Machine Learning Enabled-Localization in 5G and LTE using Image Classification and Deep Learning

by

Hind Mukhtar

A thesis
submitted to the University of Ottawa
in partial fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Examining Committee

The following served on the Examining Committee for this thesis.

External Examiner:	Marc St-Hilaire Professor, Dept. of Systems and Computer Engineering, Carleton University
Internal Member(s):	Tet Yeap Professor, Dept. of Electrical Engineering and Computer Science, University of Ottawa
Supervisor(s):	Melike Erol-Kantarci Professor, Dept. of Electrical Engineering and Computer Science, University of Ottawa

Declaration of Authorship

I hereby certify that this thesis is entirely my own original work except where otherwise indicated. I am aware of the University's regulations concerning plagiarism, including those concerning consequent disciplinary actions. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Abstract

Demand for localization has been growing due to the increase in location-based services and high bandwidth applications requiring precise localization of users to improve resource management and beam forming. Outdoor localization has been traditionally done through Global Positioning System (GPS), however it's performance degrades in urban settings due to obstruction and multi-path effects, creating the need for better localization techniques. This thesis proposes a technique using a cascaded approach composed of image classification and deep learning using LIDAR or satellite images and Channel State Information (CSI) data from base stations to predict the location of moving vehicles and users outdoors. The algorithm's performance is assessed using 3 different datasets. The first two use simulated data in the Milli-meter Wave (mmWave) band and lidar images that are collected from the neighbourhood of Rosslyn in Arlington, Virginia. The results show an improvement in localization accuracy as a result of the hierarchical architecture, with a Mean Absolute Error (MAE) of 6.55m for the proposed technique in comparison to a MAE of 9.82m using one Convolutional Neural Network (CNN). The third dataset uses measurements from an LTE mobile communication system along with satellite images that take place at the University of Denmark. The results achieve a MAE of 9.45 m for the heirchichal approach in comparison to a MAE of 15.74 m for one Feed-Forward Neural Network (FFNN)

Acknowledgements

This work would not have been possible without the supervision of Professor Melike Erol-Kantarci, who has guided me throughout the program at the University of Ottawa and inspired me to advance my research. Her leadership, collaboration and patience has truly allowed me to enjoy every step of this process and encouraged me to explore the world of research, academia and particularly, AI-enabled wireless communications. I am extremely grateful to have had the opportunity to learn and grow from her supervision.

Secondly, I would like to give credit to the other masters and PhD students as well as post doctoral fellows at the Networked systems and Communications Research Lab (NETCORE) for their constructive feedback and contributions that have assisted me in completing my thesis.

I would also like to acknowledge Professor Tet Yeap for his phenomenal course in neural networks, which provided me with the necessary background and skills to develop the proposed methodology in my thesis and inspired me to pursue the use of deep learning. I believe that I speak for many student when I state that his course was engaging, interesting and challenging in all the right ways.

Finally, I would like to thank the University of Ottawa for awarding me the Masters Admissions Scholarship, which has funded my studies and research.

Dedication

This thesis is dedicated to my parents, Eiman Ahmed and Mustafa Mukhtar, who have been a constant source of inspiration and support throughout my life. Their commitment and hard work has been an example for me to work hard for the things I aspire to achieve.

Table of Contents

List of Tables

List of Figures

Glossary

adaptive modulation Selection of coding rate and modulation scheme based on the channel quality [21] 8

AI-enabled wireless communication The concept of applying artificial intelligence and machine learning to solve intractable problems or optimize protocols and algorithms in wireless communication networks using large amounts of data [49] 5

beam forming Controlling the shape and direction of the antenna's signal beam using phased antenna arrays by interfering and constructing patterns [42] 8

beam steering Dynamically altering the antenna's signal beam by changing the phase in real time [42] 8

core network The part of the telecommunications network that facilitates routes between nodes to exchange information between sub-networks [5] 8

demodulation The process of extracting the original signal from the carrier 5

modulation The process of encoding data into a carrier signal to enable transmission of information 5

NodeB The term for a particular radio base station node in a telecommunications network. It's function is to provide communication between the user equipment and communication network 6, 56, 57

radio access network The network infrastructure in a mobile communications network that wirelessly connects users to the network. The basic structure consists of the radio unit, baseband unit and antennas. [5] 8

tensorflow An open-source platform for machine learning 52

Abbreviations

ABS Anti-Lock Braking System 17

ANN Artificial Neural Network 23

AoA Angle of Arrival 1, 2, 9, 17, 23, 52, 74, 77

AoD Angle of Departure 2, 17, 23, 52, 74, 77

AP Access Point 9, 18, 19, 29

BLE Bluetooth Low Energy 19

CNN Convolutional Neural Network iv, 3, 11, 13, 17, 21–23, 27, 28, 43, 44, 51, 52, 59–62, 65, 72–77, 84

CRS Cell-Specific Reference Signal 21

CSI Channel State Information iv, 3, 21, 23, 25, 27, 42, 43, 65, 76, 77

CU Central Unit 57

CVM-Net Cross-View Matching Network xi, 28, 29

DU Distributed Units 57

eMBB enhanced-Mobile Broadband 29

FFNN Feed-Forward Neural Network iv, xii, 3, 11, 17, 21, 27, 43, 52, 54, 55, 64–66, 69–73, 75–77, 85

GPS Global Positioning System iv, 1, 2, 4, 8, 9, 17, 18, 20, 27, 31, 74, 76

IoT Internet of Things 7, 27

ITU International Telecommunications Union 6

KNN K-Nearest Neighbors 14, 21, 23, 25

KPI Key Performance Indicators 6

LDWS Lane Departing Warning System 17

LoS Line of Sight 20, 21

LR Linear Regression 15, 21, 23, 25

LSTM Long-Short Term Memory 13, 14, 17, 21–23

MAE Mean Absolute Error iv, 58–61, 64, 65, 70–72, 74, 76

MIMO Multiple Input, Multiple Output xi, 4, 6, 8, 23, 25, 26, 32, 42, 72, 75–78

mmWave Milli-meter Wave iv, 2, 8, 21, 27, 32, 42, 75

MSE Mean Squared Error 15, 55

NVF Network Function Virtualization 8

OFDM Orthogonal Frequency Division Multiplexing 6

PCI Physical Cell ID 38

R-CNN Region Based Convolutional Neural Network 13, 27

RAN Radio Access Network 57

ReLU Rectified Linear Unit 52

RF Radio Frequency 27

RMSE Root Mean Squared Error 58–61, 64, 65, 70–72, 74, 76

RNN Recurrent Neural Network 13

RP Reference Point 18, 19, 75

RSRP Reference Signal Received Power xii, 38, 52, 65, 67, 77

RSRQ Reference Signal Received Quality 38, 52, 77

RSS Received Signal Strength xii, 1–3, 9, 17, 19, 20, 22, 23, 31, 35, 36, 43, 46, 52, 61, 74, 77

RSSI Reference Signal Strength Indicator 18, 19, 22, 38, 41, 48, 52, 65, 77

SDT Software Defined Terminal 21

SINR Signal-to-Interference-Plus-Noise Ratio 38, 52, 65, 77

SISO Single Input, Single Output 3, 32, 40, 43, 54, 64, 65, 71, 75–77

SSIM Structural Similarity Index Matrices 15, 47

SVM Support Vector Machine 13

SVR Support Vector Regression 23

ToA Time of Arrival 1, 2, 9, 17, 23, 52, 74, 77

UE User Equipment 6, 57

URLL Ulta-Reliability Low Latency 29

VLAD Vector of Locally Aggregated Descriptors 28

WLAN Wirelsss Local Area Network 5, 22

WSN Wireless Sensor Networks 19

Chapter 1

Introduction

1.1 Motivation

Accurate localization of users, devices and vehicles has been an active research area with outcomes impacting many applications such as marketing, self-driving cars, smart parking and so on. Localization is simply the process of retaining a user's location. Traditionally, outdoor localization is done through GPS. GPS has good accuracy for unobstructed-sky however its performance degrades in urban canyons as a result of severe obstruction and multipath effects. On the other hand, indoor localization can be done using Wi-Fi or Bluetooth signals, using three main methods: Trilateration, fingerprinting and proximity. Trilateration is very similar to GPS; in that it uses three transmitters with known locations to find the receiver [31]. This method uses information such as RSS, Time of Arrival (ToA), Angle of Arrival (AoA) or delay. The fingerprinting approach collects information regarding received signals and their locations during the online or training phase and compares RSS in the offline phase to estimate the location [10]. Finally, proximity methods report when devices are in range, and estimate the location of the device based on their closeness to known locations. Although effective, these techniques lack accuracy and rely on the user being connected to Wi-Fi or Bluetooth, or within range to nearby sensors to estimate their location. Moreover, GPS and other satellite signals lack precision or entirely fail indoors [27]. There is clearly a need for accurate localization both indoors and outdoors. The proposed methodology focuses on outdoor localization however the developed methods are applicable to indoor localization applications as well.

Localization through mobile network (LTE, 5G) base stations is a particularly intriguing topic as the applications go beyond navigation. Knowledge of the user's location assists

the base station with accurately estimating channel conditions, thus improving resource management and beam forming [36]. As 5G requirements go beyond daily cellphone usage and aim to achieve enhanced mobile broadband, massive machine type communications and ultra-reliable and low latency communications, there is a growing need to advance localization technologies to support high frequency spectrum [45]. The mmWave spectrum has gained significant attention in the upcoming 5G networks due to its high data transmission rates, which will be able to support such applications. However, the mmWave band comes with its own set of challenges. For example, the attenuation properties of the channel increase the effects of blockage as mmWaves cannot travel through solid mediums [19]. High mobility applications also create the need for frequent beam tracking [19]. Moreover, the narrow beams require precise localization methods as errors in location can impact the transmission.

Machine learning has been used in many applications such as speech and image recognition, financial predictions, medical predictions and diagnosis, and within the last few years, has been a hot topic in wireless networks. Recent research has shown the benefits of machine learning when applied to problems such as routing, resource allocation and adaptive modulation techniques [45] [17] [18]. Machine learning-based localization techniques have been proven to achieve state of the art results, particularly in deep learning. The integration of these techniques with multiple input, multiple output (MIMO) systems has addressed some of the issues that were present in traditional localization approaches, such as the need for precise calibration in the offline phase and the fluctuation of instantaneous RSS values. Such techniques are discussed in [13] and [40]. The proposed methodology addresses the above issues by introducing a localization algorithm using a hierarchical regional classification and CNN approach. RSS, AoA, Angle of Departure (AoD) and ToA data are used in conjunction with lidar data collected from a simulated dataset that takes place in the neighbourhood of Rosslyn in Arlington, Virginia to predict the location of moving vehicles within the provided map.

1.2 Challenges

There are various challenges to overcome in the traditional localization techniques, as well as in the emerging studies. As mentioned previously, although GPS technology is highly effective and accurate, one of the biggest drawbacks of this technology is the deterioration of GPS signals indoors and in urban canyons. Additionally, GPS requires the deployment of large satellite systems, and the use of various satellites to locate only one user, which can drive up the cost of the technology both financially and computationally. Moreover,

traditional fingerprinting based localization techniques rely on channel conditions and path loss models to achieve accurate predictions. As such, these techniques require costly site-surveying to construct the RSS map in order to match incoming RSS data with accurate locations [48]. Finally, traditional proximity methods are fairly simple, however, this simplicity comes at the cost of reduced accuracy. Another challenge with this method is that it requires users to be connected to a nearby access point or sensor.

As the applications of localization increase with the deployment of 5G networks, the challenges also grow. For example, the use of localization techniques for vehicular communications requires the real-time performance of the algorithms to meet the timing requirements as the environment is quickly changing [46]. The use of machine learning algorithms for wireless communication applications also requires the need for large datasets to achieve precise predictions. Particularly, gaining access to labelled data becomes notably difficult in wireless network applications due to the overcharging radio environments. This drives the need for algorithms that are easily adaptable, scalable and trainable.

1.3 Thesis Contributions

The proposed methodology in this thesis is a localization algorithm in 5G and LTE networks using deep learning and image classification. The contributions are summarized below.

- A cascaded approach using lidar and satellite images and deep learning is introduced where the deep learning model explores the use of CNNs and FFNNs to predict user's co-ordinates, taking CSI data as an input
- The localization area is divided into various regions that are visually distinguishable through their landmarks. An image classification algorithm is implemented using image differencing that takes LIDAR or satellite images as an input to label the user's region based on the predefined boundaries.
- One CNN or FFNN is trained for each region of the map, using training data that is distinct to that region, to predict location co-ordinates.

The algorithm is assessed using 3 different datasets, 2 take place in Arlington, Virgina and 1 at the University of Denmark. The adaptability and scalability of the algorithm is assessed by applying it to the third data set, which provides satellite images and measurements from a Single Input, Single Output (SISO) LTE system, rather than lidar images

and 5G measurements from a MIMO system as in the first two datasets. The results are showcased given various adaptations of the algorithm. The findings show promising results that are comparable to GPS accuracy, using only one base station for localization

1.4 Thesis Organization

Chapter 2 provides background information of wireless communication systems, traditional localization techniques and the theory behind any applied or referenced machine learning algorithms. Related works are surveyed in Chapter 3, providing a brief comparison of current literature to the proposed algorithm. The proposed methodology is introduced in chapter 4, starting with a detailed review of the datasets used. The proposed technique is then described in detail. The results are showcased in chapter 5, where various adaptations of the algorithm are assessed and the benefits and drawbacks of the methodology is discussed. Finally the chapter ends with the conclusion. The appendix contains the source code of the proposed algorithm for all datasets, and all adaptations of the algorithm in python.

Chapter 2

Background Theory

This chapter provides brief descriptions of topics discussed in the literature review as well any background theory relevant to the proposed methodology. As the focal point of the proposed methodology is AI-enabled wireless communication in 5G and LTE, this section will focus mainly on the communication protocols and applications of those standards. Details regarding the machine learning algorithms used and discussed are also provided.

2.1 Wireless Communication Systems

Wireless communication systems enable the transmission of information from one point to another without the use of wired connections such as cables or wave-guides. The infrastructure of the network varies depending on the communication system, however, a general system consists of transmitters and receivers that deal with the modulation, demodulation, encoding and decoding of information as well as antennas that transform electrical signals into electromagnetic waves that can be transmitted in an unguided medium. Some of the widely used wireless communication systems are satellite communication, radar, Wirelsss Local Area Network (WLAN), bluetooth and cellular (mobile) communication systems. Although wireless communication systems provide the ease of mobility and accessibility globally, the lack of a guided medium also poses a set of challenges that may vary depending on the technology and protocols used.

Multi-path propagation is caused by reflection or diffraction of signals travelling from the transmitter to receiver due to objects in the environment, which can cause fading or inter-symbol interference [33]. As the receiver cannot differentiate between the various

multi-path components, the received signal becomes a combination of all the components; causing constructive or destructive interference. Small scale fading occurs when there are fluctuations in the signal strength or phase over short distances and duration of time. Large scale fading indicates the attenuation of signals as a result of large obstacles on the path [33]. The two types of large scale fading are path loss, which refers to the natural attenuation of signals over a long distance as a result of the attenuation properties of the wavelength and shadowing, which refers to the attenuation of signals as a result of a physical obstruction. Inter-symbol interference can occur in high bandwidth applications where the arrival of the multi-path components interferes with the subsequent signals [33].

2.1.1 LTE

LTE, also known as 4G, is a wireless network specification that was introduced by the International Telecommunications Union (ITU) in 2008. Unlike the former 3G standard, LTE uses Orthogonal Frequency Division Multiplexing (OFDM) and MIMO systems, which allow for multiple antennas at the receivers and transmitters, to enable higher data bandwidth, increase throughput and resilience to interference and reflections [16]. OFDM is a frequency division multiplexing scheme that transmits a number of closely spaced subcarriers that are orthogonal to each other in parallel, rather than transmitting a high rate stream of data with one sub carrier. This approach is advantageous as the frequency diversity of the transmitted data stream creates immunity to selective fading, reduces external interference and inter-symbol interference and increases spectrum efficiency [16]. The introduction of LTE also created the need for a new system architecture, shown in Fig. 2.1 to support the data rate and latency requirements. The new architecture uses evolved NodeBs, eNodeBs, that manage the cell's mobility and radio resources to optimize the communication link between all of the User Equipment (UE) in the coverage area. This is done through the use of resource blocks, which is the smallest unit of resource that can be allocated to a user, shown in fig. 2.2. One resource block is 180 kHz wide in the frequency domain and 0.5 ms wide in the time domain, consisting of 12 sub-carriers with a bandwidth of 15 kHz or 24 sub-carriers with a bandwidth of 7.5 kHz [1]. There is a guard band between each sub-carrier. The main Key Performance Indicators (KPI) of LTE networks are down-link speed of 100 Mbps, up-link speed of 50Mbps and latency of 10 ms [16].

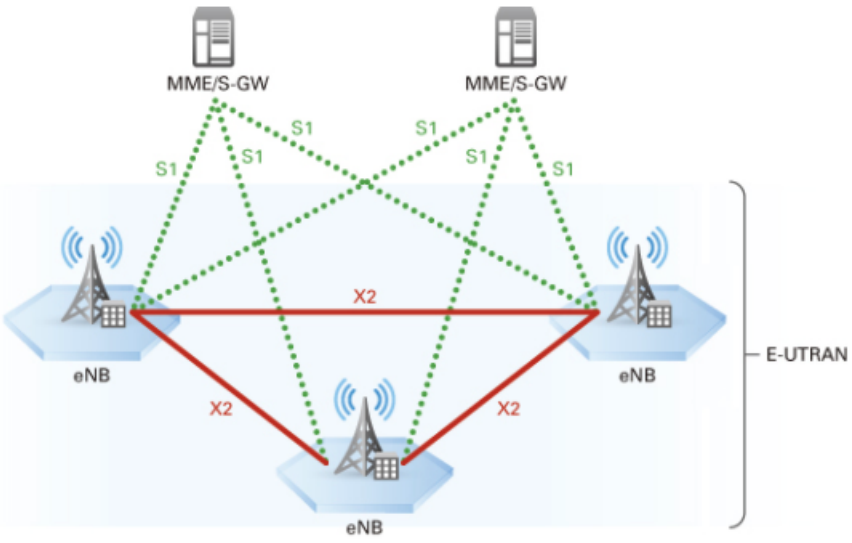

Figure 2.1: LTE system architecture [4]

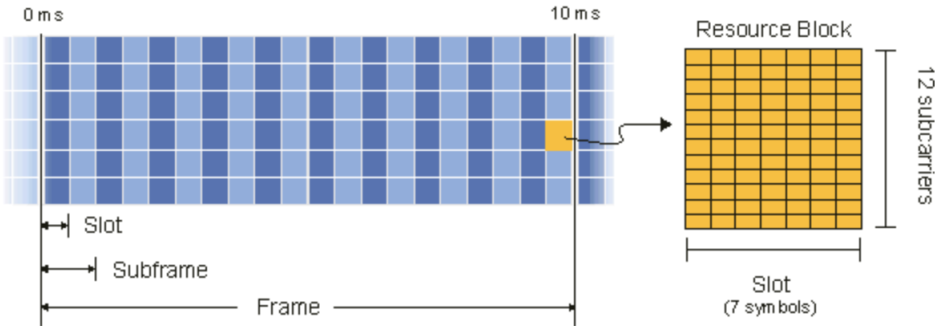

Figure 2.2: LTE resource block [1]

2.1.2 5G New Radio

The upcoming 5G network's aim to achieve massive machine type communication, enhanced mobile broadband and ultra-reliability and low latency [20]. Machine to machine communication, also known as Internet of Things (IoT), enables connection between billions of devices worldwide, without any human intervention. Such applications include

smart grids and cities, vehicular communications to support self-driving and smart farming. The enhanced mobile broadband aspect of 5G will provide significantly faster data rates and greater capacity that will be available through the use mmWave spectrum, which was previously not feasible due to some of the challenges that it brings. Finally, ultra-reliability and low latency, as low as 1 ms, allows real-time control of devices through a wireless medium. This will allow the use of wireless communication in things like robotic surgery, augmented and virtual reality and other mission-critical applications. The 5G radio access network architecture consists of small cells to support the small range of the mmWave specturm, as well macro cells that utilize massive MIMO [20]. The network also includes local and central servers to support low latency applications. Key features of the 5G core network include network slicing to enable segmentation of the core network for a specific industry and Network Function Virtualization (NVF), which allows network functions to run on virtual machines [20]. Along with the enhanced radio access and core networks, the ambitious 5G performance specifications are also made possible through optimization algorithms and techniques such as beam steering, beam forming, adaptive modulation and resource allocation.

2.2 Localization in Wireless Communication Systems

Localization is the process of retaining a user's location. Although navigation is the most common use of localization, other applications of this technology include fleet management, marketing, self- driving cars and optimization of wireless communication links Both indoor and outdoor localization are done using trilateration/ triangulation, cell allocation or pattern based techniques, which rely on either satellite based or radio frequency communication systems. The techniques used are discussed in more detail below.

2.2.1 Satellite Based Localization

Outdoor localization is traditionally done using the Global Navigation Satellite System, also known as GPS. GPS consists of a constellation of satellites that communicate with a ground control system [9]. The technology measures the distance from the user's receiver to at least 4 satellites, whose location is already known. The distance is measured by determining the time difference between the signal transmission from the satellite and the reception of the signal by the user on the ground. GPS can achieve an accuracy of 1 cm to 100 m depending on the location and environment. A major drawback of GPS is that it requires a receiver sensitivity of -130 dBm, rendering it un-servicable in forests, dense

urban areas, tunnels and indoor settings [9]. Moreover, GPS systems can be very costly and also lack power efficiency [8].

2.2.2 Radio Frequency Based Localization

Radio frequency based localization methods utilize WiFi, blue-tooth or cellular signals to estimate the location of users. The most common methods are trilateration, triangulation, fingerprinting and proximity. Trilateration and triangulation are geometric calculations that use the known location of various transmitters to estimate the location of user, similar to GPS. These methods are described in more detail below.

- **Trilateration** is used to determine the location of the receiver using geometry, given the co-ordinates of the Access Point (AP). The co-ordinates of three reference points are required to calculate the distance. A disadvantage of this method is that signal attenuation and multi-path propagation can make it difficult to estimate the distance given the signal strength, especially indoors [14].

- **Triangulation** is very similar to trilateration, however, it uses angles of arrival and departure rather than signal strength or time. The downside of this method is that the angles of the base station, i.e. the transmitter, may not always be known [14].

- **Fingerprinting** uses channel state information to predict the location of the user. The technique is divided into two steps: the offline phase which collects signal information such as RSS, ToA, AoA and positions from pre-defined reference points, and the online phase where the predictions take place [14]. During the online phase, the received signals are compared with the data base of fingerprints. The prediction algorithm could be probabilistic or deterministic. RSS based fingerprinting techniques estimate the distance based on the attenuation properties of the channel, given the location of the transmitter. Whereas time based fingerprinting techniques rely on the time of arrival, time difference of arrival or the round trip time. The distance between the transmitter and receiver is calculated using the time delay and the speed of the signal [14].

- **Proximity** is a simple localization method that estimates the location of users based on their proximity to known checkpoints or sensors. This method has a high variance in comparison to other localization methods, with a typical error of 50 m to 200 m [50].

2.3 Machine Learning Techniques

2.3.1 Neural Networks

Neural networks are a type of supervised machine learning technique that are designed to mimic the way the human brain works. The concept behind the algorithm is to take a large dataset, known as the training samples, and develop a system that can learn from these samples to solve a problem or make predictions [34]. Fundamentally, the networks are composed of layers of connected artificial neurons. Perceptions are the most basic example of neurons, that work by taking several binary inputs with associated weights to produce a single binary output. The output is determined by whether the sum of the products of the weights and inputs is greater than or equal to a defined threshold [34]. The neurons can be modelled with various activation functions, such as sigmoid, softmax and hyperbolic tan, shown in Eqs. 2.2,2.3 and 2.4, to achieve different results. In those equations, the term z refers to the sum of the weights multiplied by the inputs. The output of each neuron is shown in Eq. 2.1

$$o_j = \gamma(\sum_{k=1}^{n} w_{kj} o_k)$$
$$Where\ o_j\ is\ the\ output\ of\ j^{th}\ neuron\ in\ current\ layer \quad (2.1)$$
$$o_k\ is\ the\ output\ of\ the\ k^{th}\ neuron\ in\ previous\ layer$$
$$w_{kj}\ is\ the\ weight\ from\ k^{th}\ neuron\ in\ previous\ layer\ to\ j^{th}\ neuron\ in\ current\ layer$$

$$\gamma(z) = \frac{1}{1+e^{-z}} \quad (2.2)$$

$$\gamma(z) = log_e(1+e^{z}) \quad (2.3)$$

$$\gamma(z) = \frac{2}{1+e^{-2z}} - 1 \quad (2.4)$$

A complex network of neurons, shown in Fig. 2.3, creates a neural network. The inputs, outputs and layers in the network are connected through weights or costs. Training the network is the process of identifying the weights that will output the most accurate

predictions. This is done through back-propagation. Back-propagation works by finding the variation between the neural network's output and the expected output of the training dataset, starting with the output layer and propagating down to the input layer [34]. The error function of the output layer is calculated using Eq. 2.5. For each layer, the weights are updated by finding the partial derivative of the error function with respect to the previous weights, multiplied by a learning factor. This is shown in Eq. 2.6. The process of updating the weights is repeated either for a set of number of iterations, or until a defined error threshold is reached [34].

$$E = \frac{1}{2n} \sum_{x} |y(x) - y'(x)|^2$$
$$Where\, y(x)\, is\, the\, output\, of\, the\, output\, layer$$
$$y'(x)\, is\, the\, derivative \tag{2.5}$$

$$\Delta w_{ij} = -\eta \frac{\delta E}{\delta w_{ij}} \tag{2.6}$$

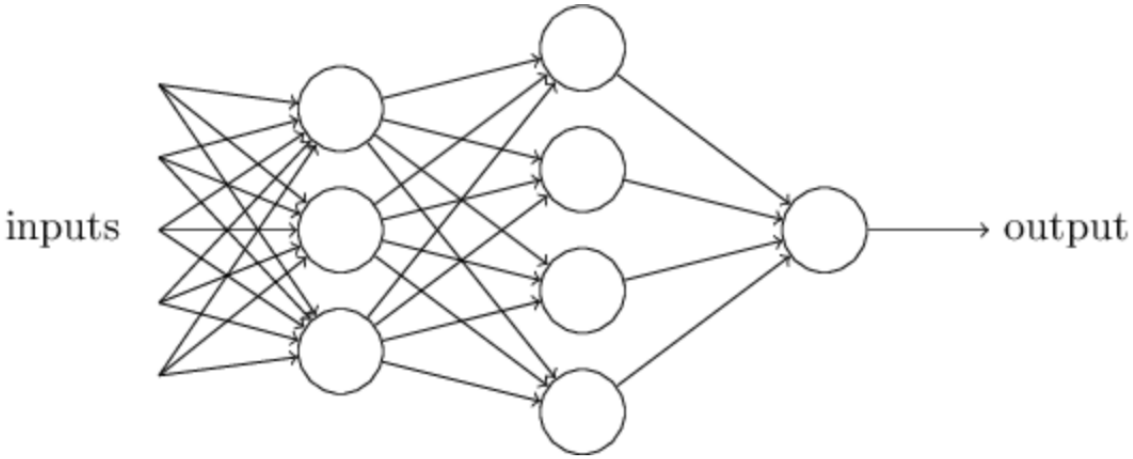

Figure 2.3: Feed-Forward Neural Network [34]

2.3.2 Convolutional Neural Networks

CNNs are FFNNs that include a feature learning stage before the classification stage, known as the convolutional layer. The convolutional layer allows the network to capture

the spatial and temporal dependencies in the input matrix through the application of relevant filters [37]. The architecture of the network is displayed in Fig. 2.4. Although the input to the matrix is typically an image, the network can intake any 2 or 3 dimensional matrix. The first stage of the convolutional layer involves the convolution of the input image with various filters of defined kernel sizes. Each convolution operation will output a matrix of the same size of the input, or smaller. The objective of this stage is to extract high level features in the input data [37]. Next, pooling is performed on each matrix produced as a result of the convolution. These matrices are referred to as channels. This stage involves sliding a kernel of defined size over each channel of the feature map to summarize the features within the area covered by the kernel. The two types of pooling are max pooling and average pooling. Max pooling returns the maximum value from the portion of the channel covered by the kernel, whereas average pooling returns the average value [37]. These stages can be repeated any number of times to achieve the desired results. Finally, the output of the last convolutional stage is flattened into a single 1 dimensional vector that is used as an input to a feed forward neural network such as the one shown in Fig. 2.3.

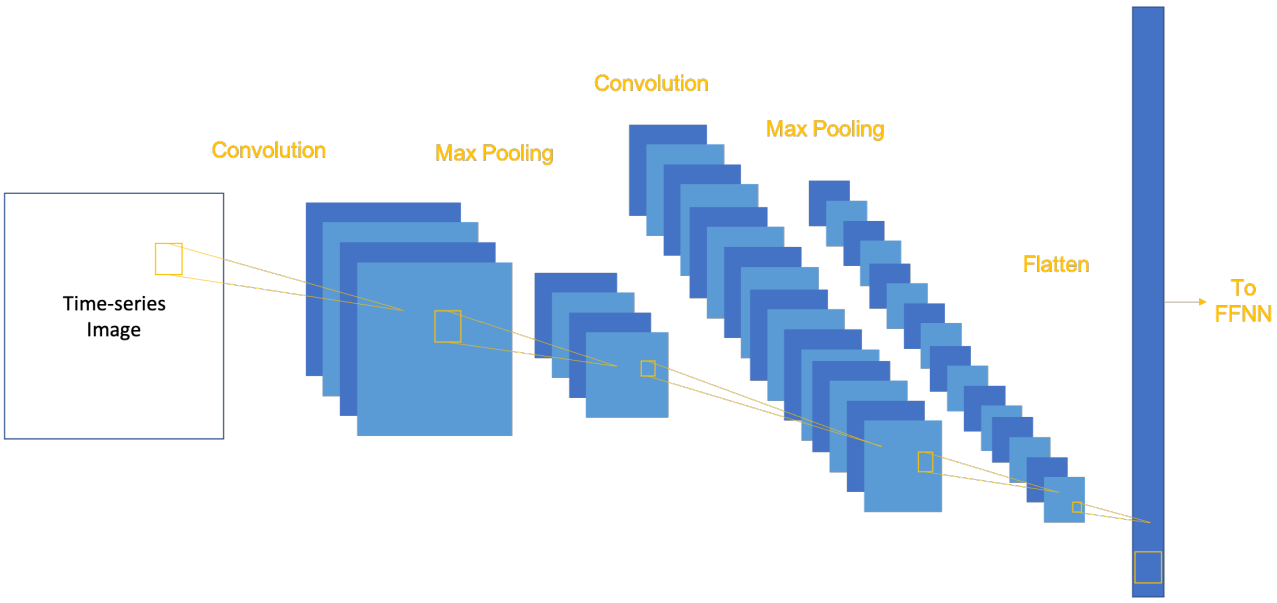

Figure 2.4: Convolutional neural network architecture

2.3.3 Region Based Convolutional Neural Networks and Faster Region Based Convolutional Neural Networks

Region based convolutional neural networks are a special type of CNNs used for object detection. Unlike image classification, object detection algorithms work by drawing a bounding box over the object of interest to identify it's presence. As such the number of bounding boxes, i.e network outputs, will vary depending on the input image which makes it difficult to use traditional CNNs for these applications as they require a predefined number of predicted outputs for all input images. To address this issue, Region Based Convolutional Neural Network (R-CNN)s work by first extracting 2000 proposed regions from the input image. This is done through a selective search algorithm that starts by generating various candidate regions and using the greedy algorithm to recursively combine similar regions into larger ones (ref). The proposed regions are then input into a CNN that produces a feature vector as an output. These features are input into a Support Vector Machine (SVM) to detect the presence of the object in the proposed regions. The disadvantage of this approach is the amount of training and prediction time due to the large number of proposed regions, which makes the implementation inconvenient (ref). This drawback is addressed in the faster R-CNN algorithm, where the input image is fed directly into the CNN to generate the feature map, rather than using the selective search algorithm. The proposed regions are then identified from the feature map. This approach significantly reduces the training and prediction times.

2.3.4 Long-Short Term Memory Networks

Recurrent Neural Network (RNN)s are a class of neural networks that allow previous outputs to be used as inputs, unlike feed forward networks [26]. This allows the computation to account for historic information, creating a memory for the network. RNNs are typically used for natural language processing and speech recognition, however, the concept can be extended to any time dependent application. One of the major drawbacks of these networks is their inability to access information after a long period of time, which makes them non-ideal for data with large sequences [26]. Long-Short Term Memory (LSTM)s are a type of RNNs that are aimed to avoid long term dependency by using internal mechanisms called gates to regulate the flow of information [43]. LSTMs are composed of the input gate, output gate, forget gate and cell states. The cell state reduces the effects of short-term memory by carrying relevant information down the chain [43]. As shown in Fig. 2.5 , the chain starts by passing the current input and information from the previous state to the forget gate. The forget gate decides on which information to keep by passing

those parameters to a sigmoid activation function that outputs a value between 0 and 1. Information that is kept outputs a value closer to 1 whereas information that is discarded outputs a value closer to 0. Next, the current input and previous state is passed on to the input gate which contains a sigmoid and hyperbolic tan activation functions. The hyperbolic tan function spans the information between -1 and 1 and the sigmoid function produces an output between 0 and 1, which decides which of the hyperbolic tan outputs to keep through the multiplication operation. The new cell state is calculated by multiplying the previous cell state by the output of the forget get. The product is then added to the output of the input gate to produce the new cell state. Finally, the output gate decides on what the new hidden state should be. This is done by passing the previous hidden state and current inputs into the sigmoid activation function and passing the new cell state into the hyperbolic tan activation function. Both outputs are multiplied to produce the new hidden state. This process describes a single LSTM cell, the neural network is composed of various cells connected through temporal weights.

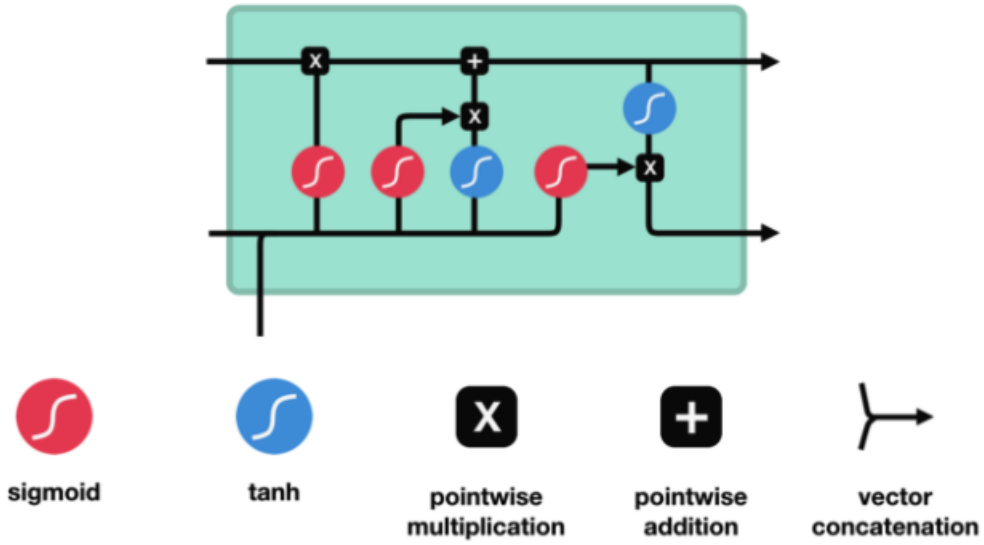

Figure 2.5: LSTM architecture [2]

2.3.5 K-Nearest Neighbours

K-Nearest Neighbors (KNN) is a supervised machine learning technique used to solve regression and classification problems. The concept involves clustering nearby data points

into regions. To start the algorithm, the number of neighbours is manually chosen, referred to as K-neighbours. Next for each training sample, the distance between the current data sample and the training data samples is calculated. The samples are then sorted based on their closeness to the current sample, from smallest to largest. The first K data points are chosen from the sorted list and their labels are obtained. The predicted class is the one that appears most frequently in the obtained list [22].

2.3.6 Linear Regression

Linear Regression (LR) simply fits a linear equation to the data to model the relationship between two variables. The model fitting process can be done using a method called least-squares regression, which minimizes the sum of the squares of the vertical deviations between the line and each data point to calculate the line of best fit. The linear regression line is shown in eq. 2.7, where b is the slope of the line, a is the intercept and X is the dependent variable [3].

$$Y = a + bX \tag{2.7}$$

2.4 Image Differencing

The proposed methodology uses image differencing to classify users' regions based on satellite or LIDAR images. Image differencing is a technique used to determine the change between images. The two common methods used to quantify the difference between images are Mean Squared Error (MSE) and Structural Similarity Index Matrices (SSIM). The MSE method, shown in Eq. 2.8, works by comparing the images pixel by pixel and calculating the squared error of the difference between the original and test image [38]. Although this method is fairly simple, the major drawback is the lack of perception of the visual system which may lead to an increase in false detections. This issue is addressed in the SSIM method which aims to extract structural information from the image. The SSIM formula, Eq. 2.12, is based on the comparison of luminance, contrast and structure shown in Eqs. 2.9, 2.10 and 2.11 respectively [38].

$$MSE = \frac{\sum_{i=1}^{n} (y - x)^2}{n} \tag{2.8}$$

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$$
$$Where\,\mu\,is\,the\,mean \tag{2.9}$$
$$c2 = (0.01L)^2$$
$$and\,L\,is\,the\,dynamic\,range\,of\,pixel\,values$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{2.10}$$
$$where\,\sigma^2\,is\,the\,variance\,and, c2 = (0.03L)^2$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \tag{2.11}$$
$$Where\,\sigma_{xy}\,is\,the\,covariance\,and\,c_3 = c2/2$$

$$SSIM(x,y) = [l(x,y)^\alpha \cdot c(x,y)^\beta \cdot s(x,y)^\gamma] \tag{2.12}$$
$$Where\,\alpha\,\beta\,and\,\gamma\,are\,assigned\,weights$$

Chapter 3

Literature Review

Localization is a highly researched topic in wireless communications. While traditional approaches explore the use of GPS for outdoor applications and trilateration, proximity and fingerprinting for indoor applications such as in [47] and [11], machine learning based localization techniques have been increasingly prominent in recent works. Some of these studies exploit the use of user data that is formerly available to the base station or access point by establishing a communication link such as RSS, ToA, AoA and AoD. Whereas other studies utilize sensors and cameras equipped on the user's devices to predict locations such as in [35]. A variety of neural network structures have been researched such as in [25], [15], [23] and [52], particularly FFNNs, CNNs and LSTMs. This section will provide a brief summary of localization studies, providing an understanding of the evolution of localization techniques over time.

3.1 Traditional Approaches

[47] integrates GPS receiver with existing vehicle sensors, such as Lane Departing Warning System (LDWS) and the wealth of sensors available from Anti-Lock Braking System (ABS), to improve localization performance. A video camera is also used to detect lane markings in order to gain accurate orientation information with respect to the road. Their proposed methodology aims to achieve autonomous vehicle navigation, which requires the vehicle to be driven through the map by a human operator in the first stage. In this stage, the vehicle creates a localization system to get an accurate map. After the learning stage, the vehicle refers to the previously created localization system, as well as the sensors and camera to successfully navigate. A mobile mapping system is used to get an accurate pose estimation

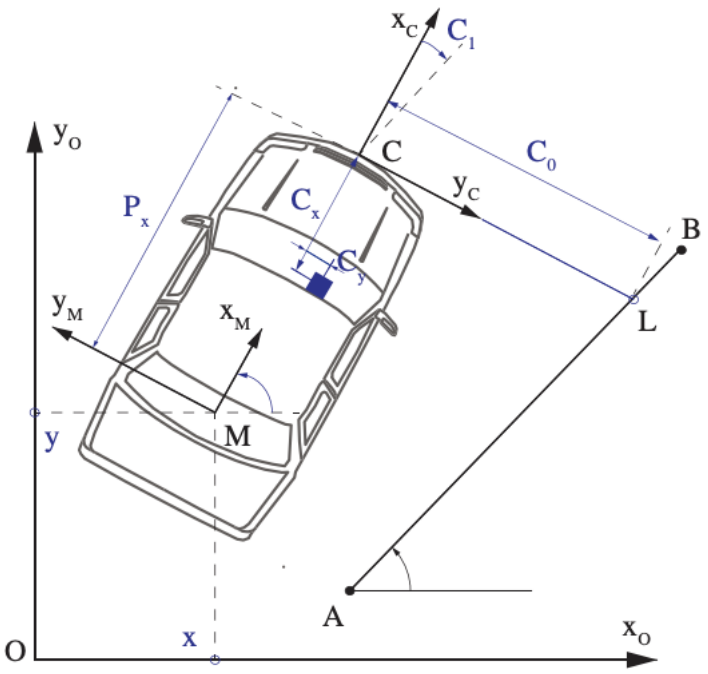

Figure 3.1: Vehicle orientation detection using lane marking segment [47]

of the body frame. The positioning data is synchronized with the measurements of the camera by trajectory interpolation. The point L in the reference frame, denoted by $[x_L y_L]^T$, is mapped using eq. 3.1. More details regarding the geometry of the orientation estimation using positioning data and lane marking segments are shown in fig. 3.1. Overall, their proposed technique achieved a lateral error of 0.11 m and longitudinal error of 1.09, proving that the manipulation and exploitation of GPS positioning data can improve localization accuracy.

$$\begin{bmatrix} x_L \\ y_L \end{bmatrix} = \begin{bmatrix} P_x cos\theta sin\theta + x \\ P_x cos\theta sin\theta + y \end{bmatrix} \tag{3.1}$$

[11] implements a wi-fi based fingerprinting method using Reference Signal Strength Indicator (RSSI) from APs. They use an indoor localization area that takes place on the second floor of a building in the National Institute of Posts and Telecommunications, which has a terrain size of 21.97 m by 10.84 m. The indoor space is equipped with three APs as well as Reference Point (RP)s. In the offline phase, the reference database is generated by

capturing the fingerprint of each RP from each AP. The fingerprints are collected during 1 minute over an area of $1m^2$ around the RP and the average of the fingerprints is stored in the database. Clustering is used to group the locations into small clusters according to the closest AP. In the online phase, the real-time localization takes place. The real time localization process involves collecting RSS values from the three APs, the cluster to which the samples belong to is the one with the highest signal strength. The euclidean distance between the observed and previously recorded RSS measurements is calculated. The location that minimizes that distance is chosen as the predicted location. Their work achieves a localization error of approximately 2 m.

The use of Bluetooth signals in an indoor environment is explored in [39], where they propose a range-based localization scheme that exploits the use of stigmergy. Their solution aims to overcome the challenges of WiFi-based fingerprinting and Wireless Sensor Networks (WSN) such as the long offline phase setup time, costly RSS calibration and high data volumes in the former and the need for costly, obstructive hardware in the latter. Their method incorporates the use of high density Bluetooth Low Energy (BLE) devices that act as anchor nodes and receivers. Stigmergy is used to predict user's locations, which is a term derived from the way ants communicate with each other to exchange information. To begin the localization process, a one-time offline calibration phase is performed where the RSSI is measured from a reference beacon at predefined steps. Based on the collected measurements, a logarithmic interpolation is performed to fit the data with the power loss model shown in eq. 3.2.

$$\begin{aligned} RSSI &= -(10nlog_{10}d - A) \\ where\, d &= distance \\ n\, is\, the\, slope & \\ A\, is\, the\, intercept & \end{aligned} \tag{3.2}$$

From there, the channel parameters are derived and the model can be used to calculate the distance. The first step in the online phase of the algorithm is the collection RSSIs from the deployed beacons. The mean is then computed to calculate an estimate of the distance. Next, an algorithm similar to Min-Max is applied to the distance estimations. The algorithm works by creating an association between each beacon's position and the previously estimated distance [32]. The mobile nodes draw horizontal and vertical lines around each beacon, such that the estimated node-beacon distance is equal to the minimum distance between each line and beacon. The node then localizes itself in the center of the rectangle. This paper manipulates the MinMax algorithm by applying the stigmergic process to the output area from the algorithm rather than estimating the output position as the center of the rectangular area. The process works by constructing a mark structure

in the rectangular area, which is then released in the spatial environment. At each step, the intensity of the mark varies as new marks superimpose on the old marks. Consecutive marks are partially superimposed if the node moves to a new location. A stigmergic map is then created using the sum of all of the existing marks, a sample map is shown in fig. 3.2. The widest area with the highest intensity is used to compute the output, the predicted location is the center of that area. This technique achieves a localization error of 2.01 m.

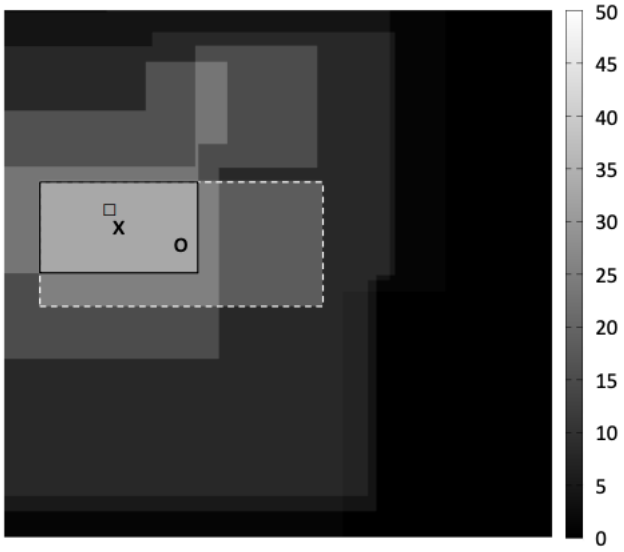

Figure 3.2: Sample image of stigmergic map from proposed methodology in [39]

3.2 Machine Learning-Based Solutions

While traditional outdoor localization techniques relies heavily on the use of GPS and indoor localization techniques explore the use of fingerprinting and proximity, there has been a push towards machine learning based solutions in recent years. The need for more accurate localization methods is due to some of the shortcomings of the current technologies. For example, while GPS can achieve considerably low localization errors in outdoor unobstructed scenarios, its performance significantly degrades indoors and in dense urban areas where Line of Sight (LoS) transmission of signals is not feasible. On the other hand, fingerprinting methods suffer from the fluctuation of RSS values due to multi-path effects. These issues become particularly imperative with the deployment of the up and coming 5G

network as it utilizes the mmWave band whose attenuation properties cannot withstand non LoS transmission and significantly degrades with long distances. Deep learning has been a popular topic in recent research and has proven to achieve state of the art localization accuracy while maintaining fairly low costs as the systems typically require fewer reference points than traditional fingerprinting techniques. The use of Neural networks, LSTM networks and CNNs are explored in [41], [13] and [40]. [13] and [40] also provide a comparison between deep learning approaches and classic machine learning techniques like LR and KNN.

[52] explores the use of fingerprint-based localization using FFNN in the LTE band. They obtain instantaneous CSI by connecting a Software Defined Terminal (SDT) that performs the standard receive process, as defined by the 3GPP LTE release 10, to a commercial LTE base station. The initial database is established by collecting CSI data and reference points in the offline phase. The features are extracted from the database using a CNN architecture to create the fingerprint map. In the online phase the CSI is passed on to the neural networks to predict the user's location. The input to the deep learning algorithm consists of the amplitude of the CSI for every antenna and sub-carrier at each location at a snapshot in time. The CSI model for a given location and time slot is shown in Eq. 3.3, where the time slot represents a slot duration of an LTE resource block. In this matrix $h_i^n(\mathcal{L}, t)$ is the amplitude and phase of the csi at the location $\mathcal{L}$ and time slot t, N_t and N_c is the total number of transmit antennas and Cell-Specific Reference Signal (CRS) and CRS_k is the subcarriers index of the k^{th} CRS.

$$\begin{bmatrix} \left|h^1_{CRS_1}(\mathcal{L},t)\right| & \left|h^2_{CRS_1}(\mathcal{L},t)\right| \cdots\cdots\cdots \left|h^{N_t}_{CRS_1}(\mathcal{L},t)\right| \\ \left|h^1_{CRS_2}(\mathcal{L},t)\right| & \left|h^2_{CRS_2}(\mathcal{L},t)\right| \cdots\cdots\cdots \left|h^{N_t}_{CRS_2}(\mathcal{L},t)\right| \\ \cdots & \cdots \quad \cdots \\ \left|h^1_{CRS_{N_c}}(\mathcal{L},t)\right| & \left|h^2_{CRS_{N_c}}(\mathcal{L},t)\right| \cdots\cdots \left|h^{N_t}_{CRS_{N_c}}(\mathcal{L},t)\right| \end{bmatrix} \tag{3.3}$$

The paper proposes two deep learning based schemes, to model slot-based localization and time domain fusion. The slot-based localization method takes a single fingerprint snapshot and outputs probabilities with respect to reference points. The final location prediction is based on the reference point with the largest probability. The time domain fusion network takes the various positions predicted by the slot-based network as an input and outputs the predicted location. The cascaded slot-based and time domain fusion networks achieve the lowest mean distance error (MDE) of 0.47 m indoors and 19.9 m outdoors over an area of 3.6 m x 6 m and 360 m x 195 m respectively.

[23] takes a similar fingerprinting-based localization approach, however, using LSTM networks rather than FFNNs. Their proposed algorithm uses RSS and wifi signals in an

indoor environment. The RNN is trained using data from consecutive locations in a trajectory at different time steps, which allows the RNN to incorporate information from the past, hence improving the localization accuracy. They propose a trajectory generation process in which RSSI values are used to create random training trajectories, given a maximum distance that a user can travel in a defined speed and time slot. They also showcase various models for the network architecture that have different inputs and outputs: Multiple RSSI inputs with single location output, multiple RRSI inputs with multiple location outputs, multiple RSSI with actual locations inputs with single location output, multiple RSSI with actual locations inputs with multiple location outputs and multiple RSSI with predicted locations inputs with multiple location output. The model using multiple RSSI with predicted locations inputs and multiple location outputs achieves the best localization error of 0.75 m over a localization area of 21 m x 16 m. The proposed methodology in this study also explores the use of neural networks for RSS and CSI based fingerprinting. However, the technique differs in that the use of this technology is explored in the mmWave band using convolutional neural networks, rather than recurrent or feed forward networks. A cascaded approach using image classification in the first step and CNN based localization in the second step is also proposed.

[15] also uses LSTM networks for indoor localization. The localization area takes place in the third floor of the IT college at Al-Zaytoonah University of Jordan, which is an L-shaped corridor that is 54 m by 32 m. Wi-fi beacons are installed on the roof of the hallway to transmit signals, and a cellphone is used to receive and store the observed RSSI in the database. Their proposed technique uses the collected data to train an LSTM model to predict the locations based on the RSSI fingerprints. They achieved a localization error of 1.009 m.

[25] proposes an approach using time-series RSS readings from WLAN access points and CNNs to predict buildings, floors and location co-ordinates. They take a hierarchical approach in which they first predict the building, then use the building's specific model to predict the floor and finally use the floor's model to predict the location co-ordinates. They create an $N \times T$ RSS feature image using time series RSS readings periodically collected from mobile nodes and use it as an input to the CNN, shown in Eq. 3.4.

$$\begin{bmatrix} RSS_{1,1} & RSS_{1,2} \cdots\cdots RSS_{1,T} \\ RSS_{2,1} & RSS_{2,2} \cdots\cdots RSS_{2,T} \\ \cdots & \cdots \quad \cdots \\ RSS_{N,1} & RSS_{N,2} \cdots\cdots RSS_{N,T} \end{bmatrix} \tag{3.4}$$

The use of the time-series RSS feature matrix allows them to overcome the challenge of randomly fluctuating RSS values as a result of shadowing, fading and multi-path effects.

Their work demonstrates the improvements in localization error as a result of using time-series RSS readings in conjunction with CNNs rather than single vector RSS readings with a feed-forward neural network. They achieve a localization mean error of 2.77 m for the RSS time-series image approach.While this approach achieves exceptional results, the concept is difficult to implement in high mobility applications, such as vehicular communications. The proposed technique uses a CNN input feature matrix composed of single-instance RSS measurements along with AoA, AoD and ToA from 25 different antennas.

[41] proposes an outdoor localization scheme using a deep learning from data collected using Long-Range Wide-Area Network (LoRaWAN) communication protocol. They aim to improve fingerprinting-based localization in complex indoor and outdoor environments. They create a system architecture consisting of LoRa sensor nodes that send data payloads to multiple Gateways or base stations, shown in fig. 3.3. The Gateway collects the RSSI, Spreading Factor (SF), and Horizontal Dilution of Precision (HDOP) and are stored in a time-series database to later be used for the training phase of the algorithm. Several readings are collected at various locations. After the data collection process, the information is processed and outliers are interpolated using linear, cubic, quadratic and denoising auto-encoder methods. This is done to ensure that missing data samples or outliers do not degrade the localization performance. The denoising auto-encoder works by shuffling the data around and then attempting to reconstruct it, which helps the network learn the features within the noise. The performance of their technique is assessed against both an outdoor dataset and an indoor experimental setup. The outdoor localization area, containing 68 base stations, takes place in Antwerp, Belgium. The indoor experimental setup takes place at the third floor of Riverside Hall at Sacramento State University. The dimensions of the floor is 8 m by 28 m. The paper also explores the use of three different neural network architectures for location estimation: Artificial Neural Network (ANN), LSTM Networks and CNNs. Their results show that the LSTM achieves the best mean localization error of approximately 191 m. The improvement in outdoor localization due to the use of deep learning techniques as opposed to LR, Support Vector Regression (SVR) and KNN is shown in fig. 3.4. The ANN achieves the best indoor localization error of 1.32 m.

[13] adopts a neural network based approach in a massive MIMO system to predict locations, the performance of the deep learning approach is then compared to a more classical KNN learning algorithm. They utilize a system where the CSI acquisition takes place in the frequency domain, which is a standard protocol for 5G systems. The instantaneous power that is measured by the baseband processor is subject to fluctuations due to small-scale fading and the inability to perfectly synchronize the transmitter and receiver oscillators. As such, statistical learning is used in the data pre-processing stage to reduce

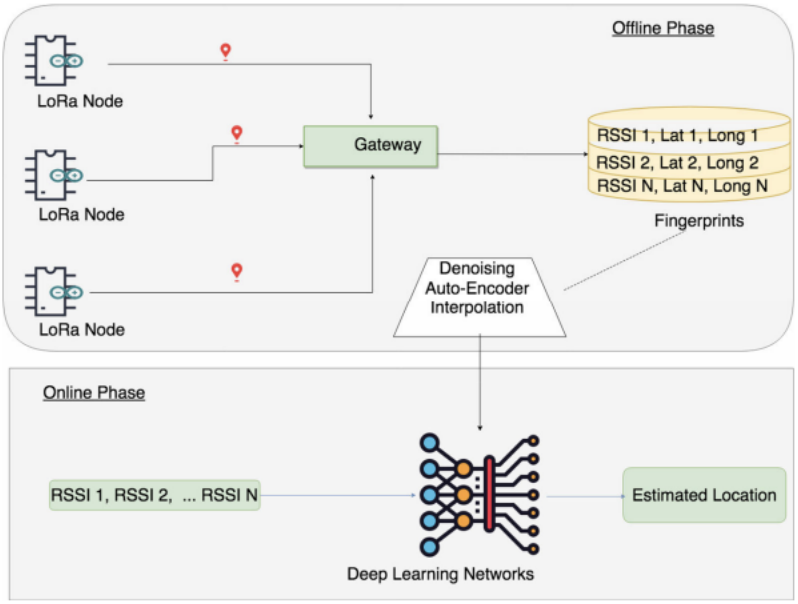

Figure 3.3: System archiecture of LoRa Based Localization [41]

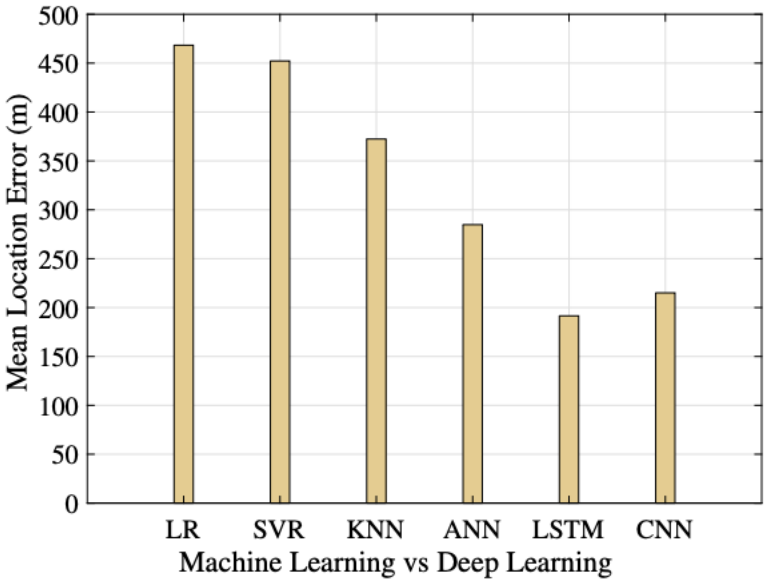

Figure 3.4: Outdoor localization results showcasing performance of machine learning vs. deep learning [41]

noise in the input data. The data was collected on the Huawei 5G test bed on the campus of the University of the Chinese Academy of Science in Huairou, P.R China. The CSI data was measured and collected at the receiver using a single antenna. The transmitter is a massive MIMO system containing s 32 dual-polarized antenna array, 8 antennas had to be discarded which left 56 antennas remaining. The base station was placed at the top of a 5 floor building. The localization area covered a terrain size of 196 m by 303 m. The data collection process took place over a time period of 90 minutes. Their proposed technique achieved an average localization error of 9.60 using neural networks and 8.16 using KNN. Although the KNN approach outperforms the neural network approach in terms of average error, the maximum localization error of the KNN approach is higher at 108.25 m in comparison to 73.26 m. This shows that the KNN approach is more susceptible to larger errors. Another benefit of the neural network approach noted in the paper is that the complexity of the KNN approach grows with the number of samples whereas the complexity of the neural network approach is only determined by the architecture and the number of neurons, making it more feasible and easily scalable. The paper also demonstrates the improvement in localization error as the number of antennas in the MIMO system increases. The relationship resembles between the two, shown in fig. 3.5, resembles an exponential decay as the localization error drops significantly until the system reaches 28 antennas. The drop in localization error after adding additional antennas still shows an improvement, however, at a slower rate.

[40] also takes a similar approach in which they implement various supervised machine learning techniques such as LR, weighted K-nearest neighbours (WKNN) and neural networks to estimate users locations outdoors by observing signals transmitted by single antenna users to a massive MIMO system. Their localization scenario takes place in an area of dimensions 316 m by 316 m, where 1000 scatterers and 50 blockers are uniformly distributed. The simulation also contains 1000 users and 1000 remote radio heads, each with 160 antennas. They adopt a sector-based processing approach where the learning models are trains with user-sector channel gains to reduce the effects of multi-path scattering. Sector-based processing allows for spatial separation of groups of propagation paths from different directions at the base station. Their results show that LR achieves the worst performance. Whereas, the neural network and WKNN models with 4 sectors is capable of predicting locations with an error of 5 m or less.

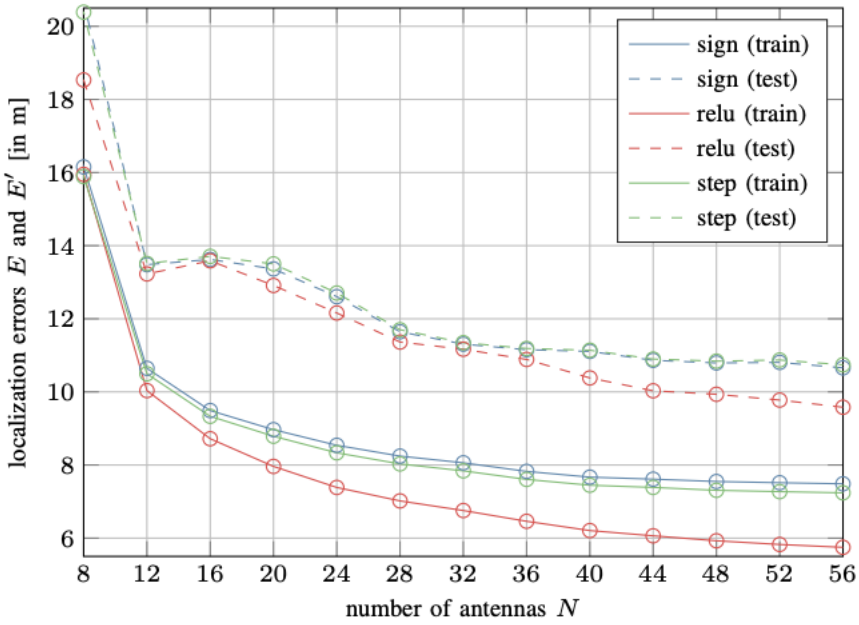

Figure 3.5: Relationship between localization error and number of antennas in a MIMO system [13]

3.3 Vision Aided Solutions

The introduction of the up and coming 5G networks has also brought along ambitious performance requirements, creating the need for better and faster localization techniques in the mmWave band to support massive machine type communication, ultra reliability and low latency and enhanced mobile broadband. The structure of the new network which creates the capacity for IoT, smart cities, and autonomous driving by connecting various devices that are fully equipped with sensors and cameras has opened up research opportunities for vision based localization techniques. Vision based localization schemes can reduce the dependency on large CSI datasets required to train machine learning algorithms or to calibrate traditional approaches. This also becomes particularly interesting as the dependency on CSI data creates some challenges as Radio Frequency (RF) environments are continuously changing. As this type of technology is just emerging, the use of only image data can still have large localization errors, such as in [35]. However, hybrid models, such as the proposed methodology in this thesis, can achieve localization errors that are comparable to GPS and other studies. The concept of vision-based techniques is also being explored in other wireless communication applications. An example of resource allocation using video sequences and images is shown in [6].

[35] uses camera images along with CNNs for landmark detection and localization to avert the use of weak or less accurate GPS signals outdoors. They propose two localization methods in which they investigate the object detection capabilities of CNNs and Faster R-CNNs. The first method uses Faster R-CNNs for landmark detection, which generates 3 features for each detected landmark. The features are then sent to a FFNN, where the co-ordinate localization takes place. The features generated consist of bounding boxes which contain the location and size of the detected landmark, labels that describe the landmark and scores that measures how well the detector identifies the objects. The second method explores a similar approach using CNNs in place of the Faster R-CNNs. This localization technique relies heavily on the images and predicts locations based on the picture's proximity to predefined landmarks. While there is still room for improvement in terms of localization error, this approach showed promising results. Their proposed technique achieves a mean localization error of 28.4739 m using the faster R-CNN approach over a localization area of 70 m by 30 m. In this study, a solution using landmark detection and image classification to predict predefined regions rather than location co-ordinates is proposed. The precise location is then predicted using a CNN that is trained using data from that distinct region. This structure provides the benefits of the image detection techniques while maintaining a lower localization error. The use of image differencing for landmark detection is also explored. This reduces the computational complexity of the

algorithm and eliminates the need for a large training data set for the images.

[24] utilizes the availability of aerial imagery with bird's eye view images, such as satellite and drone pictures to perform cross-view image matching. They propose the addition of a Net Vector of Locally Aggregated Descriptors (VLAD) layer on top of a CNN to extract descriptors that are against large viewpoint changes, referred to as the CVM-Net. They also propose a Markov localization framework to localize vehicles on the road using a video stream of ground level images. The cross-view image retrieval process works by finding a match of the ground image from a given database of satellite images using the CVM-Net. The CVM-Net framework consists of two branches, each with the same architecture. The first part is the local feature extraction and global descriptor generation, in which CNNs are used to extract local features. In the second part, the local features are encoded into a global descriptor that is invariant across large viewpoint changes. NetVLAD descriptors are embedded on top of each CNN branch. VLAD is a descriptor pooling method used in image classification that captures information about the statistics of local descriptors aggregated over the image. NetVLAD is an adaptation of the VLAD pooling method applied to a CNN framework to create a generalized, trainable layer [7]. They propose two CVM-Net based strategies, the first one contains independent NetVLADs and the second one contains NetVLADs with shared weights, shown in fig. 3.6. To account for the temporal consistency of ground view images from a video stream, the Markov localization framework is used to improve the localization performance. The Markov localization framework aims to find the probability of the vehicle's pose given past measurements. The localization scheme works by first storing a descritized version of the satellite map along with the image descriptors in the database, to be used by the CVM-Nets. The descriptor of an incoming ground image is then computed using the CVM-Nets and the Euclidean distance with respect to every satellite image in the database is used to determine the probability of the vehicle's location. This probability is then also used in the Markov localization process to determine the sensor measurement probability distribution. The performance of their proposed methodology is assessed against two datasets containing panorama ground images. The platform for the Markov localization is implemented using a vehicle with 12 fish-eye cameras mounted on the top. Their results show an average localization error of 676.7 m over a localization area of 10 Km by 5 Km without using Markov localization and an average localization error of 16.39 m using Markov localization.

The work in [6] also takes advantage of visual data like RGB images and video frames, however, for resource allocation rather than localization. Their proposed methodology aims to support heterogeneous co-existing services in wireless networks by creating a proactive resource allocation system that is aware of its surrounding environment, to help achieve the ambitious 5G and 6G goals in terms of reliability, spectral efficiency and latency. They

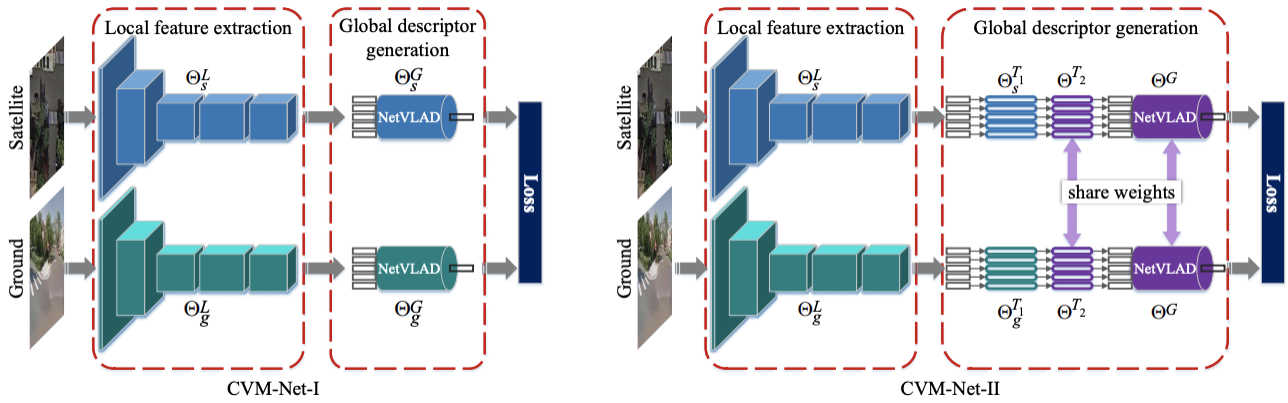

Figure 3.6: Image displaying CVM-Net architecture [24]

introduce a service identification framework using deep learning and computer vision to provide wireless networks with contextual awareness. They use an observed video sequence to predict the service type and request time prior to when the service is starting. Two deep learning neural network based algorithms are also proposed to predict the service type and request time in a spectrum sharing scenario. An example of their service identification framework is demonstrated in fig. 3.7, where the resource allocation scheme in the bottom left corner represents the under-utilized resources as a result of a reactive system, whereas the scheme in the bottom right represents a better utilized scheme due to the proactive service identification framework using a video sequence. In the example, the mounted camera in the AP is able to detect the vehicle and the mobile user. Based on this contextual information, it anticipates that the vehicle would be requesting an Ulta-Reliability Low Latency (URLL) service and the walking user would be requesting an enhanced-Mobile Broadband (eMBB) service. The speed and the distance of the users is then used to determine when each service would be required, allowing the network to allocate all of the resourced to the eMBB user first and then accommodating for the URLL users based on their anticipating service start time.

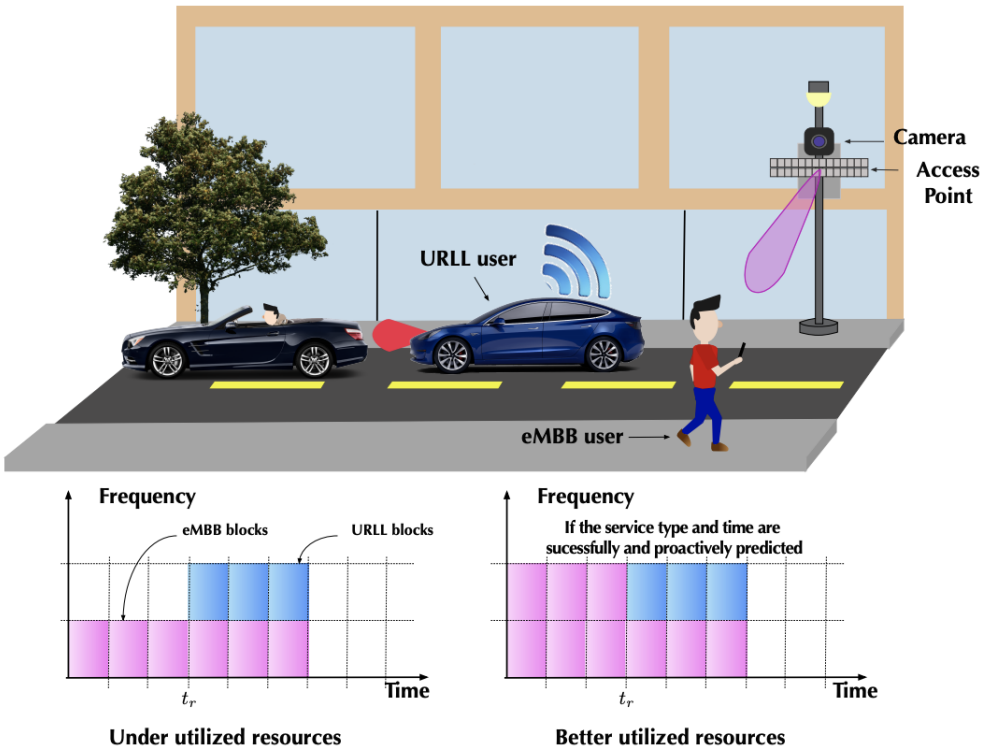

Figure 3.7: An example scenario provided by [6] demonstrating their proposed service identification framework

They propose two deep learning solutions that utilize advances in video analytics to form predictions. The first one is a two-stage neural network architecture that identifies objects of interest in the observed video sequence in the first stage and makes service type predictions in the second stage. The first neural network outputs a matrix that contains the number of objects detected in each frame and the dimensions of the objects. The matrices are then filtered and a candidate user and device are identified. The second neural network takes these vectors as an input to predict the service type based on the type of user and device. The second type of deep learning solution is an end-to-end neural network architecture that uses 3D-convolution and 3D max-pooling to utilize spatial-temporal information in the video sequence. The architecture has two different prediction layers, one to classify the service type and one to regress the request time. Their proposed technique achieves a maximum of 78% utilization and 60% reliability for the two-stage approach and 85% utilization and 98% reliability for the end-to-end solution, creating a

strong argument for vision aided and deep learning based wireless communication solutions.

3.4 Overview

Overall, this literature review showcases state of the art localization techniques as technological advancements progress. Studies have shown how the manipulation and integration of GPS signals with cameras and sensors can extend the use of the technology to urban areas and vehicular use, which aids in overcoming some of its shortcomings. While such techniques can help improve the accuracy of GPS in areas prone to performance degradation, their implementation will still fail in areas where GPS signals are not available such as in indoor environments or in tunnels. Moreover, as WiFi signals become increasingly available indoors, fingerprinting has grown to become an attractive localization method that can achieve promising results. The algorithms used to create the database and perform fingerprinting can be adjusted to improve predictions. Although fingerprinting based techniques have been able to achieve good accuracy, their implementation requires costly surveying and various reference points. The need for various reference points also makes the implementation of such systems difficult in large outdoor environments. Recent studies have experimented with applying machine learning based fingerprinting techniques rather than traditional probability and mathematical calculations, helping tackle some of the obstacles that come with traditional approaches. These studies have been able to achieve considerably low localization errors, such as in [23], which demonstrates how the use of machine learning can attain good localization predictions despite the degradation of signals indoors and the fluctuation of RSS values; a common obstacle of using the fingerprinting method [11]. However, these studies overcame the RSS fluctuation concerns by using time-series RSS measurements rather than single instance ones. The employment of these techniques can work seamlessly for fixed or slow paced receivers but is bound to malfunction in high mobility applications such as vehicular communications as the time dependent RSS measurements would not be accurate. On the other hand, more recent studies also explore the use of vision data for localization and other wireless communication applications, such as resource allocation. An example of that is shown in [35]. Although promising, the results of these studies show that the use of vision data alone can yield higher localization errors. The proposed methodology in this thesis focuses on the integration of vision data and radio parameters through the application of deep neural networks to maintain low errors while only using single instance measurements.

Chapter 4

Methodology

4.1 Datasets

The performance of the proposed methodology, as described in 4.2,is assessed against 3 different datasets. The first two are from the Raymobime dataset found at [29] which contain LIDAR images and raytracing data in the mmWave band using a MIMO system. The third dataset, found at [12], contains satellite images and channel state information in the LTE band using a SISO system. The results from the first two data sets are used to present the applications of the proposed technique in the upcoming 5G networks, whereas the results from the third dataset are used to demonstrate the adaptability and scalability of the algorithm in a different environment. The structure of the datasets is described in more details below.

4.1.1 Raymobtime Dataset

The Raymobtime-s008 and s009 datasets that are available at [29] are used. The datasets provide realistic data from a simulated 5G network using mmWaves and MIMO. The data is generated for over a terrain size of 300 m by 900 m that maps onto Rosslyn which is an urban neighborhood in Arlington, VA. The map information is collected from open-StreetMaps and converted into a 3-dimensional framework using Cadmapper 3D. The simulated environment consists of building structures, pedestrians, vehicles, multiple mobile receivers and one base station. The 3D scenario is then imported into wireless InSite, a 3D electromagnetic simulation software, where the signal propagation is simulated. Traffic

simulation is done using SUMO (Simulation of Urban Mobility), an open-source multi-modal traffic simulation package.

The structure of the dataset consists of episodes and scenes, where the episodes are composed of sequential scenes. There are 10 receivers in each scene. In this scenario, the receivers are a combination of moving cars, trucks and buses, each with a dedicated speed. The maximum vehicle speed is 64.8 km/hr. The datasets contain approximately 2000 episodes that are extracted by periodic sampling in intervals of 30 seconds, with each episode containing one scene. Fig. 4.1 displays the Rosslyn map, where the blue entities represent the building structures, the red data points represent the positions of the vehicles on the streets and the green point represents the location of the base station.

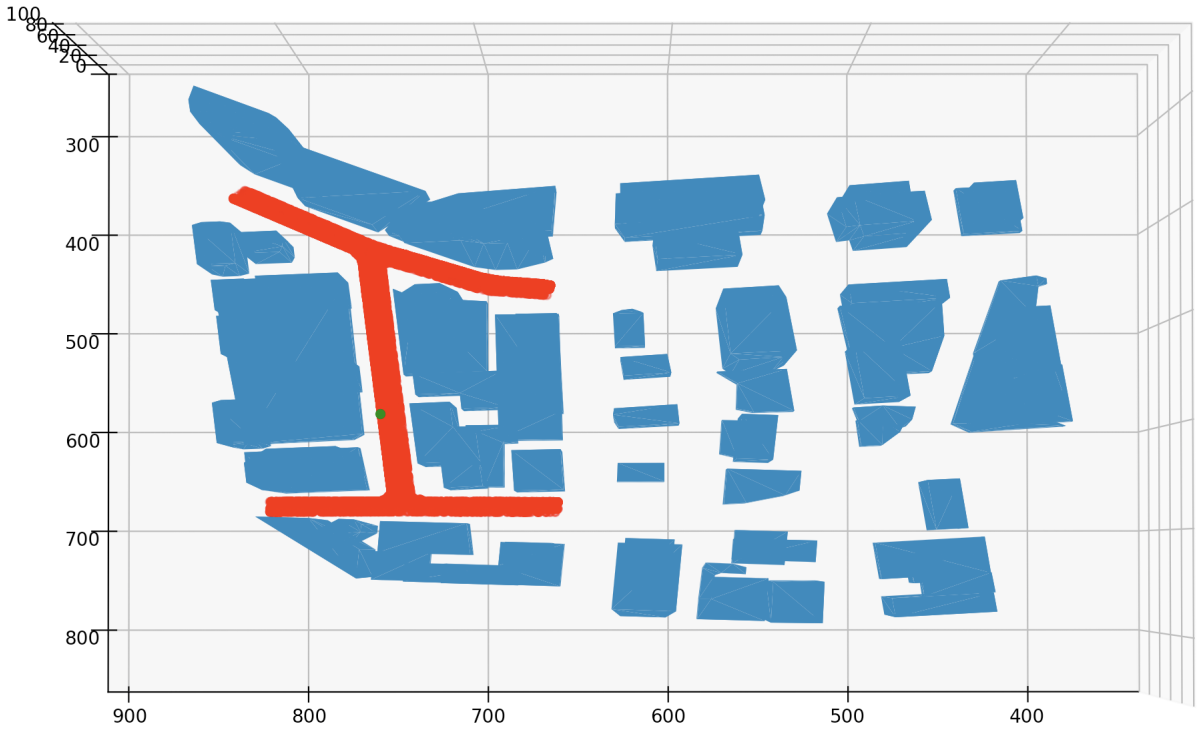

Figure 4.1: Plotted map of Rosslyn, Virginia (blue) and data set points (red), base station (green)

The dataset contains ray-tracing data, lidar data [30] and user position information. The lidar data is only available for a few vehicles in each episode and is provided in uniform

Table 4.1: Table summarizing dataset features.

Ray-Tracing Data	**User Information**
Received Power	Valid/Invalid Channel Flag
Time of Arrival	Episode ID
Elevation Angle of Arrival	Scene ID
Azimuth Angle of Arrival	Vehicle Array ID
Elevation Angle of Departure	Vehicle Name
Azimuth Angle of Departure	LOS Flag
Gain Phase	X Co-ordinate
Ray Phase	Y Co-ordinate
LOS Flag	Z Co-ordinate

quantized representation. The format of the dataset is summarized in table 4.1. Each ray-tracing parameter contains 25 different values native to one of the 25 antennas in the receivers.

The s008 dataset consists of 20860 samples, whereas the s009 dataset contains 20000 samples. While extracting the data, samples containing invalid channels were discarded, leaving a remainder of 11194 samples for the s008 dataset and 9638 samples for the s009 dataset. In this scenario, invalid channels are defined as communication channels belonging to vehicles that have left the analysis zone. The datasets are then split into a 60/40 ratio for the training and testing samples respectively. The pre-processing stage consists of replacing "NaNs" or invalid values in the dataset with 0s, min-max normalization, and the concatenation of the features into a single matrix. The matrix is composed of 7 rows and 25 columns, with each row containing a feature, and the columns representing measurements from each antenna. The structure of the matrix is shown in eq. 4.1.

$$\begin{bmatrix} AoA\_el\_Ant0 & AoA\_el\_Ant1 \cdots\cdots AoA\_el\_Ant24 \\ AoA\_az\_Ant0 & AoA\_az\_Ant1 \cdots\cdots AoA\_az\_Ant24 \\ AoD\_el\_Ant0 & AoD\_el\_Ant1 \cdots\cdots AoD\_el\_Ant24 \\ AoD\_az\_Ant0 & AoD\_az\_Ant1 \cdots\cdots AoD\_az\_Ant24 \\ RSS\_Ant0 & RSS\_Ant1 \cdots\cdots RSS\_Ant24 \\ gainPhase\_Ant0 & gainPhase\_Ant1 \cdots\cdots gainPhase\_Ant24 \\ ToA\_Ant0 & ToA\_Ant1 \cdots\cdots ToA\_Ant24 \\ RayPhase\_Ant0 & RayPhase\_Ant1 \cdots\cdots RayPhase\_Ant24 \end{bmatrix} \tag{4.1}$$

The lidar point cloud representation data was converted into coloured bird's eye view images. Several steps were taken to process the images. First, the image is converted into grayscale to reduce dimensionality. Then, the image is blurred to remove intricate detail and is then converted to black and white. The white space in the image background is cropped. Next, the images are padded again with a white background. This is done to center all of the images in a uniform manner. Finally, the images are scaled down to reduce computational complexity. Fig. 4.2 displays the evolution of the images as they are processed.

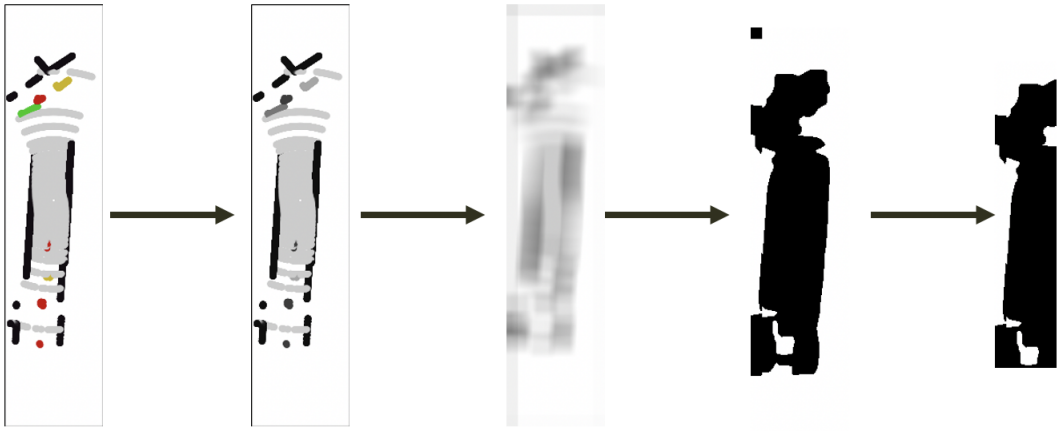

Figure 4.2: Visual demonstration of image processing technique

The RSS values are plotted to analyze the correlation between location and RSS strength. It is determined that region 1 contains the strongest RSS values in the raymobtime datasets due to its proximity to the transmitter. Regions 0 and 2 have weaker

RSS values, making them more difficult to classify using fingerprinting. This is shown in Figures 4.3 and 4.4. This information is later used in the proposed mechanism to identify the areas that can be classified into regions using the RSS value, rather than LIDAR images.

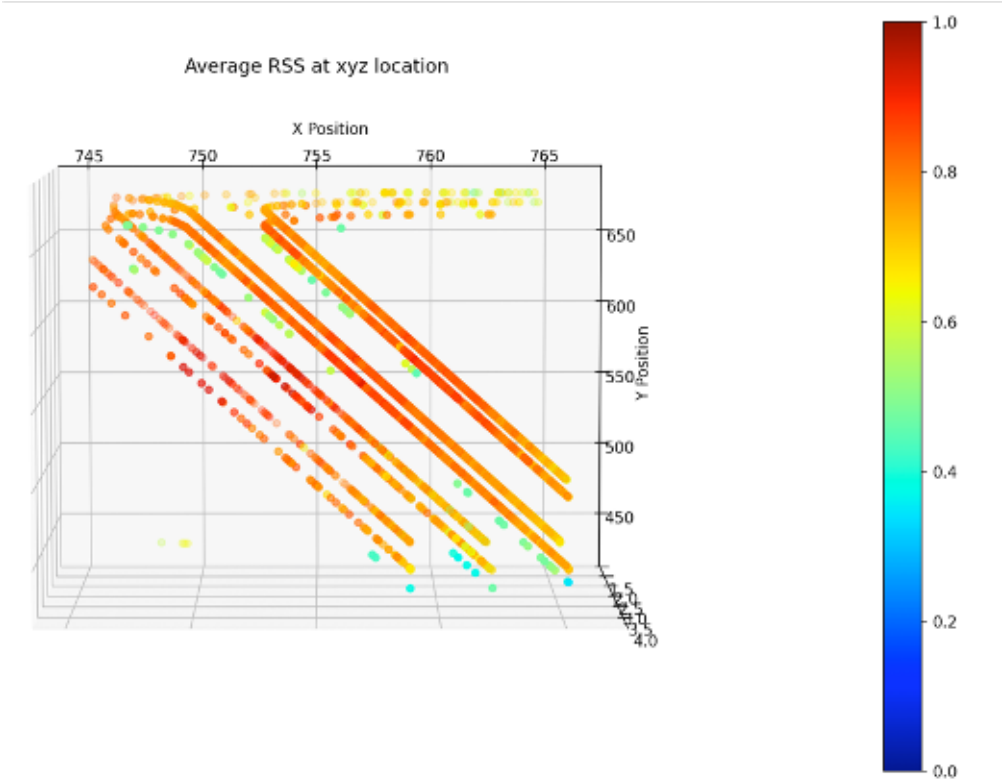

Figure 4.3: RSS values over localization area for S008 dataset

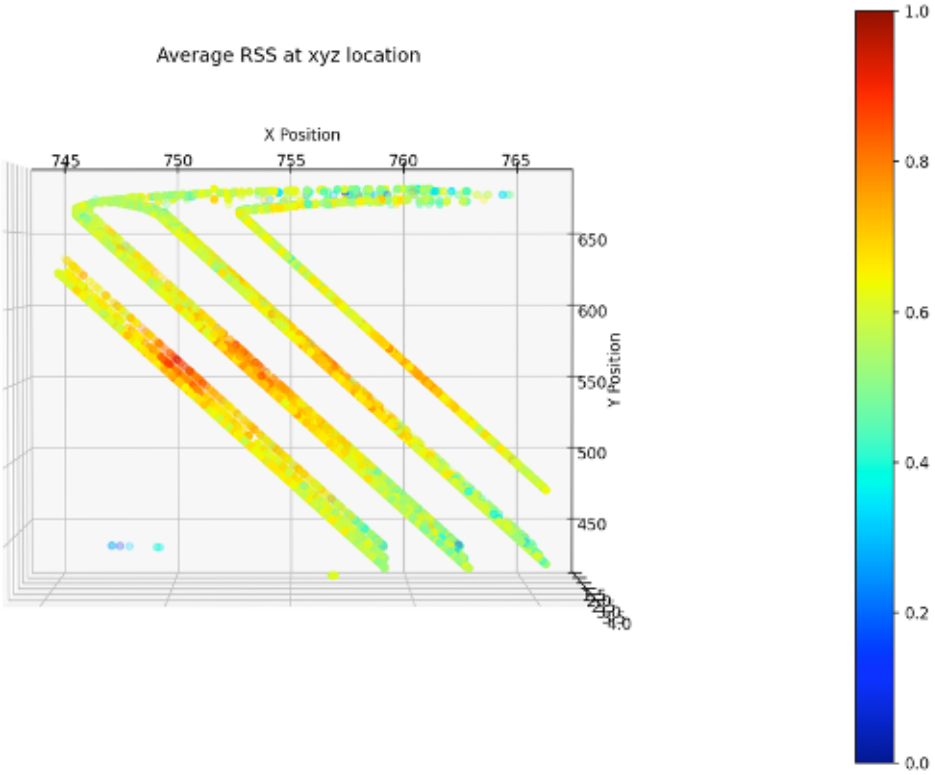

Figure 4.4: RSS values over localization area for S009 dataset

Samples of the training images used for the image classification stage are shown in 4.5.

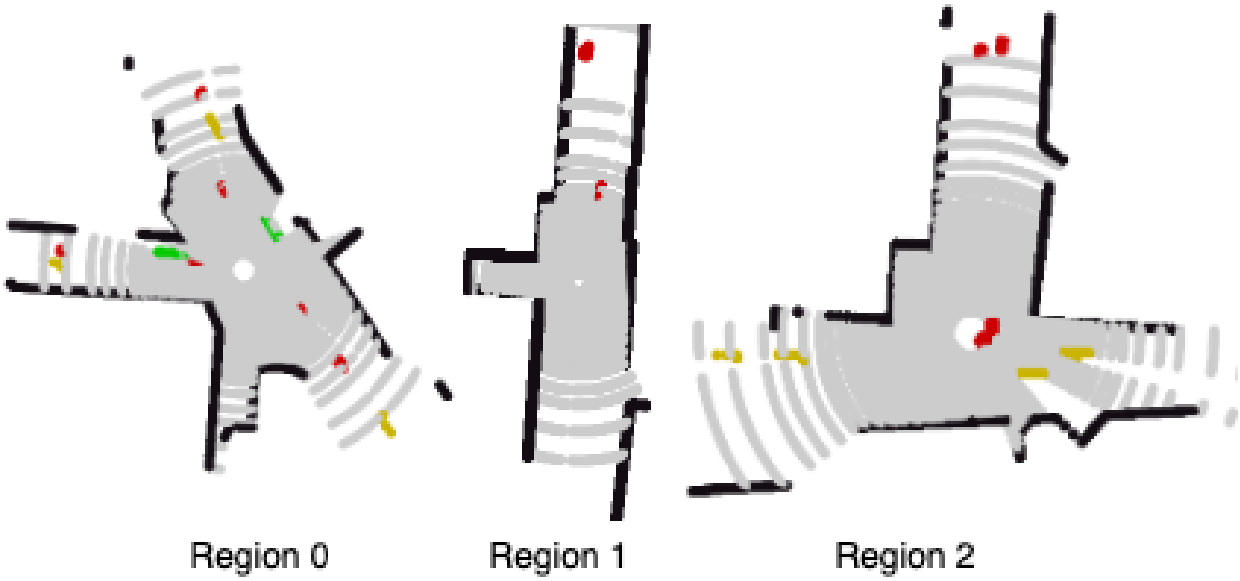

Figure 4.5: Sample lidar and training images for the 3 different map regions

4.1.2 Mobile Communication System Measurements and Satellite Images Dataset

The dataset provides real measurements from a deployed LTE-A communications system that takes place at the University of Denmark [12]. The scenario consists of one mobile vehicle with a single antenna and one fixed base station with 3 Physical Cell ID (PCI)s, 2 at 811 MHz (PCI_64 and PCI_65) and 1 at 2630 MHz (PCI_302). The overall coverage area is approximately 2.4 Km by 1.25 Km, the map of the localization area is shown in Fig. 4.6. There are 57585 samples in total, however, the overall dataset is split into 3 different sets based on the PCI index. For the purpose of this study, only PCI_64 is used which contains 11863 samples and has a coverage area of 1.446 Km by 1 Km shown in fig. 4.6. The dataset contains Signal-to-Interference-Plus-Noise Ratio (SINR), RSRP, Reference Signal Received Quality (RSRQ), Power/RSSI and a satellite image for each sample.

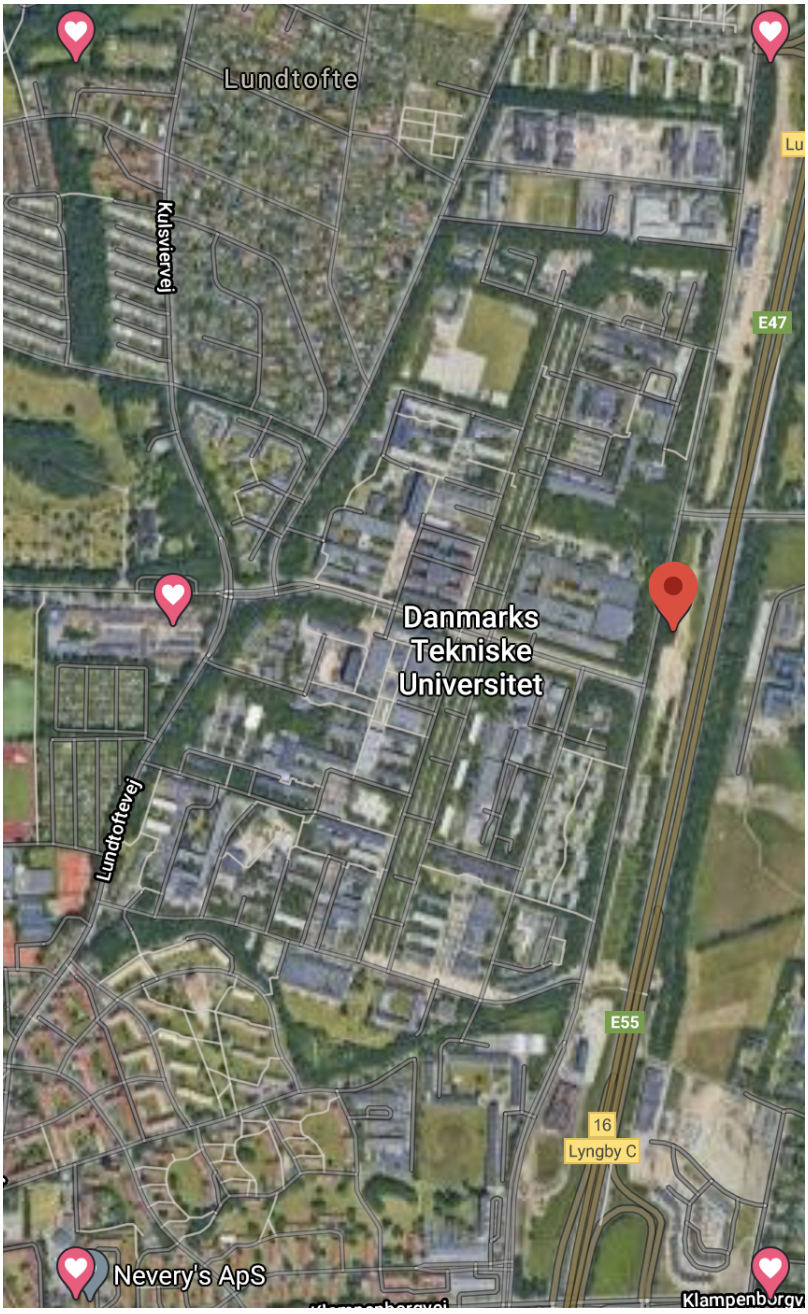

Figure 4.6: University of Denmark Localization Area

As this dataset does not contain any invalid channels or measurements, the pre-processing

stage of the channel state information consists of min max normalization and the concatenation of the data into a single vector. Unlike the raymobtime dataset, this is a SISO system, hence a one dimensional vector is used in place of the image feature matrix. The structure of the input vector is shown in Eq.5.1.

$$\begin{bmatrix} SINR & RSRP & RSRQ & Power/RSSI \end{bmatrix} \tag{4.2}$$

In this dataset, the provided satellite images are used for the image classification stage. To preprocess the images, an approach similar to the one used for the LIDAR images is taken, with the exception of cropping and padding the images. This is because all of the provided images are of the same size, 256 x 256 pixels. The preprocessing stage consists of converting the images to gray scale to reduce dimensions, blurring the images to remove sharp edges and details, converting the images to black and white based on a defined colour threshold and finally, resizing the images to reduce computational complexity. The process is shown in Fig. 4.7.

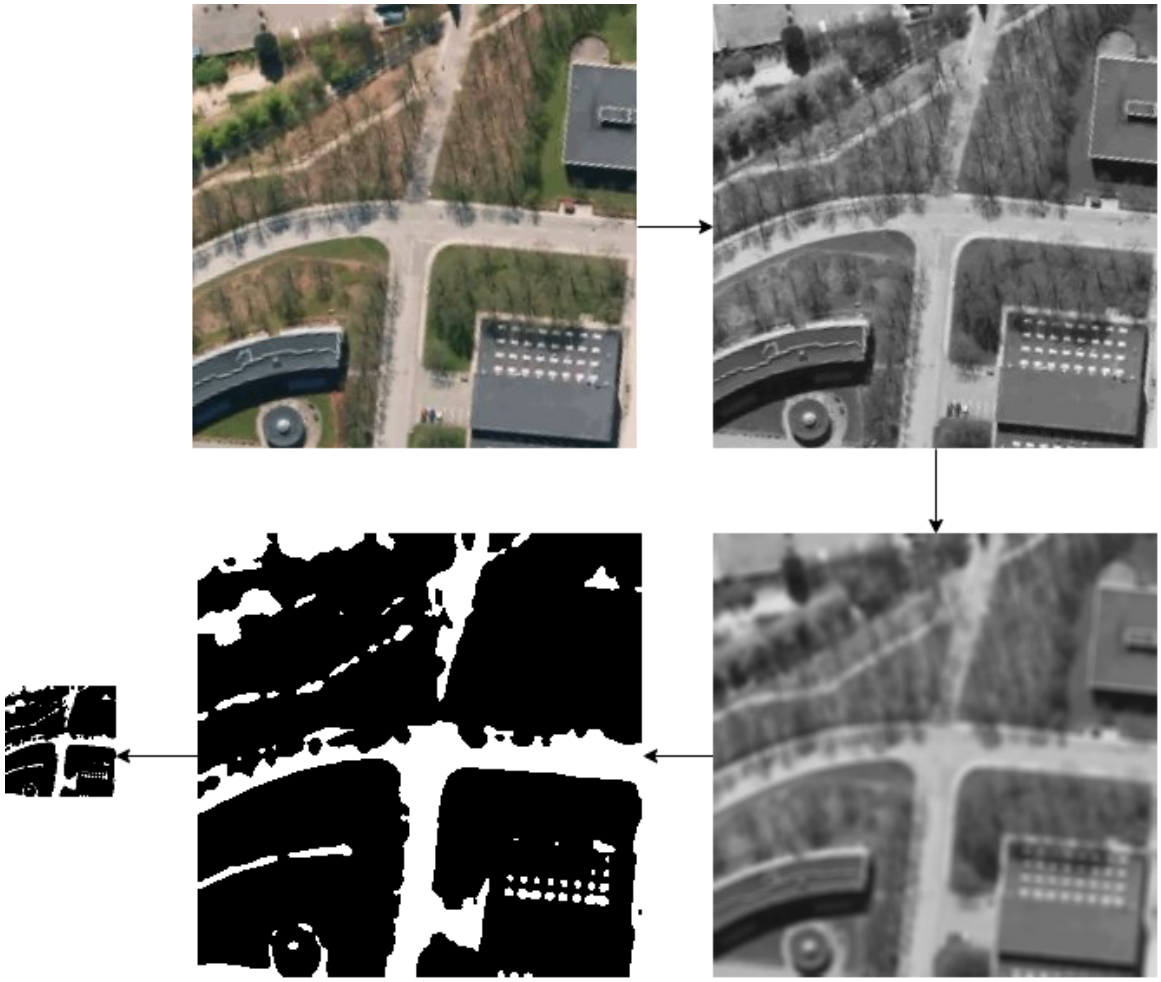

Figure 4.7: Image pre-processing stages of the mobile communication system datasets

Similar to the raymobtime datasets, the RSSI values are plotted to analyze the correlation between location and RSS strength. It is determined that region 2 contains the strongest RSSI values in these datasets. Regions 1, 3 and 4 have weaker RSSI values, making them more difficult to classify using fingerprinting. This is shown in fig. 4.8. This information is later used in the proposed mechanism to identify the areas that can be classified into regions using the RSSI value, rather than lidar images.

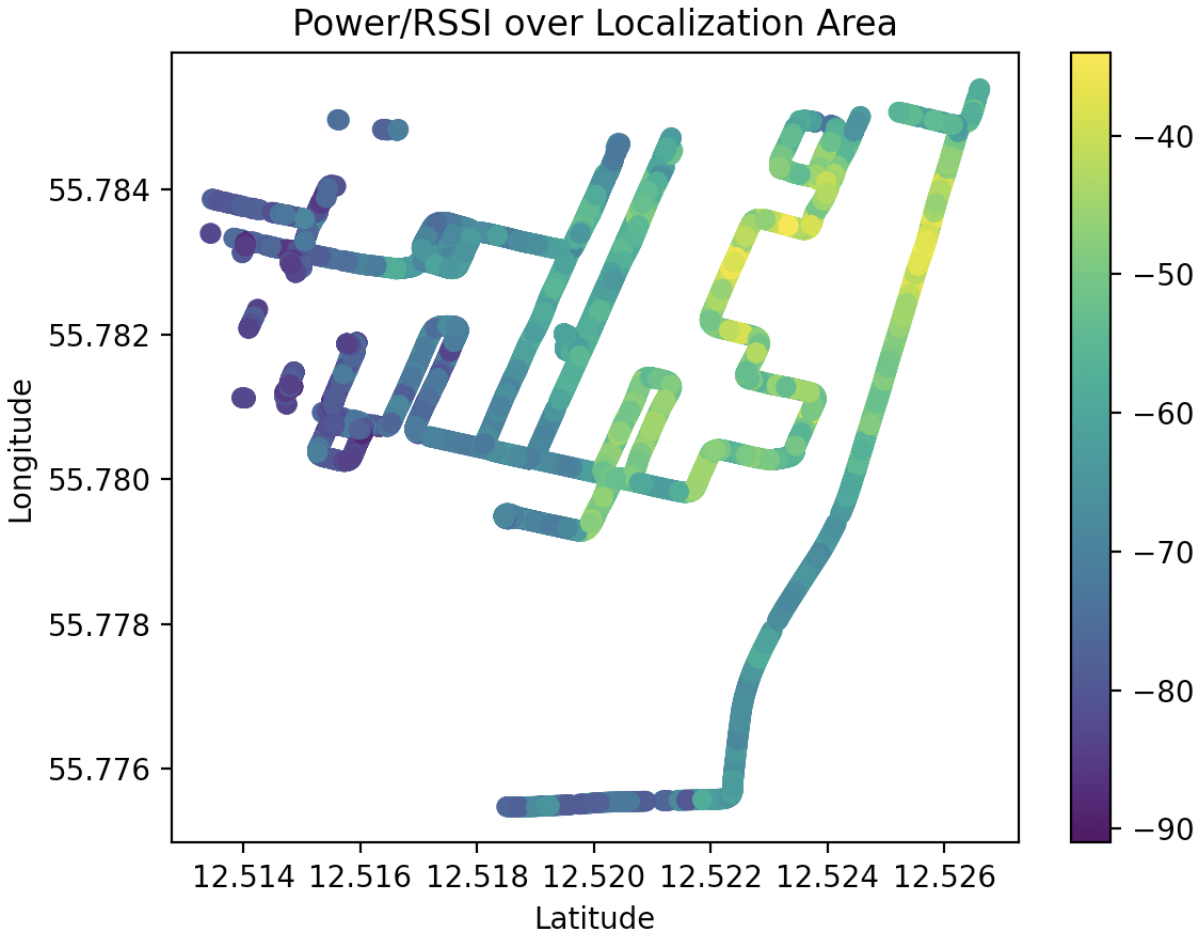

Figure 4.8: Power/RSSI measurements over the localization area

4.2 Proposed Technique

This study proposes a cascaded image classification and deep learning algorithm using LIDAR or satellite images and ray-tracing data to predict user locations. This is a centralized technique in which the vehicles or users provide the required information to a localization service provider where all the computations take place. As described in section 4.1, 3 different datasets are applied to the algorithm, 2 of which take place in the same localization area. The first two datasets, described in section 4.1.1, provide CSI data in the mmWave band in a MIMO system along with LIDAR images. Whereas the third dataset

described in section 4.1.2, provides CSI data in a SISO LTE-A system. The first stage of the algorithm uses RSS-based fingerprinting along with image detection to automatically sort the data points to the correct region. This is done through image differencing, which compares the LIDAR or satellite images provided by the vehicles to a repository of stored images of various map regions. To reduce localization error, the approach introduces neural networks that correspond to a predefined region in the map. Due to the structure of the datasets, the raymobtime datasets are applied to regional CNNs and the mobile communication system dataset is applied to FFNNs. This is because the mobile communication system and satellite images dataset only provides 4 features for each sample, which makes the implementation of this dataset using CNNs impractical as the networks will overfit given the small about of information. The regional neural networks are trained using radio parameters from samples belonging to that specific region, rather than utilizing the entire training sample set, which allows the predictor to learn the trends and outcomes distinct to that region. The regional classification algorithm is described in the flowchart in figures 4.10 and 4.11. All the regions are within the coverage area of the same base station. The scheme also considers two approaches for defining the regions. The first approach has solid defined boundaries, i.e, there is no overlap between neighbouring regions. In the second approach, the boundaries of neighbouring regions are extended by approximately 50 m to create an overlap. The algorithm is described in more detail in section 4.2.1.

4.2.1 Regional Classification

Regional classification is the first segment of the cascaded localization technique where data samples are automatically sorted into a pre-defined region in the localization area by either using the RSS value or provided image. The implementation of the proposed approach varies slightly based on the dataset used. The details of the regional classification scheme for each dataset is described in more detail below.

Raymobtime Datasets

A hybrid classification approach comprised of RSS based classification and image detection is proposed. Algorithm 4.1 starts by checking the RSS value of the sample. If the normalized RSS is greater than or equal to the defined threshold, then the sample belongs to region 1. Otherwise, the images are processed and image differencing is used to compare the LIDAR image to the repository of images. In the image processing stage, the images are first transformed to grayscale to reduce dimension, then they're converted to black and white, next they're blurred to remove excess detail and sharp edges and finally,

they're scaled to reduce the computational time of the algorithm. The best processing techniques were determined through trial and error measurements. The goal was to reduce computation time, while producing clear images for accurate regional classification. After the image processing stage, the image differencing algorithm is applied. The image differencing algorithm provides a score which defines the correlation between the two images, the closer the value is to 1, the stronger the correlation. The predicted region is the region of the image with the strongest score. If the score is lower than a pre-defined threshold then the data sample is sorted into a general CNN rather than a regional one. Here, a 70% image match is required for the LIDAR image to be classified into a region. This is done to reduce the classification error as a result of low image quality. On average, the algorithm takes 328 ms for image processing and differencing, per input image. As the image differencing stage can be timely and computationally expensive, the use of the hybrid approach reduces the complexity of the algorithm and the chances of image classification errors by only relying on images when needed. The regional boundaries are defined based on the unique landmarks that can be distinguishable through the image classification algorithm. This is done by dividing the localization area into 3 distinct regions. The regions are identifiable through the birds eye view of the street intersections of the LIDAR images. Since the localization area in those datasets is reasonable in size, the repository of training images consists of one image for each region, along with samples of imperfect LIDAR images to improve the image differencing accuracy. As mentioned before, two approaches are taken to define regional boundaries. This is done to reduce the chances of false image classification in regional boundaries. The two different decision boundaries can be seen in fig. 4.9.

Algorithm 4.1 Localization Algorithm with General Neural Networks

Data: Raytracing Data, Lidar Images
Result: X, Y, Z Coordinates
initialization **for** *Every data point* **do**
  check SNR **if** *SNR* $\geq$ *Threshold* **then**
    Append data to region 1 CNN, Predict Location
  else
    for *Every Training Image* **do**
      checkScore(currentImage, TrainingImage)
    end
    if *Image Score* $\leq$ *Threshold* **then**
      Append data to general CNN, predict location
    else
      Region = index(max(imageScore))
       Append data to regional CNN, predict location
    end
  end
end

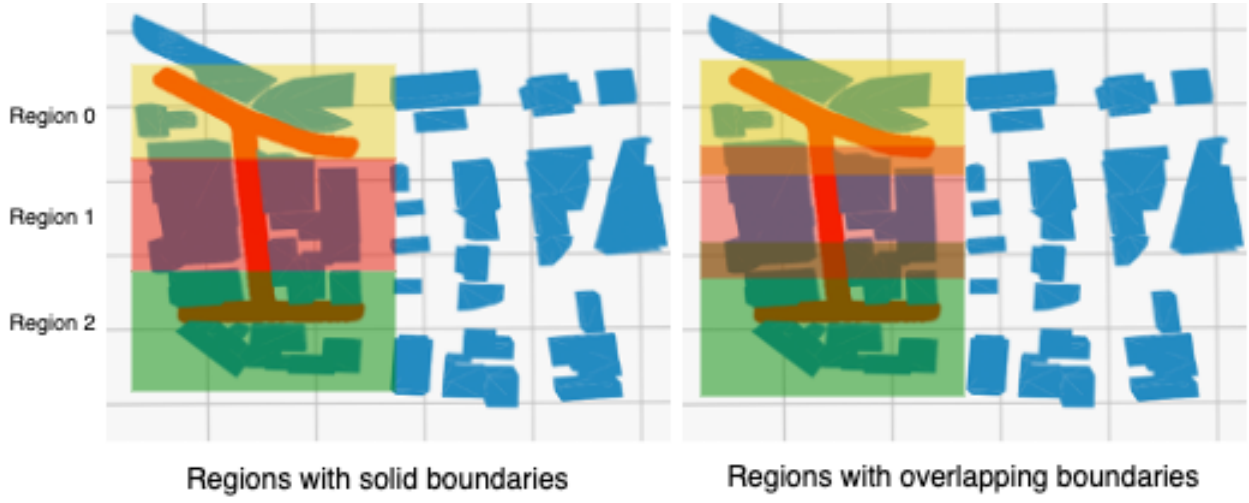

Figure 4.9: Image displaying the different regional boundary approaches

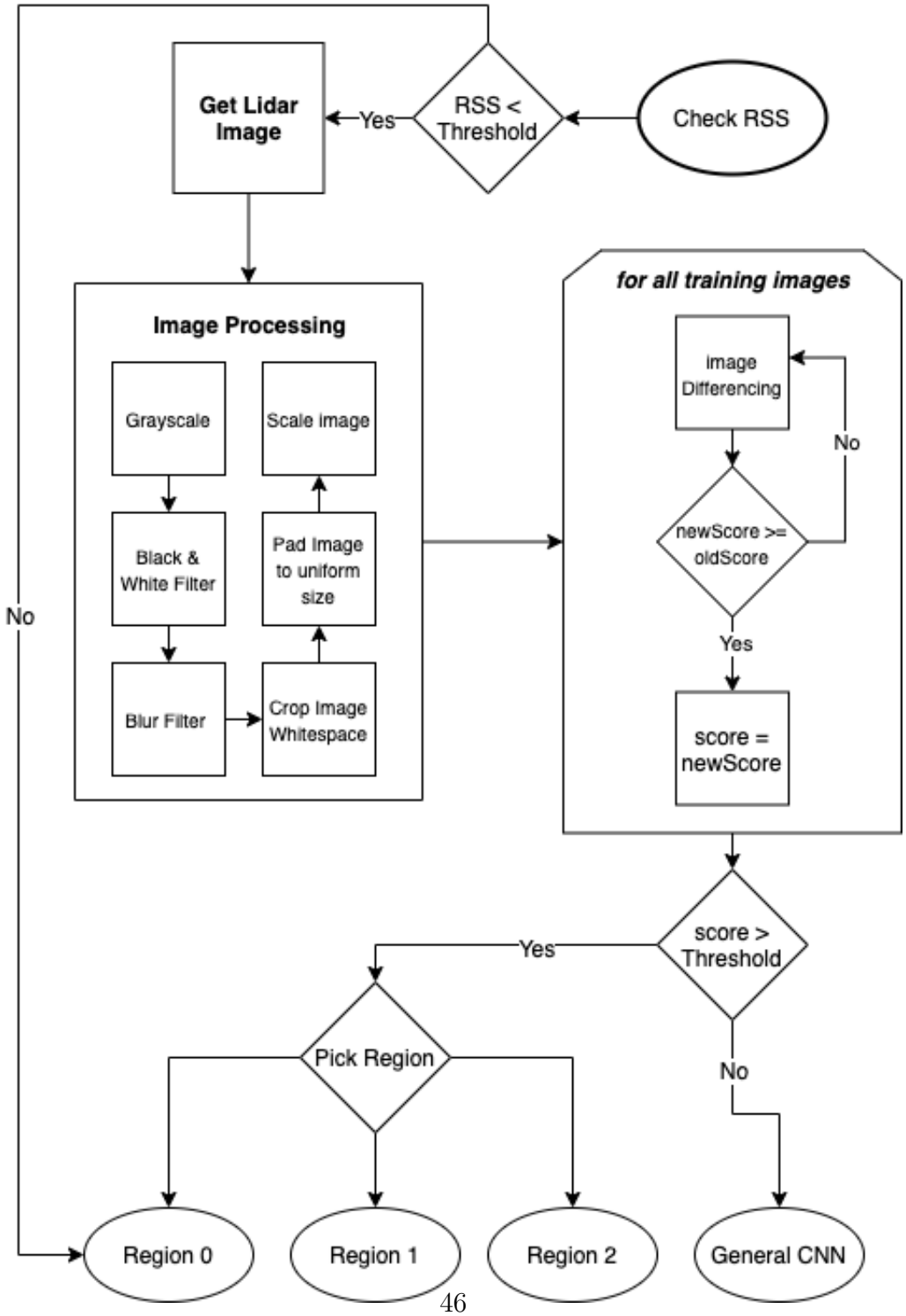

Figure 4.10: Flowchart displaying hybrid RSS based fingerprinting and image classification approach.

Mobile Communication and Satellite Images Dataset

The regional classification approach for the mobile communication and satellite images dataset is similar to the process described for the raymobtime datasets, with a few exceptions. Here, the map of the localization area is split into 4 different regions, shown in fig. 4.12. The image differencing algorithm takes the processed satellite image as an input and compares it to a stored repository of training images, each representing a different region. The SSIM score between the input image and each training images is calculated. The input image then belongs to the region of the training images that yields the highest score. Several images from varying parts of the region are used as training samples for the image training repository. 65% of the training images from each region were randomly sampled and used to represent the landmarks in that region. The number of training images selected was based on trial and error measurements. The goal was to increase the chances of true positive classifications, while reducing prediction time. For this dataset, two regional classification algorithms are proposed. The first one, is similar to the one used for the raymobtime datasets, described in algorithm 4.1. On the other hand, algorithm 4.2 omits the use of the general neural networks. This is done because the satellite images provide more detail, hence the likelihood of false classifications is reduced which allows the system to maintain a lower localization error without disposing any images with a low image differencing score.

Algorithm 4.2 Localization Algorithm without General Neural Networks

```
Data: CSI Data, Satellite Images
Result: Longitude, Latitude
initialization  for Every data point do
    check RSSI  if RSSI ≥ Threshold then
        Append data to region 1 FFNN,  Predict Location
    else
        for Every Training Image do
            checkScore(currentImage, TrainingImage)
             score = currentScore
             if newScore ≥ currentScore then
                 score = newScore
            end
        end
    end
    Region = index(max(imageScore))
     Append data to regional FFNNs, predict location
end
```

Similar to the raymobtime datasets, a second adaptation of the algorithms is proposed that uses a hybrid regional classification approach that integrates the image differencing algorithm with RSSI-based fingerprinting. To implement this, the RSSI strength over the localization area was plotted, shown in fig. 4.8. This plot showed that the RSSI values are strongest in region 1. Using this information a simple comparison algorithm that checks the sample's RSSI was created: if the RSSI is greater than a pre-defined threshold then the sample is sorted into region 1, otherwise the image differencing technique is used. Again, two ways of defining regional boundaries are used, which are regions with solid or overlapping boundaries shown in figures 4.12 and 4.13. On average, the algorithm takes 26.7 s for image processing and classification, for each sample. The prediction time of the mobile communication system dataset is notably longer as the satellite images are much larger and contain more detail

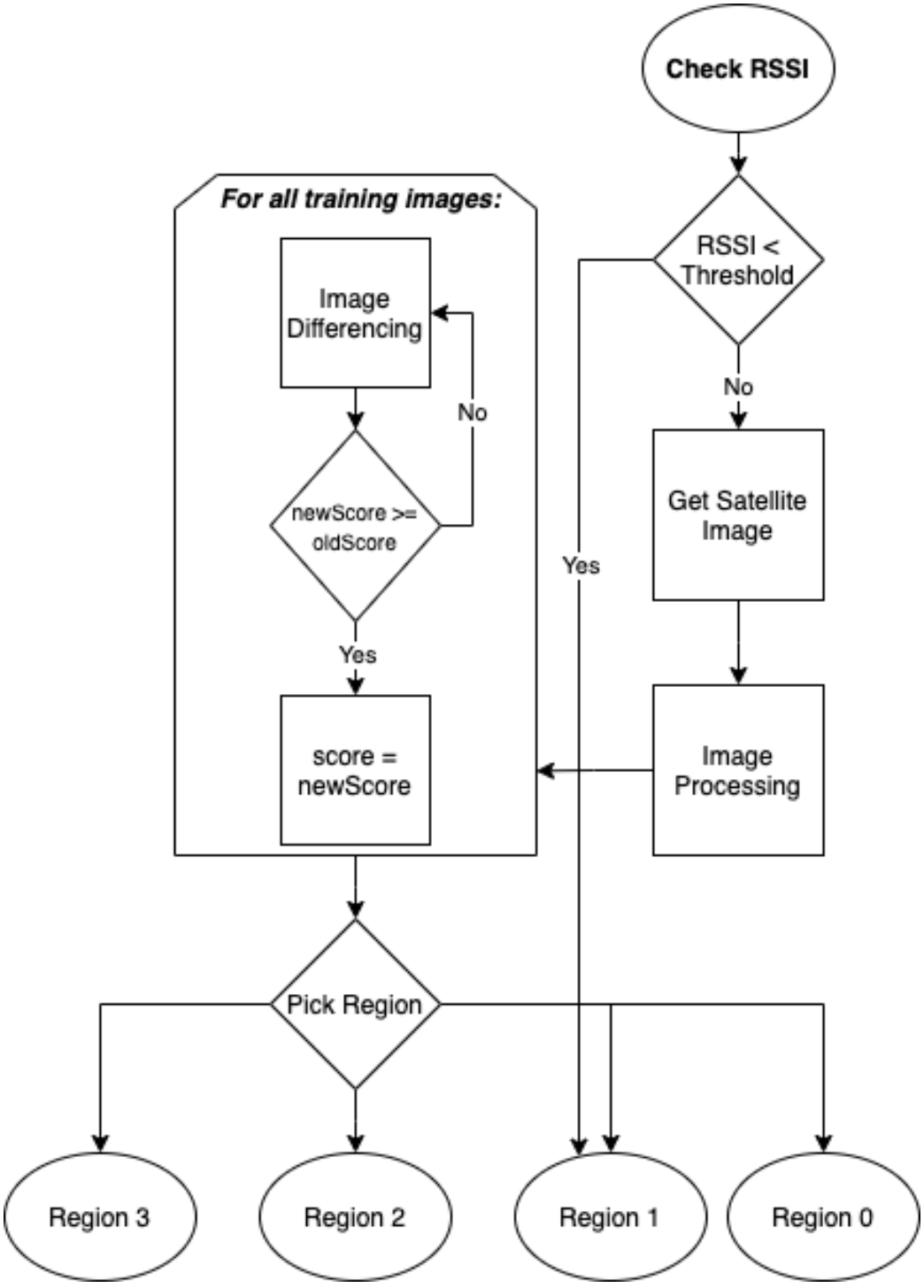

Figure 4.11: Flowchart describing Regional FFNN localaization technique

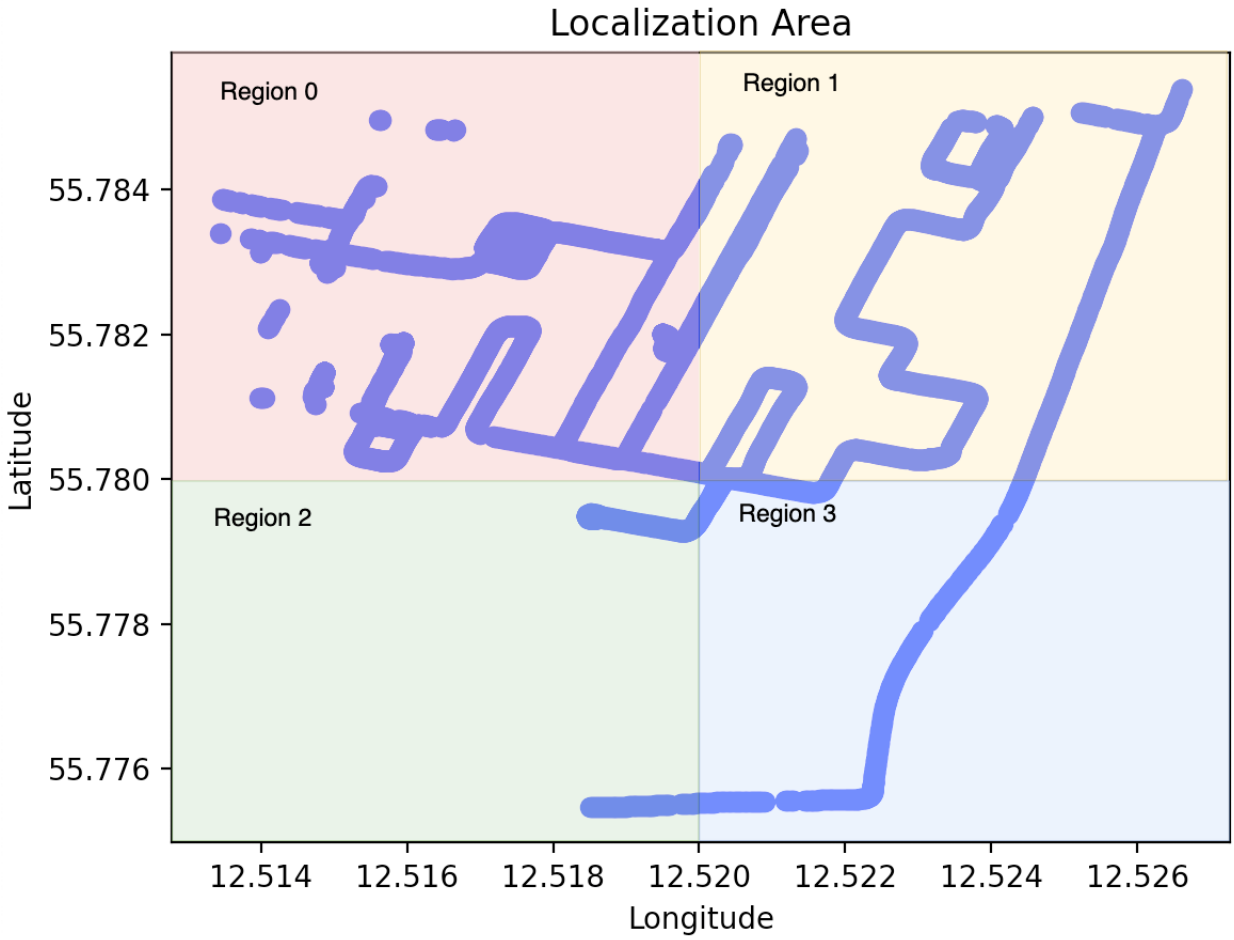

Figure 4.12: Image displaying the solid regional boundary approach for the mobile communication system dataset

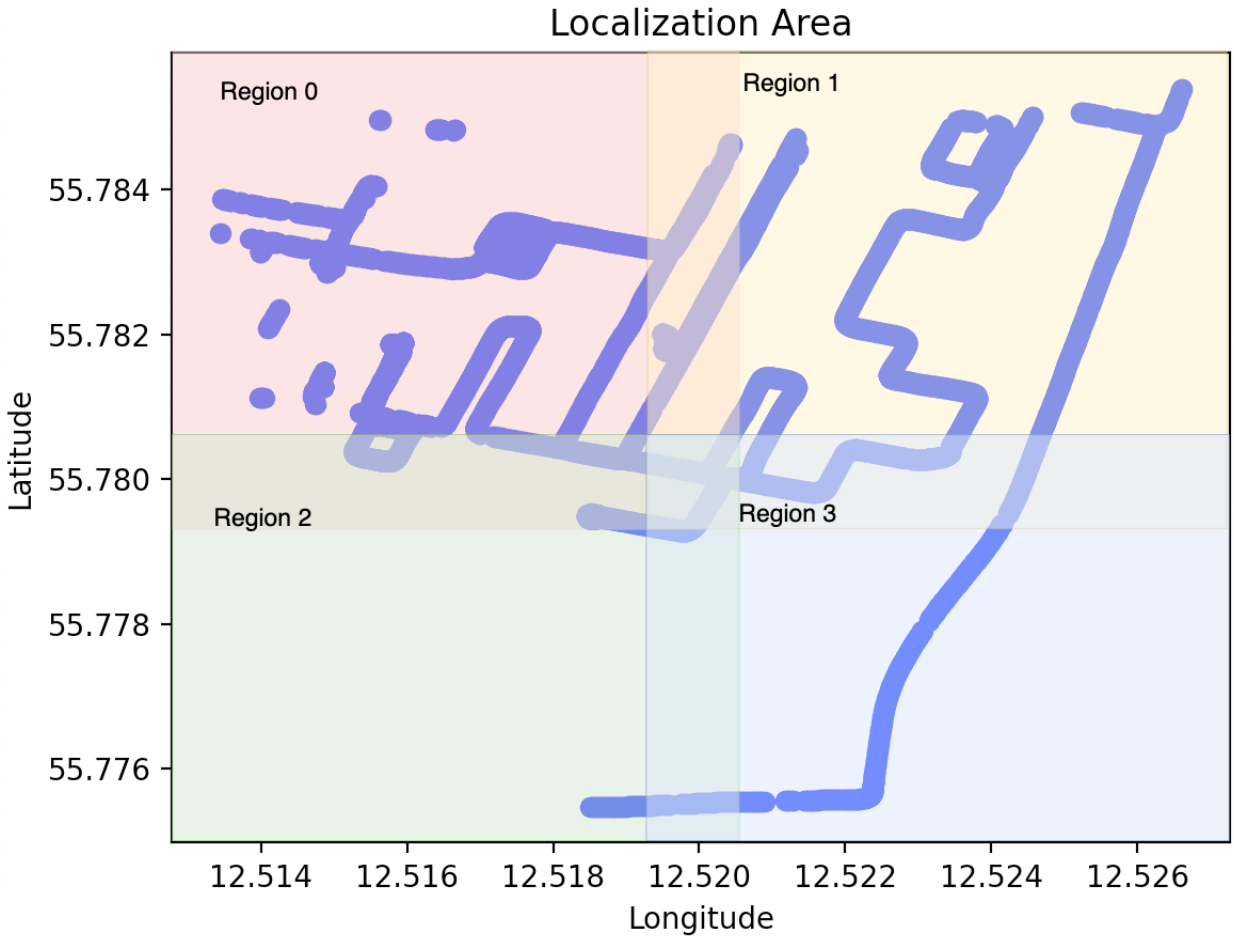

Figure 4.13: Image displaying the overlapping regional boundary approach for the mobile communication system dataset

4.2.2 Neural Network-Based Localization

The neural network-based localization stage considers two approaches. The first one has one neural network for each region in the localization area, and a general neural network for data samples that either lack images or achieve an image differencing score that lower than the define threshold, as described in the regional classification stage in algorithm 4.1. The second approach has one neural network for each region of the localization area, but omits the use of the general neural networks. The regional CNNs are trained using data points

specific to the region, whereas the general CNN is trained using all of the training samples. For the raymobtime datasets, the input to the CNN is the matrix shown in eq. 4.1 which contains the ray-tracing data associated with the data point: RSS, elevation AoA, azimuth AoA, elevation AoD, azimuth AoD, ToA and ray phase. The CNN outputs 3 values, which correspond to the x, y and z coordinates of the predicted location in normalized form. The mobile communication system dataset uses eq. 5.1 as the input to the CNN. As this is a one dimensional matrix, with only 4 data points for each sample, the convolutional layer in the CNN is bypassed as 2-dimensional convolution cannot be performed on a 1 dimensional vector. The input vector data contains SINR, RSRP, RSRQ and Power/RSSI. The FFNN for this dataset outputs 2 values that correspond to the longitude and latitude of the predicted location, in normalized form. Details regarding the neural network architecture for each dataset are described in more detail below.

Raymobtime Datasets

As mentioned previously, the raymobtime datasets, s008 and s009, use CNNs to predict the x, y and z co-ordinates of the vehicle's locations. The architecture of the networks is summarized in table 4.2. The input to the CNNs is the 2-dimensional matrix composed of radio parameters, shown in eq. 4.1. The networks then output 3 values that correspond to the x, y and z co-ordinates of the predicted locations. For all regions, the CNNs consist of only one convolutional layer that contains 4 filters of kernel size 2x2, Rectified Linear Unit (ReLU) activation, leaky ReLU with an α of 0.1 and max pooling of size 2x2. The filters are applied through python's Keras library using 2-dimensional convolution. The values of the parameters of the convolutional layer and the neural network were determined by trial and error tests that achieved the best results. Dropouts were added to the networks of the regions that were over-fitting. Dropout regularization works by randomly ignoring or dropping neurons in layers. They prevent network layers from co-adapting to prevent mistakes from previous layers, hence producing sturdier results [44]. The proposed network architecture also uses Adam optimizer for the back-propagation algorithm. Adam is a method for stochastic optimization that replaces stochastic gradient descent, as it combines the advantages of adaptive gradient algorithm and root mean square propagation [28]. The implementation of all the networks is done in python using the tensorflow library. Details regarding the python implementation of the CNNs are shown in the appendix. As seen in table 4.2, the CNNs are very similar in structure, with the main difference being the number of epochs and use of dropouts. Networks that covered a larger localization area, such as region 1 and the general CNN required more iterations and performed better with the use of dropouts. On the contrary, networks that covered a smaller localization area,

such as region 0 and region 2 either utilized dropouts or required less iterations.

Table 4.2: Table summarizing the network architecture of the CNNs for the raymobtime datasets.

Parameters	**Values**			
	Region 0	**Region 1**	**Region 2**	**General CNN**
Architecture	128x64x 32x16x3	128x64x 32x16x3	128x16x3	128x64x 32x16x3
Epochs	110	100	50	300
Batch Size	10	10	10	10
Dropout	0.2	N/A	0.2	N/A
Activation Function Hidden Layers	Sigmoid	Sigmoid	Sigmoid	Sigmoid
Activation Function Output Layer	Linear	Linear	Linear	Linear
Loss Function	MSE	MSE	MSE	MSE

To assess the performance of the network architecture, the model's training vs. validation loss is plotted over the number of iterations. The aim is to ensure the intersection of the training and validation loss curves, which indicates a good fit. A sample plot of the region 2 CNN model is shown in Fig.4.14. The Sigmoid activation function is used for the hidden layers, linear activation function for the output layer and MSE for the loss metric. On average, the training time for the regions 0, 1, 2 and the general CNN models is 2.48 minutes, 4.11 minutes, 8.61 seconds and 8.44 minutes and the prediction time per data point is 1.91 ms, 64.73 μs, 350.28 μs, 119.42 μs, respectively.

Figure 4.14: Training vs.Validation Loss for Region 2 CNN Model

Mobile Communication System Dataset

The mobile communication system dataset bypasses the convolutional layer, and uses only the FFNN to predict the locations of users. This is because this dataset uses a SISO system, hence there are only measurements from a single antenna. The FFNNs take a 1-dimensional vector composed of the radio parameters, shown in eq. 5.1, as an input and outputs two values that correspond to the longitude and latitude of the predicted locations. Since two different regional classification systems are used for this dataset, two FFNN network architectures are also considered. The first set of FFNNs, shown in table

4.3 is used for the system that includes the general FFNN. The second set of FFNNs, shown in table 4.4, is used for the system that omits the use of the general FFNN. The output of the networks is the normalized form of the longitude and latitude predictions of the users. Again, dropouts are used for the networks that tend to over fit due to their smaller coverage area that reduces their number of samples. Linear activation functions are used for the hidden layers and sigmoid activation functions are used for the output layers. Similar to the raymobtime datasets, Adam optimizer is used for the back-propagation algorithm and MSE is used as the loss metric. The training and validation loss of the networks was also plotted to optimize their performance. The networks in table 4.3 have training times of 33.6 minutes, 54.2 minutes, 25.08 minutes, 11.38 minutes, 113 minutes and prediction times of 1.001 s, 2.15 s , 1.51 s, 1.87 s, 6.3 s for region 0, 1, 2, 3 and the general FFNN. The networks in table 4.4 have training times of 33.6 minutes, 33.31 minutes, 25.08 minutes, 11.38 minutes and prediction times of 1.001 s, 1.32 s, 1.51 s, 1.87 s for regions 0, 1, 2 and 3.

Table 4.3: Table summarizing the network architecture of the FFNN for the mobile communication system dataset, including a general FNNN.

Parameters	**Values**				
	Region 0	**Region 1**	**Region 2**	**Region 3**	**General CNN**
Architecture	128x64x 32x16x3	128x64x 32x16x2	128x64x 32x16x2	128x64x 32x16x2	128x64x 32x16x2
Epochs	5000	5000	3000	6000	6000
Batch Size	10	10	10	10	10
Dropout	0.2	N/A	0.2	0.2	N/A
Activation Function Hidden Layers	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid
Activation Function Output Layer	Linear	Linear	Linear	Linear	Linear
Loss Function	MSE	MSE	MSE	MSE	MSE

Table 4.4: Table summarizing the network architecture of the FFNN for the mobile communication system dataset, without a general FFNN.

Parameters	**Values**			
	0	**1**	**2**	**3**
Architecture	128x64x 32x16x2	128x64x 32x16x2	128x64x 32x16x2	128x64x 32x16x2
Epochs	5000	3000	3000	5000
Batch Size	10	10	10	10
Dropout	0.2	N/A	0.2	N/A
Activation Function Hidden Layers	Sigmoid	Sigmoid	Sigmoid	Sigmoid
Activation Function Output Layer	Linear	Linear	Linear	Linear
Loss Function	MSE	MSE	MSE	MSE

4.2.3 System Implementation

The implementation of the proposed system can take place at either a 5G or LTE communication network. The proposed methodology is recommended for outdoor localization, although the concept can be extended to indoor use as well. The raymobtime datasets provide a realistic simulation of an outdoor localization scenario for vehicular communications, however the proposed methodology is intended for any kind of mobile receivers. The approach introduces a centralized localization scheme in which the computations take place at the base stations. Details regarding the implementation of the system in different network infrastructures are provided below:

- **LTE Network**: Computations will take place at eNodeBs. The base station will collect radio parameters while communicating with the user to use for localization predictions. The base station can also request transmissions from the user to perform

localization. Images are taken by the user and transmitted to the base station. Each base station will only perform computations within its coverage area. If the user exits the coverage area then the following base station will take over. Neighbouring eNodeBs can share information regarding the user's most recent location.

- **5G Network**: For 5G network architecture, the implementation of the system would be the same for new generation eNodeBs. As 5G networks introduce the new concept of cloud Radio Access Network (RAN)s, 5G base stations which are also known as gNodeBs are split into a Central Unit (CU) and Distributed Units (DU). In the case of gNodeBs, the computations will take place at the CU.

In the case where both LTE and 5G networks are available, the localization scheme will depend on the UEs network availability. If the UE has access to both networks, then the 5G network will take priority.

Chapter 5

Findings

5.1 Results

This section assesses the performance of the proposed methodology in section 4.2 by applying it to the algorithms described in section 4.1. To quantify the performance, the following metrics are used:

- **Localization Accuracy**: The ability of the algorithm to accurately predict locations. In this thesis, the localization accuracy is calculated using Root Mean Squared Error (RMSE) and MAE, shown in eq. 5.1 and 5.2, respectively.
- **Latency/ Delay**: The system should be able to locate users without any noticeable delay, ideally in milliseconds time-frame, in order to be feasible in a real-time localization scenario [51].
- **Availability**: Use of hardware that is readily available on users devices, rather than creating a new infrastructure [51].
- **Cost**: The cost of the localization system is also affected by the need for new hardware. Similar to the availability metric, the system should not require a new infrastructure or costly devices.
- **Energy Efficiency**: The systems ability to minimize energy consumption and not drain user's devices. Several factors can affect energy efficiency such as frequency of transmission required for localization, transmission power and computational complexity [51].

- **Reception Range**: The system's ability to overcome signal degradation and interference as a result of a larger range.
- **Scalability**: The system should be easily adaptable to new environments, and scalable to a large number of users.

The quantitative metrics, such as localization accuracy and latency are assessed again the three datasets. As previously mentions, RMSE and MAE are used as the error metric to measure localization accuracy. The results of both the Raymobtime and Mobile Communication System datasets are described in greater detail below.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y - x)^2}{n}} \tag{5.1}$$

$$MAE = \frac{\sum_{i=1}^{n} |y - x|}{n} \tag{5.2}$$

5.1.1 Raymobtime Datasets

As mentioned before, the Raymobtime-s008 and s009 data sets that are available at [29] are used. In this section, the s008 and s009 data sets are referred to as data set-1 and data set-2, respectively. First the data is preprocessed as described in section 4.2 and then the steps in the proposed algorithm are applied. To determine the performance, various adaptations of the algorithm are assessed. The results are compared with the localization error of one CNN using the input matrix of ray-tracing data described in eq. 4.1 to demonstrate the improvement in localization accuracy as a result of the regional CNN approach.

The results of the proposed technique are showcased given the two main approaches for regional classification described in section 4.2: regions with overlapping vs. solid boundaries and hybrid regional classification vs. image classification. RMSE and MAE, shown in eq. 5.1 and eq. 5.2, are used as metrics to assess the performance of our algorithm. Both metrics describe the average model prediction error, however, the RMSE assigns comparatively higher weights to large errors. As such, MAE can be used to determine the overall performance for a given data set, whereas RMSE can be used to detect the instances of outliers and false classifications. In this case, as a regional CNN approach using image classification is implemented, a higher RMSE can be linked to false classification as the regional CNNs are only trained to make predictions within their boundaries. Hence, data points that are classified into inaccurate regions can cause large errors.

The results of the different combinations of techniques are summarized in Tables 5.1 and 5.2. The results display the error for each region in the map. The first noticeable outcome is that the localization error in region 0 is significantly higher in data set-1 than data set-2. As seen in table 5.2, for region 1 and 2, the proposed approach achieved a higher accuracy. Using the same adaptation of the algorithm, region 1 has a RMSE of 8.34 m and MAE of 8.03 m for dataset-1 and a ??f RMSE 7.83 m and MAE of 8.32 m for dataset-2. Whereas, region 2 has a RMSE of 3.02 m and MAE of 4.17 m for dataset-1 and RMSE of 4.52 m and MAE of 6.22 m for dataset-2. For both regions and datasets, the localization error is fairly close in value. Meanwhile, Region 0 has a RMSE of 10.90 m and MAE of 10.91 m for dataset-1 and RMSE if 2.45 m and MAE of 3.60 m for dataset 2. The classification accuracy of data set-1 confirms that this is due to the higher number of false positive image classifications in that region (see Table 5.1). The classification accuracy is defined as the percentage of true positive classifications. Fig.5.1 displays samples of the outliers in the LIDAR images that are causing the false positives. The quality of those images is substandard in comparison to the training images shown in Fig.4.5, which makes them unidentifiable to the image differencing algorithm. The classification algorithm described in section 4.2 requires the image differencing score to be higher than the defined threshold in order for the data point to be sorted into a regional CNN. The threshold is based on trial and error computations that achieved the greatest overall localization error; in this study a 70% image match is required. This allows some variation between the vehicle's LIDAR images and the training images, while minimizing the number of wrongful classifications. As such, the lower quality images coincidentally passed the threshold and were mis-classified into region 0.

The variation in localization error between the two data sets in region 1 shows the potential of this algorithm as more image data sets become available. With a higher classification accuracy, data set 2 achieved an RMSE and MAE of 2.45 m and 3.60 m in region 0, in comparison to to the RMSE and MAE of 10.9 m and 10.38 m in data set-1. In this study, the use of street and building silhouettes to classify the regions is explored as the LIDAR images shown in Fig.4.5 which lacked a great amount of detail. However, future works can examine the use of more reliable image classification techniques such as the geographical landmarks used in [35] to reduce the chances of false positives and improve the overall error.

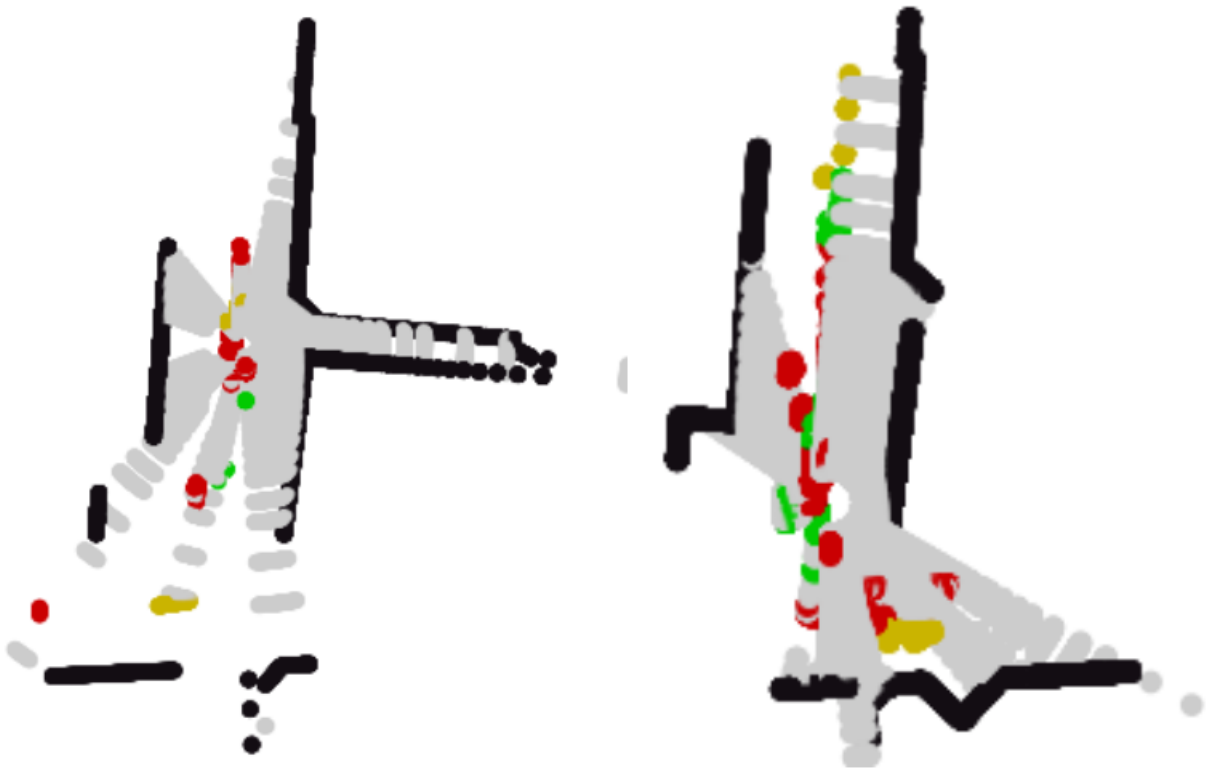

Figure 5.1: Samples of the false positive outliers in region 0

The results, summarized in table 5.2, show that the regional CNNs with the hybrid regional classification achieve the best localization accuracy with a MAE of 7.41 m and 6.55 m for data sets 1 and 2, whereas the network architecture with only 1 CNN has the highest error. This implies that the regional CNN approach allows the neural networks to better understand the trends and outcomes associated with the specific environments, hence producing more accurate predictions. The results also highlight the significance of the regional boundaries and classification techniques. The use of overlapping boundaries at neighbouring regions increased the classification accuracy from 98.14% to 99.95%, which consequently improved the localization error. This can also be seen in the localization error of data set-1 in region 0 as the use of overlapping boundaries improved the error of the region from 12.34 m to 10.9 m. However, the addition of RSS based fingerprinting to the classification technique achieved negligible improvements to the localization accuracy. This is indicated in the results as the RMSE of the regional CNNs with overlapping boundaries and image classification is 7.84 m, a drawback of only 0.05 m in comparison to the CNN with overlapping boundaries and hybrid classification. A similar trend can be seen in the

MAE as well as the results of CNNs with solid boundaries shown in table 5.2. Although the enhancement of the algorithm in terms of localization accuracy by using hybrid classification is insignificant, one added benefit that remains is the improvement with respect to computational complexity. As described in the proposed methodology, the fingerprinting based classification is a simple comparison operation with a predefined threshold, while the image classification method requires the implementation of image preprocessing and image differencing as described in Fig. 4.10. The regional classification prediction time per sample is 328 ms using the image classification approach, and 247 ms using the hybrid classification approach. Whereas the CNN prediction time is 1.91 ms, 64.73 ms, 250.29 μs and 119.42 μs for regions 0, 1, 2 and the general CNN, respectively. By comparing the computational time of the CNN predictions to the various CNN's architecture shown in table 4.2, it becomes clear that there is a trend between the required prediction time and the complexity of the network: the larger the number of neurons, hidden layers and epochs and the smaller the batch size, the greater the prediction time becomes.

Table 5.1: Table summarizing classification accuracy.

	Accuracy	
	Dataset 1	**Dataset 2**
Hybrid Regional Classification with solid boundaries	97.12%	98.14%
Image Classification with solid boundaries	97.1%	97.9%
Hybrid Regional Classification with overlapping boundaries	99.5%	99.95%
Image Classification with overlapping boundaries	98.12%	98.8%

Table 5.2: Table summarizing localization error.

	Region	RMSE (m)		MAE (m)	
Dataset		**1**	**2**	**1**	**2**
One CNN		9.44	12.51	10.91	9.82
Regional CNN with Overlapping Boundaries (Hybrid Classification)	**0**	10.90	2.45	10.38	3.60
	1	8.34	7.83	8.03	8.32
	2	3.02	4.52	4.17	6.22
	Gen. CNN	8.28	6.51	8.56	6.54
	Total	7.79	7.08	7.41	6.55
Regional CNN with Overlapping Boundaries (Image Classification)	**0**	10.78	3.07	10.46	4.33
	1	8.11	7.52	7.71	8.34
	2	2.97	3.02	4.23	4.08
	Gen. CNN	8.14	7.31	8.43	5.21
	Total	7.84	7.79	7.88	6.59
Regional CNN with Solid Boundaries (Hybrid Classification)	**0**	12.75	3.64	12.27	5.31
	1	7.68	7.63	7.50	7.75
	2	3.403	4.24	4.81	6.15
	Gen. CNN	9.09	8.59	8.81	5.48
	Total	9.36	8.08	8.43	6.05
Regional CNN with Solid Boundaries (Image Classification)	**0**	12.34	3.85	12.45	5.80
	1	8.33	7.29	8.42	8.17
	2	3.42	4.09	4.82	5.75
	Gen. CNN	8.29	9.16	7.77	6.52
	Total	8.65	8.26	8.12	7.23

5.1.2 Mobile Communication System and Satellite Images Dataset

The proposed algorithm is applied to the Mobile Communication System and Satellite Images dataset to expand the application of the technique to SISO systems and assess the performance of the image classification method against more detailed images. In this section, the dataset will be referred to as dataset-3. Similar to the two previous datasets, the performance of the algorithm is evaluated against the different adaptations of the algorithm and the singular FFNN is used as a benchmark to showcase the improvement in localization error as a result of the regional approach. As discussed in section 4.2, a new version of the algorithm that omits the use of the general neural networks was introduced for this dataset. This section will provide detailed results of the performance of both algorithms 4.1 and 4.2 when applied to the dataset. As this dataset provides the user positions in longitude and latitude format, Eq. 5.3 is used to calculate the distance between two points. The RMSE amd MAE are then calculated using Eqs. 5.1 and 5.2 respectively.

$$\begin{aligned} a &= sin^2(\Delta\psi/2) + cos\psi_1 \cdot cos\psi_2 \cdot sin^2(\Delta/2) \\ c &= 2 \cdot atan2(\sqrt{a}, sqrt1 - a) \\ d &= R \cdot c \\ &Where\, R\, is\, the\, Earth's\, radius \\ R &= 6371Km \end{aligned} \tag{5.3}$$

Localization Scheme using General FFNNs

An improvement in regional classification can be seen with the use of the satellite images in comparison to the LIDAR images. This is shown through the classification accuracy in table 5.3. The results show that the different adaptations of the algorithm have negligible effects on the classification accuracy, unlike the two previous datasets. A regional classification accuracy of 99.945%, 99.725%, 99.985% and 99.985% is obtained for the hybrid regional classification with solid boundaries, image classification with solid boundaries, hybrid regional classification with overlapping boundaries and image classification with overlapping boundaries. This is likely due to the amount of detail in the satellite images that is not present in the LIDAR images, which makes the landmarks easier to distinguish. However, similar to the Raymobtime datasets, the introduction of the hybrid regional classification also produced insignificant improvements to the classification accuracy. As such, the main benefit of this approach remains to be the reduced computational complexity.

Table 5.3: Table summarizing classification accuracy.

	Accuracy
	Dataset 3
Hybrid Regional Classification with solid boundaries	99.945%
Image Classification with solid boundaries	99.725%
Hybrid Regional Classification with overlapping boundaries	99.985%
Image Classification with overlapping boundaries	99.985%

As seen in table 5.4, this dataset shows a degradation in localization error in the deep learning stage of the algorithm in comparison to the two previous datasets. A RMSE of 24.83 m, 24.56 m, 23.35 m and 22.04 m and MAE of 11.55 m, 12.64 m, 12.04 m, 9.45 m is obtained for the hybrid classification with overlapping boundaries, image classification with overlapping boundaries, hybrid classification with solid boundaries and image classification with solid boundaries approaches. The singular FFNN, without the regional localization scheme, obtained a RMSE of 33.36 m and MAE of 15.74 m. Although there is an improvement using the regional FFNN approach with respect to the singular FFNN, it is still clear that the use of FFNNs rather than CNNs and the reduction in the number of features as a result of the SISO system had an impact on the performance of the neural networks. The increase in error is particularly evident in the localization error of the general FFNNs in the regional approach. In comparison to the above-mentioned error values for the different adaptations of the algorithm, the general FFNN achieved a RMSE of 69.08 m, 63.65 m, 68.65 m and 58.65 m and MAE of 30.43 m, 31.79 m, 30.54 m, 25.68 m, respectively. Fig. 5.2 displays an image of the predicted and actual locations, which shows that the network is unable to form an accurate mapping between the CSI data and location co-ordinates, causing false predictions. In order to further investigate this claim, the CSI data was plotted with respect to the different locations. As seen in figure 5.3, the value of the RSRP is strongest in the center of the map and the top right corner, and weakest in the top left corner and bottom right corner. Similar trends are also seen with the values of the power/RSSI shown in fig. 4.8 and the SINR. The similarity in CSI data in those locations makes it easier to falsely classify their locations. This is seen in the FFNN

predictions in fig 5.2 as the network is unable to predict the locations in the right hand side of the map. This issue can be addressed by increasing the number of antennas in the receiver, and hence increasing the number of features that are input to the neural network, allowing it to learn from more than one attribute as previously shown in the raymobtime datasets. The use of the regional networks also aids in reducing the effects of this issue as the networks are only learning the trends and outcomes of their region. By increasing the accuracy of the regional classification, and reducing the number of samples that are sorted into the general FFNN, the overall localization error of the dataset is reduced. This is shown in the results in table 5.4 as the localization error of the regional FFNNs and the overall localization error is still significantly lower than the localization error of the general FFNNs.

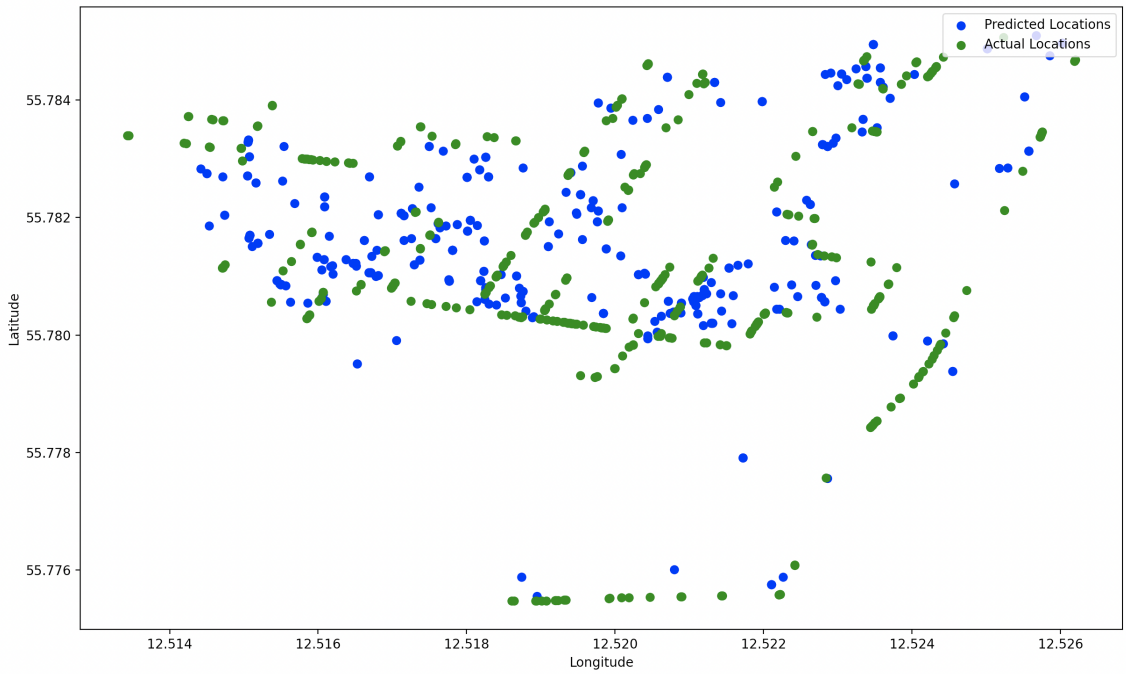

Figure 5.2: Plot displaying actual vs. predicted locations for general FFNN network

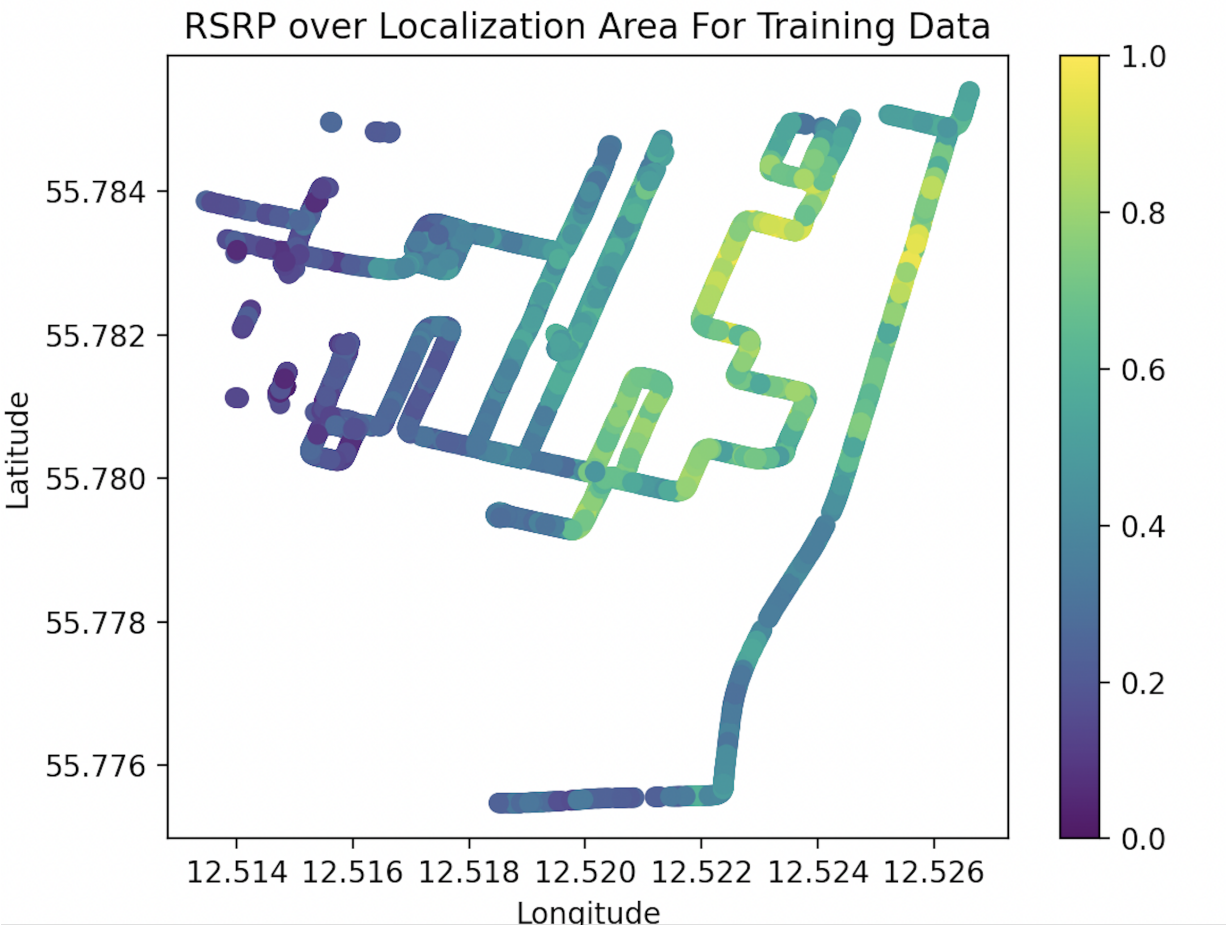

Figure 5.3: Plot displaying RSRP values over the localization area

Table 5.4: Table summarizing localization error for dataset-3

		RMSE (m)	MAE (m)
One FFNN		32.36	15.74
Regional FFNN with Overlapping Boundaries (Hybrid Classification)	**0**	7.57	3.96
	1	9.75	4.45
	2	6.29	4.04
	3	15.50	7.91
	Gen. FFNN	69.08	30.43
	Total	24.83	11.55
Regional FFNN with Overlapping Boundaries (Image Classification)	**0**	15.89	9.07
	1	7.54	3.97
	2	18.05	10.43
	3	2.59	1.85
	Gen. FFNN	63.65	31.79
	Total	24.56	12.64
Regional FFNN with Solid Boundaries (Hybrid Classification)	**0**	12.77	7.8
	1	23.60	9.31
	2	20.94	15.73
	3	1.94	1.34
	Gen. FFNN	68.65	30.54
	Total	23.35	12.04
Regional FFNN with Solid Boundaries (Image Classification)	**0**	7.80	3.09
	1	9.31	4.005
	2	20.94	13.69
	3	11.99	8.77
	Gen. FFNN	58.65	25.68
	Total	22.04	9.45

Localization Scheme without General FFNNs

To further investigate the effect of the general FFNNs on the overall localization error, the percentage of samples that were disposed into the general FFNN as a result of a lower image differencing score was calculated. The results for each dataset are summarized in table 5.7. It was observed that for dataset 3 only 6.79% and 9.99% of the data samples were being sorted into the general FFNN using the hybrid and image classification approaches, respectively. Given this information another adaptation of the algorithm, described in section 4.2, was assessed. This adaptation omits the general FFNNs and only relies on the regional networks. Table 5.5 summarizes the regional classification accuracy of this adaptation. The results show that the lack of the general FFNN had a negligible effect on the classification accuracy, as only a small number of samples was being disposed. This adaptation of the algorithm achieved a maximum regional classification accuracy of 99.98% for the image classification with overlapping boundaries approach, which is comparable to the performance of the algorithm with the general FFNNs.

Table 5.5: Table summarizing classification accuracy.

	Accuracy
Hybrid Regional Classification with solid boundaries	99.37%
Image Classification with solid boundaries	99.40%
Hybrid Regional Classification with overlapping boundaries	99.52%
Image Classification with overlapping boundaries	99.98%

Table 5.6: Table summarizing localization error, without general FFNNs

		RMSE (m)	MAE (m)
One FFNN		32.36	15.74
Regional FFNN with Solid Boundaries (Hybrid Classification)	**0**	25.77	4.533
	1	19.48	10.74
	2	23.4	9.33
	3	24.38	12.94
	Total	16.31	9.25
Regional FFNN with Overlapping Boundaries (Image Classification)	**0**	10.62	4.94
	1	18.87	8.07
	2	7.55	5.16
	3	20.18	10.34
	Total	14.94	6.55
Regional FFNN with Overlapping Boundaries (Hybrid Classification)	**0**	12.99	6.77
	1	18.79	5.98
	2	28.67	9.77
	3	20.56	10.72
	Total	15.24	8.41
Regional FFNN with Solid Boundaries (Image Classification)	**0**	8.51	2.96
	1	19.55	8.88
	2	5.61	3.65
	3	11.01	8.24
	Total	14.29	5.83

Table 5.6 summarizes the localization accuracy of the algorithm without the general FFNNs. Again, both RMSE and MAE are used as metrics to describe the error for each region. A RMSE of 16.31 m, 14.94 m, 15.24 m and 14.29 m and MAE of 9.25 m, 6.55 m, 8.41 m, and 5.83 m is obtained for the hybrid classification with solid boundaries,

image classification with overlapping boundaries, hybrid classification with overlapping boundaries and image classification with solid boundaries approaches. The results show a notable improvement from the results previously discussed in table 5.4, using the general FFNNs. A common trend between all adaptations as well as regions is the higher RMSE value with respect to MAE. For example, in the hybrid classification with solid boundaries approach, region 0 achieves a RMSE of 25.77 m and a MAE of 4.533 m. This trend continues for all regions and all adaptations of the algorithm, including the singular FFNN. As previously mentioned in this thesis, a larger RMSE to MAE ratio indicates that there are a few predictions in the testing samples that have very large errors. Various factors can contribute to this. For example, false regional classifications can result in the sample being trained by the wrong neural network, hence producing an inaccurate prediction and causing large errors. Moreover, as the proposed methodology is being applied to a SISO system, the lack of input data to the neural network limits its learning capabilities. However, an improvement in the system's learning capabilities can be seen through the use of the regional FFNNs. The results show a significant improvement in MAE and RMSE with the use of the regional FFNN approach in comparison to the singular FFNN as the singular FFNN achieves an MAE of 15.74 m, whereas the regional FFNN achieves a minimum MAE of 5.83 m.

5.2 Performance Evaluation

The performance of the algorithm can be assessed based on it's results when applied to the three datasets in sections 5.1.1 and 5.1.2. Based on the results, dataset 2 achieved the best localization accuracy in terms of RMSE, dataset 3 achieved the best regional classification accuracy and best localization accuracy in terms of MAE and datasets 1 and 2 achieved the smallest delay in terms of computational time. As the regional classification accuracy of the three datasets was fairly close in value, another metric was calculated to assess the algorithm's efficiency in predicting regions in each dataset. The percentage of samples in the general neural network was calculated. This value refers to the number of samples that were not easily distinguishable by the image differencing algorithm and were disposed to the general neural network for localization; i.e. the smaller the value, the better the performance. Table 5.7 provides a summary of this metric for both the image classification and hybrid classification adaptations of the algorithm. Although it was previously determined that dataset 3 has the best regional classification accuracy, the performance of the adaptations was not clear as the results showed an accuracy of 99.945%, 99.725%, 99.985% and 99.985% for hybrid regional classification with solid boundaries, image classification

with solid boundaries, hybrid classification with solid boundaries and image classification with overlapping boundaries. It can now be shown that the hybrid regional classification approach achieves the best regional classification accuracy as it minimizes the number of samples being disposed to the general neural network, with only 6.79% of the testing samples being disposed. This metric also provides a justification for the large variance in localization error in table 5.4 between the error of the general FFNNs and the total error. Although the general FFNNs had considerably high RMSE and MAE values, the total error of the dataset was still able to remain low as the general FFNN samples make up a very small portion of the total testing sample set. Since the percentage of samples in the general neural network was fairly low for dataset 3, another version of the algorithm was adapted for this dataset, which omits the use of the general FFNNs. The results of this new adaptation showed that the absence of the general FFNN has a negligible effect on the classification accuracy, as a maximum classification accuracy of 99.98% was achieved. This change also had a positive effect on the overall localization accuracy, demonstrating that the use of only the regional FFNNs improved the network's predictions. Dataset 3 achieved a minimum MAE of 9.45 m with the use of the general FFNNs and 5.83 m without.

Table 5.7: Table summarizing the number of samples disposed to the general neural networks

	% of Samples in General Neural Network	
	Image Classification	**Hybrid Classification**
Dataset 1	22.27%	22.27%
Dataset 2	43%	38%
Dataset 3	9.99%	6.79%

The performance of the proposed methodology is also comparable to traditional and recent outdoor localization methods discussed in section 3. Table 5.8 provides a comparison of the error per square meter of the two proposed methodologies in this thesis: localization using LIDAR images and CNNs and localization using satellite images and FFNNs, with the error of state of the art outdoor localization technologies. The error per square meter is the normalized value of the mean error, given the dimensions of the localization area. This value provides a good understanding of the performance of the algorithms in different environments. There are two common trends observed from the summarized results. Firstly, the use of only FFNN based localization without the integration of other technologies such as MIMO can yield higher errors. This is seen in [52] as they achieve a

localization error of 19.9 m and an error per square meter of 0.00028348, which is fairly high compared to other work. Secondly, the use of only vision data can also yield higher errors. This is seen in both [35] and [24] as they achieve average errors of 28.4739 m and 676.7 m, respectively. The work in [24] provides an example of the potential of vision based data in a hybrid architecture. They introduce a second adaptation of their algorithm that also utilizes Markov localization along with image detection, which significantly improves the accuracy and achieves an average error of 16.39 m. The proposed methodology also highlights the improvement of localization error as a result of integrating FFNN and vision based technologies. The outdoor localization using satellite images and FFNNs method achieved an error per square meter of 0.00000398, which is an improvement in comparison to the fingerprint-based localization using FFNN.

Table 5.8: Table summarizing comparison of results between localization technologies discussed in section 3

Method	**Localization Area (m)**	**Mean Error (m)**	**Error per Square Meter**
Fingerprint-Based Localization using FFNN [52]	360 x 195	19.9	0.00028348
Neural Network-Based Localization using Massive MIMO [13]	196 x 303	9.60	0.00016165
Supervised Machine Learning-Based Localization using MIMO [40]	316 x 316	5	0.00005007
R-CNN Vision-Based Localization [35]	70 x 30	28.4739	0.013559
Vision-Based CVM-Net with Markov Localization [24]	10,000 x 5,000	16.39	0.00000033
Vision-Based CVM-Net Localization [24]	10,000 x 5000	676.7	0.00001353
Outdoor Localization using LIDAR Images and CNNs	900 x 300	6.05	0.00002241
Outdoor Localization using Satellite Images and FFNNs	1466 x 1000	5.83	0.00000398

The performance of the proposed methodology is also assessed against the metrics described in section 5.1. This is discussed in more detail below:

- **Localization Accuracy**: The localization accuracy of the proposed methodology is applied to the three datasets and the results are described in great detail in sections 5.1.1 and 5.1.2. Dataset 2 achieves the greatest overall localization accuracy, with a RMSE of 7.41 m and MAE of 6.55 m using the regional CNN approach with overlapping boundaries. Dataset 3 obtains a MAE of 5.83 m, but a higher RMSE of 14.29 m, which can be an indication of large errors in some of the sample of the dataset. The proposed methodology shows promising results that are comparable to some of the best existing outdoor localization technologies such as GPS.

- **Latency/ Delay**: The proposed methodology achieves negligible delay in datasets 1 and 2 using LIDAR images and CNNs, with an image processing and regional prediction time of 328 ms and CNN prediction time of 1.91 ms, 64.73 μs, 350.28 μs and 119.42 μs for regions 0, 1, 2 and the general CNN. An increased delay in terms of image processing and regional classification is observed in dataset 3. Due to the larger and more detailed satellite images the image difference algorithm achieves a processing and prediction time of 26.7 s. Although this delay is considerably high, the potential of the algorithm is demonstrated using the less detailed LIDAR images. The results display a trade-off between regional classification accuracy and latency. The judgement between classification accuracy and latency can be made depending on the application of the system and the speed of receiver. For example, a system with fixed receivers is less susceptible to longer delays whereas, a vehicular communications system with fast moving cars would require negligible delays for successful localization.

- **Availability**: The proposed system takes advantage of current LTE infrastructure as well as up coming 5G infrastructure. As technology has become more advanced, LIDAR sensors and cameras are fully equipped on most vehicles and devices. Dataset 3 uses satellite images, which utilizes satellite infrastructure.

- **Cost**: There are no additional costs associated with the implementation of this localization system due to the abundant availability of LIDAR sensors and cameras in most vehicles and devices today.

- **Energy Efficiency**: The proposed system takes advantage of information that is readily available to the base station by forming a communication link, such as RSS, AoA, AoD and ToA. Hence, the number of transmissions for localization can be

reduced. The use of machine learning eliminates the need for complex mathematical computations that are traditionally used in localization techniques. Although the networks needs to be trained, which can take up to a few minutes, this process only needs to be completed once. Moreover, the localization algorithm takes place at the base station rather than the user's devices, which reduces impact to their battery life.

- **Reception Range**: The system is capable of overcoming signal degradations and environmental obstructions as the provided datasets take place in outdoor scenarios. Two of the datasets utilize the mmWave band, which particularly has significant attenuation properties. Moreover, The proposed system uses a centralized method with only one base station, without the need for any excess RPs or repeaters. The first two datasets cover a terrain size of 900 m by 300 m and the third dataset covers a terrain size of 1.446 Km by 1 Km.

- **Scalability**: The scalability of the algorithm is demonstrated through its application to 3 different datasets, each with a size of at least 10,000 samples. The performance of the proposed methodology was applied to 2 maps in different parts of the world: one in Arlington, Virginia and one at the University of Denmark. The adaptability of the system was also proven through it's application to a MIMO system in the mmWave band as well as a SISO system in the LTE band. Although the trained neural networks rely heavily on the datasets, the system can be easily updated with new data as the radio environment evolves.

5.3 Future Work

Future work will focus on exploring the use of different image classification algorithms to reduce computational complexity and improve classification accuracy. Datasets 1 and 2, which used LIDAR images, achieved good computational complexity in terms of prediction time. However, their classification accuracy was sub-par in comparison to dataset 3, which used satellite images. Particularly, dataset 2 which sorted 43% of the samples into the general CNN. Although dataset 3 obtained very high classification accuracy and also managed to maintain low localization errors without the use of a general FFNN, the size of the satellite images vastly affected the computational complexity and yielded a prediction time of 26.7 s. This makes the algorithm difficult to implement in a fast moving environment. As the implementation of the proposed technique continues, the next steps should consider

the use of an image classification algorithm that provides high classification accuracy while maintaining a low prediction time, such as CNNs.

The integration of outlier detection and data filtering techniques should also be examined. The results showed a higher RMSE than MAE which indicates large errors. This is emphasized even more in the results of dataset 3. Future work should implement outlier detection techniques to remove outliers from the dataset that can cause large errors. Examples of corrupt images were also observed in dataset 2, shown in fig. 5.1. The use of image reconstruction techniques to improve such images should be explored to make the localization system less susceptible to errors. In terms of data filtering, zero padding was applied to the CSI data in datasets 1 and 2. Future work should also experiment with implementing different filtering methods, such as interpolation, and assess the effect on the performance.

The algorithm should also be expanded to a larger localization area. As the thesis proposes a centralized method consisting of only one base station per localization area, the algorithm should be scaled and tested for the cell coverage range of 4G and 5G networks. Overall, the algorithm achieved good localization accuracy and acceptable delay which makes it comparable to GPS performance. Next, the algorithm should be extended to detect user direction as well.

Finally, details regarding the implementation of the system in LTE and 5G networks should be explored. The proposed methodology in the thesis provides a brief description of the system implementation, however, it overlooks certain details such as the handover protocol when users are moving between coverage areas.

5.4 Conclusion

This thesis introduced a localization technique using a two-stage solution comprised of regional classification and deep learning using CNNs, FFNNs and LIDAR and satellite images. Two interpretations of regional classification are proposed: a hybrid approach using fingerprinting and image differencing and standalone image differencing. The proposed methodology also explored the use of two ways of defining regional boundaries: regions with solid boundaries and regions with overlapping boundaries. The algorithm is applied to 3 datasets, the Raymobtime s008 and s009 simulated datasets [29] that model a MIMO 5G system in the neighbourhood of Rosslyn in Arlington, Virginia and the Mobile Communication System and Satellite Images dataset [12] which contains real measurements from an LTE-A SISO system at the University of Denmark.

For the Raymobtime datasets, the map of Rosslyn is split into 3 distinct regions that are distinguishable through their landmarks and the lidar images collected by vehicles are used to detect the landmarks in the localization territory to predict the receiver's region. Once the vehicle is classified into a region, the user's data is used as an input into the regional CNN to predict their location co-ordinates. An input matrix is created for the CNNs containing the user's RSS, AoA, AoD, ToA and ray phase for all 25 antennas in the vehicle. The ray-tracing data is also used as an input to a singular CNN where the predictions are used as a benchmark to assess the performance of the algorithm. The results show that the regional CNN approach with hybrid regional classification and overlapping boundaries achieved the highest classification accuracy and lowest localization error, with a MAE of 7.41 m and 6.55 m for data sets 1 and 2, respectively.

The University of Denmark's map is split into 4 regions. In this dataset, the provided satellite images are cropped to only show the the receiver's surrounding landmarks rather than the general region. As such, the algorithm was slightly adjusted to include various training images for every region to reduce the number of regional neural networks required for localization. This dataset also required the use of FFNNs rather than CNNs due to the lack of features in the SISO system. This change was implemented by simply skipping the convolutional layer in the CNN architecture. Similar to the two previous datasets, the satellite images are used to classify the user's region and the FFNN is used to predict the location co-ordinates. A 1-dimensional input vector was created for the FFNN using SINR, RSRP, RSRQ and power/RSSI. The results showed that this dataset was able to achieve better regional classification accuracy with the use of the satellite images, however, the lack of CSI data and the use of FFNNs rather than CNNs reduced the network's ability to form accurate predictions and increased the overall localization error. Due to this discovery, a second version of the proposed methodology was used for this dataset, which omit the use of the general neural networks. The implementation of this new architecture showed that the lack of the general FFNNs had a minimum impact on the regional classification accuracy and a positive influence on the localization accuracy. The proposed methodology achieved a localization error of 5.83 m

Overall, the proposed methodology shows promising results. The use of the different datasets demonstrated the the success and adaptability of the algorithm and also outlined the benefits and shortcomings of the different adaptations. The results showed that the regional classification approach is capable of achieving higher localization accuracy than a single neural network. The results also displayed the advantages of a MIMO system over a SISO system in terms of gaining better predictions from the neural networks. Finally, an improvement in regional classification accuracy was demonstrated through the use of satellite images, rather than LIDAR images, as they provide more detail. As more localiza-

tion datasets containing images and visual information become available, future works will incorporate the use of 5G MIMO systems data with camera or satellite images to improve localization error.

References

[1] Agilent 3gpp long term evolution: System overview, product development, and test challenges. https://literature.cdn.keysight.com/litweb/pdf/5989-8139EN.pdf?id=1431418.

[2] Illustrated guide to lstm's and gru's: A step by step explanation. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21, journal=Medium, publisher=Towards Data Science, author=Phi, Michael, year=2020, month=Jun.

[3] Linear regression. http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm.

[4] Lte enb tutorial: What is the lte enb? https://www.artizanetworks.com/resources/tutorials/what_lteenb.html.

[5] 3GPP. Vocabulary for 3GPP Specifications. Technical Specification (TS) 21.905, 3rd Generation Partnership Project (3GPP), 07 2020. Version 17.0.0.

[6] Muhammad Alrabeiah, Umut Demirhan, Andrew Hredzak, and Ahmed Alkhateeb. Computer vision aided URLL communications: Proactive service identification and coexistence. *CoRR*, abs/2103.10419, 2021.

[7] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition, 2016.

[8] Abebe Belay, Hsin-Piao Lin, Getaneh Tarekegn, Yirga Munaye, and Lei Yen. An indoor and outdoor positioning using a hybrid of support vector machine and deep neural network algorithms. *Journal of Sensors*, 2018:1–12, 12 2018.

[9] Ralf Bill, Clemens Cap, Kofahl M, and Thomas Mundt. Indoor and outdoor positioning in mobile environments—a review and some investigations on wlan-positioning. *Annals of Gis / Geographic Information Sciences*, 10:91–98, 12 2004.

[10] M. Bshara, U. Orguner, F. Gustafsson, and L. Van Biesen. Fingerprinting localization in wireless networks based on received-signal-strength measurements: A case study on wimax networks. *IEEE Transactions on Vehicular Technology*, 59(1):283–294, 2010.

[11] Houria Chabbar and Mouhcine Chami. Indoor localization using wi-fi method based on fingerprinting technique. In *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, pages 1–5, 2017.

[12] Jakob Thrane; Henrik Lehrmann Christiansen. Mobile communication system measurements and satellite images, 2019.

[13] Alexis Decurninge, Luis García Ordóñez, Paul Ferrand, He Gaoning, Li Bojie, Zhang Wei, and Maxime Guillaud. Csi-based outdoor localization for massive mimo: Experiments with a learning approach. In *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–6, 2018.

[14] Marina Din, Norziana Jamil, Jacentha Maniam, and Mohamad A Mohamed. Review of indoor localization techniques. *International Journal of Engineering and Technology(UAE)*, 7:201–204, 04 2018.

[15] M. Elbes, E. Almaita, T. Alrawashdeh, T. Kanan, S. AlZu'bi, and B. Hawashin. An indoor localization approach based on deep learning for indoor location-based services. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pages 437–441, 2019.

[16] A. ElNashar, M. A. El-saidny, and M. Sherif. *LTE Network Architecture and Protocols*, pages 1–45. 2014.

[17] M. Elsayed and M. Erol-Kantarci. Ai-enabled future wireless networks: Challenges, opportunities, and open issues. *IEEE Vehicular Technology Magazine*, 14(3):70–77, 2019.

[18] M. Elsayed and M. Erol-Kantarci. Radio resource and beam management in 5g mmwave using clustering and deep reinforcement learning. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6, 2020.

[19] M. Giordani, A. Zanella, and M. Zorzi. Millimeter wave communication in vehicular networks: Challenges and opportunities. In *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, pages 1–6, 2017.

[20] A. Gupta and R. K. Jha. A survey of 5g network: Architecture and emerging technologies. *IEEE Access*, 3:1206–1232, 2015.

[21] S S Hadi and T C Tiong. Adaptive modulation and coding for LTE wireless communication. *IOP Conference Series: Materials Science and Engineering*, 78:012016, apr 2015.

[22] Onel Harrison. Machine learning basics with the k-nearest neighbors algorithm. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761, Jul 2019.

[23] Minh Tu Hoang, Brosnan Yuen, Xiaodai Dong, Tao Lu, Robert Westendorp, and Kishore Reddy. Recurrent neural networks for accurate rssi indoor localization. *IEEE Internet of Things Journal*, 6(6):10639–10651, 2019.

[24] Sixing Hu and Gim Hee Lee. Image-based geo-localization using satellite imagery, 2019.

[25] Mai Ibrahim, Marwan Torki, and Mustafa Elnainay. Cnn based indoor localization using rss time-series. *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018.

[26] L. C. Jain. *Recurrent neural networks: design and applications*. CRC Press, 2000.

[27] Ojas Kanhere and Theodore S. Rappaport. Position locationing for millimeter wave systems. *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018.

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[29] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath. 5g mimo data for machine learning: Application to beam-selection using deep learning. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9, 2018.

[30] A. Klautau, N. González-Prelcic, and R. W. Heath. Lidar data for deep learning-based mmwave beam-selection. *IEEE Wireless Communications Letters*, 8(3):909–912, 2019.

[31] A. R. Kulaib, R. M. Shubair, M. A. Al-Qutayri, and Jason W. P. Ng. An overview of localization techniques for wireless sensor networks. *2011 International Conference on Innovations in Information Technology*, 2011.

[32] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: A quantitative comparison. *Computer Networks*, 43:499–518, 11 2003.

[33] A. F. Molisch. *Technical Challenges of Wireless Communications*, pages 27–36. 2011.

[34] Michael A. Nielsen. Neural networks and deep learning, 2018.

[35] Sivapong Nilwong, Delowar Hossain, Shin-Ichiro Kaneko, and Genci Capi. Deep learning-based landmark detection for mobile robot outdoor localization. *Machines*, 7(2):25, 2019.

[36] Takayuki Nishio, Yusuke Koda, Jihong Park, Mehdi Bennis, and Klaus Doppler. When wireless communications meet computer vision in beyond 5g. 10 2020.

[37] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks, 2015.

[38] Gintautas Palubinskas. Mystery behind similarity measures mse and ssim. *2014 IEEE International Conference on Image Processing (ICIP)*, 2014.

[39] Filippo Palumbo, Paolo Barsocchi, Stefano Chessa, and Juan Carlos Augusto. A stigmergic approach to indoor localization using bluetooth low energy beacons. In *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2015.

[40] Hessam Pirzadeh, Chenwei Wang, and Haralabos Papadopoulos. Machine-learning assisted outdoor localization via sector-based fog massive mimo. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6, 2019.

[41] Jait Purohit, Xuyu Wang, Shiwen Mao, Xiaoyan Sun, and Chao Yang. Fingerprinting-based indoor and outdoor localization with lora and deep learning. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6, 2020.

[42] Shahram ShahbazPanahi and Yindi Jing. Chapter 9 - recent advances in network beamforming. In Rama Chellappa and Sergios Theodoridis, editors, *Academic Press Library in Signal Processing, Volume 7*, pages 403–477. Academic Press, 2018.

[43] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, Mar 2020.

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[45] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao. Application of machine learning in wireless networks: Key techniques and open issues. *IEEE Communications Surveys Tutorials*, 21(4):3072–3108, 2019.

[46] H. Suo, J. Wan, L. Huang, and C. Zou. Issues and challenges of wireless sensor networks localization in emerging applications. In *2012 International Conference on Computer Science and Electronics Engineering*, volume 3, pages 447–451, 2012.

[47] Z. Tao, Ph. Bonnifait, V. Frémont, and J. Ibañez-Guzman. Mapping and localization using gps, lane markings and proprioceptive sensors. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 406–412, 2013.

[48] B. Viel and M. Asplund. Why is fingerprint-based indoor localization still so hard? In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, pages 443–448, 2014.

[49] Cheng-Xiang Wang, Marco Di Renzo, Slawomir Stańczak, Sen Wang, and Erik G. Larsson. Artificial intelligence enabled wireless networking for 5g and beyond: Recent advances and future challenges, 2020.

[50] F. Zafari, A. Gkelias, and K. K. Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys Tutorials*, 21(3):2568–2599, 2019.

[51] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys Tutorials*, 21(3):2568–2599, 2019.

[52] H. Zhang, Z. Zhang, S. Zhang, S. Xu, and S. Cao. Fingerprint-based localization using commercial lte signals: A field-trial study. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–5, 2019.

APPENDICES

Sample code displaying CNN structure and implementation:

```
batch_size = 10
epochs = 150
num_classes = 3

model0 = Sequential()
model0.add(Conv2D(4, kernel_size=(2, 2),activation='relu',
input_shape=(8,25,1),padding='same'))
model0.add(MaxPooling2D((2, 2),padding='same'))
model0.add(Flatten())
model0.add(Dense(64, activation='sigmoid'))
model0.add(Dense(32, activation='sigmoid'))
model0.add(Dense(16, activation='sigmoid'))
model0.add(Dense(num_classes, activation='linear'))

model0.compile(loss='mean_squared_error',
optimizer=keras.optimizers.Adam(),
metrics=['accuracy'])

model0.summary()

train = model0.fit(inputTrain0,
POS_Batch0_train, batch_size=batch_size,
epochs=epochs, validation_split=0.1)

loss = train.history['loss']
valLoss = train.history['val_loss']
```

```
plt.plot(loss)
plt.plot(valLoss)
plt.title('model train vs validation loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper right')
plt.show()

predictedPOS = model0.predict(inputTest0)
```

Sample code displaying FFNN structure and implementation:

```
batch_size = 50
epochs = 5000 #5000

model1 = Sequential()
model1.add(Dense(256, input_dim=4, activation='sigmoid'))
model1.add(GaussianNoise(0.1))
model1.add(Dense(256, activation='sigmoid'))
model1.add(Dense(128, activation='sigmoid'))
model1.add(Dense(128, activation='sigmoid'))
model1.add(Dense(64, activation='sigmoid'))
model1.add(Dense(32, activation='sigmoid'))
model1.add(Dense(16, activation='sigmoid'))
model1.add(Dense(2, activation='linear'))
print(model1.summary())

model1.compile(loss='mean_squared_error',
optimizer=keras.optimizers.Adam(),metrics=['accuracy'])

train = model1.fit(inputVectorTrain1, POS_train1 ,
batch_size=batch_size,epochs=epochs, validation_split=0.1)

loss = train.history['loss']
valLoss = train.history['val_loss']
```

```
plt.plot(loss)
plt.plot(valLoss)
plt.title('model train vs validation loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper right')
plt.show()

predictedPOS1 = model1.predict(inputVectorTest1)
```

Sample code displaying image processing implementation:

```
def processImg(img):
size = 256
scalingFactor = 0.25
#Grayscale to change to 2D image
imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imwrite("001.png", imgGray)
#Blur image
ksize = (5,5) #best blur 10
blurredImg = cv2.blur(imgGray, ksize, cv2.BORDER_DEFAULT)
cv2.imwrite("002.png", blurredImg)
#Convert to black and white
val, imgbw = cv2.threshold(blurredImg, 150, 255, cv2.THRESH_BINARY)
cv2.imwrite("003.png", imgbw)
#scale image to reduce size
scaledImg = cv2.resize(imgbw,
(int(size*scalingFactor), int(size*scalingFactor)))
cv2.imwrite("004.png", scaledImg)
return scaledImg
```

Sample code displaying image differencing implementation:

```
def imageDifference(img1, trainingImages):
path = "mapbox_api/" + str(int(trainingImages[0][0])) + ".png"
trainingImg = cv2.imread(path)
processedTrainingImg = processImg(trainingImg)
```

```
score, diff = ssim(img1, processedTrainingImg, full=True)
diff = (diff * 255).astype("uint8")
region = 0
for i in range(0, len(trainingImages)):
for j in range(0, len(trainingImages[i])):
path = "mapbox_api/" + str(int(trainingImages[i][j])) + ".png"
trainingImg = cv2.imread(path)
processedTrainingImg = processImg(trainingImg)
scoreNew, diffNew = ssim(img1, processedTrainingImg, full=True)
diffNew = (diffNew * 255).astype("uint8")
if(scoreNew > score):
score = scoreNew
diff = diffNew
region = i
if(score < 0.6): #best results 0.7
region = 100
else:
region = i
print("SSIM: {}".format(score))
print(region)
return score, diff, region
```