BEAM PREDICTION USING CONVOLUTIONAL NEURAL NETWORK AND ARTIFICIAL NEURAL NETWORK

CHARITH DISSANAYAKE

MASTER OF ENGINEERING
IN
COMPUTER ENGINEERING

SCHOOL OF INFORMATION TECHNOLOGY
MAE FAH LUANG UNIVERSITY
2023

BEAM PREDICTION USING CONVOLUTIONAL NEURAL NETWORK AND ARTIFICIAL NEURAL NETWORK

CHARITH DISSANAYAKE

THIS THESIS IS A PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING
IN
COMPUTER ENGINEERING

SCHOOL OF INFORMATION TECHNOLOGY
MAE FAH LUANG UNIVERSITY
2023

BEAM PREDICTION USING CONVOLUTIONAL NEURAL NETWORK AND ARTIFICIAL NEURAL NETWORK

CHARITH DISSANAYAKE

THIS THESIS HAS BEEN APPROVED
TO BE A PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ENGINEERING
IN
COMPUTER ENGINEERING
2023

EXAMINATION COMMITTEE

Adisorn Leelasantitham
.....CHAIRPERSON
(Assoc. Prof. Adisorn Leelasantitham, Ph. D.)

P. Temdee
.....ADVISOR
(Assoc. Prof. Punnarumol Temdee, Ph. D.)

.....EXAMINER
(Asst. Prof. Roungsan Chaisricharoen, Ph. D.)

.....EXAMINER
(Assoc. Prof. Nattapol Aunsri, Ph. D.)

Chayapol Kamyod
.....EXAMINER
(Chayapol Kamyod, Ph. D.)

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my supervisor Associate Professor Dr. Punnarumol Temdee of my Master's research study for her great guidance, immense support, knowledge and patience extended towards me. Her guidance helped me throughout the research and writing of this thesis. I have no words to express my gratitude for her unwavering encouragement. Then I would like to thank the honorary examiner panel of my master's research. I would like to thank you for providing me your valuable time. Furthermore, I would like to express my thankfulness to the president, all the academic, administrative and non-academic staff of Mae Fah Luang University, Thailand for providing me with the research facilities. Dean, all the academic and non-academic staff of the School of Information Technology, Mae Fah Luang University, I appreciate your precious helping hand given to me to make my Master's degree succeed.

Without you, my loving mother, father, and sister nothing would have been even possible. Your love, prayers and sacrifices made me who I am today. To my brother-in-law, I really appreciate your continuous encouragement and never-ending support. Last but not least, to the most compassionate wife, Minoli; I would never be able to put my thankfulness in words. Your never-ending support, encouragement, commitment and sacrifices will be forever engraved in my heart.

Charith Dissanayake

Thesis Title	Beam Prediction using Convolutional Neural Network and Artificial Neural Network
Author	Charith Dissanayake
Degree	Master of Engineering (Computer Engineering)
Advisor	Assoc. Prof. Punnarumol Temdee, Ph. D.

ABSTRACT

In the Sixth-Generation mobile network, beam training methods will be used due to energy loss that requires more computation. Therefore, this study introduces a beam prediction approach using machine learning techniques with data preprocessing methods. The model contains a convolutional neural network and a multilayer perceptron network that accepts images and global positioning system (GPS) data as inputs. This study used 15281 images and 15281 GPS data samples from an online dataset to evaluate the model. The proposed approach was tested with 64 and 32 beam indexes output classes and resulted in 88.98% and 85.72% accuracies respectively. Furthermore, the model showed precision and recall values of 89.09% and 89.15% for 64 classes and 85.87% and 85.66% for 32 classes respectively.

Keywords: Sixth-Generation Mobile Network, Beam Training, Beam Prediction, Convolutional Neural Network, Multilayer Perceptron

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

Table **Page**

LIST OF FIGURES

LIST OF FIGURES (continued)

ABBREVIATIONS AND SYMBOLS

AoA	Angle of Arrival
ANN	Artificial Neural Network
BS	Base Station
BF	Beamforming
CNN	Convolutional Neural Network
FN	False Negatives
FP	False Positives
5G	Fifth Generation
GPS	Global Positioning System
KNN	K-Nearest Neighbors
LiDAR	Light Detection and Ranging
LT	Lookup Table
MLP	Multilayer Perceptron
MIMO	Multiple-Input Multiple-Output
PCA	Principal Component Analysis
RADAR	Radio Detection and Ranging
RF	Random Forest
RNN	Recurrent Neural Network
RGB	Red- Green- Blue
6G	Sixth Generation
SVM	Support Vector Machine
SSB	Synchronization Signal Block
TP	True Positives

CHAPTER 1

INTRODUCTION

1.1 Background of the Research

Currently, data transmission over distance is a vital technology that is integrated into a large number of day-to-day equipment like mobile phones, computers, televisions and even vehicles. Data transmission can be done using wired or wireless transmission media. Today, most of the applications tend to use wireless method since it is more convenient to use due to high mobility and simplicity of installation and maintenance.

1.1.1 The 5G and 6G Communication Systems

Cellular network communication is one of the wireless communication technologies that is mainly used in mobile phones. Cellular communication technology has been developing for more than four decades (Read, n.d.). The development of wireless communication is important with respect to several aspects like data rate, latency, reliability, user density, etc. Therefore, a large number of advancements have been added to wireless communication from the first generation (1G) to the fifth generation (5G). 5G is the latest cellular network generation. The sixth generation (6G) is the next communication era. Figure 1.1 shows the difference between 5G and 6G wireless communication systems.

The 6G is still in the experimental stage and it will be ready for use in the next decade. The frequency range of 5G cellular network can be expanded up to 100 GHz which is in the mmWave band. Mmwave is a band of frequency spectrum which is between 30 GHz to 300 GHz. However, the 6G will use both mmWave and terahertz (THz) bands (Tripathi et al., 2021) as shown in following Figure 1.2.

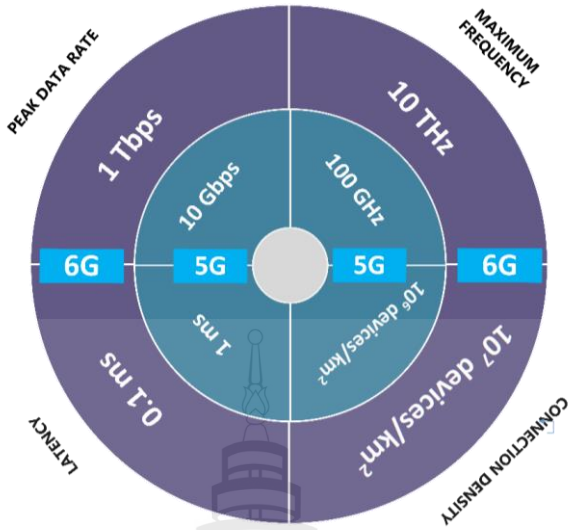


Figure 1.1 Key Features of 5G and 6G

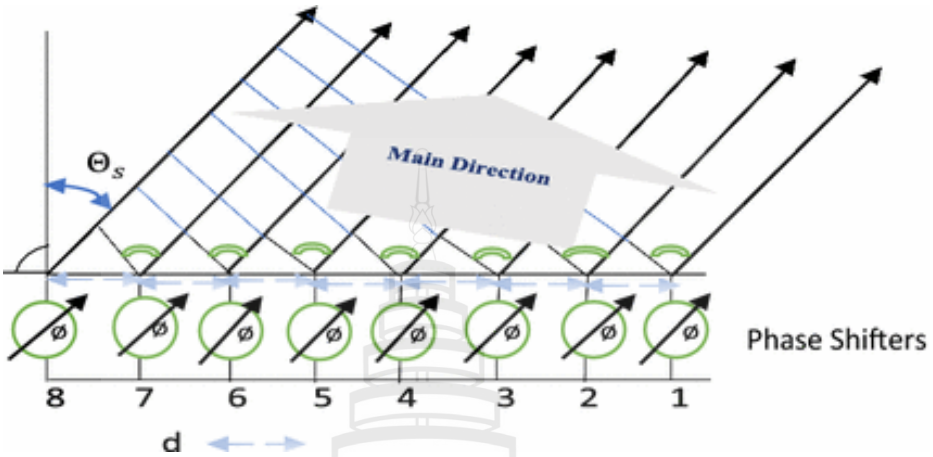
Figure 1.2 Frequency Bands of 5G and 6G

1.1.2 Beamforming

Beamforming (BF) is a technique that is used to send signals towards a specific direction that can be done in three methods. This technology is used in 5G and 6G to send beams only toward the user equipment which enhances the directional signal transfer.

Normally, BF is done by a set of phased array antennas. In phased array antennas, the phase of the signal is changed to form the beam that is emitted by individual elements to provide constructive/destructive interference in order to steer the beams in a particular direction as shown in Figure 1.3 (Banerjee & Mandal, 2018). The

phase shift (Δ∅) between two elements is a constant which is also called as phase increment.



Source Banerjee and Mandal (2018)

Figure 1.3 Beam Steering

$$x = d \sin \theta_s \quad (1)$$

$$\frac{360^0}{\Delta\emptyset} = \frac{\lambda}{x} \quad (2)$$

$$\Delta\emptyset = \frac{360^0 d \sin \theta_s}{\lambda} \quad (3)$$

Δ∅- Phase shift between two elements

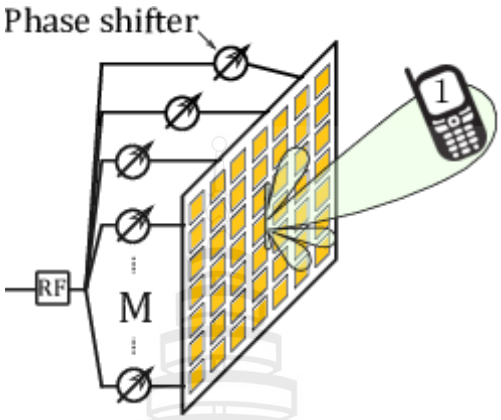
d- Distance between the radiating elements

θ_s- Beam steering angle

λ- Operating wavelength

1.1.2.1 Analog Beamforming: In this analog BF, the same signal is fed to each antenna element of a phased array antenna. Then these signals are steered using phase-shifters in order to send the signal in a desired direction as following Figure 1.4 (Rozé et al., 2015). Therefore, signals are emitted with different phases. The analog BF

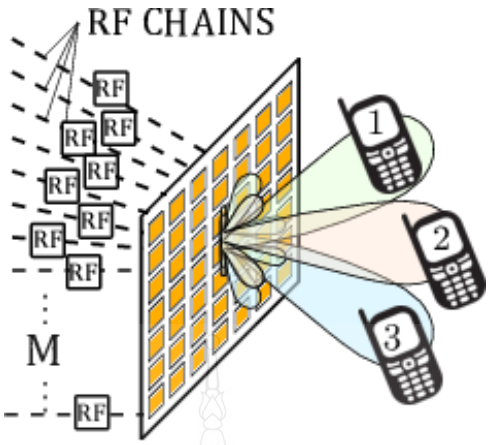
has a simple system architecture and it is low cost. On the other hand, it can steer a single stream of transmission that limits the data rate and scalability.



Source Rozé et al. (2015)

Figure 1.4 Analog Beamforming

1.1.2.2 Digital Beamforming: Digital BF is an enhanced technology than analog BF which can solve the scalability problem of analog BF. In digital BF, converters are attached to each antenna element at the transmitter's end. Along with converters, each antenna element has a dedicated RF chain. Consequently, phase and amplitude variations are done before analog to digital converters. Therefore digital BF enables sending signals with various phases and amplitudes as following Figure 1.5 (Rozé et al., 2015).



Source Rozé et al. (2015)

Figure 1.5 Digital Beamforming

1.1.2.3 Hybrid Beamforming: Hybrid beamforming is done by both analog and digital beamforming. One of the main reasons to introduce hybrid beamforming is to reduce the number of RF chains that reduce the complexity and energy consumption of the system. Therefore, the subarrays are introduced and they form a very large array in hybrid beamforming. The subarrays do analog beamforming. The subarrays share a digital baseband processor that carries out several digital signal processes.

1.1.3 Beam Sweeping and Indexing

Beam sweeping is the process of transmitting a sequence of beams by the base station to different directions which are called Synchronization Signal Block (SSB) beams. All beams that are transmitted by a particular base station are at different timings. Therefore, user devices can identify these beams using beam indices as shown in following Figure 1.6.

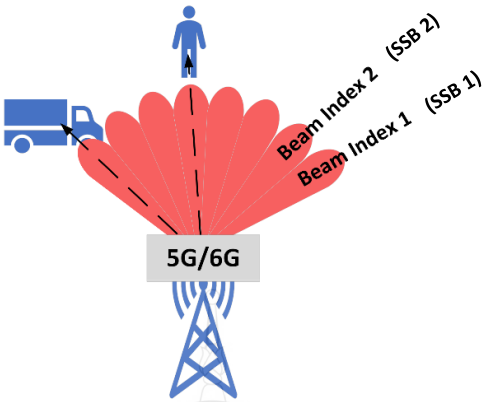


Figure 1.6 Beamforming and Beam Sweeping

1.2 Problem Statement

Though 5G and 6G provide better features, there are several challenges to overcome. One of the challenges is transmission impairment. Transmission impairment can occur due to attenuation (path loss, absorption loss and scattering loss), noise and distortion. MmWave and THz communication of 5G and 6G undergo more path loss and absorption losses since they are high-frequency signals which is a challenge for 5G, 6G and future communication systems (Misailidis & Voudouris, 2020). Hence BF technique is exerted to overcome this challenge. The 5G and 6G systems have to use large antenna arrays with narrow beam signals to obtain high BF gain due to this signal attenuation. Therefore, the 5G and 6G systems should have the ability to select the optimal beam between transmitter and receiver as shown in Figure 1.7.

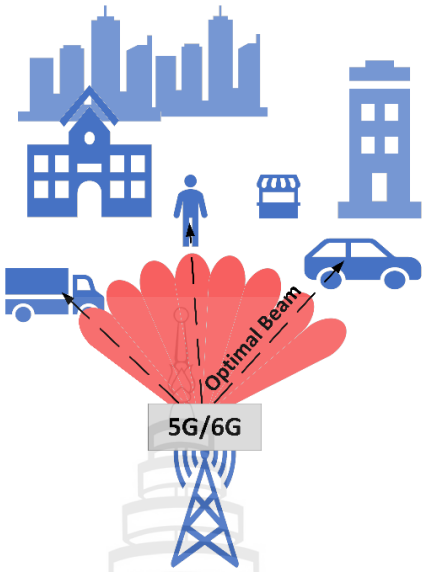


Figure 1.7 Optimal Beam Selection in 5G and 6G

In the BF, beam training algorithms are used to transmit the optimal beam between the transmitter and receiver. Generally, there are three beam training approaches (Yi et al., 2024). They are blind beam training, prior information aid beam training and machine learning-based beam training. However, the beam training algorithms need more computational resources. It is one of the main problems in 5G and 6G communication systems.

Beam tracking is another technique that can be used to improve directional communication. The misalignment of beams can affect to reduce the performance due to the loss of links and low data rate. This can happen due to the mobility of the users (Liu et al., 2020). To overcome this issue, different kinds of beam-tracking techniques are also being introduced in recent years. Accordingly, there are unsolved problems in beam training and beam tracking.

1.3 Research Objectives

Due to the aforementioned problems and challenges, it is important to introduce an effective solution for BF in current and future wireless communication systems.

Machine Learning based beam prediction is one of the alternatives for beam training overhead. The objectives of this research are;

1. To utilize deep learning techniques to develop a machine learning based approach for accurate beam prediction.

2. To evaluate the performance of the proposed model and compare with existing traditional basic fine-tuned deep learning and supervised machine learning algorithms.

1.4 Scope of the Study

The scope of this study is to develop a deep learning model for beam prediction of 5G, 6G and future communication systems, which can be trained using two types of data that are images and GPS data. The evaluation of this model was done using available online real-world dataset. Data were collected under different environments in this dataset that are called scenarios. This study utilized scenario 02, 05, 08 and 09 because previous studies that were carried out to introduce new approaches for the beam prediction using them shows less accuracy. Therefore, this approach is aimed to propose new model for beam prediction with higher accuracy. The research includes feature engineering before training the model. In the evaluation, the first step was to observe the accuracy, recall, precision and effectiveness with respect to overfitting and underfitting. In addition, the computational complexity of the proposed model was analyzed. Then the model was compared with basic deep learning algorithms and several supervised machine learning algorithms.

CHAPTER 2

LITERATURE REVIEW

2.1 Machine Learning Based Beamforming with Non-Sensory Data

The BF is an important technology that is used as a solution for path loss and absorption loss of signals in THz and mmWave high-frequency bands. However, this technology has challenges to overcome. BF needs beam training to select the optimum beam. But it requires high training overhead. Discovering an effective and efficient approach for BF is one of the leading research areas in the wireless communication domain. Several researches were carried out to obtain a solution for effective beam transmission between transmitter and receiver. Among them, some studies were conducted to introduce machine learning based beam prediction approaches that do not use more sensory information of mobile users. An algorithm called “DeepIA” was introduced in (Cousik et al., 2022) as a non-sensory approach. It is a deep neural network based approach. It proposes to sweep a subset of the codebook instead of sweeping all beams of the codebook. After that, beam measurement reports of users are employed in “DeepIA” to predict the best beam for the initial connection. In the results, it was specified that this algorithm can capture complex patterns to predict the optimum beam than traditional beam sweeping initial access. Zhang et al. (2020) introduced a machine learning framework to learn beam codebook with respect to hardware constraints. They proposed a neural network architecture in order to predict the beam vector using the structure of the channel and obtaining the hardware constraints. Another approach (Heng et al., 2021) similar to the study (Zhang et al., 2020), proposed a site-specific probing code book for beam alignment. In their approach, BSs sweep wide beams to obtain the channel matrix. Then a neural network uses this channel matrix in order to update wider beam weight. After learning the probing codebook, the neural network predicts narrow beams.

2.2 Machine Learning Based Beamforming with Sensory Data

2.2.1 Datasets

Some studies have been carried out to predict the optimum beam using the sensory information of mobile users. Researchers who studied these approaches have developed machine learning models that can be trained by different kinds of data. The most common data types used by researchers are GPS data and images. Therefore, the dataset (Alkhateeb et al., 2023) is an important asset for researchers in this field which is utilized in this study. This dataset contains mmWave, GPS, image, Light Detection and Ranging (LiDAR) and Radio Detection and Ranging (RADAR) data. Developers of the dataset implemented several testbeds and collected data under 39 scenarios.

Apart from (Alkhateeb et al., 2023), there are several datasets that can be used for beam prediction. The dataset (Alrabeiah et al., 2020) is similar to the dataset (Alkhateeb et al., 2023) mentioned above which consists of RGB images, depth data, LiDAR, RADAR, GPS, etc. Another dataset (Ahn, n.d.) contains RGB and depth images which are labels with categories and object bounding boxes. Applications of mmWave and massive MIMO (multiple-input multiple-output) can use the dataset (Alkhateeb, 2019). Developers of this dataset targeted machine learning based mmWave/massive MIMO channels applications.

2.2.2 Related Studies in Beam Prediction

Morais et al. (2023) used only latitude and longitude GPS values from the dataset (Alkhateeb et al., 2023) to predict the best beam. They used three algorithms that are K-nearest neighbors, lookup table (LT) and artificial neural network (ANN). Latitude and longitude data were used to train these algorithms after utilizing min-max normalization. They considered pre-defined beam codebook as output target classes. Pan et al. (2017) constructed an SVM model and inserted location data as the feature vector to the model. They also used beams of the transmitter as the output classes while ignoring blockages.

Some works used both position data and RGB image data for beam prediction (Charan, Hredzak et al., 2022; Charan, Osman et al., 2022) while some used only image data (Imran et al., 2023) from the dataset (Alkhateeb et al., 2023). All researchers of

these articles used machine learning algorithms for the prediction. For example, the authors of the article Charan, Osman et al. (2022) ran a fine-tuned ResNet model upon images and extracted the feature vector first. Then they concatenated normalized position data with the feature vector and inserted it into an MLP neural network that was used for the classification. They also compared this model with vision-only and position-only models. Machine learning algorithms like ResNet-50, LeNet and MLP were used for beam prediction in these work (Imran et al., 2023; Charan, Osman et al., 2022).

Researchers have used not only GPS and image data but also RADAR (Demirhan & Alkhateeb, 2022) and LiDAR (Jiang et al., 2023) data for the beam prediction. Alrabeiah, Hredzak & Alkhateeb (2020) tried to cater to two challenges which are training overhead and link blockage using the deep learning approach. Another work (Ahn et al., 2023) used multi-model sensing data such as ultrasonic sensing data, infrared data and data from thermographic camera for beam prediction. This study used the EfficientDet object detector instead of traditional algorithms like YOLO and fast region-based CNN. Researchers of this article utilized the dataset Ahn (n.d.) to evaluate their approach that allows BSs to set beams directly without codebook quantization.

Alkhateeb et al. (2018) introduced a machine learning based mmWave system to serve mobile users. In this system, all users are simultaneously served by multiple coordinating base stations. However, a particular user should be selected for one uplink pilot signal that is received at the BSs. From these signals, a signature is drawn which describes the user's location and details about its user location. Then they introduced a deep learning model that can be used to predict the beamforming vectors by using those signatures at the BSs. Rezaie et al. (2020) introduced a method to narrow down the best beam pairs using deep neural network architecture. They used the location and orientation data of the receiver which were provided into a deep neural network as input. Lin et al. (2019) suggested a new approach for beam selection using a heat map of images. In their study, images were formed by the different beam powers and searched to obtain peak heat of images. On the other hand, drones will contribute to the THz and mmWave communication systems to extend coverage and increase security in the future. There are several studies that were carried out to predict beams in the drone

context (Charan et al., 2023). They developed machine learning based approach that uses images that were captured by drones.

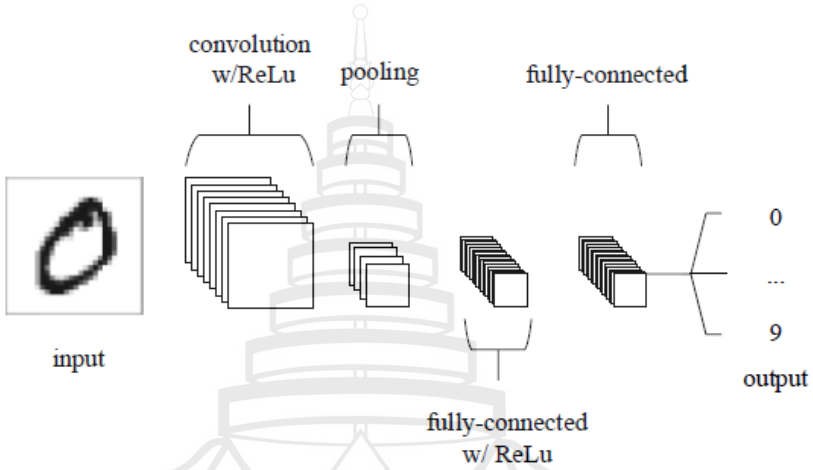
2.2.3 Related Studies in Beam Tracking

Beam tracking is also a technology that is used to target users. However, it is also a complicated process. Researchers introduce different kinds of solutions to overcome this problem. Among them, some studies use machine learning based solutions. Burghal et al. (2019) proposed an effective method for beam tracking using a Recurrent Neural Network (RNN). The authors used this RNN to estimate the angle of arrival (AoA). Jiang and Alkhateeb (2022) proposed an encoder-decoder model to enhance beam tracking. They used visual data that were captured at the base station for the beam tracking. Similar to the research Burghal et al. (2019), Tian and Wang (2021) conducted a study in which images were used in a deep leaning framework for beam tracking. They utilized the dataset (Alrabeiah, Hredzak, Liu et al., 2020) in their research. Three approaches were used in the study namely CNN, autoencoder and Principal component analysis (PCA). Some studies used location data to enhance beam tracking technology in vehicular communications (Moon et al., 2020; Nasim et al., 2020; Guo et al., 2019; Va et al., 2017).

2.3 Machine Learning Algorithms and Techniques

ANN is a collection of computational nodes that collectively learn by input data and optimize its corresponding outputs. Inputs are inserted as multidimensional vector to the input layer. Then it is distributed to hidden layers by the input layer (O'Shea & Nash, 2015). In the hidden layers, inputs are multiplied with weights and sum all weighted inputs and bias. Then it passes through an activation function (Kenji, 2011). Training of ANN contains forward and backward data passing. At the last layer of the ANN, predicted outputs and actual outputs are compared and error is calculated. Then the error is used to alter the weights during the backward pass to reduce the error (Domingos, 2005). MLP is an ANN that contains multiple layers. It is a feedforward neural network that consists of one or more hidden layers (Sonawane & Patil, 2015). CNN is also a feedforward neural network that deals with specific input type which is

images. A main difference of neurons of layers in the CNN is, that they are organized in a three-dimensional way. CNN consists of three types of layers that are convolutional layers, pooling layers and fully connected layers as shown in following Figure 2.1. (O'Shea & Nash, 2015). The MLP and CNN are the two main algorithms used in this study.



Source O'Shea and Nash (2015)

Figure 2.1 Basic CNN Architecture

In addition to that, the PCA algorithm was used to reduce the dimensionality of features in this study. PCA utilizes vector space transform to reduce the dimensionality. This algorithm finds the maximum variance of high dimensional data and they are projected to low-dimensional space while remaining highly important information (Salem & Hussein, 2019). In this algorithm, a covariance matrix is used to identify how much each pair of features varies together. The covariance of two features is calculated as following equation (4) where x and y are training samples of features x and y respectively, x-bar and y-bar represent the mean value of feature x and the mean value of feature y respectively, i refers to the i^{th} sample and n represents the total amount of samples (Raschka, 2015). If the inserted dataset has n features, then the covariance matrix is n × n where each element is covariance between two features. For example, if we have two features, the elements are covariance (x, x), covariance (x, y), covariance

(y, x) and covariance (y, y). Then it calculates the eigenvectors and eigenvalues of the covariance matrix to find the principal components according to equation (5) where V is a matrix in which columns are eigenvectors of D and C is a diagonal matrix where eigenvalues of D represent diagonal elements (Dhalla, 2021). After that, the eigenvalues are ordered by the highest to lowest value. The eigenvalue represents how much variance is indicated by the eigenvector. Therefore, the algorithm selects eigenvectors that correspond to the highest K eigenvalues in the order. Finally, to project original data onto dimensions of selected principal components, they are multiplied by the matrix of eigenvectors (Sena, 2024). In addition to the PCA, the random oversampling technique was also used to balance the dataset. It randomly selects instances from minority classes and duplicates them to balance the dataset (Ashraf, 2023).

$$\text{Covariance}_{xy} = \frac{1}{(n-1)} \sum_{i=1}^{n} (x_i - \bar{x})((y_i - \bar{y}) \quad (4)$$

$$V^{-1}CV = D \quad (5)$$

On the other hand, the proposed model of this study was compared with SVM which is a powerful supervised machine learning algorithm. SVM classification can be divided into two types which are linear and non-linear. In linear classification, input data can be linearly separable by a single straight line while in non-linear classification, data can’t be linearly separable by a single straight line. However, most of the applications are linearly separable. In linear SVM, D-dimensional input data space is separated using a decision hyperplane (Nalepa & Kawulok, 2019) as shown in Figure 2.2. The hyperplane is referred to as a decision margin of data where the maximum margin is selected as shown in Figure 2.2. In the figure data in two features are represented in two colors. In the linear problem that is represented by Figure 2.2, hyperplane 2 is selected since it has the maximum margin.

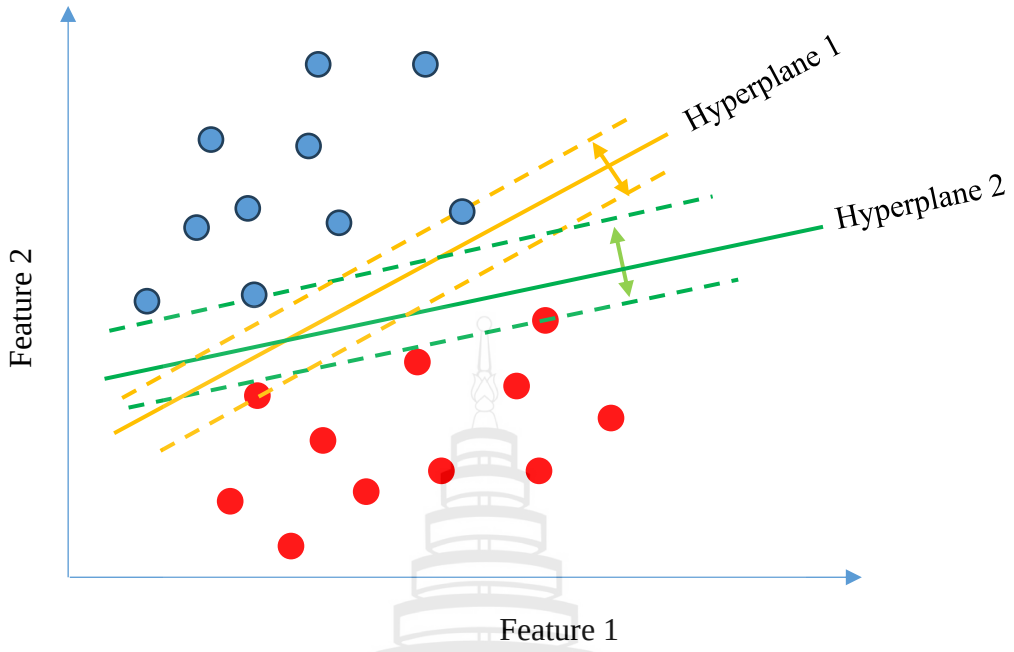


Figure 2.2 Representation of Linear SVM Classification

In addition to SVM, the RF algorithm was also used for the comparison in this study. RF algorithm is one of the bagging algorithms that constructs a large collection of de-correlated trees and averages them (Cousik et al., 2022). In the RF, first, it selects bootstrap samples from the training data. Then the algorithm creates an ensemble of trees. Then during the classification, it selects the majority vote (Breiman, 2001) as shown in Figure 2.3. Finally, the proposed model was compared with the Adaptive Boosting algorithm. Similar to bagging, adaptive boosting also creates a combination of models that are weak classifiers. However, classifiers are connected sequentially and allow models to learn from mistakes using weights. Initially, weights are assigned equally. Then models are trained iteratively while increasing the weights of misclassified examples as shown in Figure 2.4. It continues until maximum prediction accuracy or maximum models that can be added (Margineantu & Dietterich, 1997).

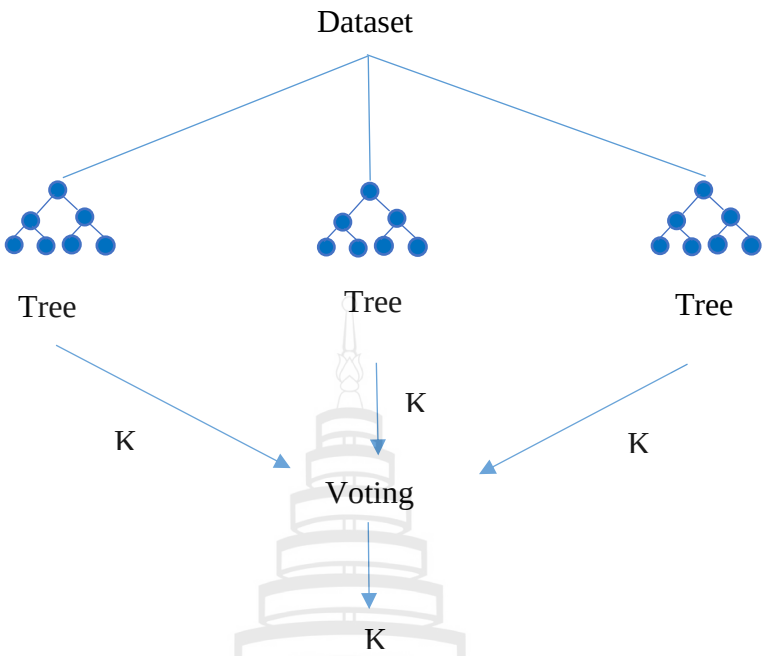


Figure 2.3 Structure of Random Forest Classification

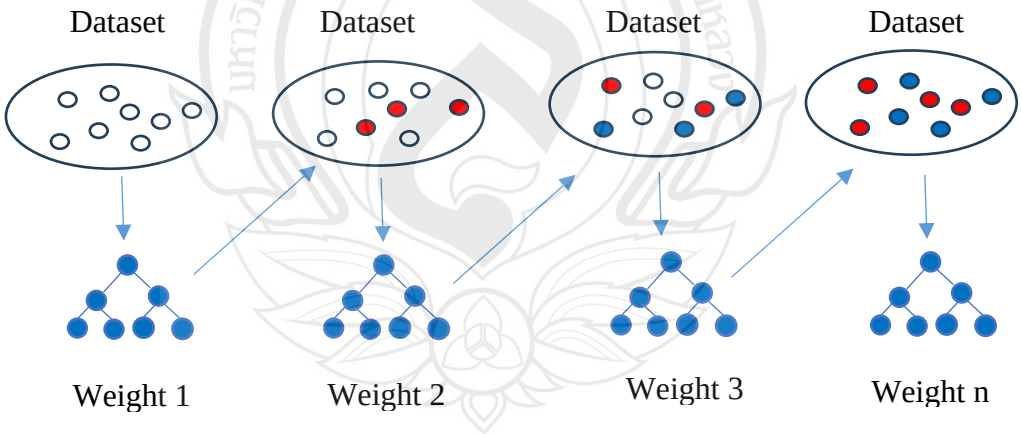


Figure 2.4 Process of Adaboosting

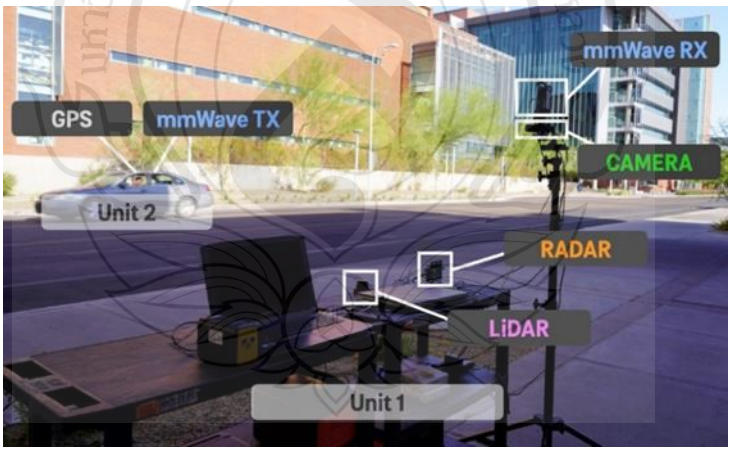
The main focus of this study was to develop a machine learning model for beam prediction that accepts real-world multiple data types. According to the literature, there are a small number of studies that were carried out to observe multi-input machine learning models. In this study, both GPS and image data were utilized for the beam prediction. A CNN was used to extract a feature vector from images. Then it was concatenated with GPS data. This study employed PCA on this concatenated feature vector without class labels which isn't tested in previous studies. The objective of applying the PCA is to identify the most important features for the training process that was done using an MLP network. Therefore, this approach aims to increase the beam prediction accuracy.

CHAPTER 3

METHODOLOGY

3.1 Dataset

This research used a publicly available dataset which is called “DeepSense 6G” (Alkhateeb et al., 2023). The dataset is available on the internet and users can log in to their website using an email address to download the dataset. According to the license of this dataset, researchers can use this dataset for non-commercial purposes along with appropriate credits. Therefore, the dataset was used for this research without any special permission. The developers implemented several testbeds to collect data under several scenarios. Following Figure 3.1 from Data Collection – DeepSense (n.d.) shows a testbed that they used to collect the data.



Source Data Collection – DeepSense (n.d.)

Figure 3.1 A Testbed to Collect Data of the Dataset

Scenarios are different test environments in which they collected data. In testbeds, vision data were collected at the base station using a RGB camera. The base station of these testbeds also contained a 16-element phased-array receiver. Position data are the latitude and longitude information of a mobile vehicle (unit 2). The vehicle also consisted quasi-omni antenna that transmitted signals at the 60 GHz band. Accordingly, the base station received this transmitted signal utilizing 64 pre-defined beams of the codebook. This pre-defined beam codebook was used for the target variable of this model.

3.2 Overview of Selected Input Data

The vision and position data of this dataset were utilized to train the proposed model since they are the most used data types in previous work. Moreover, these data provide more convenience in the data collection process of real-world applications. The dataset contains 39 scenarios and this study used four scenarios namely 02, 05, 08 and 09. They were selected since they contain higher data samples than other scenarios and they ensure the variety of data since data were collected during both daytime and nighttime as shown in Table 3.1.

Table 3.1 Selected Scenarios of the Dataset

	No. of GPS samples	**No. of images**	**Time Period**
Scenario 02	2974	2974	Night-time
Scenario 05	2300	2300	Night-time
Scenario 08	4043	4043	Day-time
Scenario 09	5964	5964	Day-time
Total	15281	15281	

Following Figures 3.2, 3.3, 3.4, 3.5 and Table 3.2 are examples of images and GPS data. Among them, images are contained in scenarios 02, 05, 08 and 09 respectively. More images of four scenarios are included in Appendix C.

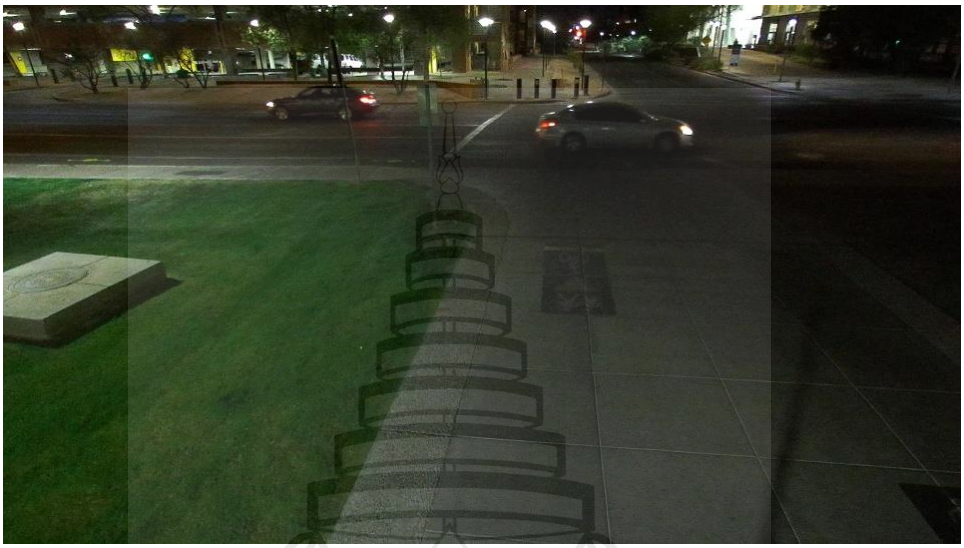


Figure 3.2 An Example Image of Scenario 02

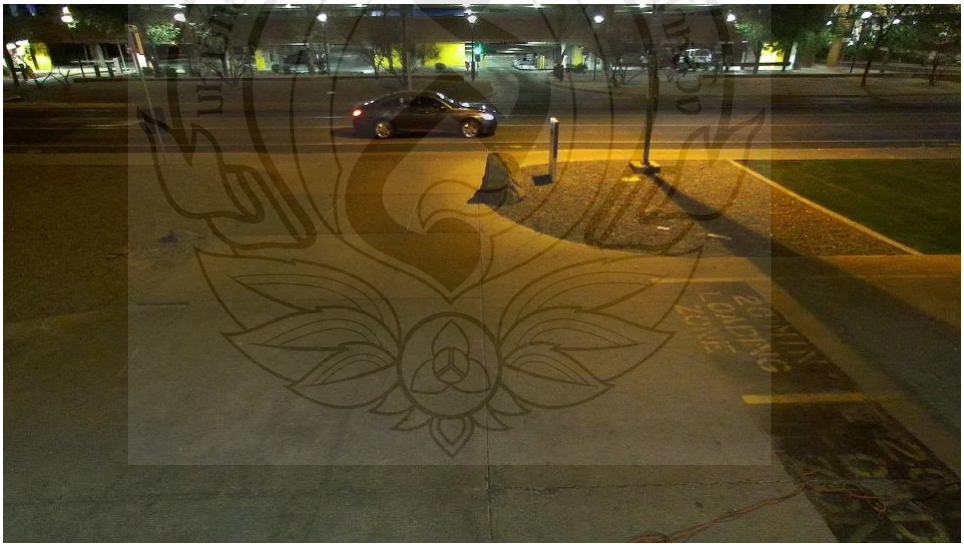


Figure 3.3 An Example Image of Scenario 05

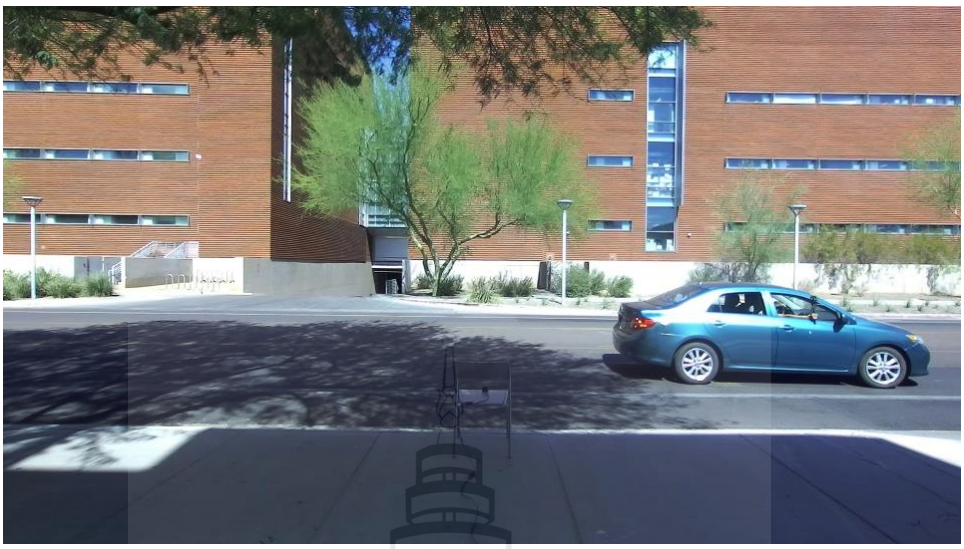


Figure 3.4 An Example Image of Scenario 08

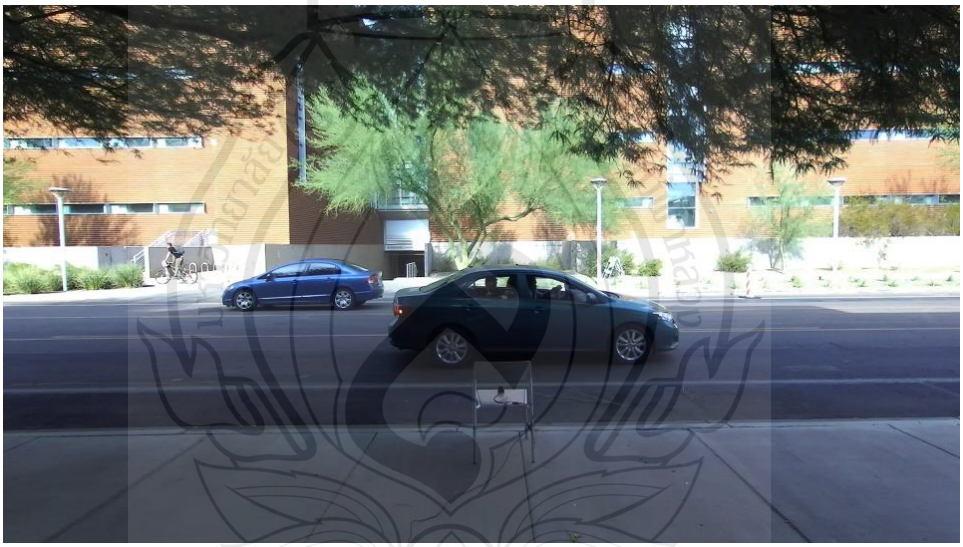


Figure 3.5 An Example Image of Scenario 09

According to the images, it is clear that scenarios 02 and 05 contain daytime images while scenarios 07 and 08 consist of night-time images in the dataset. They are RGB images with dimensions of 960 × 450 (Data Collection – DeepSense, n.d.).

Table 3.2 Examples of Selected GPS Data

Latitude	Longitude
33.419404	-111.928920
33.419401	-111.928920
33.419397	-111.928920
33.419393	-111.928920
33.419390	-111.928921
33.420232	-111.928978
33.420227	-111.928978
33.420222	-111.928977

Developers of the dataset provide GPS data in two ways, that are as text files and NumPy files. In this study, NumPy files were used to obtain GPS data. Table 3.2 shows a few samples of them. According to Table 3.2, GPS data of each scenario have two types that are latitude and longitude samples.

3.3 Classification Outputs

This study aimed to implement a model that is able to classify the best beam. In the 5G cellular communication system, base stations transmit beams using BF and beam sweeping methods. Beam sweeping is an extended process of BF in which beam indices are used, as described in the introduction section. However, the base station should select the best beam when user equipment communicates with the base station. Beam indexes are the target variables of the dataset as shown in following Figure 3.6.

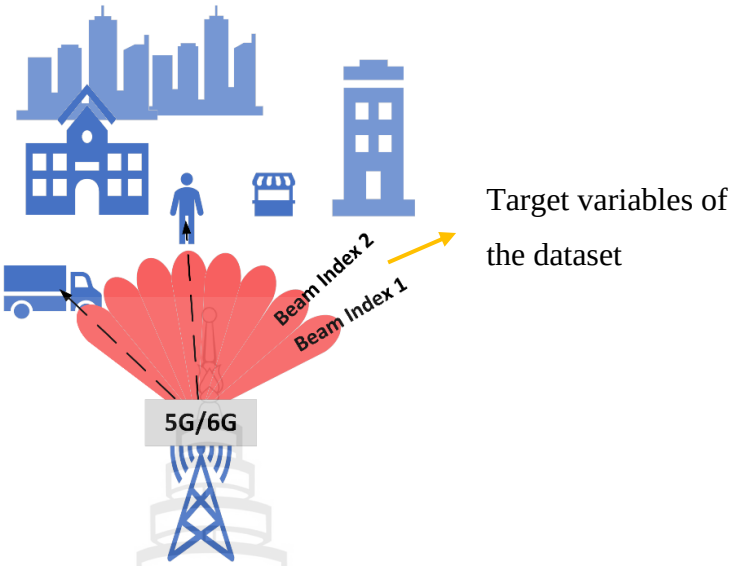


Figure 3.6 Target Variables of the Dataset

The dataset contains 64 beam indexes as shown in Figure 3.6 and corresponding image and GPS data for each beam index. Each scenario of the dataset contains beam indexes and corresponding other data like images, GPS data, time stamps, satellite details, power, etc. However, this study selected beam indexes, images and GPS data.

3.4 Feature Scaling

The proposed model of this study accepts both images and GPS data. Therefore, total GPS data and images of selected scenarios were used for the training and testing of the model. The first step of this research was to normalize the data. Since the GPS data are not in a normal distribution, the study used the min-max normalization technique instead of standardization. Furthermore, images were read using the OpenCV library and reduce the image size to reduce the computational complexity. Then they were divided by 255 to scale between 0 and 1 as following figure since they are color images as shown in Figure 3.7. Furthermore, scaling between 0 and 1 helps in the process of backpropagation which is used in the proposed model.

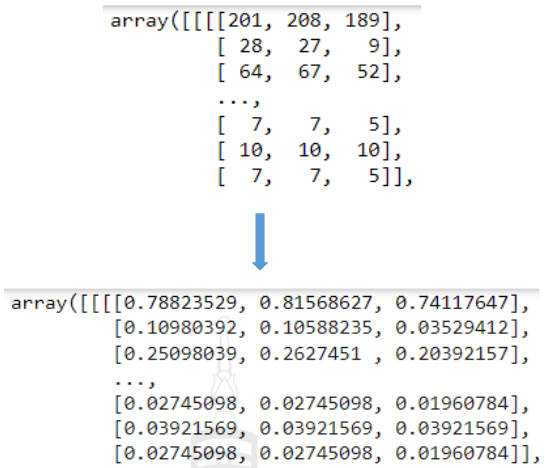


Figure 3.7 Scaling Pixel Values of Images between Zero and One

After the scaling, the rescaled image values were inserted into the first section of the proposed model which is a CNN as following Figure 3.8.

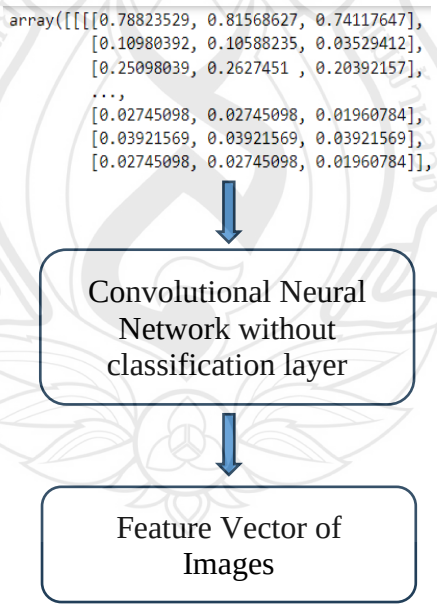


Figure 3.8 Extraction of Features from Images

The following sub-section describes the complete approach of this study along with the above-mentioned steps.

3.5 Approach of the Study

Figure 3.9 represents the overview of the study that was used in this research. It has several steps from data scaling to final classification. This approach used two machine learning techniques which are CNN and MLP.

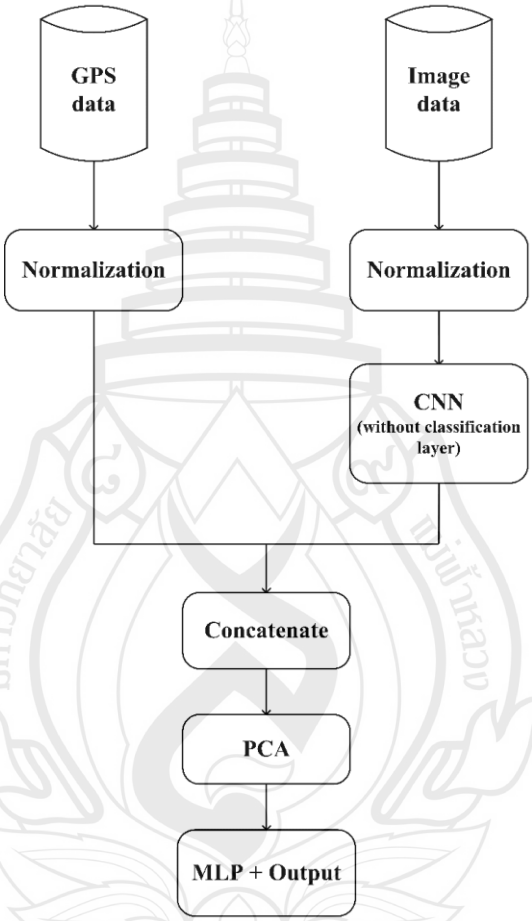


Figure 3.9 Approach of the Study

After the extraction of features from scaled image values by the CNN as shown in Figure 3.8, both GPS data and image features were concatenated as shown in Figure 3.10. After that, the Principal component analysis (PCA) technique was applied to reduce the dimension of the concatenated feature vector which is a novel experiment in this research area.

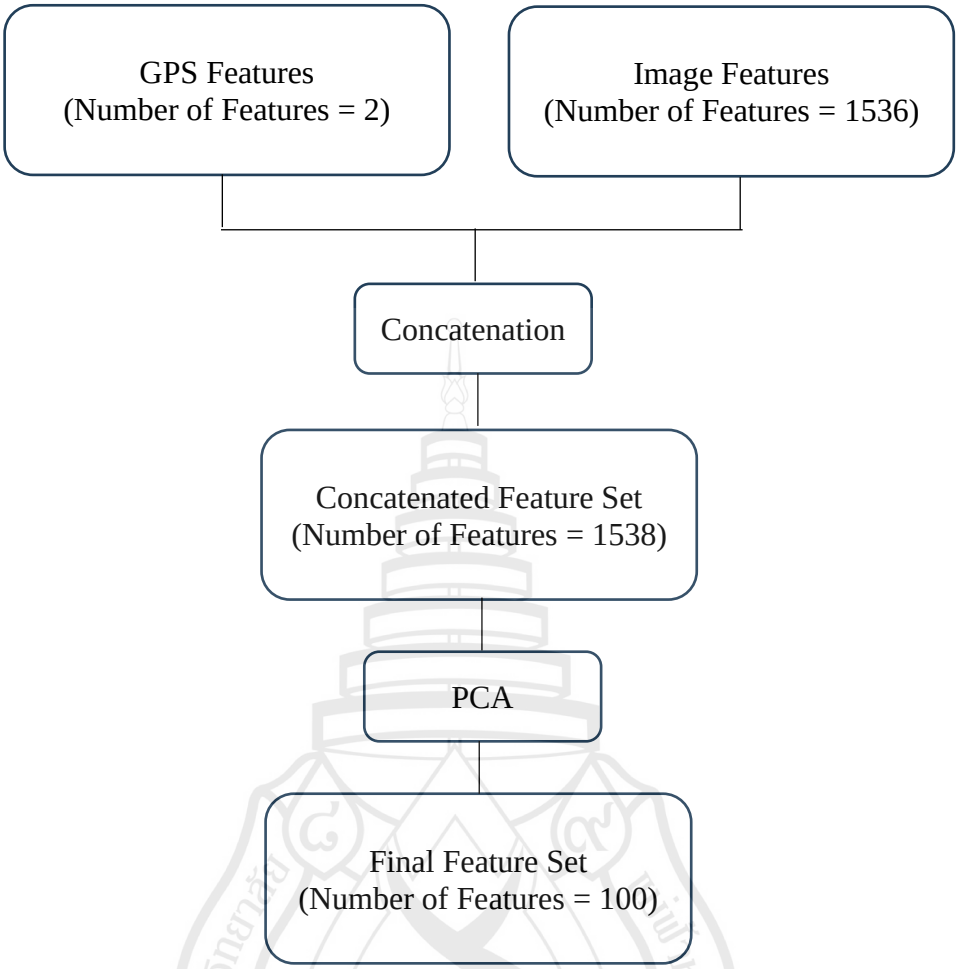


Figure 3.10 Feature Concatenation and Reduction

At the end of the above step of this approach, the most important hundred features were selected. However, the classes of this dataset are imbalanced. Therefore, the dataset was balanced using the random oversampling technique to ensure an effective training process. Finally, the MLP network was used for the training. Figure 3.11 shows the deep learning model that is used after the feature scaling step.

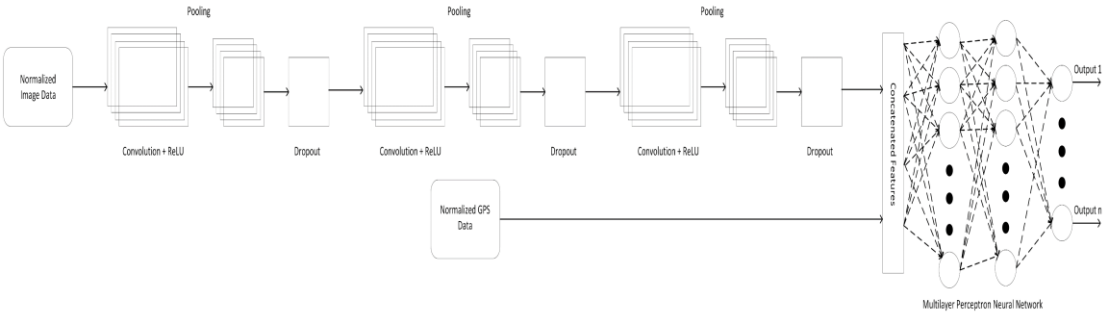


Figure 3.11 Model Architecture

However, before the training of the model, features were split into training and testing sets. Accordingly, the data were split into 70% for training and 30% for testing sets along with shuffling which ensures effective integration of data into the training process.

3.5.1 Convolutional Neural Network of the Proposed Model

Normally CNN contains three types of layers which are convolutional layers, pooling layers and fully connected layers. However, in this study, the CNN was utilized only for the feature extraction of images. Therefore, the CNN doesn't contain fully connected layers in this study. However, it has three convolutional layers followed by pooling layers and dropout layers. In convolutional layers, features are extracted using several parameters such as amount of filters, size of filters, activation function, etc. This study employed 16, 32 and 64 amount of filters in three convolutional layers respectively. However, all filters are in the size of 3×3. Each convolutional layer contains the ReLU activation function. After each convolutional layer, a max pooling layer is used to reduce the size of the feature map. Dropout layers are used after each max pooling layer to improve the performance of a network by reducing overfitting. It drops neurons in layers randomly during the training (Srivastava et al., 2014).

3.5.2 Multilayer Perceptron of the Proposed Model

In this study, an MLP was used for the training which predicts the beam index. The MLP network was trained using the final feature vector and output labels (beam indexes). The MLP network used in this study has two hidden layers. Each hidden layer

of this MLP takes advantage of the ReLU activation function and He_uniform kernel initializer. The output layer uses the softmax activation function with glorot_uniform initializer since they are more suitable for multiple classes.

3.6 Model Validation

The proposed model was validated using two methods in the study. The most important method is K-fold cross-validation. In this approach, the data is split into K number of subsets. This study utilized 5-fold cross-validation. Therefore, the data were split into five subsets. Then the model was trained. During the training, four folds were used for training and the remaining fold was used for testing in each iteration, Finally, the mean value of all iterations was obtained. Usually, the scikit-learn cross-validation function can be used for general machine learning algorithms like SVM, RF, etc. However, there is a wrapper that allows to use of deep learning models in scikit-learn. This KerasClassifier wrapper was used for the cross-validation of the proposed model. The second approach was done during the final training of the model. This option is provided by the Keras fit method. It calculates loss and accuracy at each epoch and it is important to observe the overfitting and underfitting of the model through accuracy and loss curves.

3.7 Model Development

The aforementioned model was developed using the Jupyter Notebook which is a web-based platform and the Python programming language. The Anaconda distribution was used that contains Python and other libraries. This study also used several libraries like Numpy, Pandas, Scikit-Learn, Keras, TensorFlow, Matplotlib and OpenCV.

3.8 Model Evaluation

3.8.1 Evaluation Metrics

The evaluation of the proposed model was done in multiple ways. First, three important evaluation metrics which are accuracy, precision and recall of the proposed model were obtained. This study is a multiclass classification problem. Therefore, it is needed to get the summation of precisions and recalls of all classes. There are two methods to obtain the summation which are called “micro” and “macro”.

1. Macro-averaging

In the macro-averaging, precisions and recall values of each class are averaged to calculate precision and recall respectively.

$$\text{Precision} = \frac{1}{N} \sum_{i=1}^{N} \text{Precision}_{\text{class i}} \tag{6}$$

$$\text{Recall} = \frac{1}{N} \sum_{i=1}^{N} \text{Recall}_{\text{class i}} \tag{7}$$

2. Micro-averaging

In the micro-averaging, true positives (TP), false positives (FP), and false negatives (FN) of each class are calculated first. After that, precision and recall are calculated using the following equations.

$$\text{Precision} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + FP_i} \tag{8}$$

$$\text{Recall} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + FN_i} \tag{9}$$

Where 1, 2, 3, ……., N = Classes of the dataset

Though there are two methods, this study focused on the macro-averaging method since macro-averaging is a class-based metric that can be identified by the equations too. Therefore, it provides equal weight for each class which is consequential

in the real world since each class is equally important in this application. On the other hand, micro-averaging is an instance-based metric.

During the training of the model, the accuracy and loss curves were captured to ensure the overfitting and underfitting characteristics of the model. After training the MLP against 64 classes as shown in the above figures, the model was tested against 32 classes too. The classes were reduced without affecting the coverage area.

3.8.2 Performance Comparison

The model was compared with five machine learning algorithms. These algorithms were trained using the same data that were utilized in the study. Hyperparameter tuning was done using the GridSearchCV function in the model_selection package of the Scikit-learn library. A CNN algorithm was selected first for the comparison. This CNN algorithm was trained using the same set of images that had been used to train the proposed model. Next, an MLP algorithm was selected. The MLP algorithm was trained using only GPS data. Finally, it was compared with the following supervised machine learning algorithms too.

3.8.2.1 Support Vector Machine: The SVM algorithm was also used to compare with the proposed model. The SVM model was trained using only GPS data that were used in the proposed model. Major parameter values of the SVM algorithm (regularization parameter, kernel and gamma) were selected by the hyperparameter tuning.

3.8.2.2 Ensemble Learning Algorithms: Finally, two ensemble learning algorithms under bagging and boosting approaches were used to compare the proposed model. For the bagging approach, the RF algorithm was selected. It was also trained on the same GPS data to compare the performance. For the boosting approach, the adaptive boosting algorithm was selected since it is more suitable for smaller or medium-sized datasets. It was also trained using GPS data.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Classification Results

The dataset contains 64 classification classes which are beam indexes. This study tested the performance of the approach with respect to 64 classification classes first. The proposed methodology was also tested using 32 classification classes to verify the performance. Table 4.1 shows the amount of samples after selecting 32 classes.

Table 4.1 Amount of Data after Selecting 32 Classes

	No. of GPS samples	**No. of images**
Scenario 02	2540	2540
Scenario 05	1910	1910
Scenario 08	3402	3402
Scenario 09	4742	4742
Total	12594	12594

The validation and test accuracies of the model with respect to 64-classes and 32-classes are shown in the following Table 4.2. According to the table, one of the main strengths of the proposed model is the satisfactory level of accuracy. On the other hand, 64 classes achieved more accuracy than 32 classes. As mentioned in the methodology, oversampling was carried out due to the imbalanced dataset. Consequently, 64 classes have more well-balanced data than 32 classes. Therefore, the accuracy of the 64 classes is higher than 32 classes. In addition, significant cross-validation results led this study to use this model for the training. The study received average cross-validation results of 89.38% with respect to 64-classes and 85.94% with respect to 32-classes.

Table 4.2 Validation and Test Accuracies of the Proposed Model

Number of classes	Validation Accuracy (%)	Test Accuracy (%)
64	89.05	88.98
32	86.41	85.72

Learning curves are good tools to diagnose the learning process of a machine learning model. There are two types of learning curves that are accuracy curve and the loss curve. They represent how the model learns over the training time. Figures 4.1, 4.2, 4.3 and 4.4 show the accuracy and loss curves of the model with respect to 64-classes and 32-classes contexts which were obtained during the training of the proposed model.

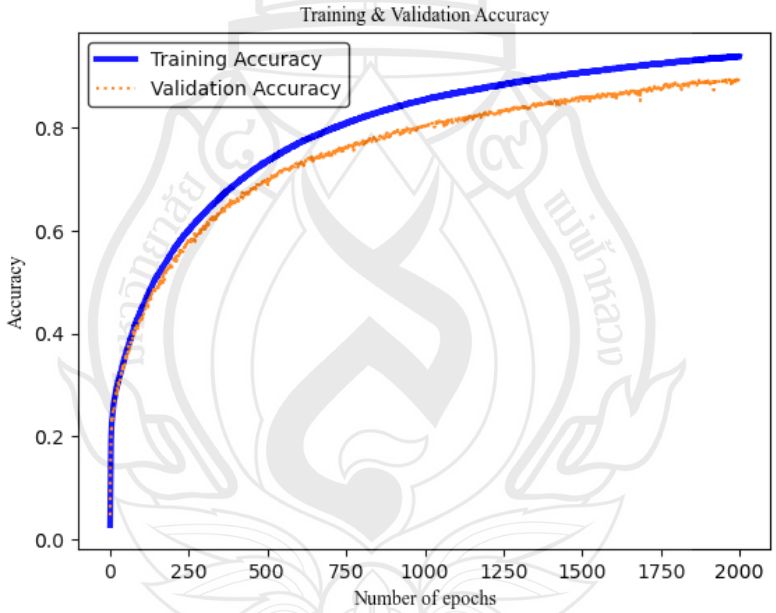


Figure 4.1 Accuracy Curve of the 64-classes Context

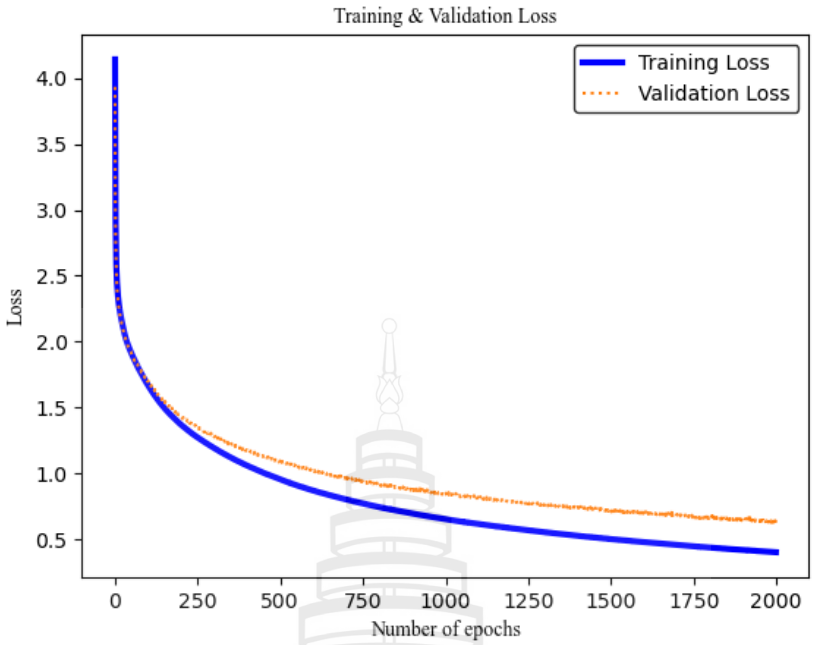


Figure 4.2 Loss Curve of the 64-classes Context

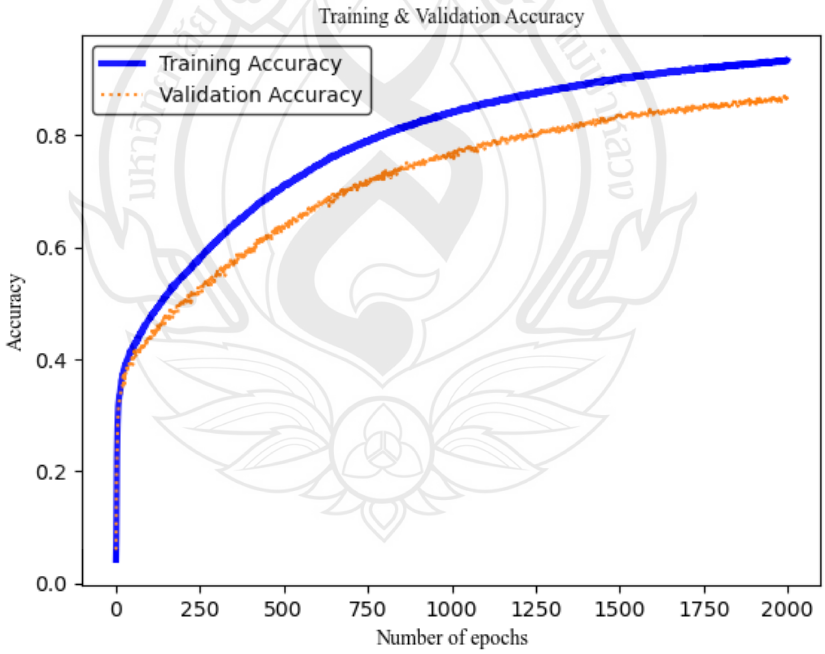


Figure 4.3 Accuracy Curve of the 32-classes Context

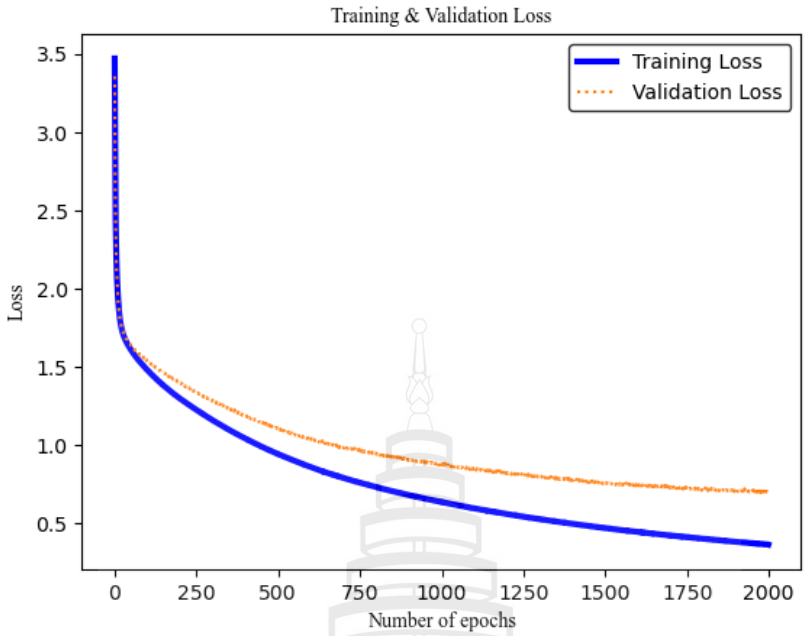


Figure 4.4 Loss Curve of the 32-classes Context

The above figures were obtained during the training of the model to observe the performance of the model regarding overfitting and underfitting. Figures 4.2 and 4.4 show the loss of the model on training data. If the model's training loss is high, it indicates that the model didn't learn the data accurately which is called "underfitting". However, the loss values are reduced in the Figures 4.2 and 4.4 which indicates the proposed model learned the training data correctly. Therefore, it has low underfitting. On the other hand, if both training and testing loss are low or if both training and testing accuracy are high, it indicates that the model has low overfitting. Both these conditions are fulfilled by the above four figures. Therefore, the proposed model has low overfitting. Training with more data helps to reduce the overfitting and increase the accuracy. Although the model is trained using a large amount of data, the selection of the best parameter values of the model is important for high accuracy.

The proposed model was evaluated using precision and recall evaluation metrics too and considered only the macro-averaging method as described in the methodology section. since it expresses how the model performs across all classes. The following table shows precision and recall values with respect to 64 and 32 classes.

Table 4.3 Precision and Recall of the Proposed Model

Number of classes	Precision (%)	Recall (%)
64	89.09	89.15
32	85.87	85.66

In addition to accuracy values, precision and recall values were also obtained to observe how the proposed model behaves with regard to the classes of the application. According to Table 4.3, the model is approximately 89% and 86% correct when predicting the positive class with respect to 64 and 32 classes respectively. Furthermore, recall value indicates how often the model can identify positive instances from all positive examples of the dataset. Sending the beam to the correct receiver is important in this application. The model shows approximately 89% and 86% ability to send the beam correctly to the correct receiver. However, missing a positive instance is a high loss for this application. Therefore, recall value should be increased along with precision which can be achieved by more hyperparameter tuning of the model.

4.2 Comparison of Model Performance

As described in the methodology section, the proposed model was compared with five machine learning algorithms that are CNN, MLP, SVM, RF and Adaboosting. Table 4.4 shows testing accuracies of CNN, MLP, SVM, RF and Adaboosting techniques. According to Table 4.4, it is clear that models that were trained by MLP, SVM, RF and adaboosting using GPS data obtained very low accuracy. These four algorithms were trained using only GPS data. GPS data contains only two features which are latitude and longitude feature columns. This might be the most affected reason for the low accuracy. In addition, GPS data contains six decimal points which might affect the accuracy too. However, the CNN model obtained more accuracy than other models since it was trained using more features that were extracted by that CNN algorithm. However, training of CNN, MLP, SVM, RF and Adaboosting models didn't use any oversampling technique to balance the dataset. This reason might affect the low

accuracy too. This is another important finding of this study that is, a balanced dataset is highly important for the training of any model. However, the accuracy of these five models can be improved by deep hyperparameter tuning. Despite deep hyperparameter tuning, MLP, SVM, RF and Adaboosting might show low accuracy due to a very small amount of features.

Table 4.4 Testing Accuracy Comparison between the Proposed Model, CNN, MLP, SVM, RF and Adaboosting

Test Accuracy of Proposed Model (%)	Test Accuracy of CNN (%)	Test Accuracy of MLP (%)	Test Accuracy of SVM (%)	Test Accuracy of RF (%)	Test Accuracy of Adaboosting (%)
88.98	57.18	37.90	34.05	34.66	23.79

The following Figure 4.5 illustrates a graphical representation of the test accuracies of the above algorithms.

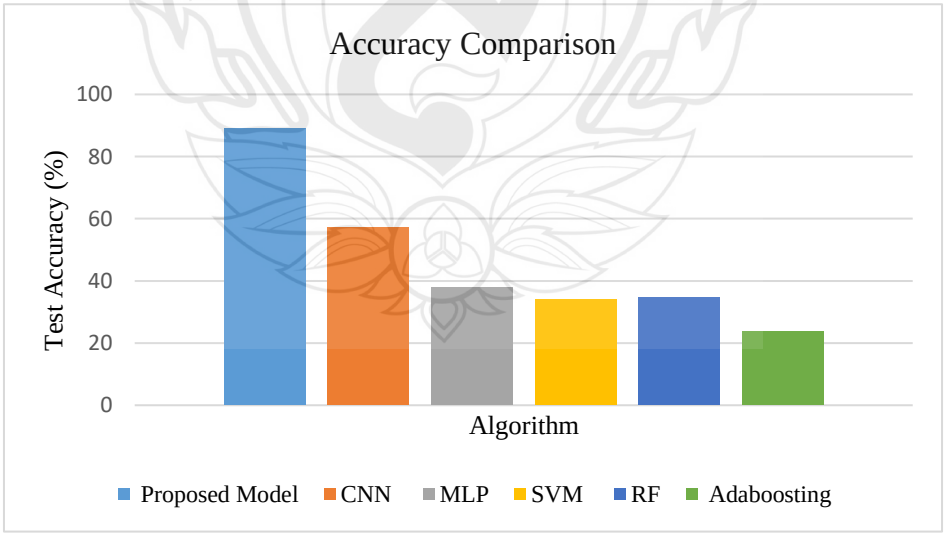


Figure 4.5 Comparison of Test Accuracies

4.3 Computational Complexity of the Model

The computational complexity of the proposed model was calculated in three steps. First, the complexity of CNN is calculated where features of images were extracted. Then the complexity of MLP was calculated that was used for the classification. Finally, the overall complexity of the model was calculated.

4.3.1 Computational Complexity of the Convolutional Neural Network

The CNN segment of the proposed model has three convolutional layers followed by max-pooling layers and dropout layers. Dropout layers reduce the computational complexity of the model. In order to identify the maximum complexity of the model, dropout layers are ignored. Pooling layers do not have learnable parameters. Therefore, the computational complexity of pooling layers are very low compared to convolutional layers (Dinata, 2023). According to the study carried out by He and Sun (2015), pooling layers take 5%-10% computation. Hence, the computational complexity of three convolutional layers are considered.

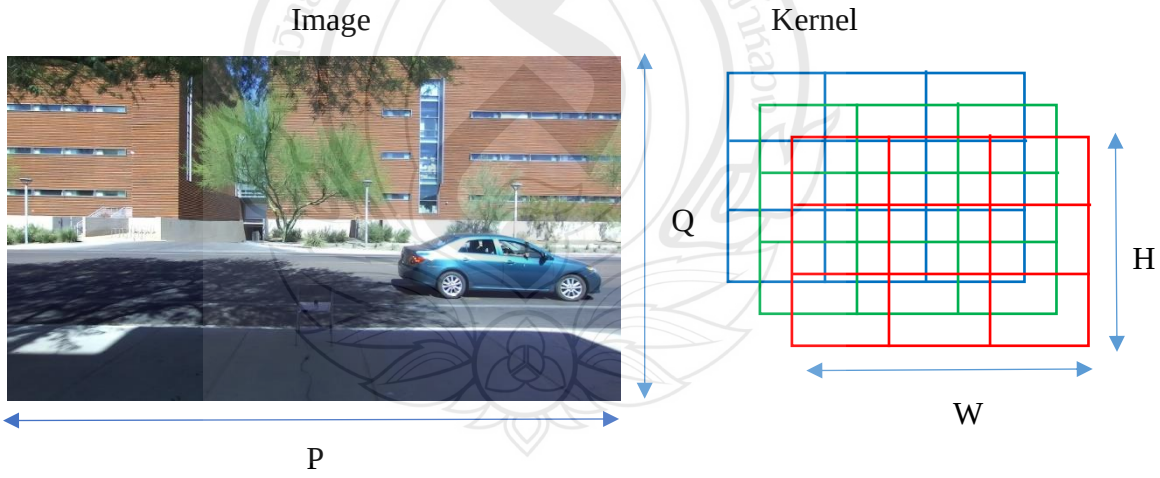


Figure 4.6 Size of the Image and Kernel

Figure 4.6 shows a sample of images that were used in this study and the general structure of a CNN kernel with defined sizes. According to Figure 4.6, the width and

height of the image are P and Q respectively while W and H are the width and height of the kernel respectively. If the depth of the kernel is defined as C, the size of the kernel can be derived as C×W×H.

During the convolutional operation as following Figure 4.7, two operations happen which are multiplication and addition. When the kernel is superposed on the image, multiplications happen between each element. This multiplication is continued until it finishes stride which gives an array of values. Finally, all elements of this array are added.

Figure 4.7 Convolutional Operation

Number of kernels = K

Number of Multiplications = $CWH \times (P - W + 1) \times (Q - H + 1)$ (10)

Number of Additions = $CWH \times (P - W + 1) \times (Q - H + 1)$ (11)

Total Number of Operations per layer

$= 2 \times CWH \times (P - W + 1) \times (Q - H + 1) \times K$ (12)

Computational Complexity per layer = $O\,(CWHK(P - W)\,(Q - H))$ (13)

Number of Layers = L

Therefore,

Computational Complexity of CNN = $O(\sum_{l=1}^{L} (C_l W_l H_l K_l (P\text{-}W_l)(Q\text{-}H_l))$ (14)

4.3.2 Computational Complexity of the Multilayer Perceptron

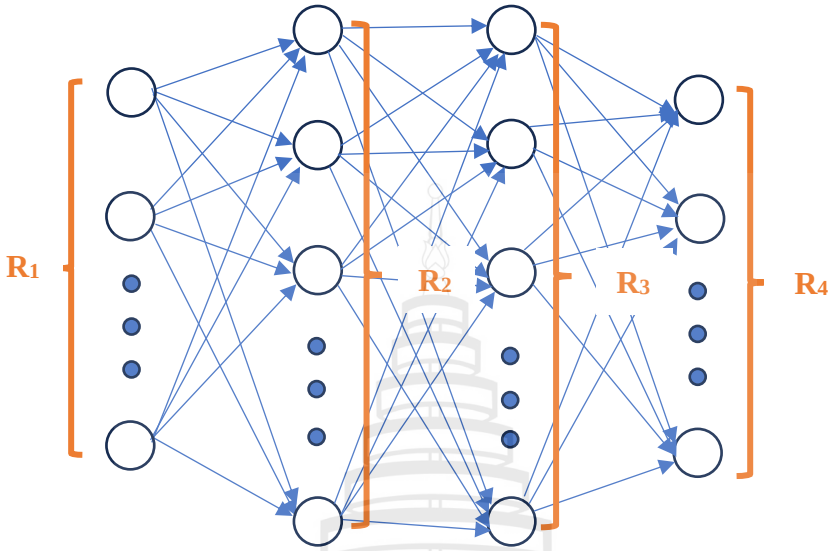


Figure 4.8 A Sample Multilayer Perceptron Neural Network

4.3.2.1 Computational Complexity of Feedforward Propagation: According to Figure 4.8, the following equations can be derived regarding the feedforward process.

Number of multiplications between inputs and weights per one element (neuron) = R_2

Number of Additions of multiplied elements per element = $R_2 - 1$ (15)

Number of bias per neuron = 1

Therefore,

Number of Additions of multiplied elements per element = R_2

Total Number of Operations per layer = $R_1 \times (R_2 + R_2)$

$= 2R_1R_2$ (16)

Batch Size = B

Therefore,

Total Number of Operations per layer = $2R_1R_2B$ (17)

Computational Complexity per layer = $O(RSB)$ (18)

Total Number of Layers = M

Therefore,

Computational Complexity of the Forward Process = $O(\sum_{m=1}^{M-1}(R_m R_{m+1} B))$ (19)

4.3.2.2 Computational Complexity of Backpropagation: In the process of backpropagation, errors are calculated first. After that, "delta weights" are calculated and weights are updated. Computational complexity between two layers is the sum of all these processes as follows.

Regarding the backpropagation process, the following equations can be derived according to Figure 4.8.

Computational Complexity per layer = $O(BR_4+BR_4+BR_4R_3+R_4R_3)$

$= O(2BR_4+R_4R_3(B+1))$

$= O(BR_4+BR_4R_3)$

$= O(BR_4R_3)$ (20)

Computational Complexity of the Backpropagation = $O(\sum_{m=M}^{2}(BR_m R_{m-1}))$ (21)

After finding both feedforward and backpropagation complexity, the total computational complexity of the MLP can be calculated by summation of them as follows.

Computational Complexity of MLP:

$O(\sum_{m=1}^{M-1}(R_m R_{m+1} B))+O(\sum_{m=M}^{2}(BR_m R_{m-1}))$ (22)

4.3.3 Computational Complexity of the Proposed Model

Finally, the complexities of the CNN and MLP are summed up to calculate the total computational complexity of the proposed model.

Total Computational Complexity of the Proposed Model:

$O(\sum_{l=1}^{L}(C_l W_l H_l K_l (P-W_l)(Q-H_l))+O(\sum_{m=1}^{M-1}(R_m R_{m+1} B))+O(\sum_{m=M}^{2}(BR_m R_{m-1})$ (23)

4.4 Discussion

According to the results, it can be seen that the proposed model has significant classification accuracy, precision and recall values. If training and validation loss values of any model decrease to the lower level, it indicates that the model is not underfitted. Therefore, according to Figure 4.9, the proposed model has less underfitting. On the other hand, the gap between training and validation curves implies the overfitting of a model. The proposed model has considerably low overfitting since there is a smaller gap between training and validation curves as shown in Figure 4.9. According to Table 4.4 in section 4.2, the test accuracy of the proposed model dominates among other algorithms.

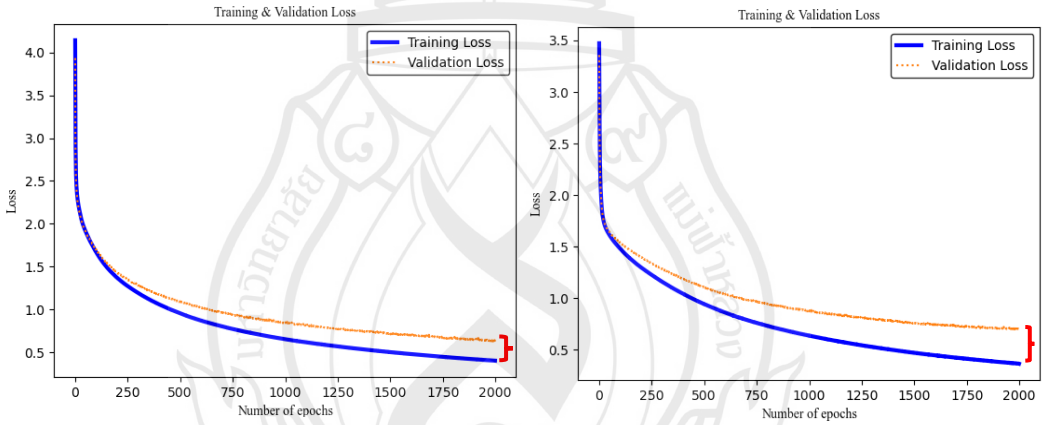


Figure 4.9 Gaps between Training and Validation Curves

The total number of operations per iteration of CNN of the proposed model can be calculated as shown in Appendix A. Similarly, the total number of operations per iteration of MLP of the proposed model can be calculated as shown in Appendix B.

According to Appendix A and B, the total number of operations per iteration of the CNN is 9, 955, 200 while the total amount of operations per iteration of the MLP is 497, 810 respectively. Thus, the total number of operations per iteration of the proposed model can be calculated by the summation of the above two numerical values.

Therefore, the total number of operations per iteration of the proposed model is approximately 7.453×10^{6}.

Table 4.5 shows floating point operations per second (FLOPS) which is one of the methods to study the complexity of a model that quantifies floating point operations. Table 4.5 shows the FLOPS of GooLeNet (Szegedy et al., 2016), VGG16 (Simonyan & Zisserman, 2015), AlexNet (Krizhevsky et al., 2012) and ResNet50 (He et al., 2016).

Table 4.5 FLOPS of GooLeNet, VGG16, AlexNet and ResNet50

Model	**FLOPS**
GooLeNet	1.57×10^{9}
VGG16	1.55×10^{10}
AlexNet	7.25×10^{8}
ResNet50	3.80×10^{9}

According to Table 4.5, the amount of operations per iteration of the proposed model is low compared to the other complex deep learning algorithms like GoogLeNet, VGGNet, AlexNet and ResNet.

CHAPTER 5

CONCLUSION AND SUGGESTION

5.1 Conclusion

This study was conducted to identify an effective approach for machine learning based beam prediction. The study used an existing real-world dataset. Images and GPS coordinates are two inputs of the proposed model. Both inputs were normalized to obtain a common scale during the study. A CNN was used for the feature extraction from images. Then, outputs of CNN and GPS data were concatenated and inserted into an MLP for beam prediction. After that, the PCA technique was applied to reduce the dimensionality of the data which is a novel tryout in machine learning based beam prediction. Furthermore, the random oversampling technique was used to obtain a balanced feature set. The model was assessed on 64-classes and 32-classes contexts. It showed 88.98% and 85.72% test accuracies for 64-classes and 32-classes contexts respectively. Apart from that the proposed model showed less overfitting and underfitting. Furthermore, the proposed model showed the best accuracy among CNN, MLP, SVM, RF and Adaboosting techniques.

5.2 Suggestion

The proposed model can be tested in a real-time testbed that contains data collection equipment which is connected to the model. In the real world, the proposed model can be used to predict beams using input data coming from the BS. However, GPS data can be inaccurate in reality. GPS data can be received by several paths as in the following Figure 5.1.

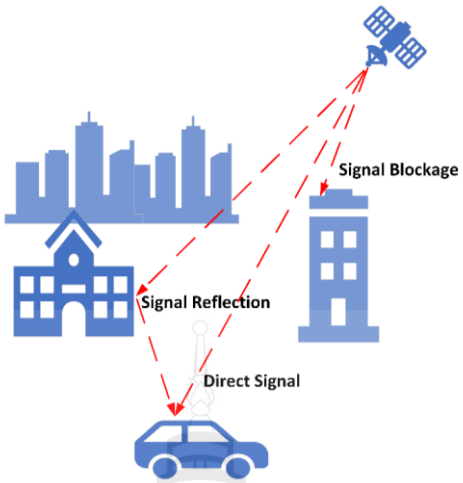


Figure 5.1 Multipath Propagation of Signal

One of the main reasons for inaccurate GPS coordinates is satellite signal reflections and blockages. As a result of that multipath fading can be occurred. Therefore, beam prediction using GPS data may reduce the performance of the model. Hence, future studies could discover new approaches to utilize GPS data effectively for beam prediction.

REFERENCES

REFERENCES

Ahn, Y., Kim, J., Kim, S., Shim, K., Kim, J., Kim, S., … Shim, B. (2023). Toward intelligent millimeter and terahertz communication for 6G: Computer vision-aided beamforming. *IEEE Wireless Communications*, *30*(5), 179–186. https://doi.org/10.1109/MWC.007.2200155

Ahn, Y. (n.d.). *Dataset: Towards intelligent millimeter and terahertz communication for 6G: Computer vision-aided beamforming*. ISLAB. Retrieved August 20, 2023, from https://isl.snu.ac.kr/

Alkhateeb, A. (2019). DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications. *ArXiv*, 1–8. https://arxiv.org/abs/1902.06435v1

Alkhateeb, A., Alex, S., Varkey, P., Li, Y., Qu, Q., & Tujkovic, D. (2018). Deep learning coordinated beamforming for highly-mobile millimeter wave systems. *IEEE Access*, *6*, 37328–37348. https://doi.org/10.1109/ACCESS.2018.2850226

Alkhateeb, A., Charan, G., Osman, T., Hredzak, A., Morais, J., Demirhan, U., … Srinivas, N. (2023). DeepSense 6G: A large-scale real-world multi-modal sensing and communication dataset. *IEEE Communications Magazine*, *61*(9), 122–128. https://doi.org/10.1109/MCOM.006.2200730

Alrabeiah, M., Hredzak, A., & Alkhateeb, A. (2020). Millimeter wave base stations with cameras: Vision-Aided beam and blockage prediction. In *IEEE Vehicular Technology Conference* (pp. 1-5). IEEE. https://doi.org/10.1109/VTC2020-SPRING48590.2020.9129369

Alrabeiah, M., Hredzak, A., Liu, Z., & Alkhateeb, A. (2020). ViWi: A deep learning dataset framework for vision-aided wireless communications. In *IEEE Vehicular Technology Conference* (pp. 1-5). IEEE. https://doi.org/10.1109/VTC2020-SPRING48590.2020.9128579

Ashraf, A. (2023). *Oversampling — Handling imbalanced data.* https://medium.com/@abdallahashraf90x/oversampling-for-better-machine-learning-with-imbalanced-data-68f9b5ac2696

Banerjee, S., & Mandal, D. (2018). Array pattern optimization for steerable circular isotropic antenna array using cat swarm optimization algorithm. *Wireless Personal Communications*, *99*(3), 1169–1194. https://doi.org/10.1007/s11277-017-5171-6

Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.

Burghal, D., Abbasi, N. A., & Molisch, A. F. (2019). A machine learning solution for beam tracking in mmwave systems. In *Asilomar Conference on Signals, Systems and Computers* (pp. 173–177). IEEE. https://doi.org/10.1109/IEEECONF44664.2019.9048770

Charan, G., Hredzak, A., & Alkhateeb, A. (2023). Millimeter wave drones with cameras: Computer vision aided wireless beam prediction. In *2023 IEEE International Conference on Communications Workshops: Sustainable Communications for Renaissance, ICC Workshops 2023* (pp. 1896–1901). IEEE. https://doi.org/10.1109/ICCWORKSHOPS57953.2023.10283784

Charan, G., Hredzak, A., Stoddard, C., Berrey, B., Seth, M., Nunez, H., … Alkhateeb, A. (2022). Towards real-world 6G drone communication: Position and camera aided beam prediction. In *Proceedings - IEEE Global Communications Conference, GLOBECOM* (pp. 2951–2956). IEEE. https://doi.org/10.1109/GLOBECOM48099.2022.10000718

Charan, G., Osman, T., Hredzak, A., Thawdar, N., & Alkhateeb, A. (2022). Vision-Position multi-modal beam prediction using real millimeter wave datasets. In *IEEE Wireless Communications and Networking Conference, WCNC* (pp. 2727–2731). IEEE. https://doi.org/10.1109/WCNC51071.2022.9771835

Cousik, T. S., Shah, V. K., Erpek, T., Sagduyu, Y. E., & Reed, J. H. (2022). Deep learning for fast and reliable initial access in AI-driven 6G mm wave networks. In *IEEE Transactions on Network Science and Engineering* (pp. 1-12). IEEE. https://doi.org/10.1109/TNSE.2022.3201748

Data Collection – DeepSense. (n.d.). Retrieved August 26, 2023, from https://www.deepsense6g.net/data-collection/

Demirhan, U., & Alkhateeb, A. (2022). Radar aided 6g beam prediction: Deep learning algorithms and real-world demonstration. In *IEEE Wireless Communications and Networking Conference, WCNC* (pp. 2655–2660). IEEE. https://doi.org/10.1109/WCNC51071.2022.9771564

Dinata, D. (2023). *CNN, KNN, and SVM Analysis. Models Introduction.* https://medium.com/@deonatan27/cnn-knn-and-svm-analysis-f7679da5358b

Dhalla, A. (2021). *The math of principal component analysis (PCA).* https://medium.com/analytics-vidhya/the-math-of-principal-component-analysis-pca-bf7da48247fc

Domingos, P. O., Silva, F. M., & Neto, H. C. (2005). An efficient and scalable architecture for neural networks with backpropagation learning. In *International Conference on Field Programmable Logic and Applications* (pp. 89-94). IEEE. https://doi.org/10.1109/FPL.2005.1515704

Guo, Y., Wang, Z., Li, M., & Liu, Q. (2019). Machine learning based mmWave channel tracking in vehicular scenario. In *2019 IEEE International Conference on Communications Workshops, ICC Workshops 2019 – Proceedings* (pp. 1-6). IEEE. https://doi.org/10.1109/ICCW.2019.8757185

He, K., & Sun, J. (2015). Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 5353–5360). IEEE. https://doi.org/10.1109/CVPR.2015.7299173

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 770–778). IEEE. https://doi.org/10.1109/CVPR.2016.90

Heng, Y., Mo, J., & Andrews, J. G. (2021). Learning probing beams for fast mmWave beam alignment. In *Proceedings - IEEE Global Communications Conference, GLOBECOM* (pp. 1-6). IEEE. https://doi.org/10.1109/GLOBECOM46510.2021.9685512

Imran, S., Charan, G., & Alkhateeb, A. (2023). Environment semantic aided communication: A real world demonstration for beam prediction. In *2023 IEEE International Conference on Communications Workshops: Sustainable Communications for Renaissance, ICC Workshops 2023* (pp. 48–53). IEEE. https://doi.org/10.1109/ICCWORKSHOPS57953.2023.10283602

Jiang, S., & Alkhateeb, A. (2022). Computer vision aided beam tracking in a real-world millimeter wave deployment. In *2022 IEEE GLOBECOM Workshops, GC Wkshps 2022 - Proceedings* (pp. 142–147). IEEE. https://doi.org/10.1109/GCWKSHPS56602.2022.10008648

Jiang, S., Charan, G., & Alkhateeb, A. (2023). LiDAR aided future beam prediction in real-world millimeter wave V2I communications. *IEEE Wireless Communications Letters*, *12*(2), 212–216. https://doi.org/10.1109/LWC.2022.3219409

Kenji, S. (2011). *Artificial neural networks: Methodological advances and biomedical applications*. https://www.intechopen.com/books/117

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*(2). https://doi.org/10.1145/3065386

Lin, C. H., Kao, W. C., Zhan, S. Q., & Lee, T. S. (2019). BsNet: A deep learning-based beam selection method for mmwave communications. In *IEEE Vehicular Technology Conference* (pp. 1-6). IEEE. https://doi.org/10.1109/VTCFALL.2019.8891363

Liu, X., Yu, J., Qi, H., Yang, J., Rong, W., Zhang, X., … Gao, Y. (2020). Learning to predict the mobility of users in mobile mmWave networks. *IEEE Wireless Communications*, *27*(1), 124–131. https://doi.org/10.1109/MWC.001.1900241

Margineantu, D. D., & Dietterich, T. G. (1997). Pruning adaptive boosting. *ICML*, 97, 211–218.

Misailidis, I., & Voudouris, K. N. (2020). A Review on propagation related issues for mm-Wave and THz next generation cellular networks. *ACM International Conference Proceeding Series*, 1–16. https://doi.org/10.1145/3437120.3437264

Moon, S., Kim, H., & Hwang, I. (2020). Deep learning-based channel estimation and tracking for millimeter-wave vehicular communications. *Journal of Communications and Networks*, *22*(3), 177–184. https://doi.org/10.1109/JCN.2020.000012

Morais, J., Bchboodi, A., Pezeshki, H., & Alkhateeb, A. (2023). Position-Aided beam prediction in the real world: How useful GPS locations actually are?. In *IEEE International Conference on Communications* (pp. 1824–1829). IEEE. https://doi.org/10.1109/ICC45041.2023.10278998

Nalepa, J., & Kawulok, M. (2019). Selecting training sets for support vector machines: a review. *Artificial Intelligence Review*, 52(2), 857–900. https://doi.org/10.1007/s10462-017-9611-1

Nasim, I., Ibrahim, A. S., & Kim, S. (2020). Learning-Based beamforming for multi-user vehicular communications: A combinatorial multi-armed bandit approach. *IEEE Access*, *8*, 219891–219902. https://doi.org/10.1109/ACCESS.2020.3043301

O'Shea, K., & Nash, R. (2015). *An introduction to convolutional neural networks*. arXiv. https://doi.org/10.48550/arXiv.1511.08458

Pan, A., Zhang, T., & Han, X. (2017). Location information aided beam allocation algorithm in mmWave massive MIMO systems. In *2017 IEEE/CIC International Conference on Communications in China, ICCC 2017* (pp. 1–6). IEEE. https://doi.org/10.1109/ICCCHINA.2017.8330410

Raschka, S. (2015). *Principal component analysis*. https://sebastianraschka.com/Articles/2015_pca_in_3_steps.html

Read, S. (n.d.). *5G Signals Dawn of New Era for Mobile Communications*. Retrieved June 24, 2023, from https://www.forbes.com/uk/advisor/mobile-phones/5g-signals-dawn-of-new-era-for-mobile-communications/

Rezaie, S., Manchon, C. N., & De Carvalho, E. (2020). Location- and orientation-aided millimeter wave beam selection using deep learning. In *IEEE International Conference on Communications* (pp. 1-6). IEEE. https://doi.org/10.1109/ICC40277.2020.9149272

Rozé, A., Hélard, M., Crussière, M., & Langlais, C. (2015). *Millimeter-Wave digital beamsteering in highly line-of-sight environments for massive mimo systems*. https://univ-rennes.hal.science/hal-01252090/document

Salem, N., & Hussein, S. (2019). Data dimensional reduction and principal components analysis. *Procedia Computer Science*, 163, 292–299. https://doi.org/10.1016/j.procs.2019.12.111

Sena, M. (2024). *Principal component analysis made easy: A step-by-step*. https://towardsdatascience.com/principal-component-analysis-made-easy-a-step-by-step-tutorial-184f295e97fe

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.

Sonawane, J. S., & Patil, D. R. (2015). Prediction of heart disease using multilayer perceptron neural network. *2014 International Conference on Information Communication and Embedded Systems, ICICES 2014*, *978*, 1–6. https://doi.org/10.1109/ICICES.2014.7033860

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Tian, Y., & Wang, C. (2021). Vision-Aided beam tracking: Explore the proper use of camera images with deep learning. In *IEEE Vehicular Technology Conference* (pp. 1-5). IEEE. https://doi.org/10.1109/VTC2021-FALL52928.2021.9625195

Tripathi, S., Sabu, N. V., Gupta, A. K., & Dhillon, H. S. (2021). Millimeter-wave and terahertz spectrum for 6G wireless. In *6G Mobile Wireless Networks* (pp. 83–121). Springer International Publishing. https://doi.org/10.1007/978-3-030-72777-2_6

Va, V., Shimizu, T., Bansal, G., & Heath, R. W. (2017). Position-aided millimeter wave V2I beam alignment: A learning-to-rank approach. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC* (pp. 1–5). IEEE. https://doi.org/10.1109/PIMRC.2017.8292679

Yi, W., Zhiqing, W., & Zhiyong, F. (2024). *Beam training and tracking in mmWave communication: A survey*. China Communications. https://doi.org/10.23919/JCC.EA.2021-0873.202401

Zhang, Y., Alrabeiah, M., & Alkhateeb, A. (2020). Learning beam codebooks with neural networks: Towards environment-aware mmWave MIMO. In *IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC* (pp. 1-5). IEEE. https://doi.org/10.1109/SPAWC48557.2020.9154320

APPENDICES

APPENDIX A

TOTAL NUMBER OF OPERATIONS PER ITERATION OF CNN OF THE PROPOSED MODEL

C = 3

P = 48

Q = 27

W = 3

H = 3

Table A1 Total number of operations per iteration of CNN of the proposed model

Kernel value (K)	**Equation for the number of operations**	**Outcome of the equation**
16	First layer $2\times CWH \times (P\text{-}W + 1) \times (Q\text{-}H+1)\times K$	993,600
32	Second layer $2\times CWH \times (P\text{-}W + 1) \times (Q\text{-}H+1)\times K$	1,987,200
64	Third layer $2\times CWH \times (P\text{-}W + 1) \times (Q\text{-}H+1)\times K$	3,974,400

Total Number of operations of the CNN = 993, 600 + 1, 987, 200 + 3, 974, 400

= 6,955,200

APPENDIX B

TOTAL NUMBER OF OPERATIONS PER ITERATION OF MLP OF THE PROPOSED MODEL

R_1=100 R_2=81 R_3=54 R_4=64

B = 10

Table B1 Total number of operations per iteration of MLP of the proposed model

Equation for the number of operations	Outcome of the equation
First layer $2R_1R_2B + (2BR_2+R_2R_1(B+1))$	252, 720
Second layer $2R_2R_3B + (2BR_3+R_3R_2(B+1))$	136, 674
Third layer $2R_3R_4B + (2BR_4+R_4R_3(B+1))$	108, 416

Total number of operations of the MLP = 252, 720 + 136, 674 + 108, 416

= 497, 810

APPENDIX C

IMAGE DATA OF THE DATASET

The following figures C.1, C.2, C.3, C.4, C5, C6, C7 and C8 represent image data of scenarios 02, 05, 07 and 08 of the dataset (Data Collection – DeepSense, n.d.).

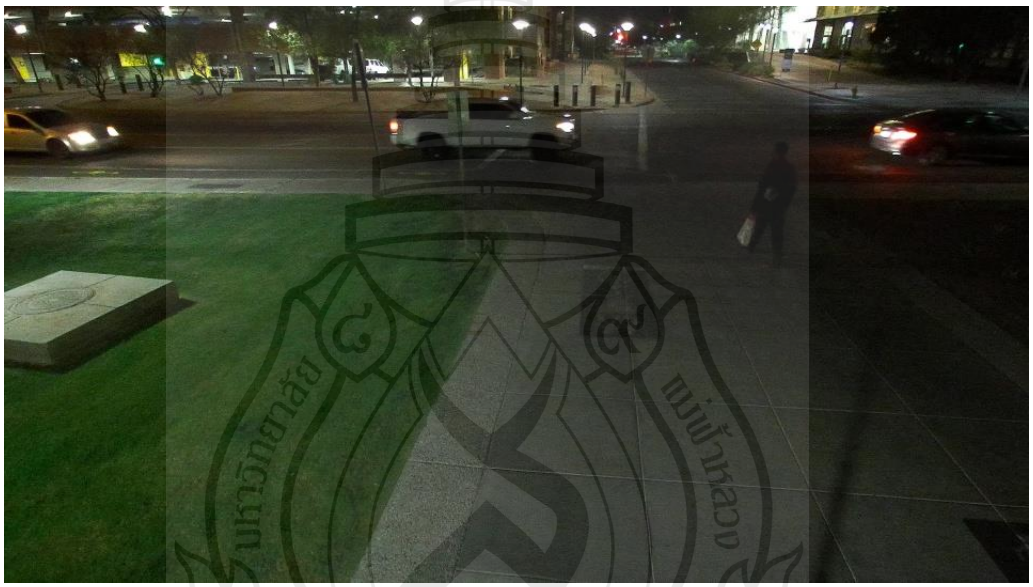


Figure C1 A Sample Image of Scenario 02

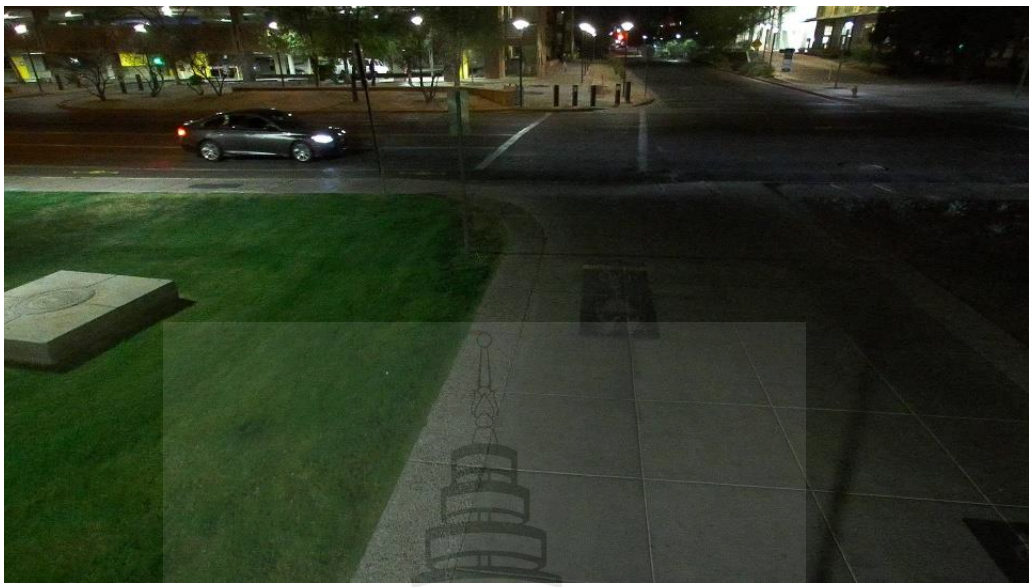


Figure C2 A Sample Image of Scenario 02

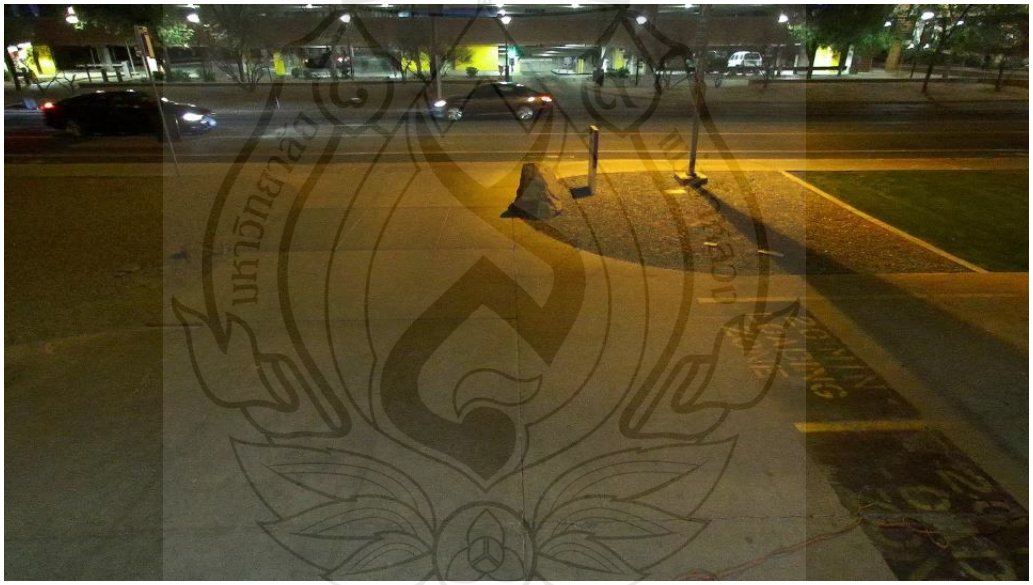
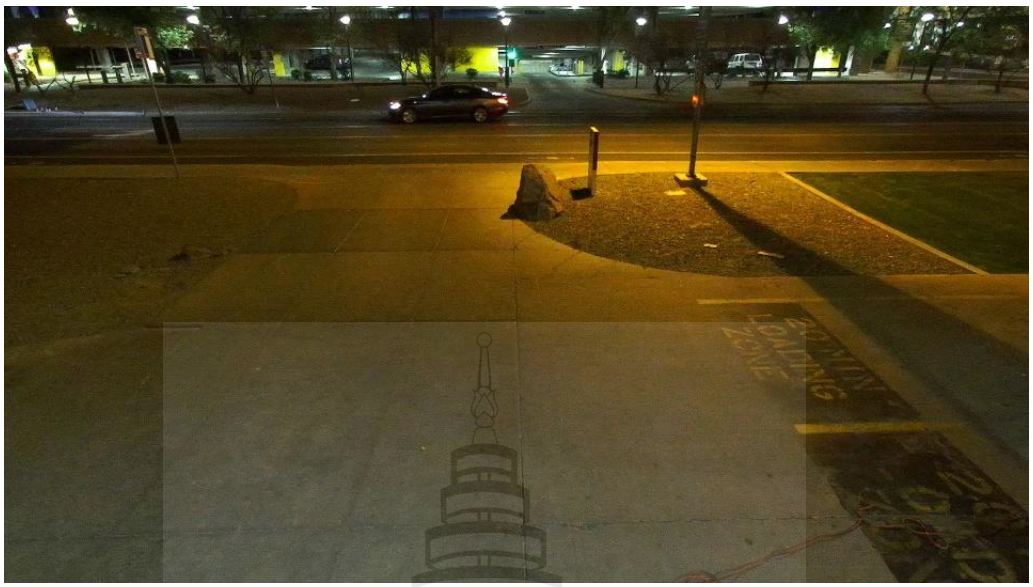


Figure C3 A Sample Image of Scenario 05



Figures C4 A Sample Image of Scenario 05

Figure C5 A Sample Image of Scenario 08

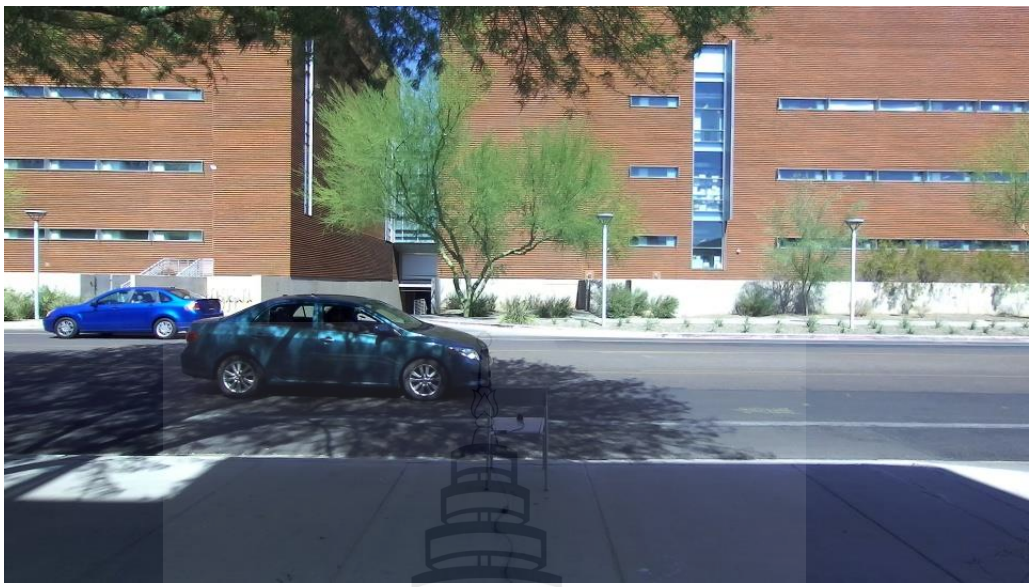


Figure C6 A Sample Image of Scenario 08

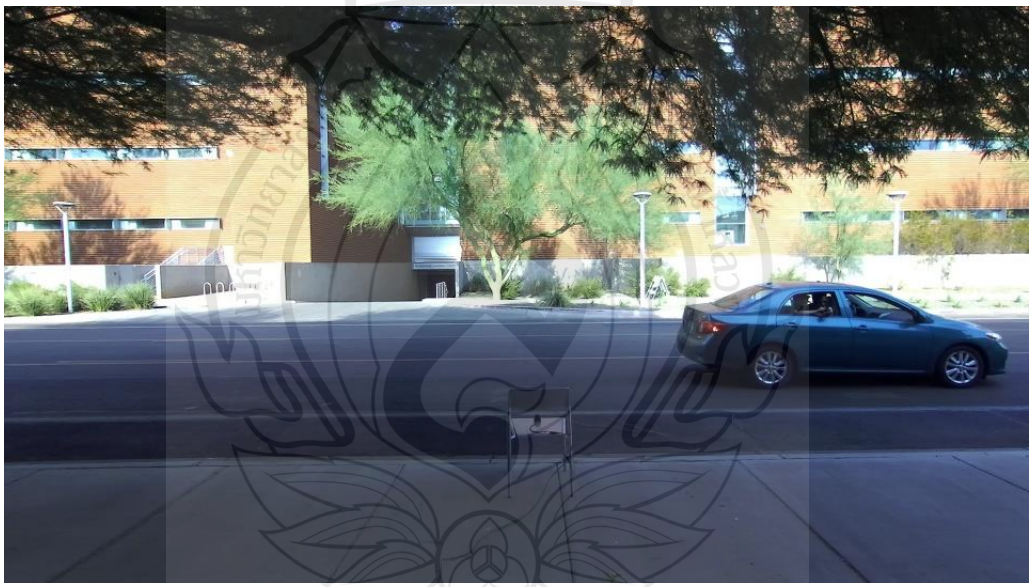


Figure C7 A Sample Image of Scenario 09

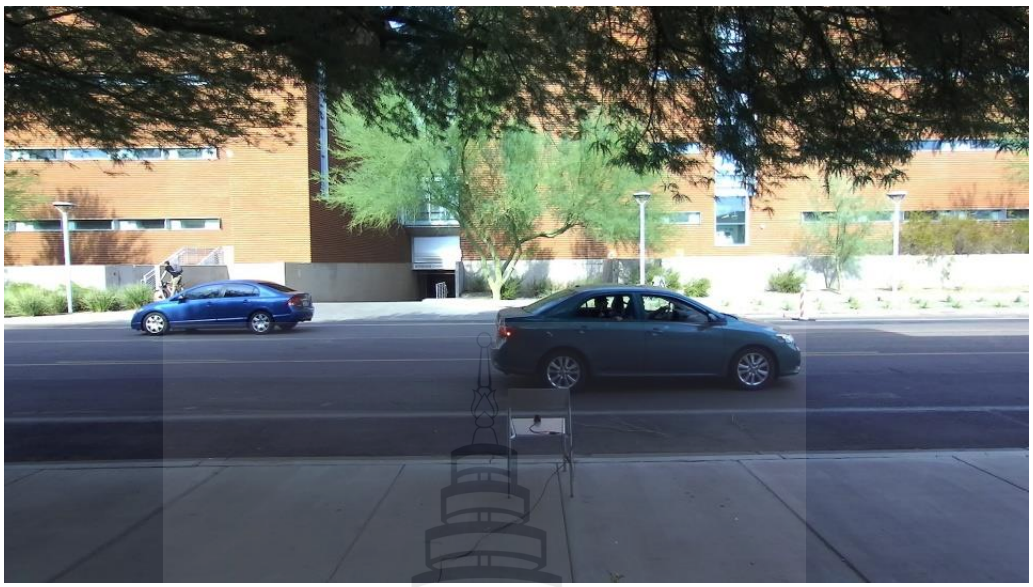


Figure C8 A Sample Image of Scenario 09

CURRICULUM VITAE

CURRICULUM VITAE

NAME Charith Dissanayake

EDUCATIONAL BACKGROUND

2017 Bachelor of Science (Special)
Computing and Information Systems
Sabaragamuwa University of Sri Lanka

WORK EXPERIENCE

2021-Present Lecturer (Probationary) (On study leave)
Gampaha Wickramarachchi University of Indigenous Medicine, Sri Lanka

PUBLICATION

Dissanayake, D. M. C., & Temdee, P. (2023). Beam prediction using convolutional neural network and artificial neural network. *Proceedings of the 7th International Conference on Information Technology (InCIT 2023)*, Chiang Rai, Thailand, 15- 17 November, 2023.