first step of the reconstruction process, in the latter part of the work we investigate the relationship of the classification accuracy to the reconstruction accuracy. A comparative quantitative analysis of reconstruction accuracies corresponding to different classification accuracies confirms the strong correlation between the two. We present the results which show a consistent and considerable reduction in the reconstruction accuracy.

The work presented in this thesis has been published in the 16^{th} Conference on Computer and Robot Vision 2019.

Acknowledgments

I would like to take the opportunity to convey my gratitude towards the people who have played an important role in this journey. I would like to express my sincere gratitude and respect towards my thesis supervisor, Dr. Charalambos Poullis for giving me the opportunity to work with ICT lab. This work would not have been possible without his guidance, continuous support and motivation. His valuable advice has always helped me in every phase of this journey, from carrying out the research to writing the paper and this thesis. I have been very fortunate to have a great supervisor like him.

The research done as part of this thesis is based upon work supported by the Natural Sciences and Engineering Research Council of Canada Grants DG-N01670 (Discovery Grant) and DND-N01885 (Collaborative Research and Development with the Department of National Defence Grant). I would like to thank Jonathan Fournier from Valcartier DRDC, and Hermann Brassard, Sylvain Pronovost, Bhakti Patel, and Scott McAvoy from Presagis Inc Canada, for their invaluable discussions and assistance in processing maps for Montreal. I would also like to thank Defence Research and Development Canada and Thales Canada for providing orthophoto RGB images of Montreal used for testing.

I would like to thank the respectable committee members and other professors and staffs of Concordia University for the help and support I have received from them. Last but not the least, I would like to thank my family and my wife for their continuous love and support that have always been the inspiration in every phase of my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Reconstructing large-scale urban areas is an inherently complex problem which involves a number of vision tasks. Typically, the first step is classification where the objective is to label each pixel into an urban feature type e.g., building, road, tree, car, ground, vegetation, etc. Next, the pixel-level labels are used to cluster the pixels into contiguous groups corresponding to instances of the urban features they represent. Finally, the reconstruction is performed on each cluster. A reconstruction algorithm is applied on each cluster according to the urban feature type the cluster corresponds to. In the case of clusters corresponding to buildings, a boundary refinement process is typically performed prior to extruding the building facades.

The objectives and contributions of this work are twofold. Firstly, we address the problem of the classification of buildings in remote sensor imagery. We investigate a number of state-of-the-art deep neural network architectures and present a comparative study of the results along with a reasoned justification on the design decisions for the proposed network named ICT-Net: a novel network with the underlying architecture of a fully convolutional network infused with Dense feature re-calibrated blocks at each layer. We demonstrate that this combination of components leads to superior performance. The proposed network is ranked first (since January 2019) at two international benchmark competitions: (a) the INRIA Aerial Image labeling challenge with more than 1.5% difference in terms of performance from the second best and other ensemble networks, and (b) the Aerial Imagery for Roof Segmentation(AIRS) challenge with more than 1.8% difference in terms of performance from the second best by Pyramid Scene Parsing Network [65] (PSPNet), which is one of the state-of-the-art deep learning models for semantic image segmentation and the winner of ImageNet scene parsing challenge 2016.

Secondly, we address the problem of reconstruction of the classified buildings and in particular study the relationship between the classification accuracy and reconstruction accuracy. We perform a comparative quantitative analysis on the reconstructions corresponding to classifications of different accuracies and report the results. Due to the lack of depth information, reconstructing 3D models is not feasible therefore the accuracy of the border localization is used as proxy for the evaluation since it is tightly coupled to the reconstruction accuracy i.e. buildings are extruded using their boundaries. As anticipated there is a strong correlation between the classification accuracy and the accuracy of the reconstruction however the analysis has shown that there is a consistent and considerable decrease in the reconstruction accuracy in terms of the per-pixel and per-building Jaccard indices. To the best of our knowledge this is the first time a quantitative analysis is performed in order to establish how the classification accuracy relates to the accuracy of the reconstruction as determined by the accuracy of the border localization.

Thesis organization: This Thesis is organized as follows: Chapter 2 presents an overview of state-of-the-art in the area of image classification, semantic segmentation, capsule networks and building foot-print extraction in satellite images using deep neural networks. Chapter 3 summarizes our work related to building classification using a capsule-based architecture. The proposed neural network ICT-Net is explained in chapter 4 including a reasoned justification of the design decisions, and details on the training and testing of the network. Chapter 5 presents a quantitative analysis of the reconstruction accuracies resulting from different classification accuracies, and chapter 6 concludes the work and discusses future directions.

Chapter 2

Literature Review

Starting from a summer project at MIT in 1966 to a complete field, computer vision has evolved and matured to a stage where recent systems have super human level accuracy on image classification tasks like ImageNet [12]. A typical pipeline for most computer vision systems involve feature extraction from the imagery followed by processing of those features to accomplish a task. Initially for the most common task of image classification, hand-engineered features like (SIFT [40], SURF [4], Spatial pyramid features [6]) were extracted from the imagery and a classifier (SVM [22], Decision tree [49], Neural Networks [51]) was trained using the extracted features to classify the image.

Yann Lecun's designed architecture LeNet [35] was the first to have an end-to-end trainable system where the feature extraction was trained using Convolutional layers followed by fully connected layers to recognize (classify) hand-written digits on cheques. At the time of its invention scaling the LeNet architecture to larger images was challenging due to the amount of computational power required, but with the advent of Graphics Processing Units (GPU) it became feasible to run massively parallel computation and more complicated architectures of Convolutional Neural Networks started to takeover the ImageNet leader-board. Other tasks like object localization and semantic segmentation are also influenced by the classification network architecture research as they are used as the backbone or building block for most object localization and semantic segmentation networks. In the next few sections we discuss about the most popular CNN architectures for image classification and semantic segmentation networks, followed by a new approach to image classification called Capsule networks and a brief overview of current state-of-art techniques for building footprint extraction.

2.1 Image Classification

Image classification is the task of classifying a whole image into a category. Supervised deep learning based approaches require a huge amount of labeled data to train a model to perform this task. ImageNet [12] dataset was proposed in 2009 for this purpose, it has over 15 million labeled high-resolution images belonging to roughly 22,000 categories. The images were collected from the web and were manually labeled using Amazon's Mechanical Turk crowd-sourcing tool. Starting in 2010, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held. ILSVRC uses a subset of ImageNet with roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images. Since then many breakthroughs in image classification have been reported on this dataset. We will look at some of the important deep neural network architectures which were able to able to improve the state-of-art on ILSVRC as well as image classification in general.

2.1.1 AlexNet

Krizhevsky *et al.* [33] started the deep learning revolution in computer vision. The authors in this paper define a Convolutional Neural Network architecture (now popularly known as AlexNet) which was architecturally similar to Yann Lecun's designed architecture LeNet-5 but outperformed the previous state-of-art on ILSVRC challenge by a large margin. AlexNet introduced a lot of new concepts which later became the standard for training of deep neural networks. It consists of 8 layers with 60 million parameters. The first 5 are convolutional layers followed by 3 fully connected layers. An architecture diagram of AlexNet can be seen in Figure 2.1. Few of the most notable contributions of this paper were

- use of non-saturating non-linearity, ReLU [44] for training of convolutional networks
- very efficient implementation of the network spread over 2 GPU
- data augmentation as a technique to increase the size of dataset by many folds

- Dropout as an additional technique to reduce over fitting

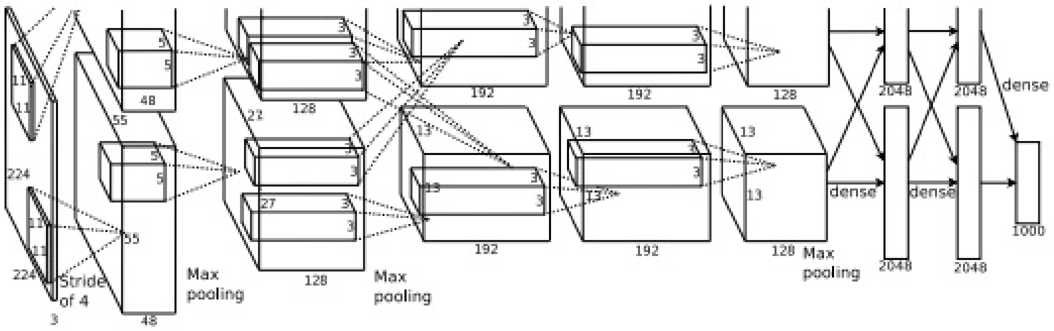


Figure 2.1: An illustration of the AlexNet [33] architecture, explicitly showing the delineation of responsibilities between the two GPUs.

2.1.2 VGG

Simonyan *et al.* [56] have observed that deeper networks have the capacity to learn better features. The authors were able to train a 16 layer and a 19 layer network by (popularly known as VGG) with only last 3 layers being fully connected and the rest were all convolutional layers. Another very significant contribution of this paper was the authors show that the use of (3x3) convolutional filters is sufficient to train a deep neural network instead of (11x11) convolutional filters used in AlexNet [33] or (7x7) convolutional filters used in **Visualizing and Understanding Convolutional Networks** [64]. Reducing the size of convolutional filters reduces the number of parameters learnt by the network which in-turn reduces over-fitting to the training data. A diagram of the architecture of the VGG network can be seen in Figure 2.2.

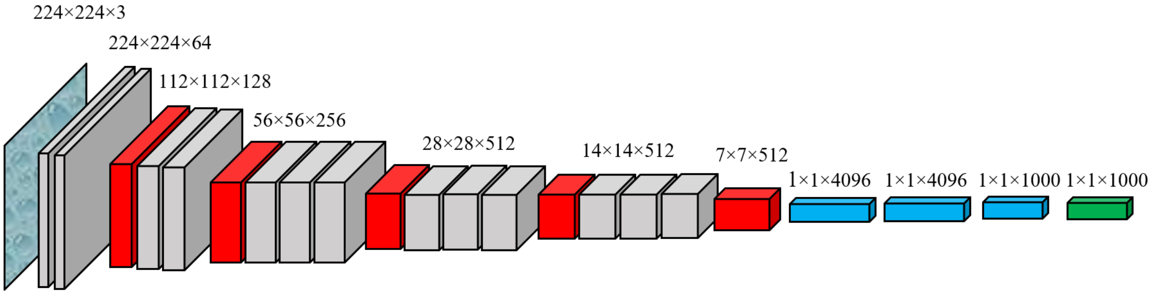


Figure 2.2: An illustration of the VGG-16 architecture. Image courtesy to wikipedia [60]

2.1.3 Inception

Szegedy *et al.* [58] were able to train a 22 layer deep neural network (named as Inception) to get state-of-art results on ILSVRC. The authors emphasized on the efficiency of the network architecture design as it has 12 times less parameters than [33] but have much higher accuracy. The network introduced the concept of modules (inception blocks) inside the network which run multiple convolutions using different filter sizes and can be done in parallel. One of the most notable contributions of this paper is that the design of Inception Network allows to have classification results without using a fully connected layers at the end which was not common for most classification network of that time. Removal of the fully connected layers played the most important role in reducing the number of parameters in the network. Inception network also introduced the use of multiple sized convolutional filters in the same network and showed how convolutional layers with (1x1) filters can be used for dimensionality reduction to remove computational bottlenecks. An architecture diagram of GoogLeNet (Inception) network can be seen in Figure 2.3.

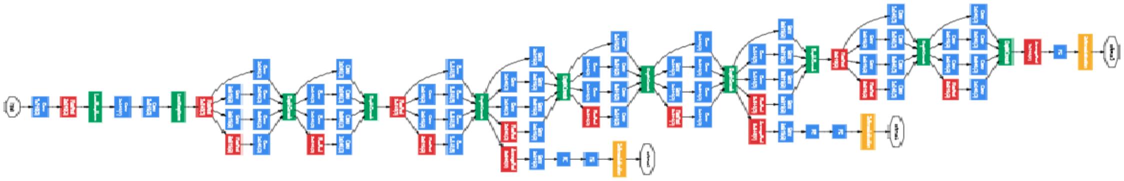


Figure 2.3: An illustration of the GoogLeNet architecture. [58]

2.1.4 ResNet

He *et al.* [21] were able to train a 152 layer convolutional network on the ImageNet dataset and improve the ILSVRC state-of-art significantly by introducing residual learning with the help of Residual blocks. They also show experimentally that just increasing the depth of the network does not improve classification performance. Using Residual blocks the authors were able to train a variant of the ResNet network which is 1000 layers deep on the CIFAR dataset. Some variants of this network like ResNet-34 and ResNet-50 have been very commonly used as a backbone in segmentation networks or to extract learned features from an image. Due to the high performance of the network there are a huge number of variants of this network like ResNext [61], WideResNext [63], Inception-ResNet [57], etc. An architecture diagram of 2 different ResNet blocks can be seen in Figure 2.4.

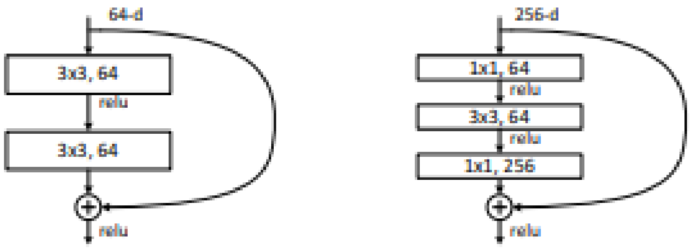


Figure 2.4: An illustration of the 2 different types of ResNet [21] block. Left: a ResNet building block for ResNet34. Right: a "bottleneck" ResNet building block for ResNet-50/101/152

2.1.5 DenseNet

Huang *et al.* [27] introduced the idea of Dense Convolutional blocks (DenseNet), which connects each layer to every other layer in a feed-forward fashion. They were able to outperform the state-of-art on ImageNet and many other image classification challenges. The advantages of this architecture are that they alleviate the vanishing-gradient problem and strengthen feature propagation by reusing lower layer features, which also substantially reduces the number of parameters in the network as the convolutional layers are very narrow with a growth rate of k which is significantly lower than traditional convolutional layers. An architecture diagram of 5 layer Dense blocks can be seen in Figure 2.5.

2.1.6 Squeeze and Excitation Network

Most deep neural networks for object recognition consider all extracted features at each layer to be of equal importance. This was until the method proposed in [25] showed that adaptive re-calibration of channel-wise feature i.e. weighing of the features, can be used effectively to model inter-dependencies between channels and produce even better performance with little computational

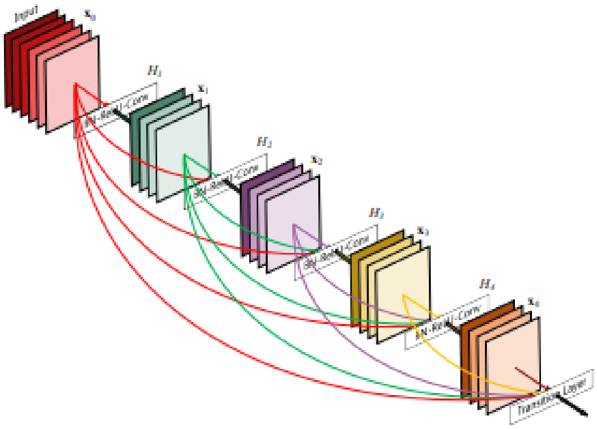


Figure 2.5: An illustration of 5-layer dense block [27] with a growth rate of k = 4. Each layer uses all preceding feature-maps as input

overhead. It can be used as a drop-in replacement block with most commonly used CNN architectures. The authors used SE blocks with multiple base networks and were able to achieve better performance than the base networks. Using SE-ResNet-154 the authors were able to achieve better than previous state-of-art on ILSVRC 2017. An architecture diagram of SE blocks can be seen in Figure 2.6.

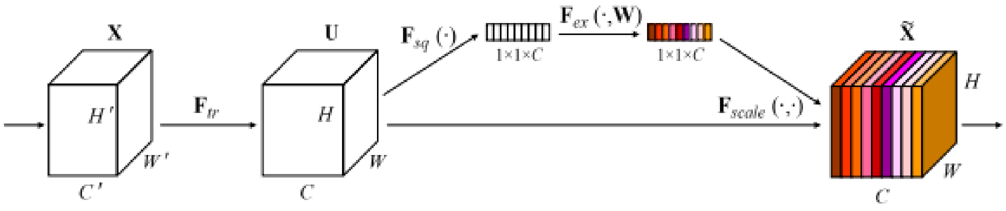


Figure 2.6: An illustration of a Squeeze-and-Excitation [25] block.

2.2 Semantic Segmentation

With super-human performance on image classification tasks, the current focus of computer vision research has shifted towards more challenging tasks like object localization or per-pixel semantic object segmentation. Prior to deep learning, semantic labeling required extraction of hand engineered features. One of these methods [54] proposed the generation of features that were classified into unary potentials and fed into conditional random fields (CRF), localizing the label and segmenting objects.

Recent techniques using deep neural networks have demonstrated excellent results. It is useful to have per-pixel semantic label in many situations like shape or geometry analysis of objects, 3D modeling of objects from 2D imagery, medical imaging, Self-Driving cars, etc. Some of the benchmark datasets in this domain are Pascal Visual Object Classes (VOC) Challenge [15] and Microsoft COCO: Common Objects in Context [38]. Cityscapes is another very renowned dataset for Semantic Urban Scene Understanding [10] which provides semantic labels on urban scenes focused on self-driving automobile research. Recently there has been an increasing interest on pixel-wise classification and extraction of urban features from Satellite or Aerial imagery. Few renowned datasets for classification of urban features are SpaceNet [14], Inria Aerial Image Labeling Dataset [41], ISPRS dataset for Potsdam, Vaihingen and Toronto.

In recent years there has been a plethora of work on the design of Semantic Segmentation architectures. Typical semantic segmentation architectures comprise of a down-sampling path responsible for feature extraction and an up-sampling path to restore the resolution of the semantic labels. Skip connections between the two paths help to have a smooth gradient back propagation and fast training of the network. Below we provide a brief overview of the state-of-the-art related to the area of semantic labeling with an emphasis on how the architectures evolved overtime. We discuss the architectures which are closely related to our work but a comprehensive review of neural network architectures for semantic segmentation can be found in [17].

2.2.1 Fully Convolutional Network (FCN)

Long *et al.* [39] adapt classification networks (AlexNet [33], the VGG net [56], and GoogLeNet [58]) into fully convolutional networks and transfer their learned representations to the segmentation task. They define a skip architecture that combines semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to get high quality segmentation results. Multiple variants of the network were proposed by the authors from which FCN-8s (using VGG backbone) delivered the best performance and was able to outperform the previous state-of-art by a significant margin. An architecture diagram of Fully convolutional networks(FCN) can be seen in

Figure 2.7.

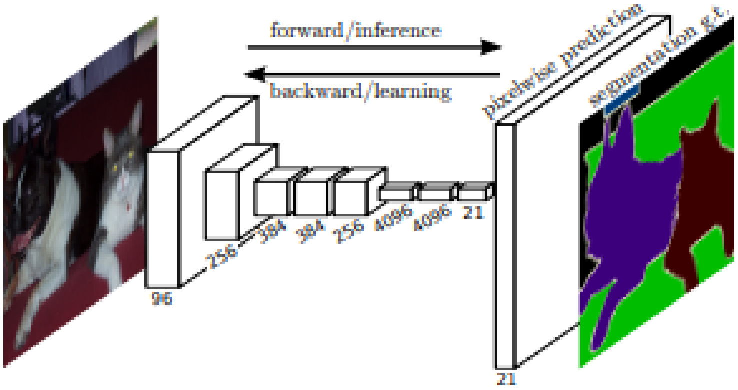


Figure 2.7: An illustration of Fully convolutional networks [39] can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation

2.2.2 SegNet

Badrinarayanan *et al.* [1] has an encoder-decoder architecture (SegNet) where the encoder part of the network is topologically identical to 13 convolutional layers of VGG16 network [56]. The role of the decoder network is to map the low resolution encoder feature maps to full input resolution feature maps for pixel-wise classification. The decoder uses pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling. This eliminates the need for learning to upsample. The sparse upsampled maps are then convolved with trainable filters to produce dense feature maps. The authors of SegNet dataset have shown very good results on scene understanding benchmark datasets. An architecture diagram of SegNet can be seen in Figure 2.8.

2.2.3 U-Net

Ronneberger *et al.* [50] proposed the U-Net architecture which was able to achieve end-to-end semantic labeling with high accuracy in the field of medical image segmentation. Architecturally it is very similar to SegNet but instead of pooling indices from the encoder it concatenates the encoder activations with the upsampled feature maps. The original version of U-Net used VGG style backbone as the encoder but since then the U-Net [50] architecture has been extensively used and

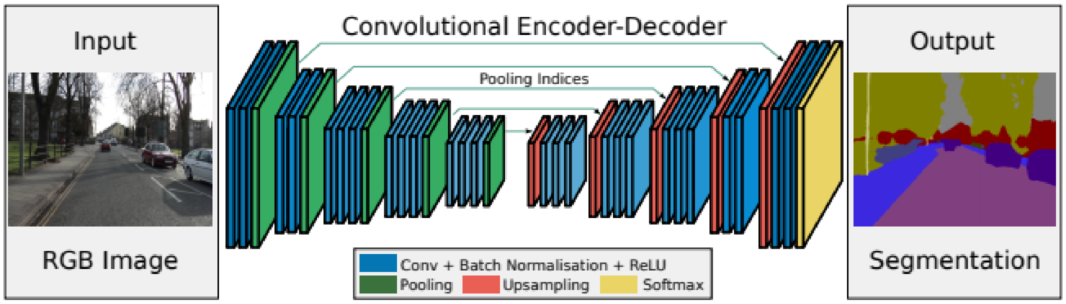


Figure 2.8: An illustration of the SegNet architecture. A decoder upsamples its input using the transferred pool indices from its encoder to produce a sparse feature map(s). It then performs convolution with a trainable filter bank to densify the feature map.

adapted to many other domains with different variants of ResNet as the backbone especially labeling of buildings from aerial imagery as in [20,28]. An architecture diagram of U-Net can be seen in Figure 2.9.

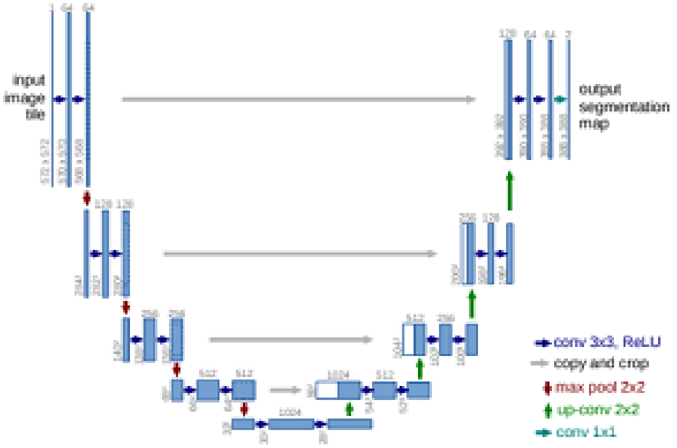


Figure 2.9: An illustration of the U-Net [50] architecture.

2.2.4 Fully Convolutional DenseNets for Semantic Segmentation

Jegou *et al.* [30] proposed a U-Net style segmentation network with a very promising pattern known as Dense blocks proposed in [27] for the problem of image classification. In a Dense block every layer is connected to every other layer in a feed forward fashion. This provides implicit deep supervision and feature reuse which in turn improves the feature extraction power without making it difficult to train the network. The one hundred layers Tiramisu network architecture proposed in [30] extended the use of Dense blocks for semantic segmentation and was able to outperform state-of-the-art on two benchmark data sets: Gatech and CamVid. An architecture diagram of Fully Convolutional DenseNets for Semantic Segmentation can be seen in Figure 2.10.

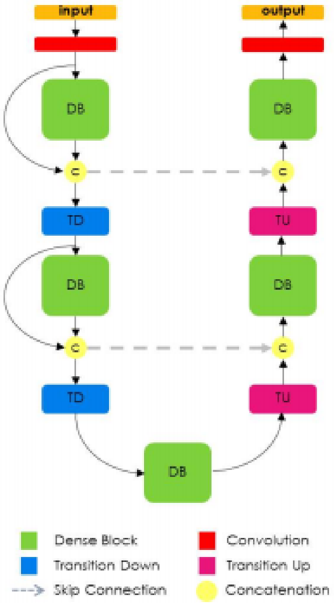


Figure 2.10: An illustration of the Fully Convolutional DenseNets for Semantic Segmentation [30] architecture. It is built from dense blocks. The diagram is composed of a downsampling path with 2 Transitions Down (TD) and an upsampling path with 2 Transitions Up (TU)

2.3 Capsule Networks

Convolutional Neural Networks(CNN) have been able to dominate in most fields of computer vision for the last decade but there are few weaknesses to the CNN approach to computer vision. CNN architectures need pooling layers to achieve local invariance where they should actually strive for equivariance. Another important property of CNN architectures is it does not account for parts-to-whole relationships for objects. Also there is growing interest in the vulnerability of neural networks to adversarial examples; inputs that have been slightly changed by an attacker to trick a neural net classifier into making the wrong classification [18].

Capsule networks [23] proposed by Hinton et al. tries to resolve these problems by introducing

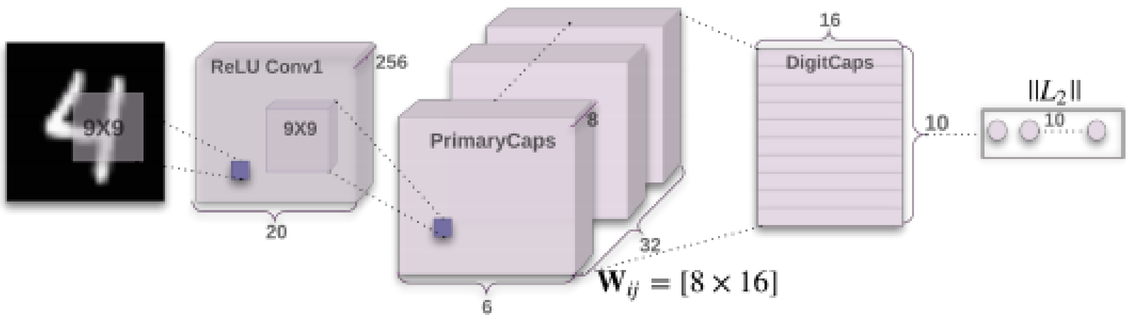


Figure 2.11: An illustration of a simple Capsule Network [52] with 3 layers. The length of the activity vector of each capsule in DigitCaps layer indicates presence of an instance of each class and is used to calculate the classification loss.

capsules, which are group of neurons whose activity vector represents the instantiation parameters of a specific type of entity such as an object or an object part. Activation maps produced by capsule layers is an n-dimensional array of vectors instead of scalars where the Vector Length signifies the estimated probability of presence of object or object part and the vector dimensions contains estimated pose parameters of object or object part. An architecture diagram of Capsule Networks can be seen in Figure 2.11. The pooling operation of CNN architectures is replaced by a voting mechanism where active capsules at one level make predictions , via transformation matrices, for the higher-level capsules. When there is consensus on a higher level capsule by multiple lower level capsules, it becomes active. To date, the known techniques for voting mechanisms include Dynamic routing [52] by Sabour *et al.* and EM routing [24] by Hinton *et al.* Capsule networks have achieved state-of-art results on MNIST Hand written Digit recognition [34] dataset, even with multiple overlapping digits and on the smallNORB [36] dataset. Hinton et al. [24] also show that capsule networks are significantly less vulnerable to adversarial attacks.

Although Capsule networks look like a very promising direction, the currently known routing algorithms are iterative in nature which makes the training of these networks very time consuming and the number of iterations for the best performance of the routing algorithm is a hyper-parameter that needs to be tuned. Also scaling Capsule layers to large images is a very challenging task as capsules require large chunks of memory.

2.4 Building Footprint Extraction

With respect to urban reconstruction, the extraction of urban geospatial features such as buildings from remote sensor imagery has also been an area of research interest for a very long time [19, 53, 59]. Automatic reconstruction of 3D models from the extracted features is extremely useful for many applications ranging from urban and community planning, development and architectural design, training of emergency response personnel, military personnel, etc. In [45] the authors propose a novel, robust, automatic segmentation technique based on the statistical analysis of the geometric properties of the data as well as an efficient and automatic modeling pipeline for the reconstruction of large-scale areas containing several thousands of buildings. With the recent advances in deep neural network architectures the pipeline has been upgraded to feature extraction using a semantic labeling CNN followed by clustering the points based on their label, and specialized processing for each of the labels of geospatial objects as proposed in [16].

Recently there has been a lot of interest for semantic labeling of buildings [28, 31, 43] fueled by the release of very large datasets like INRIA Aerial Image Labeling dataset [41], and SpaceNet where a corpus of commercial satellite imagery with labeled training data was made publicly available for use in machine learning research. In [28] the authors use a variant of the aforementioned U-Net network architectures replacing the VGG11 [56] encoder with a more powerful activated Batch Normalized [7] WideResnet-38 [63] in the context of instance segmentation of buildings for DeepGlobe-CVPR 2018 building detection sub-challenge, and were able to get very good results.

Conclusion

Although many architectures have been proposed for classification, there still remains a large gap between the current state-of-the-art and the ultimate goal of semantic segmentation of large-scale remote sensed images. In this thesis, we focus on bridging the gap by exploring different architectures and proposing ICT-Net: a novel network architecture that combines the strengths of deep neural network architectures (UNet) and building blocks (DenseNet block, SE block) which when applied to the problem of semantic labeling of buildings is proven to achieve better classification accuracy than state-of-the-art on the INRIA Aerial Image Labeling dataset. As of Nov 2019 the proposed network is top ranked on the competitions' leaderboard with more than 1.5% difference from the second best entry on INRIA and 1.8% difference AIRS benchmark datasets.

Chapter 3

Building Classification with Capsule Network

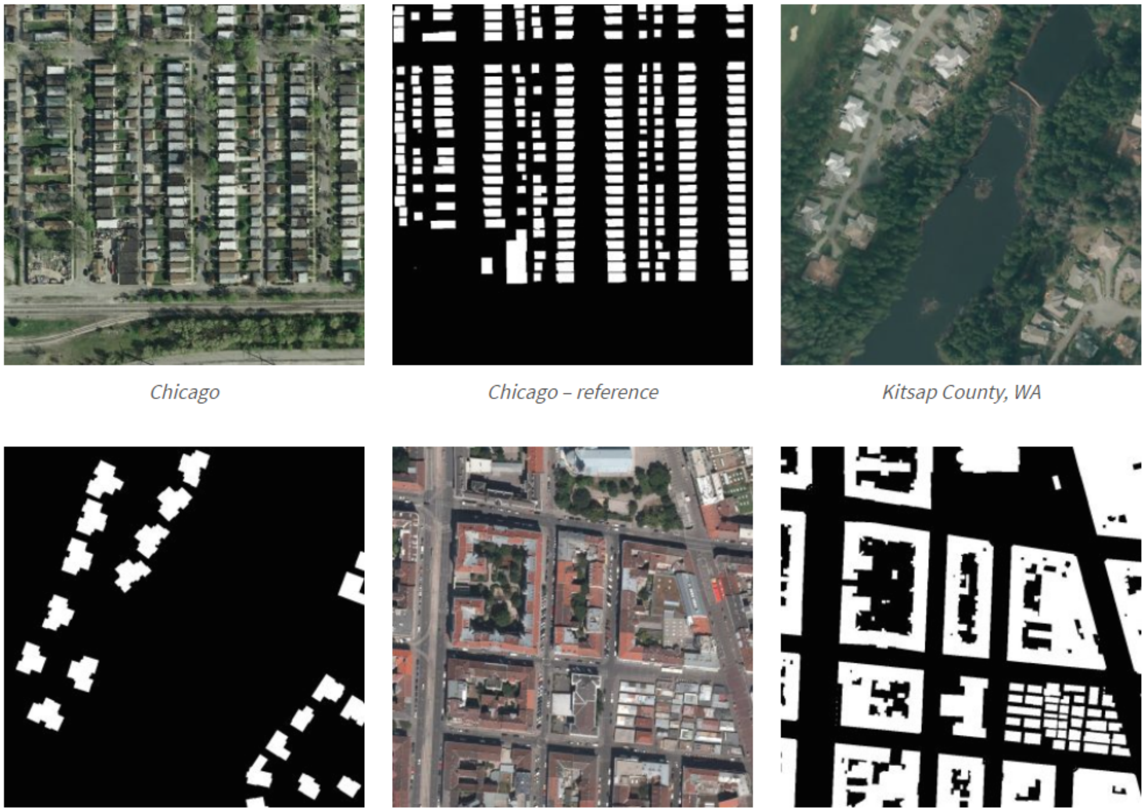
In this chapter we describe the details of our approach to pixel-wise classification of building from aerial imagery using our proposed capsule network architectures. We also include the justification for our design decision choice of using capsule network. Along with the architectures we also discuss about the second most important aspect of any supervised classification system - the dataset. We used the INRIA Aerial Image labeling [41] benchmark datasets to evaluate the performance of our networks. We also provide extensive details of training/validation, and testing of the model and end the chapter with some quantitative and qualitative results obtained by the models.

3.1 Dataset

The Inria benchmark datasets is organized as an open challenge where the publishers provided imagery and ground truth with per-pixel building vs non-building labels. Only a part (training data) of the dataset's ground truth is publicly available for training of neural networks and to maintain fairness of evaluation they have made it mandatory to submit the prediction on the other part (test data). It is evaluated and published on a leaderboard which is publicly accessible.

The training of the networks is performed using the INRIA Aerial Image labeling dataset which consists of pixelwise labeled aerial imagery for building classification. The dataset covers $810km^2$

area across 10 different cities with spatial resolution of $30cm$, and is split into two equal sets ($405Km^2$ each) for training and testing. The dataset consists of 3-band orthorectified RGB images and the training labels consist of ground truth data for two semantic classes: building and non-building. The training data covers parts of the cities of Austin, Chicago, Kitsap county, western Tyrol, and Vienna. The test data covers parts of the cities of Bellingham, Bloomington, Innsbruck, San Francisco and Eastern Tyrol. There are 36 tiles with resolution of 5000×5000 pixels for each city, each tile covering $1500 \times 1500m^2$ area on the ground. The training data is further divided into two sets: (1) the validation set which comprises of the first 5 tiles of each city, and (2) the training set which consists of the rest of the tiles as suggested in [41]. An example image from the dataset can be seen in Figure 3.1.



Chicago

Chicago – reference

Kitsap County, WA

Kitsap County, WA – Reference

Vienna

Vienna – Reference

Figure 3.1: Sample imagery from INRIA Aerial Image labeling dataset.

We have chosen the INRIA benchmark dataset over other available options because it uniquely offers two significant advantages. Firstly, the training and testing datasets are from *completely different cities* with no overlap i.e. *all* images of 5 cities (Austin, Chicago, Kitsap, Western-Tyrol, Vienna) are provided for training, and *all* images of another 5 different cities (Bellingham, Bloomington, Innsbruck, San Francisco, Eastern-Tyrol) are used for testing. Secondly,the dataset covers *dissimilar urban settlements* e.g., European, American, etc, with large variations in building density, architecture, and overall characteristics e.g., red shingles, flat roofs, etc. For these reasons, we have chosen this benchmark dataset because it is ideal for assessing the *generalization capacity of the network.*

3.2 Network Architecture

Convolutional neural networks are very good at classifying objects by detecting the presence of features related to object parts and objects in a hierarchical fashion inside the network for any input image. The activation in a CNN architecture are scalar values so it can only indicate presence or absence of a feature in an image. Capsule networks [23] proposed by Hinton et al. introduces entities in a network, these entities are known as capsules which indicate the presence and instantiation parameters of an object or object parts. In capsule-based architectures the activation maps produced by capsule layers is an n-dimensional array of vectors instead of scalars where Vector Length signifies the estimated probability of presence of object or object part and the vector dimensions contains estimated pose parameters or other properties of object or object part.

Requirement: An important aspect of 3D reconstruction of urban features(ie. buildings in this case) using the per-pixel semantic classification from the network and depth information, is more information (like roof-type, roof-materials, etc) about each building can improve the quality of 3D reconstruction.

Decision: Based on the promise of capsule networks, it is perfect fit to the situation. The magnitude of a capsules activation vector can indicate the presence or absence of a buildings and if present other dimensions of the vector can encode the properties of the building.

Architecture: Scaling capsule layers to large images is a very challenging task as capsules require large chunks of memory. To get rid of the scaling problem we take inspiration from Sabor *et al.*

[52] and try a combination of convolution and capsule layers where the core or bottleneck of the network is made up of capsule layers and it is enclosed with convolution and up-convolution layers. The network takes as input a patch of size 128×128. It is followed by 5 layers of convolution operation which extracts features from the image. For 2^{nd} and 4^{th} convolution layer we use a strided convolution with a stride of 2 which helps us extract features as well as reduce the spatial extent of features to $\frac{1}{4}^{th}$ of the input size. The activations from convolution layer 5 is fed to the *primary capsule layer* whose activations are 16 dimensional vectors. It leads to a *classification capsule layer* which has 32 dimensional activation vectors. It is followed by a fully connected decoder module which uses the activation from classification capsule to reconstruct a 16×16 segmentation mask. It is followed by 2 up-convolution blocks where each block consists of one transpose convolution followed by a convolution layer. The output of last convolution block is a 128×128 segmentation mask.

3.3 Training and Validation

The network is trained on 155 tiles each with resolution 5000×5000 from the available training data with their corresponding ground truth. The training is performed for 30 epochs on a single nVidiaGTX 1080Ti. We used Tensorflow API for the development and training/testing of the network. Due to the iterative process in dynamic routing algorithm the training process is very slow and requires approximately 20 hours to complete 1 epoch of training. Every epoch was divided into 31 sub-epochs each consisting of 5 tiles (1 from each city). Limited by GPU memory we had to choose a small batch size of 4 to have a comparatively larger patch size of 128×128 as we observed context is very important for semantic labeling of buildings.

Implementation details: The network was trained using a Margin loss as in the dynamic routing [52] paper. Our margin loss was defined as a combination of cross-entropy classification loss and regression reconstruction loss with equal weight for both the losses. Adam Optimizer with a learning rate of 0.0001 was used to train the network for 30 epoch.

Data input: Our network takes in patches of 128×128 out of the entire tile with 50% overlap. At the time of training a patch is classified as building if its mask has at least 100 pixels labeled as building. Among all the (approximately 30000) patches, 5000 randomly sampled patched are selected for each mini-epoch of training. At testing the same input patch size of 128×128 had to be used.

Network output: Generally CNN architectures are slow to train but at inference time they are much faster but Capsule based architectures are slow even at inference time due to the iterative routing algorithm. The output produced by the network is a 1-channel gray-scale image of the same size as the input image where each pixel has a probability score of being a building in the range $[0, 1]$. We convert the probability map into a binary mask by applying threshold. Predictions for each patch are generated and then they are combined to form a 5000×5000 segmentation mask. It takes approximately $40min$ to generate prediction for 1 input tile of 5000×5000.

3.4 Results

The INRIA dataset uses two main performance measures: Intersection over Union (Jaccard index) and Accuracy. Intersection over Union (IoU) is defined as the number of pixels labeled as buildings in both the prediction and the reference, divided by the number of pixels labeled as buildings in the prediction or the reference. Accuracy is defined as the percentage of correctly classified pixels. On the validation set of 25 image tiles we achieved 70.42% (IoU) and 95.14% (accuracy). An illustration of prediction by the network can be seen in Figure 3.2. The test set evaluation is done by the organizers of the competition and involves the classification of 5 cities for which no images have been used for training and validation, and for which no ground truth is available to the participants. We achieved 65.83% (IoU) and 94.80% (accuracy) on the test set which can be found on the competition's leaderboard [1]. City-wise performance on the test dataset can be found in table 3.1.

> Our capsule-based architecture achieves 65.83% IoU and 94.80% accuracy on the overall test dataset of INRIA.

[1] https://project.inria.fr/aerialimagelabeling/leaderboard/

City	IoU (%)	Accuracy (%)
Bellingham	66.14	96.53
Bloomington	50.30	95.33
Innsbruck	64.50	95.60
San Francisco	71.05	90.11
East Tyrol	63.94	96.46
Overall	**65.83**	**94.80**

Table 3.1: Performance evaluation of capsule based architecture on the test dataset.

Conclusion

The concept of Capsule networks look very promising but the currently known iterative routing algorithms make both training and inference very slow for these networks. It took us approximately 5 days to generate inference prediction on the test dataset of 180 image tiles of 5000×5000. Another challenge with Capsule layers is to use large patch size as input to the network because it requires large chunks of memory. However, context is very important in the classification of remote sensing imagery due to variation in shape and size of objects like buildings and the ability to increase the patch size can have a huge impact on final prediction as demonstrated by the AMLL team in [26]. Due to the challenges discussed above we continued exploring other CNN-based architectures for segmentation of buildings the best of which is discussed in chapter 4.

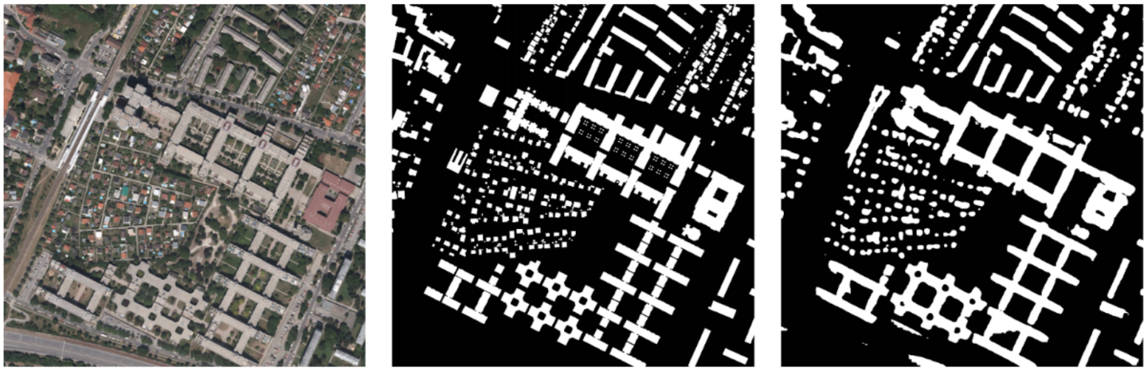


Figure 3.2: Building Classification of a patch from vienna city, tile 02. Result for an image from the validation dataset. Left: A patch from vienna city as input image. Middle: The semantically labeled ground truth image. Right: The binary map prediction resulting after the thresholding of the probability output from the capsule based segmentation network.

Chapter 4

Building Segmentation with ICT-Net

In this chapter we describe our approach to pixel-wise classification of building from aerial imagery using our proposed neural network architectures ICT-Net. In the discussion of network architecture we include the justification for all design decision choices. We will also provide a detailed overview of the datasets used to benchmark our network, namely INRIA Aerial Image labeling [41] and Aerial Imagery for Roof Segmentation(AIRS) [9]. We will also provide extensive details of training/validation, and testing of the model on both datasets and end the chapter with some qualitative and quantitative results of the models.

4.1 Dataset

The Inria benchmark dataset is one of the most popular dataset for Building segmentation as it has a huge variety of imagery as it is acquired from 10 different cities with different urban settlement types and density. So we choose to that as our primary dataset for training and experiments. More details about this dataset can be found in chapter 3.1 and some sample imagery from the dataset can be seen in Figure 3.1.

In addition to INRIA, we also use the AIRS dataset for benchmarking ICT-Net. Similar to the INRIA dataset, publishers for the AIRS benchmark dataset also provide imagery and ground truth with per-pixel roof vs non-roof labels. Only a part (training and validation) of the dataset's ground truth is publicly available for training of neural network and to maintain fairness of evaluation they have made it mandatory to submit the prediction on the test data. It is evaluated and published on a leaderboard which is publicly accessible.

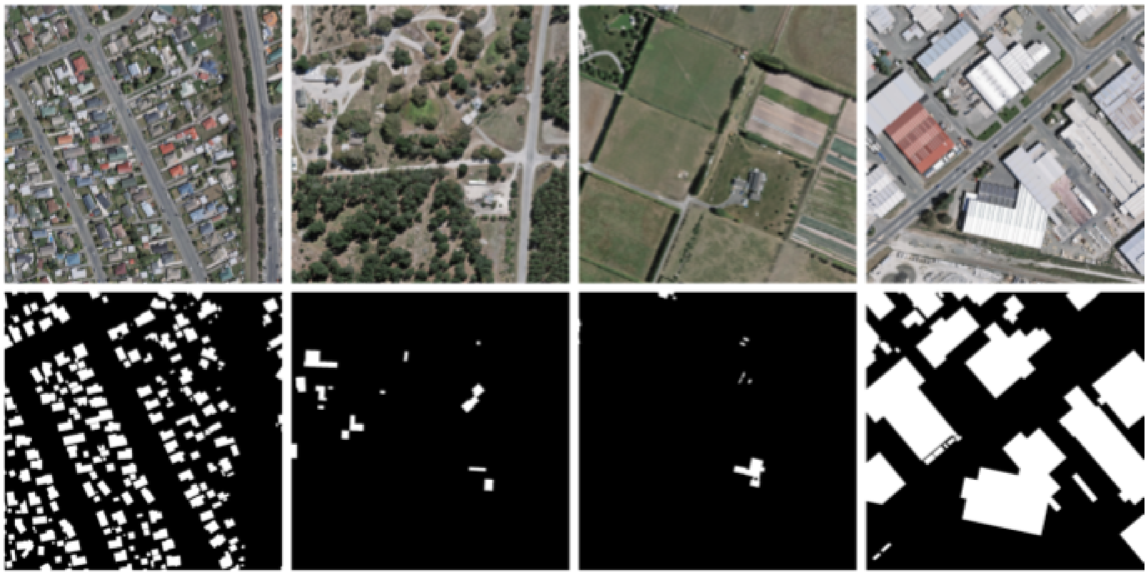


Figure 4.1: Sample imagery from Aerial Imagery for Roof Segmentation (AIRS) dataset.

AIRS (Aerial Imagery for Roof Segmentation) is a public dataset that aims at benchmarking the algorithms of roof segmentation from very high-resolution aerial imagery. AIRS dataset covers almost the full area of Christchurch, the largest city in the South Island of New Zealand. Although the aerial imagery is from one city but there is huge variety of settlement types. An illustration of sample images from the AIRS dataset can be seen in 4.1. The imagery contains 3-band orthorectified RGB images at $7.5cm$ ground resolution. It has a coverage of $457Km^2$ aerial images with over 220,000(approx.) buildings and refined ground truths that strictly align with roof outlines.There are 1046 tiles with resolution of 10000×10000 pixels which are already split into train, validation and test split of 857, 94 and 95 as the dataset has been published. The ground truth for buildings is carefully refined to align with their roofs and the segmentation task for AIRS contains two semantic classes: roof and non-roof pixels. We have chosen the AIRS benchmark dataset as it covers a completely different geographic location with different urban settlements and the aerial imagery is very high resolution so we are able to validate the *generalization capacity* of the trained neural network.

4.2 Network Architecture

A vast number of networks has been proposed for image classification and semantic labeling. State-of-the-art performance is generally achieved with deep networks however these are difficult to train due to vanishing or exploding gradients. Many networks [21, 27, 30, 50] have shown that skip connections play an important role in having good gradient propagation through the network. In our work, as part of the network design process, we first identified the requirements for the particular task at hand i.e. semantic segmentation of buildings from remote sensor images, and then decisions were made to address these:

- **Requirement 1:** An important aspect of semantic segmentation of buildings is to have high localization accuracy and take into account as much context information as possible. This is necessary in order to address the wide variability in buildings typically relating to their function e.g., shape, size, color and/or region they appear in e.g., density in urban/rural, etc.

  Decision: To that end, the U-Net architecture [50] takes into account spatial information and combines it with contextual information via the direct downsampling-upsampling links.

- **Requirement 2:** In order to be able to process large chunks of data at a time it is imperative that the network contains as few parameters as possible.

  Decision: Dense blocks connect every layer to every other layer in a feed-forward fashion. Along with good gradient propagation they also encourage feature reuse and reduce the number of parameters substantially as there is no need to relearn the redundant feature maps. At the end of every Dense block all the extracted features accumulate creating a very diverse set of features. As a result of this feature redundancy there is a substantial reduction in the network parameters leading to faster training times. This allows the processing of larger patch (and batch) sizes (which also addresses Requirement 1) therefore allowing additional contextual information during each feed-forward pass.

- **Requirement 3:** The contribution of the feature maps at each layer to the output must depend on their importance.

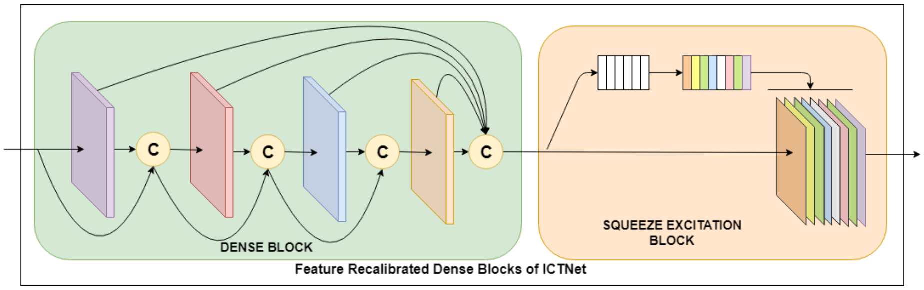


Figure 4.2: Proposed feature recalibrated Dense block with 4 convolutional layers and a growth rate $\kappa = 12$ used by the ICT-Net. c stands for concatenation.

Decision: Using the Squeeze-and-Excitation (SE) blocks the dynamic channel-wise feature re-weighting mechanism provides a way to upweigh important feature maps and downweigh the rest. In [25] authors show adaptive re-calibration of channel-wise feature responses by explicitly modelling inter-dependencies between channels using squeeze and excitation block on existing architectures [21, 58, 61] results in improved performance.

The proposed network architecture is distinct and combines the strengths of the U-Net architecture, Dense blocks, and Squeeze-and-Excitation (SE) blocks. This results in improved prediction accuracy and it has been shown to outperform other state-of-the-art network architectures such as the ones proposed in [26] which have a much higher number of learning parameters on the INRIA benchmark dataset. Figure 4.2 shows a diagram of the proposed feature recalibrated Dense block with 4 convolutional layers and a growth rate $\kappa = 12$ used by the ICT-Net. The proposed network has 11 feature recalibrated dense blocks with [4,5,7,10,12,15,12,10,7,5,4] number of convolutional layers in each dense block, respectively.

Perhaps the closest architecture to the one proposed was discussed in section 2.2.4 [30] which uses 103 convolutional layers. If SE blocks are introduced at the output of every layer this will cause a vast increase in the number of parameters which will hinder the training. In contrast, in our work we have chosen to include an SE block only at the end of every Dense block in order to re-calibrate the accumulated feature-maps of all preceding layers. Thus, the variations in the information learned at each layer - in the form of the features maps - are weighted by the SE block according to their importance as determined by the loss function.

Discussion: To verify the validity of the above design decisions we performed a comparative study involving a number of state-of-the-art architectures and blocks. Following the same training procedure for all architectures reported, and without any data augmentation the ICT-Net was compared with U-Net [50] and Tiramisu-103 [30]. The results on the validation dataset are shown in Table 4.1 where it is evident that the proposed architecture outperforms both U-Net and Tiramisu-103.

Paper	Method	Overall IoU (%)	Overall Accuracy (%)
[50]	UNet	70.86	95.51
[30]	Tiramisu-103	73.91	95.71
Ours	ICT-Net	**75.5**	**96.05**

Table 4.1: Performance evaluation of SOTA architectures (U-Net [50] and Tiramisu-103 [30]) on the validation dataset

4.3 Training and Validation on INRIA dataset

The network is trained on 155 tiles each with resolution 5000×5000 from the available training data with their corresponding ground truth. The training is performed for 100 epochs on a single nVidiaGTX 1080Ti. We used Tensorflow API for the development and training/testing of the network. Due to the large size of the dataset it requires approximately 6 hours to complete 1 epoch of training. Every epoch was divided into 31 sub-epochs each consisting of 5 tiles (1 from each city). Limited by GPU memory we had to choose a small batch size of 4 to have a comparatively larger patch size of 256×256 as we observed context is very important for semantic labeling of buildings.

Implementation details: The network was trained using cross-entropy loss with RMSProp Optimizer with an initial learning rate of 0.001 and decay of 0.995 for the first 50 epochs. After the 50^{th} epoch the learning rate was reduced to 0.0001 and trained for another 50 epochs. Instead of using dropout as a regularization technique we applied a large number of data augmentations in order to restrict the network from overfitting to the training dataset.

Data input: Our network takes in patches of 256×256 out of the entire tile with 50% overlap. The patches are selected sequentially for every odd epoch and the same number of patches is selected randomly for every even epoch during the training. We use the alternating patch generation strategy to restrict the network from overfitting while still having the opportunity to learn all the features from every tile. At testing the input patch size is increased to 768×768 (the maximum that could

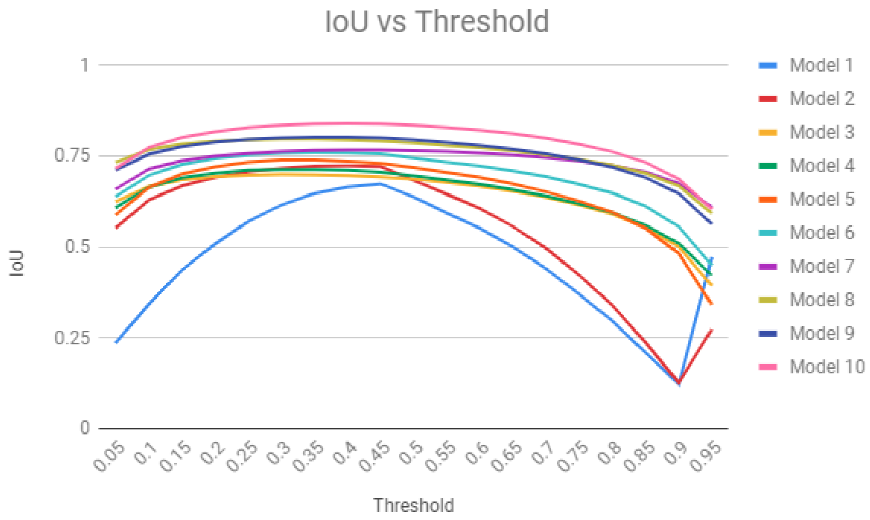


Figure 4.3: Empirical study to determine the optimal thresholding value for converting the grayscale classification map produced by the network to a binary map. The models shown correspond to the same network ICT-Net at different training snapshots for which the classification accuracy (i.e. IoU in the graph) was calculated **after** the thresholding at every 0.05 intervals as shown. The optimal threshold value is $\tau = 0.4$.

fit in the GPU memory) so that we are able to increase the context for large building in every patch. During testing, the patches are selected using 50% overlap similar to what is done during training.

Network output: The output produced by the network is a 1-channel gray-scale image of the same size as the input image where each pixel has a probability score of being a building in the range $[0, 1]$. We convert the probability map into a binary mask by thresholding. We conducted an empirical study on the validation dataset and have chosen $\tau = 0.4$ as the optimal threshold value for converting the gray-scale image to a binary map as shown in Figure 4.3. The output patches are then assembled into tiles of size 5000×5000 by weighted average and overlapping areas near the edges are down-weighted. During the testing, the standard test time augmentations are applied to each tile and they are merged back using an average of the probability scores.

ICT-Net achieves 75.50% IoU and 96.05% accuracy on the overall validation dataset of INRIA.

Data augmentations: Based on the validation results we used the pretrained weights and trained our network with the following data augmentations with a probability of 70% to be applied to every patch: random rotations in the range $[0°, 360°]$ using reflection padding, random flip, random selection of a patch in the range of $[0.75, 1.25]$ of the image patch size and re-size it to original patch size of 256. Data augmentations significantly improved the performance of the network in terms of accuracy.

4.4 Results - INRIA dataset

The INRIA dataset uses two main performance measures: Intersection over Union (Jaccard index) and Accuracy. Intersection over Union (IoU) is defined as the number of pixels labeled as buildings in both the prediction and the reference, divided by the number of pixels labeled as buildings in the prediction or the reference. Accuracy is defined as the percentage of correctly classified pixels.

The measures are calculated by the organizers of the competition and involve the classification of 5 cities for which no images have been used for training and validation, and for which no ground truth is available to the participants. As of Nov 2019 the proposed architecture is ranked as the top performing in terms of both IoU (80.32%) and accuracy (97.14%) on the competition's leaderboard [1] since February 2019. Figure 4.4 shows an example of a result for a small area of an image from the test dataset (top left). The probability image produced by the network is shown as a heat map (bottom right) overlaid on top of the RGB image (bottom left). The binary map resulting after the thresholding is shown in the top right image.

> ICT-Net achieves 80.32% IoU and 97.14% accuracy on the overall test dataset of INRIA, which is 1.5% IoU above the second best on the leaderboard.

As previously mentioned, the proposed network is currently ranked as the top performing network with the second best having more than 1.5% difference in terms of the IoU. The authors in [26] provide details of the next 4 top performing techniques on the INRIA aerial image labeling benchmark dataset. All 4 methods are Convolutional Neural Networks(CNNs), amongst which 3 of them are based on U-Net architecture. Table 4.2 shows a quantitative comparison between the proposed network ICT-Net and these other techniques on the test dataset as reported by the competition organizers.

[1] `https://project.inria.fr/aerialimagelabeling/leaderboard/`

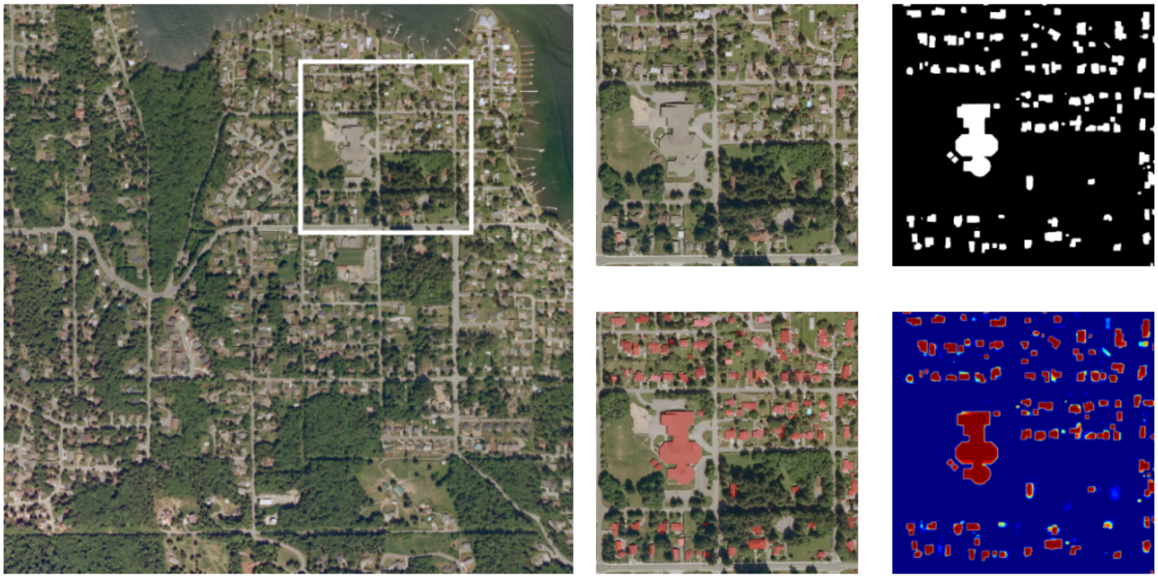


Figure 4.4: Building Classification of Bellingham city, tile 17. Result for an image from the test dataset. Top left: A closeup of a small area of an input image. Bottom left: The probability image overlaid on top of the RGB image. Bottom right: The probability image shown as a heat map. Top right: The binary map resulting after the thresholding of the probability image.

Paper	Method	Overall IoU	Overall Accuracy
[26]	Raisa	69.57	95.30
[26]	ONERA	71.02	95.63
[26]	NUS	72.45	95.90
[26]	AMLL	72.55	95.91
[29]	N/A	78.31	96.76
[29]	N/A	78.39	96.84
[29]	N/A	78.45	96.74
[29]	N/A	78.80	96.91
Ours	ICT-Net	**80.32**	**97.14**

Table 4.2: Performance evaluation of the best performing networks on the test dataset. First 4 entries are defined as State of the art by INRIA [26]. Next 4 entries are other top performances from the leaderboard [29]. ICT-Net outperforms all others with more than 1.5% difference in terms of the IoU.

Below we provide a brief overview of the main characteristics of these four other networks. Stacked U-Nets by **Raisa** [32] uses a U-Net based architecture where instead of using a single U-Net they use a stack of two U-Nets arranged end-to-end. The second network works as a post-processor for the previous one to enhance its predictions. The network uses a loss function that combines both binary cross entropy and a differential form of Intersection-over-Union (IoU) [42].

Signed distance transform regression by **ONERA** - uses a standard fully convolutional network [1] architecture, which is adapted to include spatial context in the optimization process by adding a regularization loss computed on the Euclidean signed distance transform (SDT) [62]. The network also outputs a regression of the SDT along with the standard classification.

The authors of Dual Resolution U-Net (**NUS**) propose dual resolution images as input to the U-Net architecture with a combined loss of sigmoid cross-entropy and soft Jaccard loss [42]. One high-resolution 384×384 patch from the original image and a crop of 768×768 with the same center and down-sampled to a twice lower resolution 384×384 image is fed to the network. Features from both high and low resolution patches are extracted by a U-Net, then score maps for each resolution are computed. A weight map is further learned on merging score maps from different resolutions. This weight map determines, for each pixel, how much the network relies on different resolution inputs. To summarize, the final result is a weighted sum of dual-resolution score maps.

Applied Machine Learning Lab **AMLL** at Duke University - proposed the use of the original U-Net architecture with half as many filters at each layers to reduce the chances of overfitting to the training dataset. To reduce poor performance of the network at the edge of the patches the patch size was increased to 2636×2636 during inference time.

City	2^{nd} Best IoU (%)	2^{nd} Best Accuracy (%)	Our IoU (%)	Our Accuracy (%)
Bellingham	74.15	97.44	**74.63**	**97.47**
Bloomington	75.55	97.72	**80.80**	**98.18**
Innsbruck	78.62	97.43	**79.50**	**97.58**
San Francisco	80.65	93.63	**81.85**	**94.08**
East Tyrol	80.80	98.31	**81.71**	**98.39**
Overall	*78.80*	*96.91*	***80.32***	***97.14***

Table 4.3: Comparison of performance evaluation of ICT-Net and the previous best entry on the leaderboard for the test dataset.

We compare the performance of ICT-Net with the 2^{nd} best entry on the leaderboard and observe that ICT-Net outperforms the previous best entry in overall(average over 5 cities) as well as in every city. This demonstrates the generalization capacity of our proposed ICT-Net. The comparison results can be found in table 4.3.

4.5 Training and Validation on AIRS dataset

Initially we use the pre-trained ICT-Net trained on INRIA dataset to test the generalization ability of the network on AIRS validation dataset of 94 tiles. Then the network is trained on 857 tiles of AIRS training dataset where each tile is of resolution 10000×10000 with their corresponding ground truth. The training is performed for 5 epochs on a single nVidiaGTX 1080Ti. Due to huge size of the dataset it requires approximately 30 hours to complete 1 epoch of training. Every epoch was divided into sub-epochs where each sub-epoch consisting of 5 tiles from the training dataset. We use the same batch and patch size of 256×256 and 4 respectively as we used for INRIA dataset.

Implementation details: Loss remain the same as training for INRIA dataset. The optimization hyper-parameters mostly remain same and the learning rate used was 0.0001. The network was trained until the loss saturated which took 5 epochs. Data augmentations were used in order to restrict the network from overfitting to the training dataset.

Data input: ICT-Net takes in patches of 256×256 out of the entire tile with 50% overlap. At testing the input patch size in increased to 768×768 to increase the context for large building in every patch. The patch generation strategies remain the same as was used for INRIA dataset.

Network output: The output produced by the network is a 1-channel gray-scale image of the same size as the input image where each pixel has a probability score of being a building in the range $[0, 1]$. We convert the probability map into a binary mask by thresholding. The output patches are then assembled into tiles of size 10000×10000 by weighted average and overlapping areas near the edges are down-weighted.

Data augmentations: On the AIRS dataset we use the same set of data augmentation techniques that were used while training ICT-Net on INRIA dataset. The details for training on the INRIA dataset can be found in 4.3.

4.6 Results - AIRS dataset

The AIRS dataset uses the following performance measures: Intersection over Union (Jaccard index), F1-Score, Precision and Recall. The evaluation metrics of intersection over union (IoU) and F1-score are used to reflect the overall performance of the baseline methods. On the other hand, precision and recall indicate the correctness and completeness of the roof segmentation results respectively.

ICT-Net achieves 91.70% IoU and 95.70% F1-score on the test dataset of AIRS, which is the highest accuracy so far; 1.8% IoU, and 1% F1-score above the second best on the leaderboard.

The evaluation provided by the organizers of the competition involves the segmentation of roofs for 95 tiles of imagery for which no images have been used for training and validation, and no ground truth is available to the participants. As of Nov 2019 the proposed architecture is ranked as the top performing in terms of both IoU (91.70%) and F1-score (95.70%) on the competition's leaderboard [2]. Figure 4.5 shows an example of a result for a small area of an image from the test dataset (top left). The probability image produced by the network is shown as a heat map (bottom right) overlaid on top of the RGB image (bottom left). The binary map resulting after the thresholding is shown in the top right image.

As already mentioned, the proposed network is currently ranked as the top performing network with the second best having 1.8% difference in terms of the IoU and 1% difference in terms of F1-score. The authors in [9] provide details of the top performing techniques on the AIRS (Aerial Imagery for Roof Segmentation) benchmark dataset. All of the methods are Convolutional Neural Networks based approaches(CNNs). Table 4.4 shows a quantitative comparison between the proposed network ICT-Net and these other techniques on the test dataset as reported by the competition organizers.

Paper	Method	IoU	F1-score	Precision	Recall
[37]	FPN	0.882	0.937	**0.963**	0.913
[9]	FPN+MSFF	0.888	0.941	0.958	0.924
[65]	PSP	0.899	0.947	0.961	0.933
[8]	ICT-Net	**0.917**	**0.957**	0.955	**0.959**

Table 4.4: Performance evaluation of the top performing networks on the AIRS test dataset. ICT-Net outperforms all others with 1.8% difference in terms of the IoU and 1% in terms of F1-Score.

[2] https://www.airs-dataset.com/leaderboard/

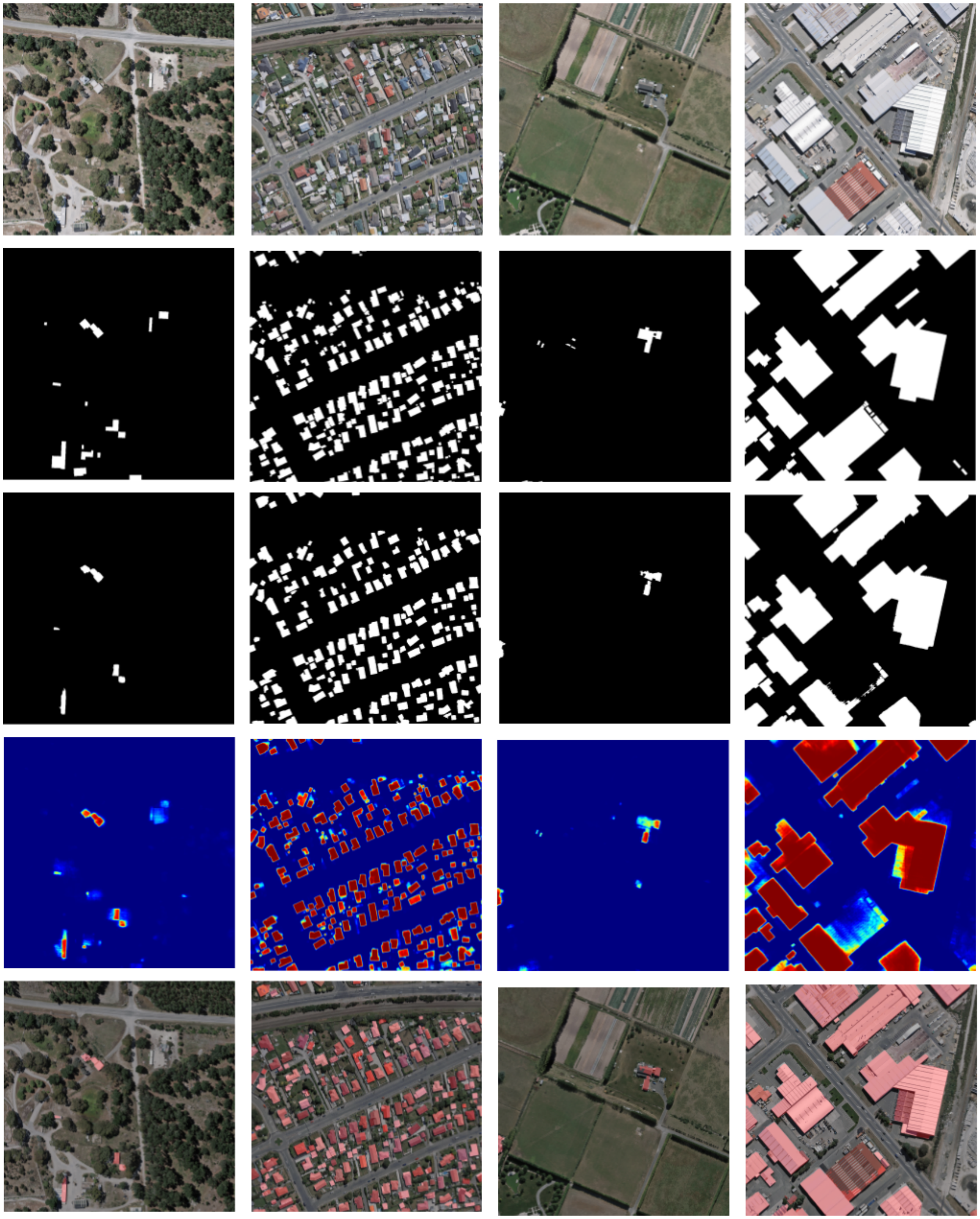


Figure 4.5: Building Classification of AIRS sample data. 1^{st} row: sample RGB input image. 2^{nd} row: ground truth label for the sample images. 3^{rd} row: The prediction from ICT-Net 4^{th} row: The probability map from ICT-Net shown as a heat map. 5^{th} row: The prediction from ICT-Net overlayed on the image.

Conclusion

We present ICT-Net: a novel network architecture which combines the strengths of U-Net, Dense blocks and feature recalibration using SE blocks. We evaluate the performance of ICT-Net on 2 benchmark datasets: INRIA and AIRS, and validate our design choices. The proposed network outperformed all other techniques on the competitions' leaderboard with more than 1.5% and 1.8% IoU difference from the second best entry on INRIA and AIRS benchmark datasets, respectively. Additional predictions from INRIA test dataset and Google imagery used by Bastani *et al.* [2] can be found here[3]. Our final goal is to have rapid and automatic 3D reconstruction of urban scene, so we took at a technique to reconstruct urban features (buildings in this case) and compare the classification accuracy to reconstruction accuracy in chapter 5.

[3] https://theictlab.org/lp/2019ICTNet/START_HERE.html

Chapter 5

3D Reconstruction

As previously stated the contributions of our work are two-fold. In chapter 4 we proposed a novel, top ranked architecture for classifying buildings from remote sensor imagery. This binary classification map is typically used as a first step to the reconstruction process since it allows the application of specialized reconstruction algorithms according to the classified type of the pixels. In this chapter we use a very well known technique by Poullis *et al.* [45] for reconstructing urban features (buildings in this case) using an RGB orthophoto image, the depth map and the building prediction mask from ICT-Net. Using the same technique on INRIA dataset we analyze the relation between classification accuracy and the accuracy of the reconstruction, with by boundary localization as a proxy to depth (since depth information was not available for INRIA dataset).

5.1 Dataset

For the purpose of 3D reconstruction of buildings using [45] we need Orthorectified RGB imagery, ground truth or prediction masks of building and the depth map of the image tile. We were able to acquire one tile of 30cm orthophoto RGB imagery of downtown Montreal (image courtesy of Defence Research and Development Canada and Thales Canada) and a depth map for the same tile (depth map courtesy Presagis Inc Canada). Using our proposed ICT-Net architecture we were able to predict the building masks for the same tile, an illustration of building segmentation mask can be see in figure 5.1.

Figure 5.2 shows an example of the downtown Montreal. The building classification is generated

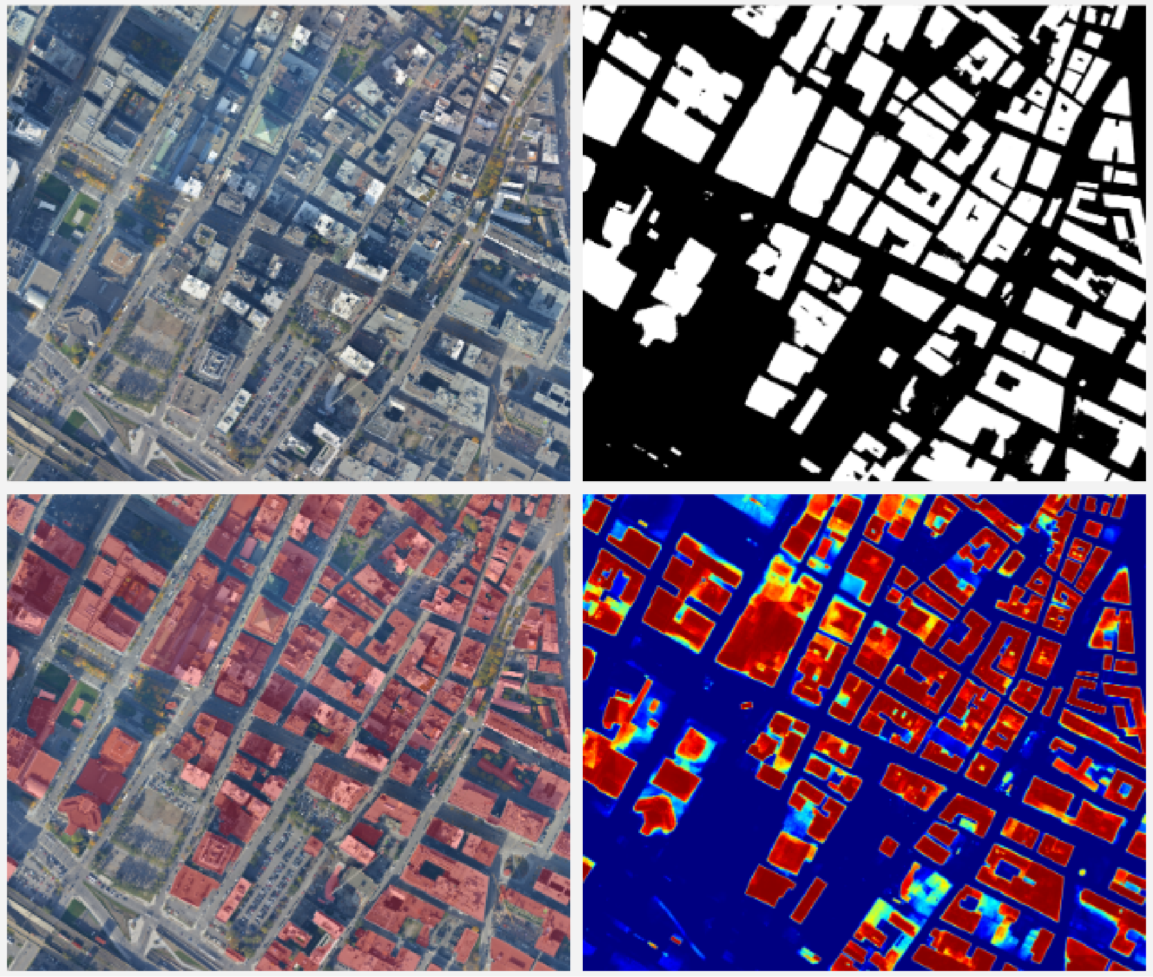


Figure 5.1: Building Classification of Montreal data. Top left: The orthophoto RGB input image. Bottom left: The prediction of ICT-Net overlaid on top of the RGB image(Red-Prediction). Bottom right: The probability mask of ICT-Net shown as a heat map. Top right: The binary segmentation map resulting after the thresholding of the raw ICT-Net output. **The orthophoto RGB image is courtesy of Defence Research and Development Canada and Thales Canada.**

with the proposed ICT-NET network and refined as explained above. In this example, LiDAR information was available which after resampling at the same resolution as the orthorectified image was used to extrude the 3D buildings from the extracted boundaries. The result shown is fully automated and no post-processing was performed. It should be noted that no images of the city of Montreal have been used in the training. We have manually evaluated the result by counting the number of buildings and confirming that all of them have been classified correctly by the network and therefore reconstructed. The accuracy of the classification is also evident from the fact that

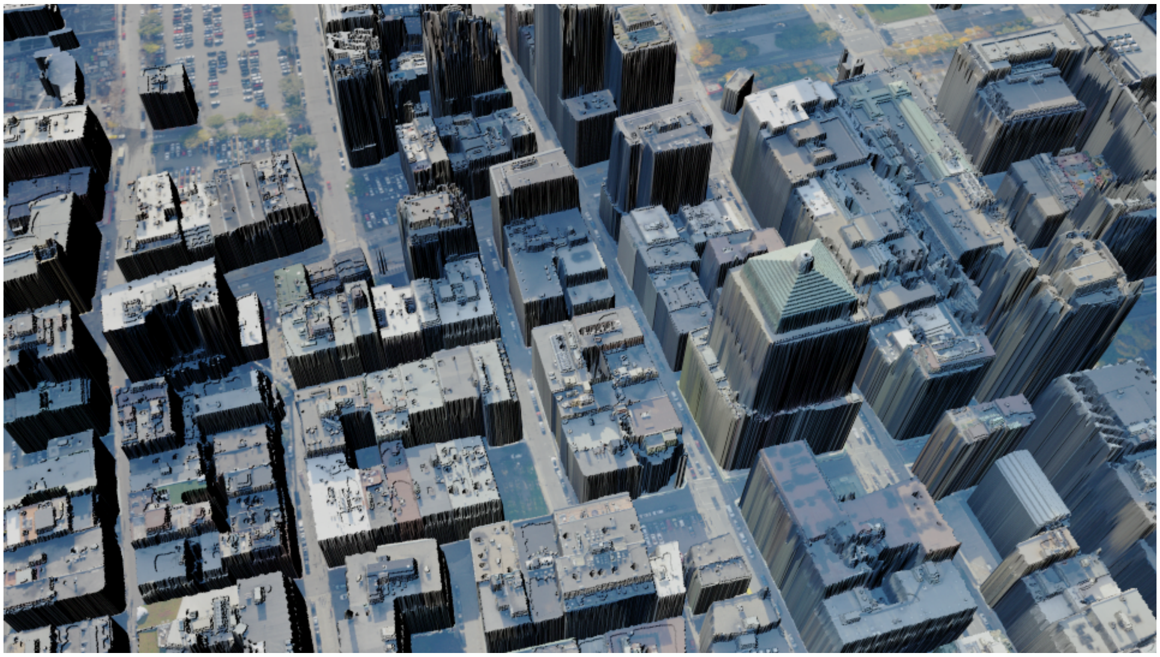


Figure 5.2: A fully automated result without any post-processing. Downtown Montreal for which no training images were used and no ground truth is available. Classification by ICT-Net and reconstruction by extruding the extracted boundaries of the buildings using the LiDAR pointcloud corresponding to the same area. The elevation of all non-building points is set to zero. All buildings have been manually verified that they are correctly classified. The accuracy of the classification can also be visually verified since there is no "bleeding" between the buildings and any other urban features e.g., roads, trees, cars, etc. **The Depth map for Montreal data was provided by Presagis Inc Canada.**

there is no "bleeding" between the buildings and any other urban features e.g., roads, trees, cars, etc in the final result. A fly-through video animation of our 3D reconstructed downtown Montreal can be found at our website [1].

5.2 Methodology

Since it is extremely difficult to acquire building blueprints or CAD models for such large areas, and no 3D/depth information is available as part of the benchmark dataset we posit that the building boundaries extracted from the classification binary map can serve as a *proxy* to the quality of the reconstruction since the boundaries are typically extruded in order to create the 3D models corresponding to the buildings. More specifically, the procedure for quantitatively evaluating the

[1] https://theictlab.org/lp/2019ICTNet/

accuracy of the reconstruction is as follows:

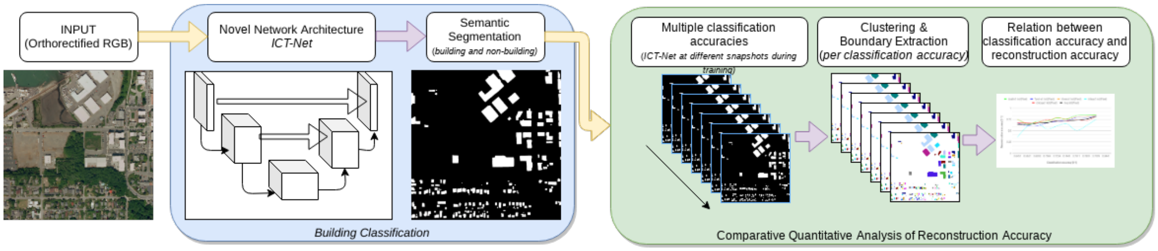

INPUT
(Orthorectified RGB)
Novel Network Architecture
ICT-Net
Semantic
Segmentation
(building and non-building)
Building Classification
Multiple classification
accuracies
(ICT-Net at different snapshots during
training)
Clustering &
Boundary Extraction
(per classification accuracy)
Relation between
classification accuracy and
reconstruction accuracy
Comparative Quantitative Analysis of Reconstruction Accuracy

Figure 5.3: The diagram summarizes the work presented in this paper. Firstly, we focus on the building classification and propose a novel network architecture which outperforms state-of-the-art on benchmark datasets and is currently top-ranking. Secondly, we investigate the relation between the classification accuracy and the reconstruction accuracy and conduct a comparative quantitative analysis which shows a strong correlation but also a consistent and considerable decrease of the reconstruction accuracy when compared to the classification accuracy.

- Building boundaries B_g are extracted from the ground truth provided as part of the training dataset.

- The RGB image corresponding to the ground truth above is used as input to the ICT-Net. The binary classification map C_b resulting from feeding forward the RGB image classifies pixels into buildings and non-buildings.

- The binary classification map C_b is refined $C_b^{refined}$ using a CRF-based technique where an energy function is minimized via graph-cut optimization for finding an optimal labeling f_p for every pixel p such that $f_p \rightarrow l$, where l is the new label. The data term of the energy function of a pixel p with label l_{p_i} is defined as,

$$E_d = \begin{cases} 10, & \text{if } f(p_i) \neq l_{p_i} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The smoothness term of the energy function of two neighbouring pixels p_1 and p_2 with labels l_{p_1} and l_{p_2} respectively is defined as,

$$E_s = \begin{cases} 20, & \text{if } l_{p_1} == l_{p_2} \text{and } f(p_1) \neq f(p_2) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The values of 10 and 20 in the equations were selected such that smoothness is favored over the observed data.

- Building boundaries B_b are extracted from the refined classification map $C_b^{refined}$. A simplification process i.e. Douglas-Pecker approximation with a tolerance of $\tau = 0.5$, is applied to the boundaries. This simplification process is a step applied to the building boundaries prior to extruding the 3D model if 3D/depth information is available [48], [46], [47].
- The simplified boundaries B_b^{approx} are finally converted back to a binary classification map and quantitatively compared to the ground truth B_g. This comparison involves IoU metrics on (i) a per-pixel and (ii) a per-building bases. In the case of the per-building IoU metric, a true positive is considered only if a building has at least 75% of its pixels overlap the pixels of the same building in the ground truth.

5.3 Comparative Quantitative Analysis of Reconstruction Accuracies

The procedure described above is followed for all input images with no changes to the values and thresholds used; the only varying condition is the classification accuracy. In our experiments, the input images are processed by the proposed ICT-Net at different training snapshots having different classification accuracies. Thus, multiple binary classification maps were produced each with a different classification accuracy.

Table 5.1 shows the quantitative results of the comparison. A total of 5 cities were processed using the aforementioned procedure. Figures 5.4 and 5.5 show the relation between the reconstruction accuracy with respect to the classification accuracy. We have used increasing classification accuracies based on the same architecture (ICT-Net) at different snapshots during the training. Using the binary classification maps we have followed the aforementioned procedure which is typical to the reconstruction process. Two metrics have been used to assess the reconstruction accuracy, namely per-pixel IoU and per-building IoU (with 75% threshold for being considered a true positive). As expected, the graph shows a strong correlation between the classification accuracy and the reconstruction accuracy. However the reconstruction accuracy is consistently lower than the classification

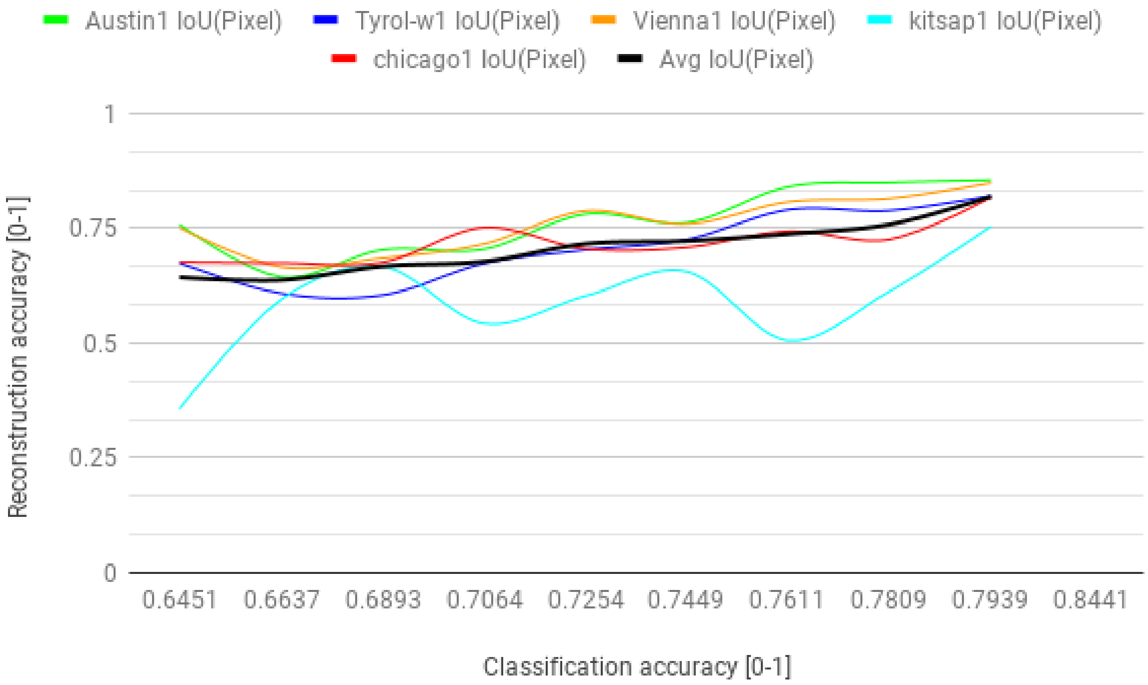


Figure 5.4: Reconstruction vs Classification accuracy (per-pixel IoU). The classification accuracy ranges from [0.6451, 0.8441] as calculated on the validation test. There is an average decrease of 4.43% ± 1.65% (confidence level 95%) in per-pixel IoU of the reconstruction accuracy. The reported averages are calculated across the accuracy levels.

Classification Accuracy	**Austin1 IoU**		**Tyrol-W1 IoU**		**Vienna1 IoU**		**Kitsap1 IoU**		**Chicago1 IoU**		**Average IoU**	
	per-pix.	**per-bldg**	**per-pix.**	**per-bldg**	**per-pix.**	**per-bldg**	**per-pix.**	**per-bldg**	**per-pix.**	**per-bldg**	**per-pix.**	**per-bldg**
0.6451	0.7038	0.5004	0.4683	0.1887	0.7291	0.3880	0.1063	0.02384	0.6445	0.4952	0.5304	0.3192
0.6637	0.7583	0.7583	0.6749	0.4213	0.7514	0.4776	0.3575	0.1354	0.6765	0.6481	0.6437	0.4881
0.6893	0.6443	0.3451	0.6084	0.2944	0.6660	0.3080	0.5949	0.3770	0.6747	0.5734	0.6377	0.3796
0.7064	0.7034	0.5240	0.6046	0.2780	0.6855	0.3728	0.6671	0.3704	0.6760	0.5420	0.6673	0.41745
0.7254	0.7049	0.5520	0.6735	0.4325	0.7160	0.4820	0.5432	0.3194	0.7516	0.7386	0.6778	0.5049
0.7449	0.7812	0.6926	0.7032	0.4643	0.7881	0.5829	0.6026	0.3230	0.7056	0.7011	0.7162	0.5528
0.7611	0.7630	0.5976	0.7256	0.4850	0.7597	0.5128	0.6561	0.4286	0.7088	0.7336	0.7226	0.5515
0.7809	0.8408	0.7914	0.7907	0.5756	0.8078	0.5879	0.5059	0.3973	0.7436	0.7782	0.7378	0.6261
0.7939	0.8498	0.7936	0.7891	0.6016	0.8153	0.6202	0.6131	0.4328	0.7259	0.7703	0.7586	0.6437
0.8441	0.8549	0.8073	0.8212	0.6634	0.8490	0.6519	0.7541	0.5714	0.8179	0.8050	0.8194	0.6998

Table 5.1: The ICT-Net at different training snapshots having different classification accuracy vs the reconstruction accuracy measured using two metrics: per-pixel IoU, and per-building IoU (with a threshold of 75% overlap for true positives)

accuracy by an average of 4.43% ± 1.65% (confidence level 95%) on the per-pixel IoU and an average of 21.7%± 4.21% (confidence level 95%) on the per-building IoU. This discrepancy can be

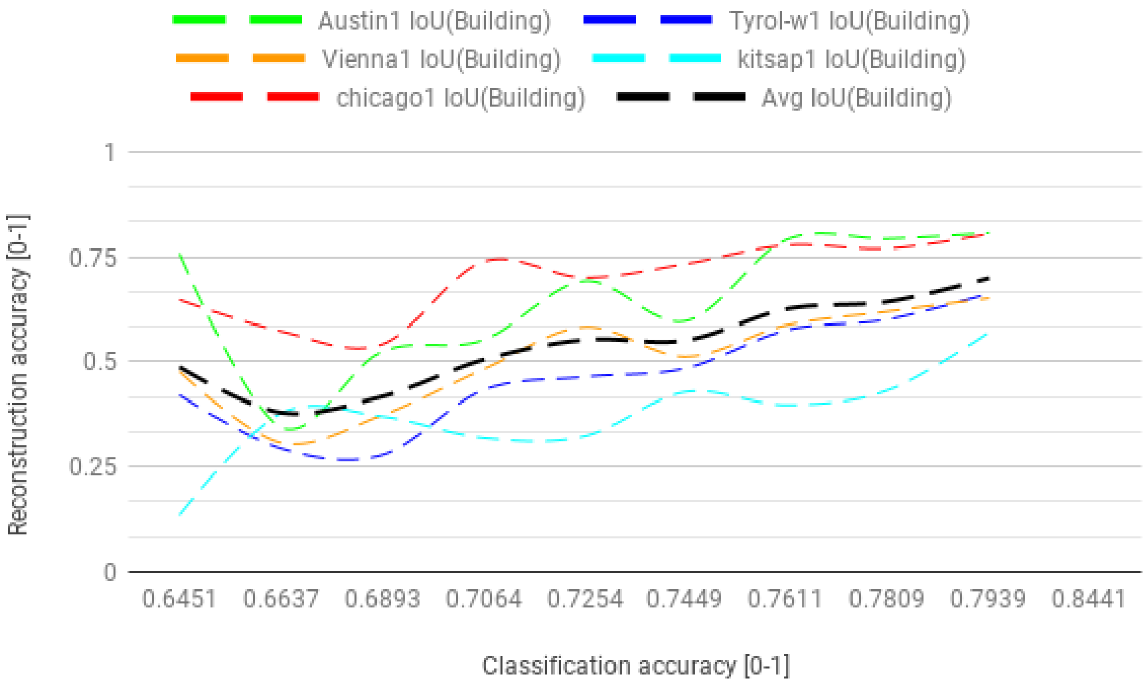


Figure 5.5: Reconstruction vs Classification accuracy (per-building IoU). The classification accuracy ranges from [0.6451, 0.8441] as calculated on the validation test. There is an average decrease of an average decrease of 21.7%± 4.21% (confidence level 95%) in per-building IoU of the reconstruction accuracy. The reported averages are calculated across the accuracy levels.

attributed to the fact that the ground truth images used for training the network may contain errors and are in most cases manually created which results in much higher classification accuracy than the reconstruction accuracy. Moreover, the high discrepancy on the per-building IoU can be attributed to the fact that a threshold must be used i.e. 75%, when calculating the true positives.

Conclusion

The results of this analysis clearly indicate *that high classification accuracy does not translate into high reconstruction accuracy*. More importantly though, the results of the analysis clearly indicate that the reconstruction accuracy must be taken into account as part of the loss function along with the classification accuracy during the training of the network.

Chapter 6

Conclusion and Future work

In this work we investigated different network architectures including a capsule based neural network to design a robust and generalizable building segmentation architecture. We justify the design choices made for our proposed network and validate our design choice using 2 benchmark datasets. We also investigated the relation between the classification accuracy and the reconstruction accuracy for 3D of reconstruction of building in urban scene.

6.1 Concluding Remarks

We have presented a novel network which combines the strengths of state-of-the-art techniques like Dense blocks in fully convolutional networks and feature recalibration using SE blocks. We have identified the requirements for the particular task and based our decisions on the actual characteristics and observations. We have shown that the proposed architecture outperforms other state-of-the-art including ensemble techniques.

Furthermore, we investigated the relation between the classification accuracy and the reconstruction accuracy. Due to the extreme difficulty of acquiring blueprints for such large areas and the unavailability of 3D information we have used the building boundaries as a proxy to the reconstruction accuracy. The proposed ICT-Net at different training snapshots was used to generate binary maps of different classification accuracies which were then used for extracting the boundaries. We presented a comparative quantitative analysis which shows a strong correlation between the two but also a consistent and considerable decrease of the reconstruction accuracy when compared to the classification accuracy.

6.2 Future Work

With respect to the future work, we plan on extending this work to (i) the classification of multiple urban feature types with limited amount of labeled data available with techniques like [11, 55], (ii) exploit the feature space used for building classification to find building properties like roof type, roof materials, etc [3, 5, 13] and (iii) conduct a comparative quantitative analysis using ground-truth 3D information acquired by LiDAR and manually processed.

Bibliography

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015.

[2] Favyen Bastani, Songtao He, Sofiane Abbar, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, David J. DeWitt, and Sam Madden. Unthule: An incremental graph construction process for robust road map extraction from aerial images. *CoRR*, abs/1802.03680, 2018.

[3] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3), June 2008.

[5] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*. Springer, 2016.

[6] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, New York, NY, USA, 2007. ACM.

[7] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. *CoRR*, abs/1712.02616, 2017.

[8] Bodhiswatta Chatterjee and Charalambos Poullis. On building classification from remote sensor imagery using deep neural networks and the relation between classification and reconstruction accuracy using border localization as proxy. In *2019 16th Conference on Computer and Robot Vision (CRV)*. IEEE, 2019.

[9] Qi Chen, Lei Wang, Yifan Wu, Guangming Wu, Zhiling Guo, and Steven L. Waslander. Aerial imagery for roof segmentation: A large-scale dataset towards automatic mapping of buildings. *CoRR*, abs/1807.09532, 2018.

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] Adrian V. Dalca, Evan M. Yu, Polina Golland, Bruce Fischl, Mert R. Sabuncu, and Juan Eugenio Iglesias. Unsupervised deep learning for bayesian brain MRI segmentation. *CoRR*, abs/1904.11319, 2019.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[13] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *CoRR*, abs/1808.00033, 2018.

[14] Adam Van Etten, Dave Lindenbaum, and Todd M. Bacastow. Spacenet: A remote sensing dataset and challenge series. *CoRR*, abs/1807.01232, 2018.

[15] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2), June 2010.

[16] Timothy Forbes and Charalambos Poullis. Deep autoencoders with aggregated residual transformations for urban reconstruction from remote sensing data. *2018 15th Conference on Computer and Robot Vision (CRV)*, 2018.

[17] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and José García Rodríguez. A review on deep learning techniques applied to semantic segmentation. *CoRR*, abs/1704.06857, 2017.

[18] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2014.

[19] Timothy L Haithcoat, Wenbo Song, and James D Hipple. Building footprint extraction and 3-d reconstruction from lidar data. In *Remote Sensing and Data Fusion over Urban Areas, IEEE/ISPRS Joint Workshop*. IEEE, 2001.

[20] Ryuhei Hamaguchi and Shuhei Hikosaka. Building detection from satellite imagery using ensemble of size-specific detectors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[22] Marti A. Hearst. Support vector machines. *IEEE Intelligent Systems*, 13(4), July 1998.

[23] Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. Transforming auto-encoders. In Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[24] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. 2018.

[25] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *CoRR*, abs/1709.01507, 2017.

[26] Bohao Huang, Kangkang Lu, Nicolas Audebert, Andrew Khalel, Yuliya Tarabalka, Jordan Malof, Alexandre Boulch, Bertrand Le Saux, Leslie Collins, Kyle Bradbury, et al. Large-scale semantic classification: outcome of the first year of inria aerial image labeling benchmark. In *IEEE International Geoscience and Remote Sensing Symposium–IGARSS 2018*, 2018.

[27] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[28] Vladimir I. Iglovikov, Selim S. Seferbekov, Alexander V. Buslaev, and Alexey Shvets. Ternausnetv2: Fully convolutional network for instance segmentation. *CoRR*, abs/1806.00844, 2018.

[29] Inria Aerial Image Labeling Benchmark. Inria Aerial Image Labeling Benchmark Leader-Board. `https://bit.ly/2GC88nr` (last accessed 18th Feb. 2019).

[30] Simon Jégou, Michal Drozdzal, David Vázquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. *CoRR*, abs/1611.09326, 2016.

[31] Andrew Khalel and Motaz El-Saban. Automatic pixelwise object labeling for aerial imagery using stacked u-nets. *CoRR*, abs/1803.04953, 2018.

[32] Andrew Khalel and Motaz El-Saban. Automatic pixelwise object labeling for aerial imagery using stacked u-nets. *arXiv preprint arXiv:1803.04953*, 2018.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, USA, 2012. Curran Associates Inc.

[34] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[35] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*. Springer, 1999.

[36] Yann LeCun, Fu Jie Huang, Leon Bottou, et al. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR (2)*. Citeseer, 2004.

[37] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[38] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[39] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[40] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2), November 2004.

[41] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre 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.

[42] Gellért Máttyus, Wenjie Luo, and Raquel Urtasun. Deeproadmapper: Extracting road topology from aerial images. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[43] Volodymyr Mnih. *Machine learning for aerial image labeling*. University of Toronto (Canada), 2013.

[44] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, USA, 2010. Omnipress.

[45] C. Poullis and S. You. Automatic reconstruction of cities from remote sensor data. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June.

[46] Charalambos Poullis. A framework for automatic modeling from pointcloud data. *IEEE transactions on pattern analysis and machine intelligence*, 2013.

[47] Charalambos Poullis. Large-scale urban reconstruction with tensor clustering and global boundary refinement. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2019.

[48] Charalambos Poullis and Suya You. Automatic creation of massive virtual cities. In *2009 IEEE Virtual Reality Conference*. IEEE, 2009.

[49] J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1), March 1986.

[50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[51] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 1958.

[52] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, 2017.

[53] Aaron K Shackelford, Curt H Davis, and Xiangyun Wang. Automated 2-d building footprint extraction from high-resolution satellite multispectral imagery. In *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, volume 3. IEEE, 2004.

[54] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1), Jan 2009.

[55] Mennatullah Siam and Boris N. Oreshkin. Adaptive masked weight imprinting for few-shot segmentation. *CoRR*, abs/1902.11123, 2019.

[56] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[57] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17. AAAI Press, 2017.

[58] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[59] Oliver Wang, Suresh K Lodha, and David P Helmbold. A bayesian approach to building footprint extraction from aerial lidar data. In *3DPVT, Third International Symposium on*. IEEE, 2006.

[60] Wikipedia. File:VGG neural network.png — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=File%3AVGG%20neural%20network.png&oldid=913770640`, 2019. [Online; accessed 06-November-2019].

[61] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016.

[62] Q. . Ye. The signed euclidean distance transform and its applications. In *[1988 Proceedings] 9th International Conference on Pattern Recognition*, May 1988.

[63] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

[64] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[65] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.