Pakistani Traffic-sign detection using Deep Learning

Author

Zain Nadeem

Regn. Number

319776

Supervisor

Dr. Muhammad Jawad Khan

MS ROBOTICS & INTELLIGENT MACHINE ENGINEERING

DEPARTMENT OF ROBOTICS & AI

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

AUGUST 2022

Pakistani Traffic-sign detection using Deep Learning

Author

ZAIN NADEEM

Regn Number

319776

A thesis submitted in partial fulfillment of the requirements for the degree of

MS ROBOTICS & INTELLIGENT MACHINE ENGINEERING

Thesis Supervisor:

DR. MUHAMMAD JAWAD KHAN

Thesis Supervisor's Signature: ____

MS ROBOTICS AND INTELLIGENT MACHINE ENGINEERING
DEPARTMENT OF ROBOTICS & AI
SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY
ISLAMABAD
AUGUST 2022

Thesis Acceptance Certificate

It is certified that the final copy of MS Thesis written by Zain Nadeem (Registration No. 319776), of Department of Robotics & AI (SMME) has been vetted by undersigned, found complete in all respects as per NUST statutes / regulations, is free from plagiarism, errors and mistakes and is accepted as a partial fulfilment for award of MS Degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in this dissertation.

Signature: ______________________________

Date: ___________________________________

Dr. Muhammad Jawad Khan (Supervisor)

Signature HOD: _________________________

Date: ___________________________________

Signature Principal: ______________________

Date: ___________________________________

FORM TH-4

MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by **Zain Nadeem** having **Regn. No. 319776**, titled "**Pakistani Traffic-sign detection using Deep Learning**", be accepted in partial fulfillment of the requirements for the award of MS Robotics & Intelligent Machine Engineering degree.

Examination Committee Members

1. Dr. Hassan Sajid Signature: _______________

2. Dr. Karam Dad Signature: _______________

3. N/A Signature: _______________

Supervisor: Dr. Muhammad Jawad Khan Signature: _______________

_______________________ Date

_______________________ Head of Department

COUNTERSINGED

_______________________ Date

_______________________ Dean/Principal

Declaration

I certify that this research work titled "*Pakistani Traffic-sign detection using Deep Learning"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student
Zain Nadeem
319776

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

Zain Nadeem

319776

Signature of Supervisor

Dr. Muhammad Jawad Khan

Copyright Statement

Acknowledgements

I am truly thankful to Allah the Almighty for all the blessings throughout my life especially in my educational career. Allah has always guided me towards the path which was best for me in all aspects. Pursuing my master’s degree and completing the course work, research work would not be possible without the guidance of Almighty Allah.

I would like to express my heartily thanks to thesis supervisor Dr. Muhammad Jawad Khan for trusting in me, by providing me the opportunity to work with him. He was not only my supervisor but also taught me 04 subjects during my Masters’ coursework which I am proficient in, thanks to him. His guidance, support and availability at every part of my research phase made me complete this work. I would like to thank him for guiding me in all the events other than my research work which has helped me in grooming my technical skills and being motivated. His efforts and dedication throughout my research work were the key element for this accomplishment.

I would like to acknowledge my committee members Dr. Hasan Sajid and Dr. Karam Dad for their support and encouragement throughout my course work and research work.

I am very thankful to my parents for supporting and loving me throughout my life and always motivating me by providing a practical example of hard work, honesty, dedication and commitment towards every phase of life.

Dedication

Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment.

Abstract

Traffic-sign Detection is one of the major aspects of the working of a modern car, more so in the case of a self-driving car. They need to be detected and recognized up to a certain degree of accuracy. This research revolved around the detection of Pakistani Traffic-sigs. The research was conducted in 3 phases; firstly, a fixed camera was used to collect video feed from real-world car rides. These videos were then extracted and manually annotated using pertinent software tools. This helped create a dataset of images of traffic signs which was important as the model being used is Deep Learning-based, which require colossal amounts of data to function properly. Secondly, this data is used to train a Deep Learning model to detect and classify the type of traffic sign. The trained model produced a mean average precision (mAP) of 75.636% on the training dataset and 49.699% on the validation dataset while the mAP stood at 43.453% for the test dataset. All these results are state-of-the-art and strong enough for implementation as real-world models. The model was cross-validated and regularized to help improve the model's working. The final model was tested in real-world scenarios and tweaked according to requirements. cl

Key Words: Traffic-sign recognition, Deep Learning, Object Detection, Faster-RCNN

Table of Contents

List of Figures

List of Tables

List of Acronyms

1. CNN	Convolutional Neural Network
2. ReLU	Rectified Linear Unit
3. FC Layer	Fully Connected Layer
4. SGD	Stochastic Gradient Descent
5. ANN	Artificial Neural Network
6. ML	Machine Learning
7. DL	Deep Learning
8. AI	Artificial Intelligence
9. CV	Computer Vision

CHAPTER 1: INTRODUCTION

The world has moved towards Industry 4.0—Artificial Intelligence—while Pakistan is still playing catch-up with the rest of the world. According to an article published by Dawn Newspaper on February 13th, 2019, road accidents in Pakistan claimed 36,000 lives in 2018 alone and these numbers have continued to rise ever since. Therefore, measures must be taken to minimize road accidents. To accomplish this, there is a need to make the cars smarter and the driver more able to use the information to make better decisions on the road [1].

Traffic signs are a universal way for traffic regulations to be enforced and these self-driving cars need to recognize them for them to work safely. Initially, these traffic signs along with traffic types were detected using conventional image processing systems which were both slower and less accurate. These systems worked based on visual features such as colors, and shapes with algorithms such as Color Segmentation used widely [2], [3]. Oher notable algorithms include Scale Invariant Feature Transform, Speeded-up Robust Features, and Binary Robust Invariant Scalable Keypoints among others [4]–[6].

More recently learning-based algorithms have replaced them and have successfully been implemented on traffic type and traffic-sign recognition problems such as the use of CNNs of German Dataset – GTSRB [7]. CNNs require a huge no. of images to work efficiently, and there is an absence of any maintained dataset containing traffic-sign images from Pakistan. Furthermore, Pakistani traffic signs differ from other signs around the world hence an indigenous dataset is required. There is a need to gather a diverse set of images from across Pakistan, in different lighting conditions and using various cameras and imaging modes. Labeling the acquired data to accurately detect and classify traffic-sign images will turn it into an excellent benchmark for future research.

1.1 Problem Statement

Self-driving cars are the next step in the evolution of the automobile industry. Although they were meant to be a sign of luxury, they carry a lot more benefits. These range from environmental impacts to better traffic system which in turn brings a net positive change in the society as a whole. These self-driving cars require a certain number of elements to work properly including, but not limited to, traffic signs, nearby vehicles, and pedestrians.

1.2 Proposed Solution

To cater for arrival of self-driving cars in Pakistan a computer vision system needs to be made which can detect traffic signs and types of vehicles among other things. This system is a deep learning-based model and needs thousands of images to train properly. Firstly, the data is collected in form of videos, using different cameras and in different lighting conditions. The video keyframes are then extracted from the collected videos and subsequently annotated against the set classes. These images (keyframes) are compiled into a single dataset—Pakistani Traffic Sign Recognition Dataset (PTSD). Secondly, the aforementioned self-collected dataset is used for training a Deep Learning-based model for the detection and recognition of traffic signs. The model is then cross-validated and regularized to help improve the model's working. The final model is tested in real-world scenarios and, thereafter, tweaked according to requirement.

1.3 Expected Outcome

The aim of this project was multifold and pertinent to real-world problems faced on the roads in Pakistan. These include, but are not limited to, real-time monitoring of conditions around the drivers' vehicle using an Advanced Driver Assistance System—ADAS—which can later be evolved and geared towards Self-driving Cars and Smart City initiatives. This research was intended to, firstly, provide a massive dataset for the training of other models related to traffic signs. Secondly, the research focused on a trained model for the detection of traffic signs.

1.4 Methodology

The research was conducted in 2 distinct phases. Firstly, the data was collected in form of videos. The different cameras used for the videography include smartphone cameras and a dashcam, all mounted on the car windshield. The videos were taken in different lighting conditions to avoid low variance in the evaluation model. From these videos, the keyframes were extracted and subsequently annotated against the set classes. Secondly, the self-collected video frame dataset was used for training a Deep Learning-based model for the detection and recognition of traffic signs. The model was then cross-validated and regularized to help improve the model's working. The final model was tested in real-world scenarios and, thereafter, tweaked according to requirement.

1.5 Thesis Overview

The thesis is further divided into the following chapters; firstly, the current literature present on the topic is reviewed in detail, in the **Literature Review** chapter, to extract the shortcomings and research gaps in relevant state-of-the-art solutions. Afterward, based on an in-depth analysis of these issues, the process used to reach the solution has been described along with the simulation setup in the **Methodology** chapter, which also explains the process of data collection and all the pre-processing which has gone in to make the data ready for the detection models. Next is the **Results** chapter which discusses the output of the models, its training, validation, and testing results along with other performance metrics. The **Discussion** chapter revolves around the novelty, improvements over the state-of-the-art, and future work possible in the research area. Finally, the **Conclusion** chapter rounds up the article's key achievements and outcomes of the conducted research.

CHAPTER 2: LITERATURE REVIEW

2.1 Related Work

The idea of a system that assists the driver to possible blind spots and missed cues has been discussed as an easier way to automation and autonomy. Some of the basic elements of such Driver Assistance Systems included sensing the environment, obstacles, traffic-signs, pedestrians, and other vehicles [8]–[12].

Traditional Image Processing techniques were previously used [13]–[17]. However, these methods are way slower compared to the current state-of-the-art practices, and that is cause for hindrance. Considering the application of our project, the accuracy and precision of the ADAS is paramount to the safety of the user, the vehicle, and the traffic on the road.

Research on automated detection of traffic signs was initiated in Japan back in 1984 as per [18]. Several other techniques which utilized the spatial aspects of the image among other such as shapes, and color have been used for related research and several publications endorse this idea [19]–[22].

Several conventional image processing-based approaches [23], [24] have been used for identification of traffic signs and types. Detection of traffic signs and types is a very time sensitive process where small delays can result in fatality and conventional techniques are not applicable here. This is due to the slower processing times of these traditional techniques which at a point become a bottleneck.

When working around learning-based techniques huge number of images are required and several organized datasets exist such as LISA Traffic-sign Dataset comprising of American traffic-signs [25], German Traffic-sign Dataset [7], and Belgian Traffic-sign Dataset [26] to name a few. All these datasets have differing number of classes, but the thing common theme is that these datasets contain huge number of images like 39,000 images in the German Dataset mentioned above.

Furthermore, these datasets are purposefully built and curated and regularly used to inspire new learning-based model by pitting them against each other in competitions [7]. This has proved fruitful as these competitions have resulted in a great deal of literature and state-of-the-art computer vision models being developed all of which seem to agree on the assumption that more data mean better performance.

2.1.1 Traffic sign Recognition

Traffic signs recognition has been an active area of research for quite some time with various degree of performance metrics publish depending on several factors. These factors are both intrinsic and extrinsic, meaning that not only the quantity, quality and other image aspects but also the type of algorithm used for recognition play a role.

The survey paper titled “Traffic Signs in the Wild” [27] published in 2017 summarized multiple approaches for traffic-sign recognition used during “Video and Image Processing Cup 2017”. These included, but were not limited to, traditional image processing and ranged up to state-of-the-art deep learning and convolution-based networks. The main theme of nearly all approaches discussed here was that they needed several thousand images to work. Furthermore, these models worked on a single image bases and not on continuous video frames.

Another paper titled “Transfer Learning Based Traffic Sign Recognition Using Inception-v3 Model” [28] introduced transfer learning using Inception-v3 and data augmentation on Belgian traffic sign dataset to improve the accuracy to 99%. This was starting of the transfer learning trend in Computer Vision which pushed the academia towards adopting deeper pre-trained networks and fine tuning them on relatively smaller datasets. The model implemented already had thousands of images which were then augmented to further increase them. In case of Pakistani traffic sign dataset, the low number of images present a bottleneck which even augmentation will not be able to circumvent.

A very recent paper titled “Traffic Sign Classification Comparison Between Various Convolution Neural Network Models” [29] presented an interesting comparison study. The paper compares 3 different CNN architectures with a custom 8-Layer CNN producing the best accuracy of 96%, the other architectures used areVGG-16 and ResNet50. As is the case with most other paper of this domain, we can see that they are working on German/Belgian/US or some other well-maintained datasets having thousands of images. The same results are not replicable on Pakistani dataset due to lack of organized datasets available.

2.1.2 Pakistani Traffic sign Dataset

The bases of research specifically in terms of traffic signs in Pakistan was set in 2014 by researchers from NUCES, Islamabad in their paper titled "Detection and Recognition of Traffic Signs from Road Scene Images" [12]. They collected a custom dataset and used color-based segmentation followed by application of Hough transform to find circles, triangles, or rectangles in the images. Recognition was done using feature matching techniques of conventional image processing such as SIFT, SURF and BRISK. This resulted in slow processing times due to long pipeline of implementation and model was not robust enough as accuracy fell when the images were not cropped or contained a background.

The second available work done on Pakistani traffic sign images was in paper published in 2018 [30] in which detection and identification of traffic signs through marker-based technique was done. The process of recognition was augmented with audio instructions to the driver to introduce a holistic system. Although, no detection technique or accuracy numbers mentioned in the paper explicitly a dataset was collected, and the system required vast improvements due to slow processing times of the proposed techniques.

The first real attempt to introduce learning-based algorithms in Pakistan was done in "A Transfer Learning based approach for Pakistani Traffic-sign Recognition; using ConvNets" [31]. A dataset of 359 images was collected, cropped and labelled according to classes defined as per local standards. A model pre-trained on German dataset was fine-tuned on this Pakistani dataset to induce transfer learning. The pilot project achieved a low class accuracy of 41% due to small number of images 359. Authors from NUCES, Islamabad increased number of images to 579 in turn improving the class accuracy to 72%, but significant overfitting was observed in this case [32].

2.2 Convolutional Neural Networks

According to Li Deng and Dong Yu in [33], deep learning is a class of machine learning algorithms that uses raw input to extract features by utilising several layers. The features extracted are of higher level, incorporating greater detail into the model. Deep learning was introduced to machine learning by Rina Dechter in 1986 in [34]. Since then various developments have taken place overtime such as the daw of neural networks working in both supervised and unsupervised conditions. Deep learning or convolutional neural networks (CNN) are part of the unsupervised realm of machine learning.

Among the pioneers of the development were Y. Lecun, L. Bottou, Y. Bengio and P.

Haffner who developed the LeNet [5]. It was a 7-level convolutional network used to classify handwritten digits on cheques. Its major constrain was the high computational requirement. Later in 2012 Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton developed the CNN, AlexNet [35] that won the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) held by ImageNet. The network had an architecture akin to LeNet but was deeper and consisted of more filters. Filters included, 11x11, 5x5, 3x3, convolutions, max pooling, dropout, data augmentation, ReLU activations and SGD with momentum.

Then in 2013, the ILSVRC was won by ZFNet, also a CNN, developed by Zeiler, Matthew D., and Rob Fergus [36] achieving a top-5 error rate of 14.8 percent, better than last year's AlexNet which had a top-5 error rate of 15.3 percent. This was achieved by primarily fine-tuning and tweaking the hyper-parameters of the AlexNet architecture. In 2014, the competition was won by GoogleNet, codenamed in the journal as Inception V1 [37]. It made the first big leap after the AlexNet in terms of a top-5 error rate of 6.67 percent. It is based on the LeNet architecture and used batch normalisation, image distortions ad RMSprop. This novelty is dubbed as the Inception Module. This worked on reducing the number of parameters, using a 22-layer deep CNN to reduce the parameter from 60 million of AlexNet to 4 million. The runner-up to the GoogleNet was the VGGNet developed by Simonyan, Karen and Zisserman, Andrew [38] of the Oxford Robotics Institute.

The VGGNet consisted of 16 convolutional layers of 3x3 convolutions with more filters than the AlexNet. Its uniform structure makes it a go-to for various applications as a baseline feature extractor. Then in 2015, ResNet took the the ILSVRC crown, formally called the Residual Neural Network (RNN) [39]. RNN introduces skip connections, also called gated units allows this neural network to uses 152 layers while retaining a computational complexity less than VGGNet. It achievd a top-5 error rate of 3.57 percent.

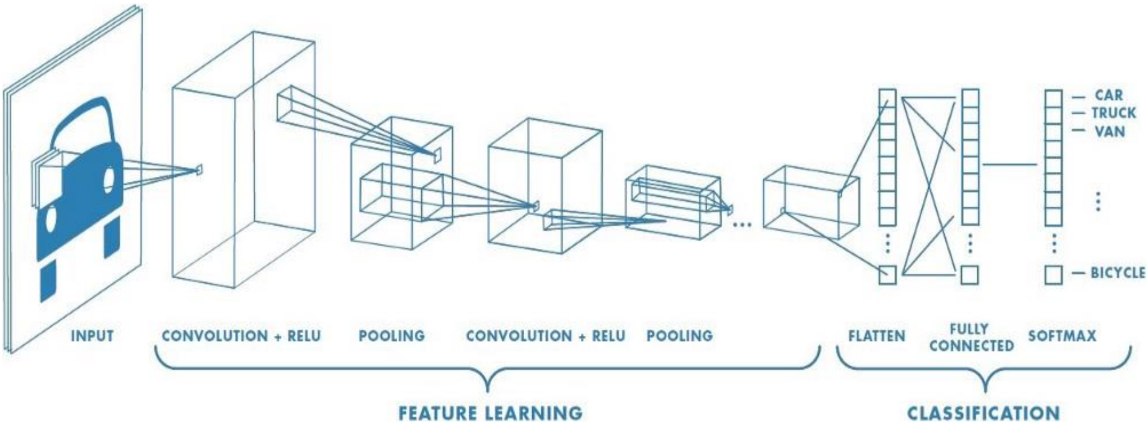

Figure 1: Convolutional Neural Network [40]

2.2.1 Convolution Layers

Convolutional Layers are responsible for the convolution of the Input Image and the filter to extract the required features and generates a feature map according to the filter size. Filter size is determined by the size of the Input Image. Filter consists of two parts the filter size *F* and the total amount of filters *K*. The input of the convolutional Layer would be the Input Image dimensions (*W(i)* * *H(i)* * *D(i)*) and the output (*W(o)* * *H(o)* * *D(o)*) where *D(o)* is equal to the total amount of filters *K* and *W(o)* and *H(o)* can be calculated by the following equation [41].

$$\frac{((W(i),\ H(i)) - F) + 2p}{Stride + 1}$$

Where,
W(i), H(i) is the Input Size of the square image
F is the Filter Size
p is the Padding

2.2.2 Parameter Calculation

Parameters for each convolution layer are calculated to get the overall trainable and non- trainable parameters in the model and to calculate the complete memory consumption of the network. If we have an input of (*W(i)* * *H(i)* * *D(i)*) and a convolution filter *(W(f)* * *H(f)* * *D(f))* where *W(i)*, *H(i)* and *D(i)* are the Width, Height and Dimension of the input to the convolutional layer and *W(f)*, *H(f)* and *D(f)* are the width, height and total number of feature maps in a convolution filter. Thus the parameters can be calculated by using the following formula:

$$(W(f) * H(f) * D(i) + 1) * D(f)$$

2.2.3 Pooling Layers

Pooling Layers are used to reduce the total number of parameters which will be used further in the network and it also reduces the overall computational cost [41]. Most commonly used pooling techniques include Average Pooling and Max Pooling.

2.2.4 Dropout Layers

Dropout Layers were made to avoid overfitting or under fitting of the model on the given dataset. It chooses the amount of nodes which will be used in the training process. These Layers are commonly used after fully connected layers which are prone to overfitting [41].

2.2.5 Activation Function

Activation function helps in providing the non-linear relation between the class of image and Image Data. They determine that which neuron should be fired or not depends upon the relevancy of the neuron towards the required output [41]. Various activation functions are being used which includes tanh, sigmoid, Relu, Leaky Relu etc.

2.2.6 Optimization Techniques

Optimization techniques are used to calculate the weights for your model. They update the weights in the learning process until you reach towards your desired output. Various optimization techniques are used which includes SGD, SGD with momentum, NAG, Adagrad, RMSprop and Adam [67].

2.2.7 Flatten Layers

Flatten Layers are responsible for converting the data into one dimensional vector so it could be fed to Fully Connected Layers where classification will be completed.

2.2.8 Fully Connected Layers

These are the feed forward neural networks. The first FC layer collects the data from the last convolution Layer after getting flattened into one dimensional vector to compute the classification and the Last FC Layer provides the final probabilities calculated for each label.

2.2.9 Accuracy Calculation

It is calculated by using the f1 Score which has two metrics Precision and Recall. Precision describes the number of true class predictions which truly belongs to the true class whereas recall defines the number of true class predictions completed out of all the true samples in the complete dataset.

Formulas for each is given below,

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$f1\ Score = 2 . \frac{Precision\ .\ Recall}{Precission\ +\ Recall}$$

Where, *TP* is True Positive, *FP* is False Positive, *FN* is False Negative

CHAPTER 3: METHODOLOGY

3.1 Simulation Setup

The training process is performed on an Nvidia Tesla P100 GPU provided by the Google Collaboratory. The training has been performed for 10,000 epochs/iterations to obtain precise and stable results. The training time is 2:35:11 (0.9313 s / it) on the aforementioned GPU.

Table 1: Simulation Setup details

Operating system	Ubuntu 18.04.5 LTS
GPU	Tesla P1000 – PCIe – 16GB
RAM	26 GB
Programming Language	Python 3.7.10
CUDA	Version 10.2.228
PyTorch	1.10.0

3.2 Data Collection

Deep Learning-based learning algorithms require massive amounts of data to be able to generalize. This is due to their inherent properties of modeling around the available training data. This is an area in which we as a country lack and there is a severe shortage of available data. To rectify this issue this research included a data collection phase where videos were collected from across Pakistan.

The videos collected were 30 in number and deliberately collected with varying properties such as framerate (fps), brightness, exposure, and lighting settings. These properties are known to affect the results of any learning-based model ultimately. The framerate is responsible for the number of frames being extracted from each second and will affect the number of images in total.

The total runtime of these 30 videos amounted to 05 hours, 42 minutes, and 01 seconds.

Of these 30 videos, 23 were collected from across a few cities in Pakistan including, but not limited to, Quetta, Karachi, Lahore, Islamabad, and Rawalpindi, and totaled 02 hours, 35 minutes, and 58 seconds of video footage. Further videos were fetched from various open-source video-sharing platforms with a total of 03 hours, 06 minutes, and 03 seconds.

3.3 Preprocessing

The next part after data collection is getting that ready for training and it starts with extracting individual frames from the video footage. The video framerate dictates the number of extracted frames from each second of that video. Considering various framerates of each video and the total runtime all the videos equate to approximately 0.56 million frames/images. Even though it was stated above that the number of training images is generally directly proportional to model performance, in this specific case a lot of the extracted frames had little or no visual change and would only prove expensive processing-wise. This is because the spatial features present in these adjacent frames are usually very similar and will not add any benefit.

To cater to this, ‘key frames’ were extracted according to equation (1). This resulted in 109,463 final number of frames. Furthermore, the input dimensions of all the images/frames being passed on to any learning-based algorithms need to be constant. In the case of this research, this resolution was fixed at 640 × 380. Another reason for setting the resolution to this specific value was that the videos also differed in aspect ratios and resolutions and some of these were very high. Higher resolution, while carrying better spatial features, also mean longer training time and use of precious computational resources. After a certain point, it becomes important to look at the cost-benefit analysis of the input resolution. It is generally observed that after a certain resolution, any increase will return a very negligible increase in model performance, but it will take a significantly longer time to train.

3.4 Data Annotation

After preprocessing the video to set requirements, the frames need to be annotated for the presence of relevant objects. This is important as the annotations are the labels that are passed to the deep learning model in a supervised learning scenario. The annotations are done using the Computer Vision Annotation Tool—CVAT—from Intel. It can output the annotations in various formats depending on the type of model being trained.

The types of objects to be detected were divided into five main classes, where four of them are for traffic types and the remaining one is for traffic signs. The four types of traffic being considered in this research are pedestrians, bikes (bicycles and motorbikes), LTVs, and HTVs. The traffic sign superclass is then further divided into 35 subclasses based on the exact type of traffic sign.

Table 2: Class names for Traffic signs

#	**Sign Name**
1	Bridge Ahead
2	Cross Roads
3	Give Way
4	Left Bend
5	No Horns
6	No left turn
7	No Mobile Allowed
8	No Overtaking
9	No Parking
10	No right turn
11	No U-Turn
12	Parking
13	Pedestrians
14	Railway Crossing
15	Right bend
16	Road Divides

17	Roundabout Ahead
18	Sharp Right Turn
19	Slow
20	Speed Breaker Ahead
21	Speed Limit (20 kmph)
22	Speed Limit (25 kmph)
23	Speed Limit (30 kmph)
24	Speed Limit (40 kmph)
25	Speed Limit (45 kmph)
26	Speed Limit (50 kmph)
27	Speed Limit (60 kmph)
28	Speed Limit (65 kmph)
29	Speed Limit (70 kmph)
30	Speed Limit (80 kmph)
31	Steep Descent
32	Stop 1
33	Stop 2
34	U-Turn
35	Zigzag Road Ahead

3.5 Flow Diagram

The overall flow of the data and all the individual steps are shown in the flow diagram in figure 2 below. The process starts with key preprocessing as detailed in section 3.3 above, it includes the key frame extraction, resizing and train/test split steps. The train/test split is done to distribute the data into two parts, one used for training the data and the other used to test its performance of the model by emulating real-world condition where the model will encounter unseen traffic signs and types. The preprocessed data is then annotated and then passed on to the proposed Convolutional Neural Network—CNN.

The proposed CNN will carry out three steps in general—using varying techniques based on type of CNN being used—extracting relevant features, detecting objects, and classifying them. The CNNs being used for the task of this research is Faster RCNN and the detailed architecture has been further detailed in the next section. Consequently, predictions are made, and further decision are taken based on them.

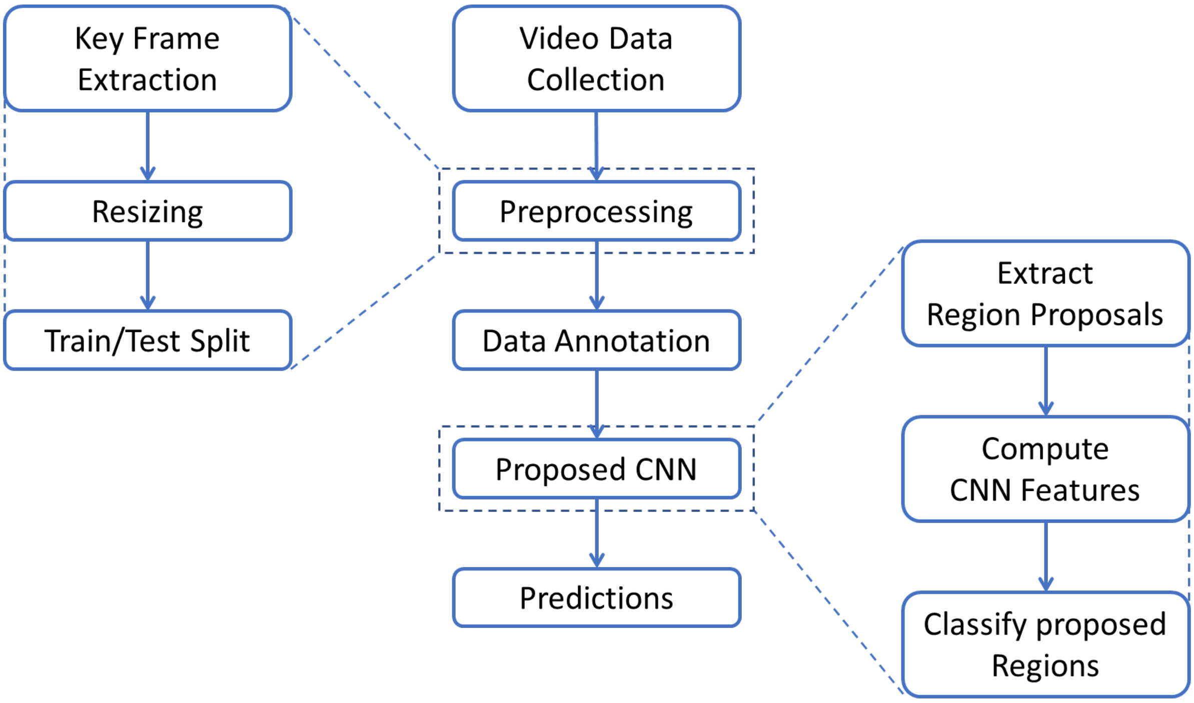

Figure 2: Flow Diagram

3.6 Model Architecture

The R-CNN—Region-based Convolutional Neural Networks—is a family of CNNs based on the idea of dividing an image into regions and then detecting objects in them instead of the whole image. There are multiple types and flavors of the RCNNs, but the inherent concept remains the same and consists of three main modules.

The first module in the vanilla RCNN proposes 2,000 regions from the input image using the Selective Search Algorithm. After the regions are extracted and resized, the proceeding module extracts features from each region in the form of a vector having size 4096 × 1. This feature vector is then passed on to the classifier layer which in this case is a Support Vector Machine—SVM—pre-trained on publicly available image dataset. The SVM then classifies the region into one of the pre-defined object classes or as background in case it does not detect any object in the given region.

The Faster-RCNN which is used in this research, differs considerably from the vanilla RCNN in that it changes the architecture into a 2-module implementation. First being the Region Proposal Network—RPN and second being the classifier network. The performance of the model is improved by the fact that the most computationally expensive task of calculating the convolution feature maps is used by both of these networks.

Faster-RCNN uses a RPN instead of the selective search algorithm, this is a sizeable improvement as now a traditional neural network is used to generate proposals with various sizes and aspect ratios. As the feature maps used by the RPN are same as those used by the detection network, the RPN deduces an objectness score for each region. This basically tells the probability of an object being present in the region. The Region of Interest—ROI pooling layer used already extracted feature maps to then pool the proposed regions and classification is performed on only the regions with a high enough probability of having an object. This further improves the detection speed as only a small part of all the regions are passed on the classification layer.

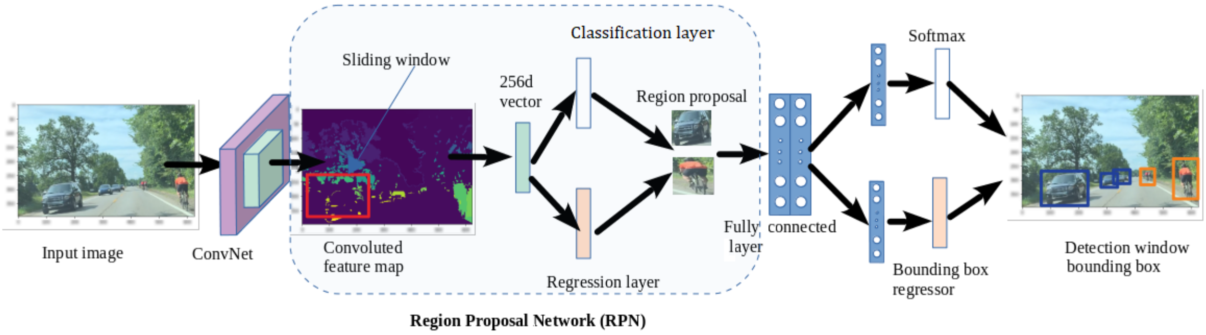

Figure 3: Model Architecture [37]

Faster-RCNNs used a backbone CNN which can be changed depending on application and requirement. Some of the most commonly used backbone networks include ResNet-50/101, VGG-16/19, RetinaNet, and Xception.

CHAPTER 4: RESULTS & DISCUSSION

4.1 Performance Metrics

It is pertinent to mention discuss the performance metric being used to characterize the predictions of the model. The primary metrics are the True Positive—TP, False Positive—FP, True Negative—TN, and False Negative—FN. These are further explained in the table 3 below:

Table 3: Performance Metrics

True Positive	It is when a model **makes** a prediction and correctly identifies the object
False Positive	It is when a model **makes** a prediction even though no object was present
True Negative	It is when a model **does not make** a prediction when there is no object
False Negative	It is when a model **does not make** a prediction even though an object was present

Another aspect which needs to be considered when an object detection model is being used is Intersection over Union—IoU. IoU is a measure of how much the predicted bounding box overlaps the original—or ground truth—bounding box in the case of a prediction being made, i.e., it is a ratio of the area of overlap and the total area covered by the original and predicted bounding boxes as given by the equation below

IoU = Intersection of bounding boxes' areas/Union of bounding boxes' areas

This metric is used along with a threshold to classify bounding boxes in accordance with one of the primary metrics. So as, if a bounding box is below the required threshold, it is classified as false positive because it made a prediction, but that prediction did not have enough quality in it to be called a correct prediction or a true positive. The IoU threshold can be varied depending on various situations and applications as well the size of the object under observation, but the default or generally accepted value is set at 0.5.

Furthermore, the primary metrics combine to form secondary metrics which are Precision and Recall and are given by the equations below:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Precision is a measure of accuracy of the model’s predictions, i.e., the number—or percentage—of correct predictions made by the model with respect to the total number of predictions made. While recall is the measure of how well the model is predicting the presence of objects, i.e., the number—or percentage—of objects detected with respect to all the objects present.

This is then succeeded by the tertiary metric called Average Precision which is generally defined as area under the precision-recall curve. This can be calculated by simple integration as per the first equation given below. The main metric being used to characterize the findings of this research is the mean Average Precision (mAP). mAP is the cumulative mean of the Average Precision across all the classes of object being predicted and is given by the second equation below.

$$p_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r})$$

$$\boldsymbol{mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k}$$
$$\boldsymbol{AP_k = \; the\ AP\ of\ class\ k}$$
$$\boldsymbol{n = \; the\ number\ of\ classes}$$

4.2 Results

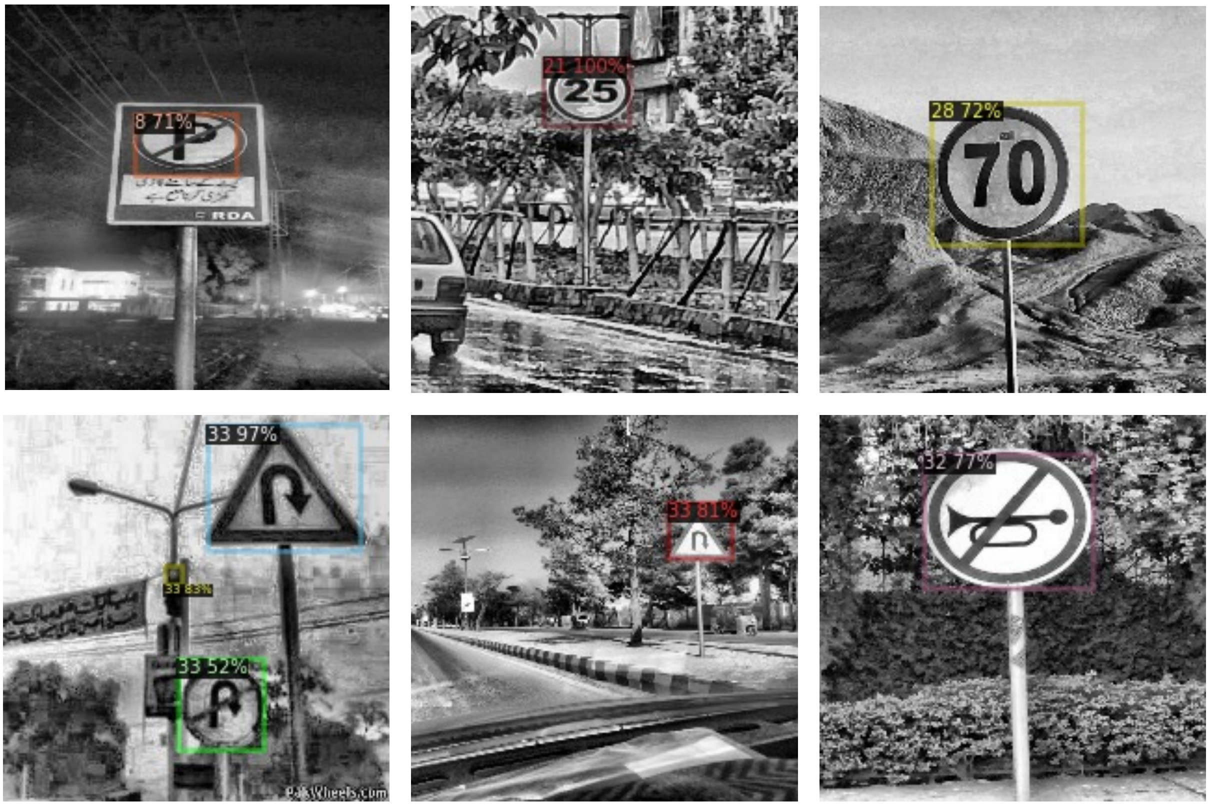

Figure 4. Detection Results

Traffic sign images were trained on multiple state-of-the-Art networks and model architecture of the Faster-RCNN family along with several different backbone networks. All the selected networks have shown remarkable results and groundbreaking mAP numbers on the COCO Dataset over the year. As we can see in the table below the Faster-RCNN model architecture with ResNet – 101 and Feature Pyramid Network as the backbone network produces the highest values for all the relevant benchmarks. Here the "AP | 50" means that the IoU value is more than 50%, "AP | s / m / l" means average precision of small, medium or large objects.

Table 4: Performance Metrics for the trained model

Model	mAP	AP | 50	AP | 75	AP | s	AP | m	AP | l
Faster RCNN (R50-FPN)	70.268	98.363	84.603	53.830	72.578	84.555
ResNet – 50 (Dilated Convolutions)	73.207	96.694	87.366	49.594	77.036	**85.514**
Xception - 101	72.289	72.289	72.289	72.289	72.289	72.289

Faster RCNN (R101-FPN) | train	**75.636**	**99.087**	**92.617**	**65.667**	**77.636**	84.649
Faster RCNN (R50-FPN) | val	49.699	64.612	59.148	46.945	40.180	69.612
Faster RCNN (R50-FPN) | test	43.453	58.242	52.686	15.00	44.763	38.119

4.2.1 Graphical Results

The graphical results of several other common and useful metric for the best performing model are given below.

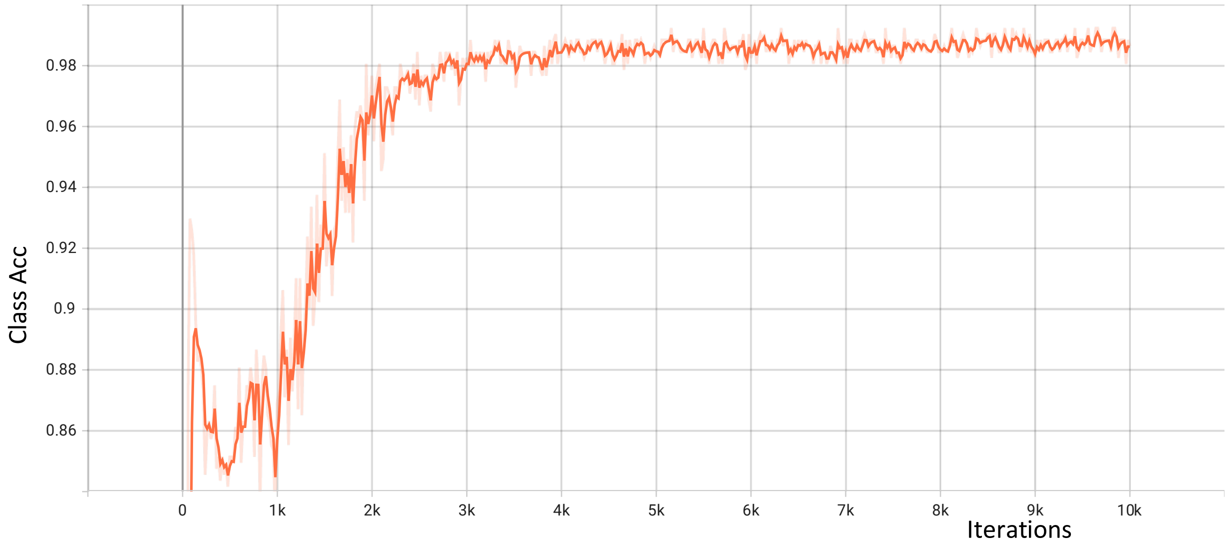

Figure 5: Average Class Accuracy for Faster RCNN (R101-FPN)

As visible from the graph above the class accuracy flutters in the initial 1000 iteration after which it starts to improve and ultimately flattens out after about 5000 iterations with a final accuracy score of 98.57%.

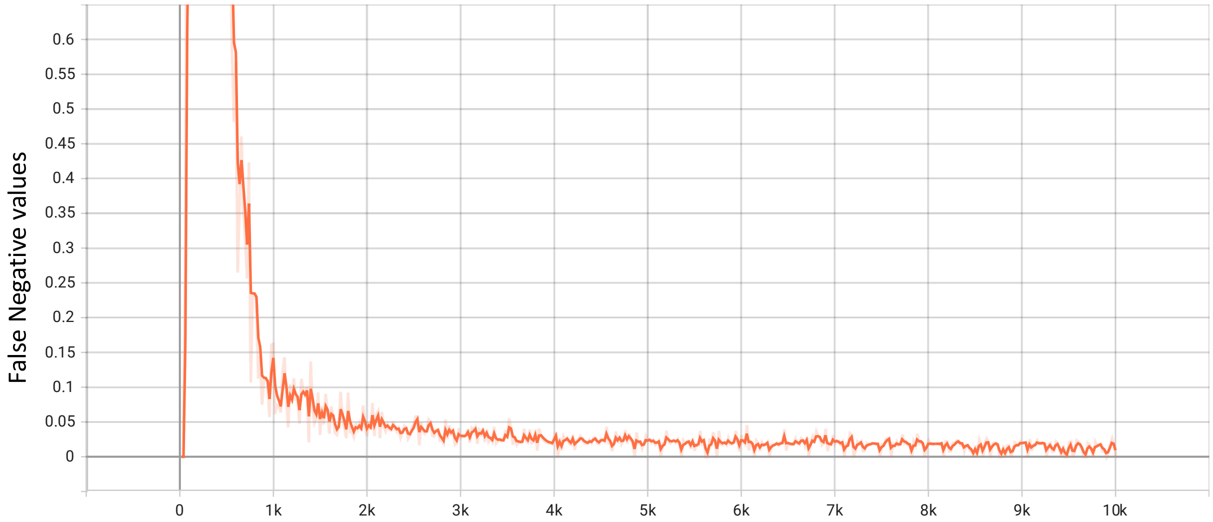

Figure 6: False Negative values for Faster RCNN (R101-FPN)

This is also a very important metric as explained in detail above, the final value for the false negative samples should be as low as possible for the model to work efficiently. This exactly the situation in the case of our best performing model, the graph show a steady decrease as the model extract better features for detection.

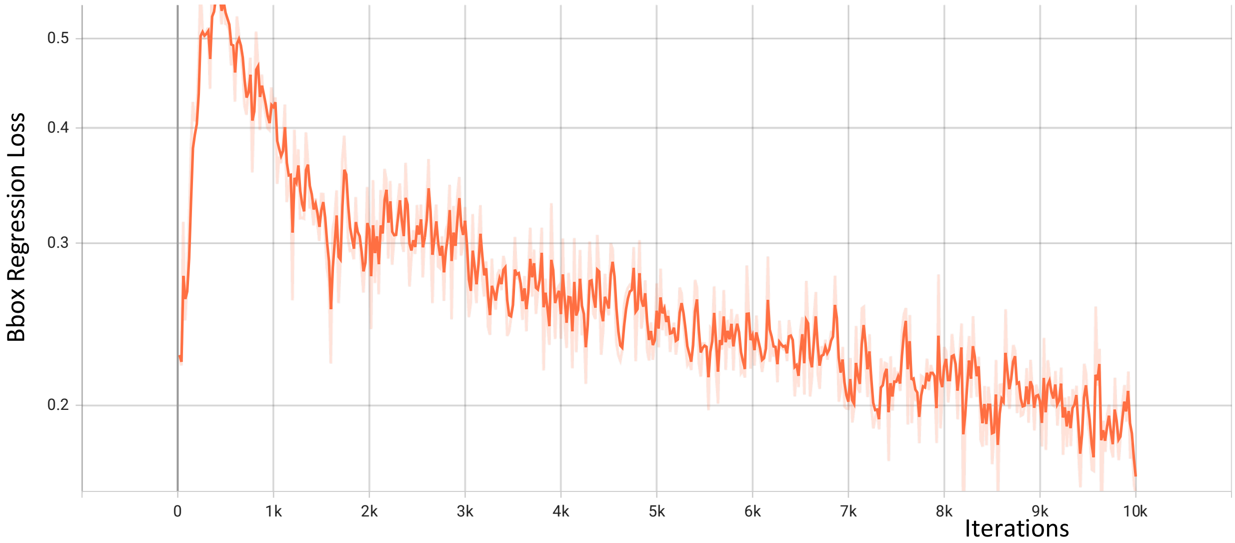

Figure 7: Bounding Box Regression Loss for Faster RCNN (R101-FPN)

Bounding boxes are the rectangles drawn around the detected object, as mentioned in the model architecture the bounding box coordinates are regressed in a branch of the network. The loss of this regression is a strong indicator of how well these boxes are ‘bounding’ the objects. The values of regression loss decrease continuously to a very negligible value, showing the improvements as the training progresses.

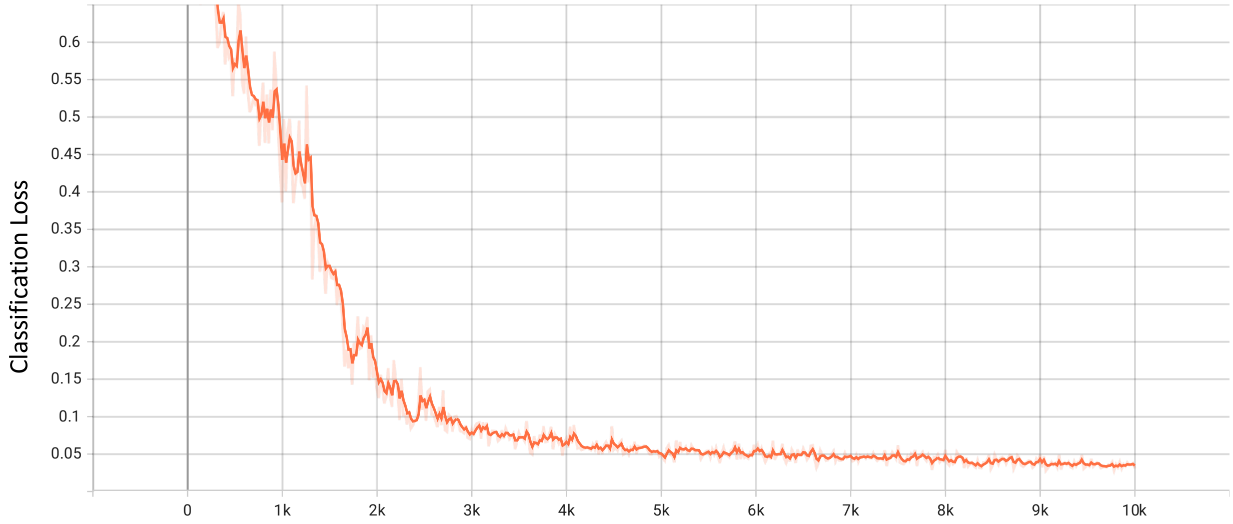

Figure 8: Classification Loss for Faster RCNN (R101-FPN)

The Faster-RCNN family divides into 2 branches after the final fully connected layer, one of these branches is the classification branch which essentially classifies the detected objects into one of the predefined classes. This part of the architecture uses softmax and the loss generally trends downwards quite early and stays more or less steady after about 4000 iterations or so.

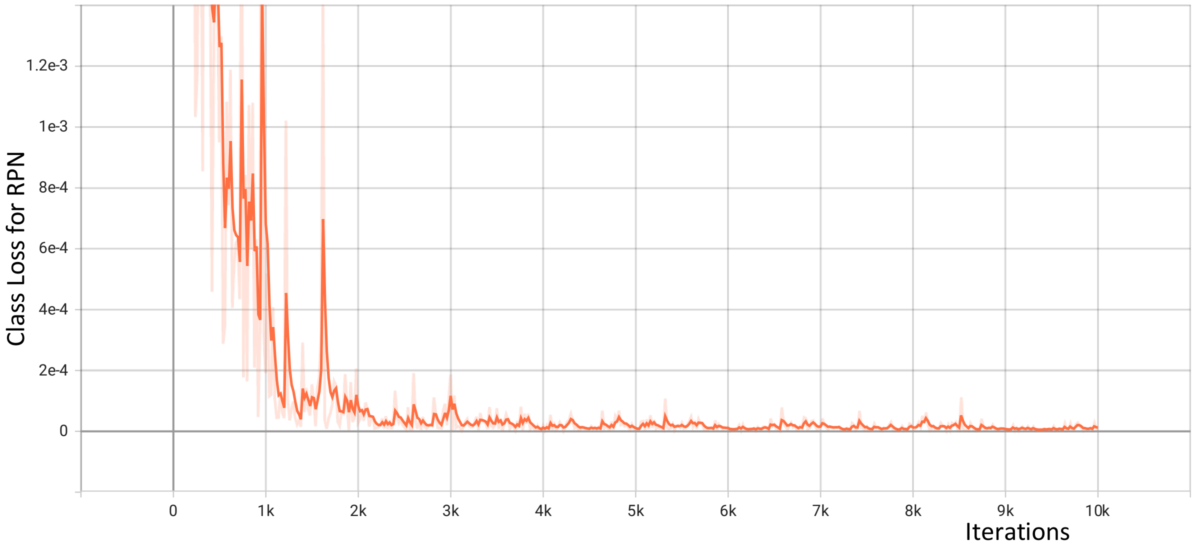

Figure 9: Class Loss of Region Proposal Network (RPN) for Faster RCNN (R101-FPN)

As explained in the model architecture subsection above Region Proposal Network is an integral part of the working of Faster-RCNN we have used here. It is a standalone network and given in fig 5 above is the expected downwards trend of the loss of the network as iterations increase.

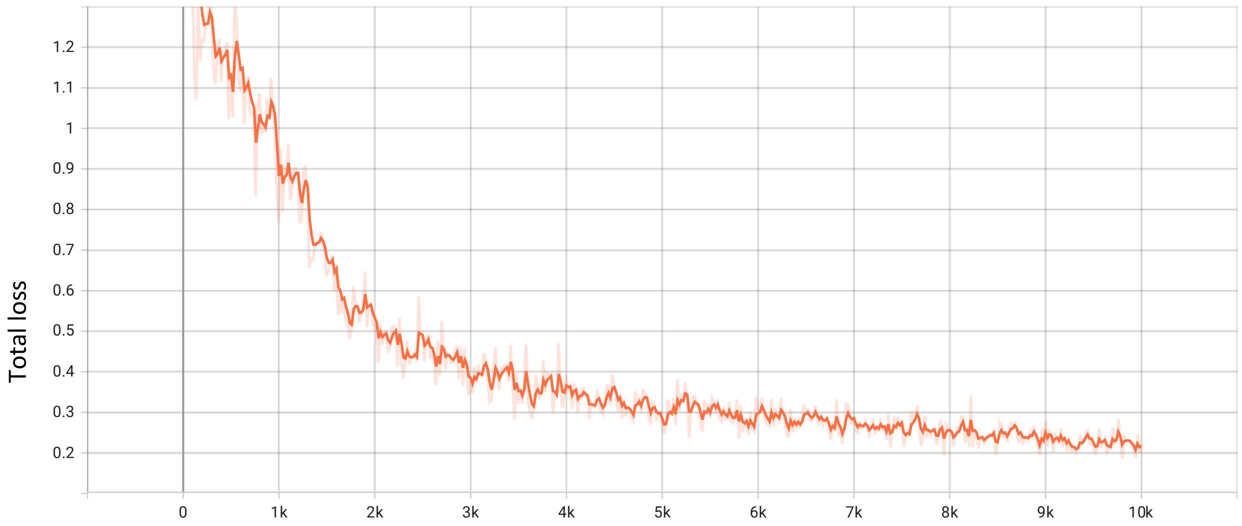

Figure 10: Total Loss for Faster RCNN (R101-FPN)

Model loss or error is another major identifier of a model's performance, it is in simple terms the cumulative sum of the difference between the actual and predicted values. The graph above shows a steady decrease as the iterations increase showing the improvement in model's performance. This is the total loss which includes both the bounding box regression loss as well as the classification loss discussed above.

4.2.1.1 Faster RCNN with Dilated Convolutions

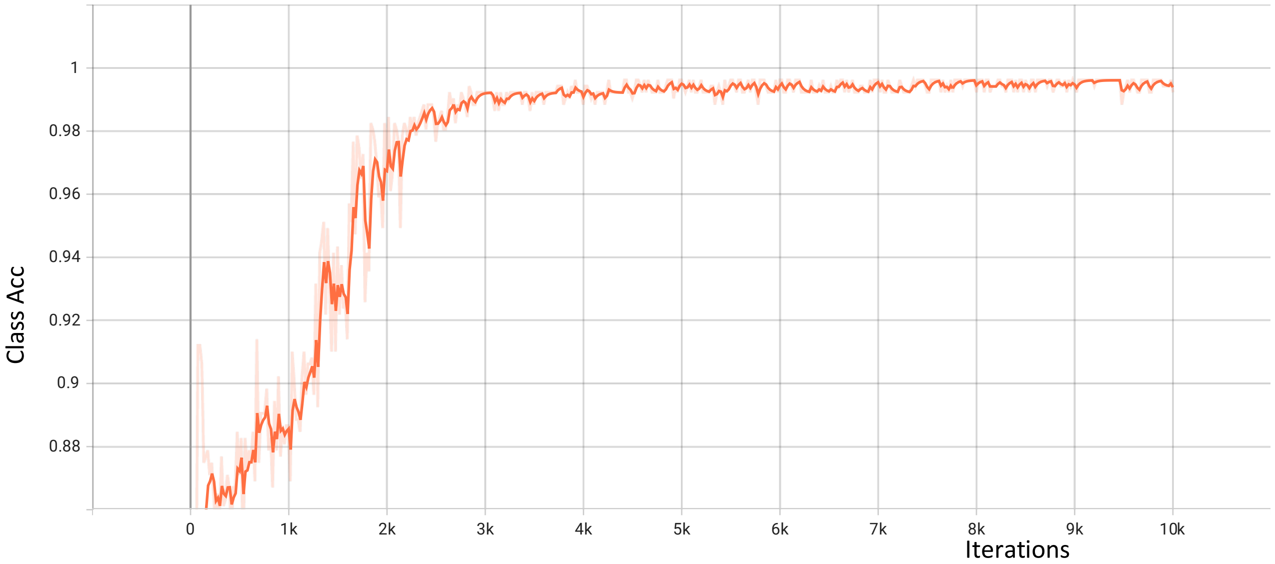

Figure 11: Average Class Accuracy for Faster RCNN (with Dilated Convolutions)

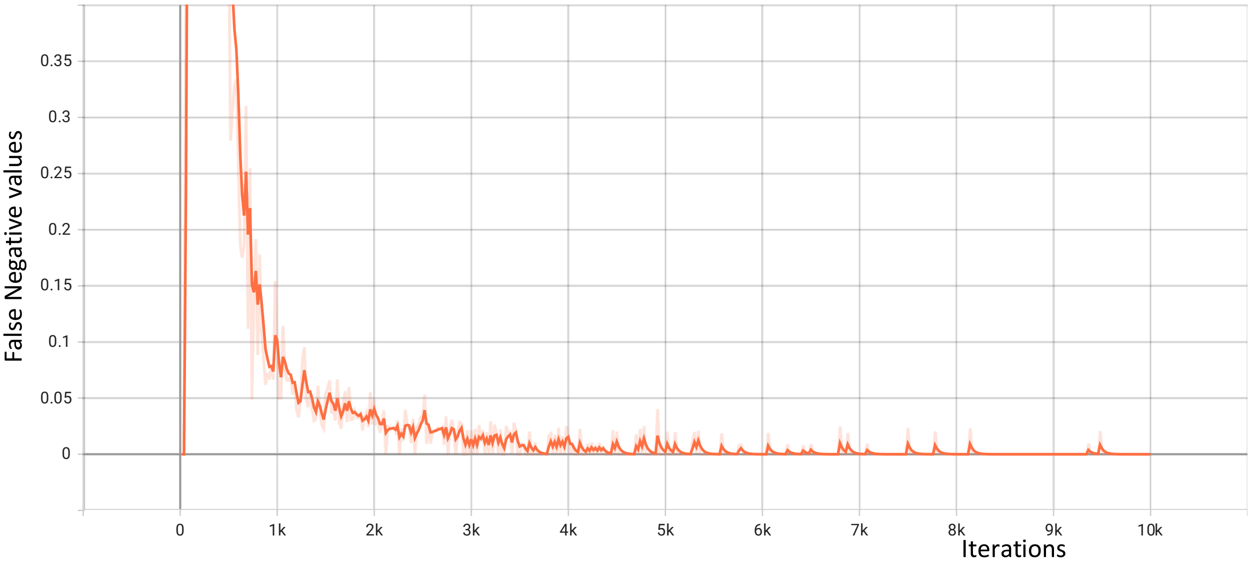

Figure 12: False Negative values for Faster RCNN (with Dilated Convolutions)

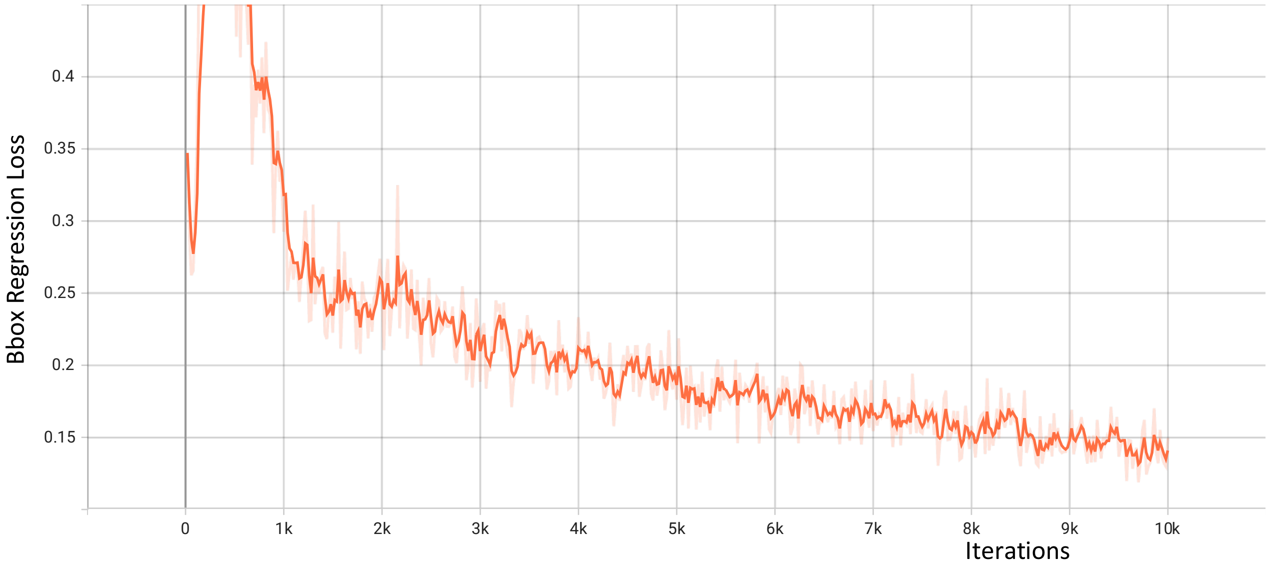

Figure 13: Bounding Box Regression Loss for Faster RCNN (with Dilated Convolutions)

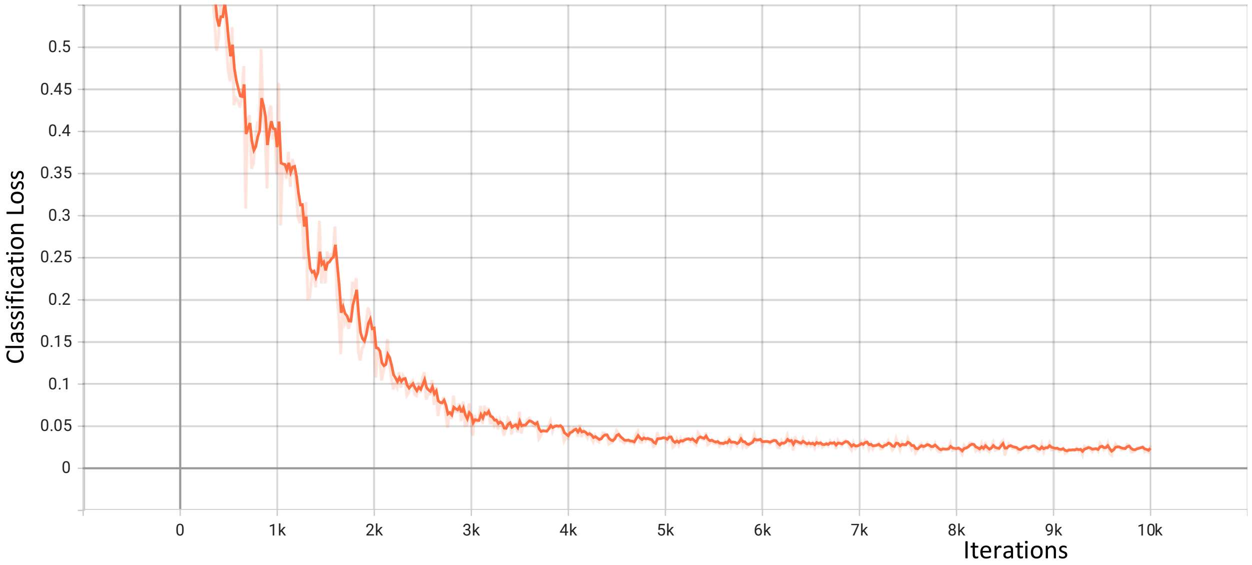

Figure 14: Classification Loss for Faster RCNN (with Dilated Convolutions)

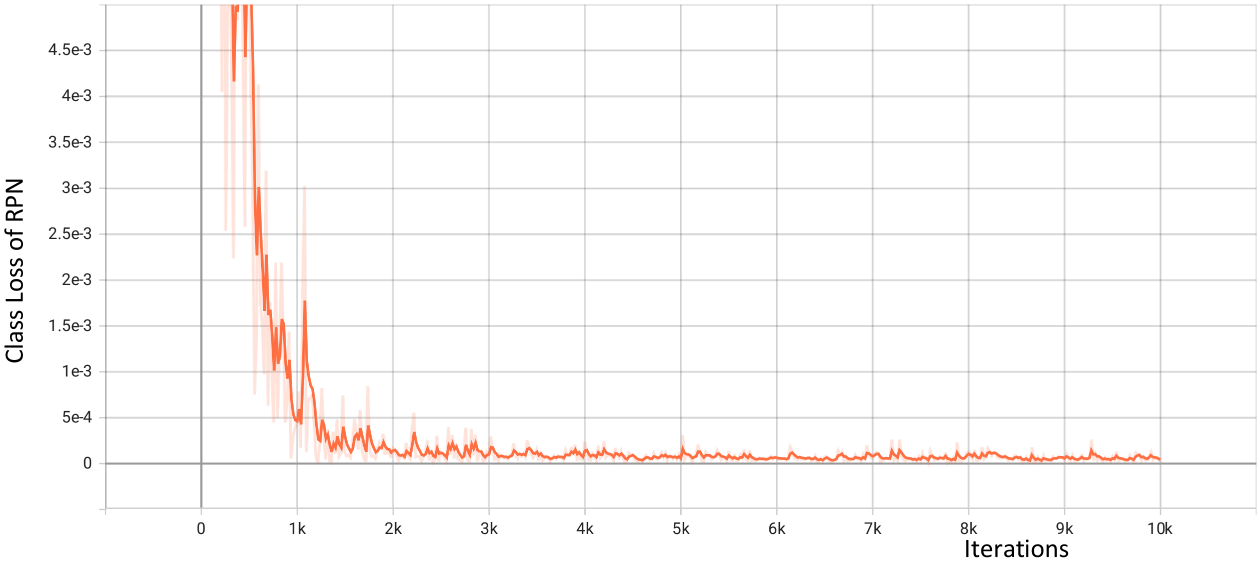

Figure 15: Class Loss of RPN for Faster RCNN (with Dilated Convolutions)

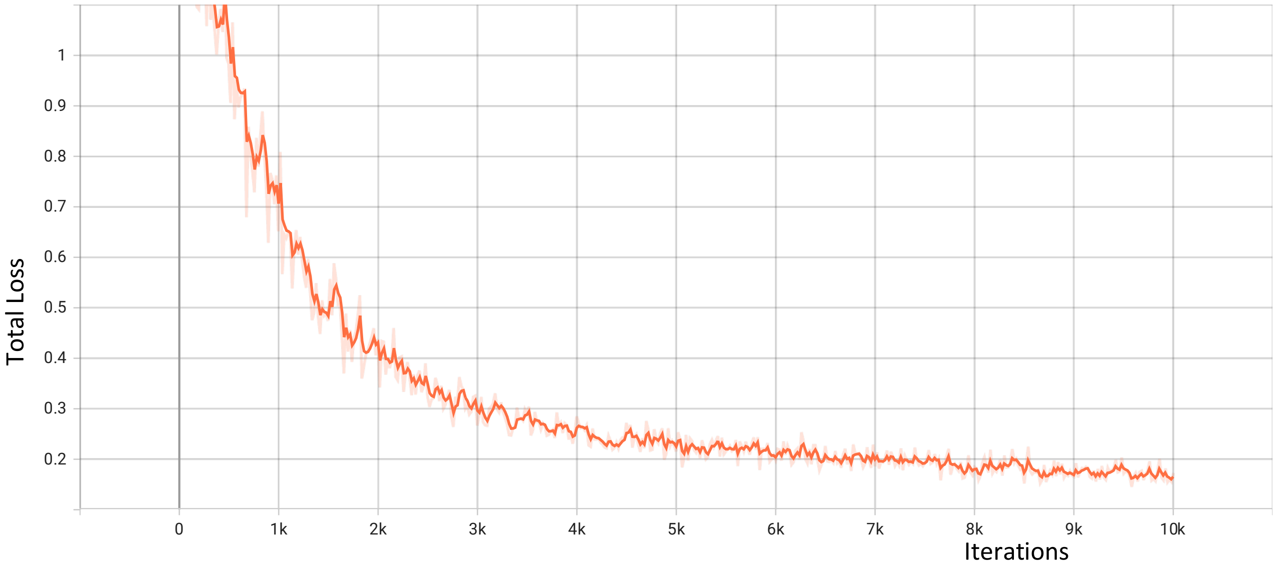

Figure 16: Total Loss for Faster RCNN (with Dilated Convolutions)

4.2.1.2 Faster RCNN (R-50 FPN)

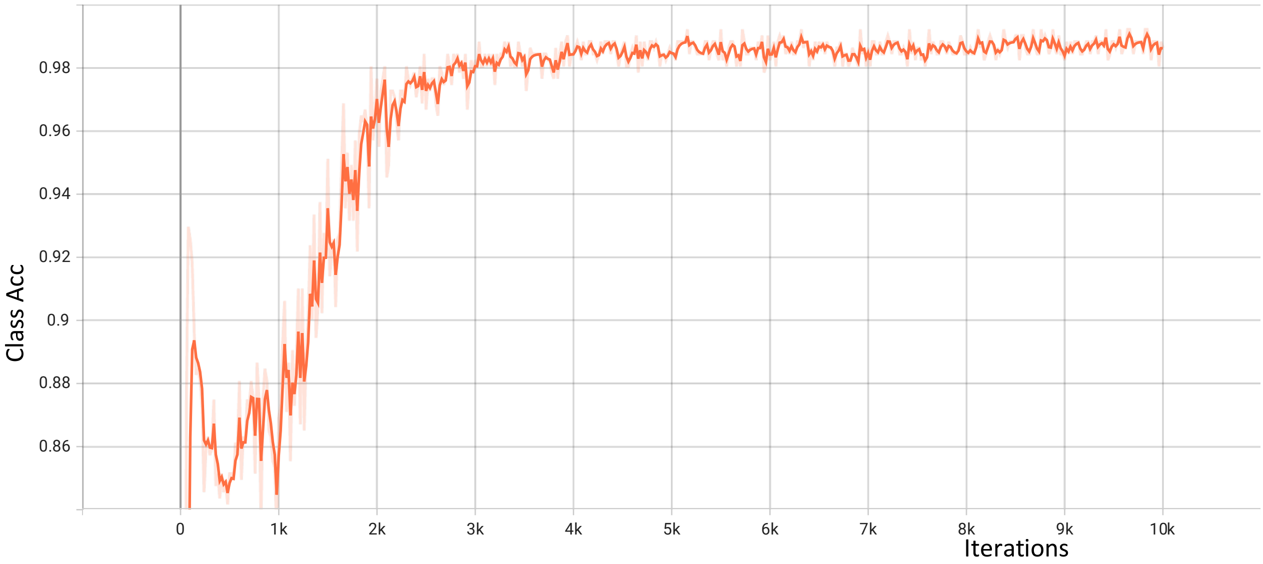

Figure 17: Average Class Accuracy for Faster RCNN (R-50 FPN)

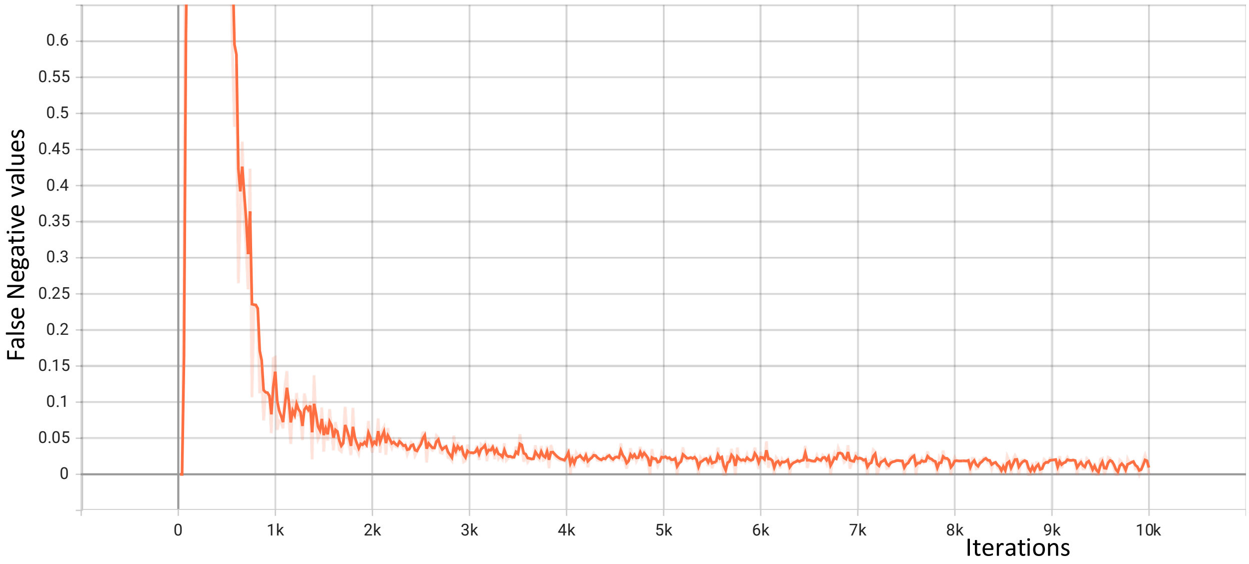

Figure 18: False Negative values for Faster RCNN (R-50 FPN)

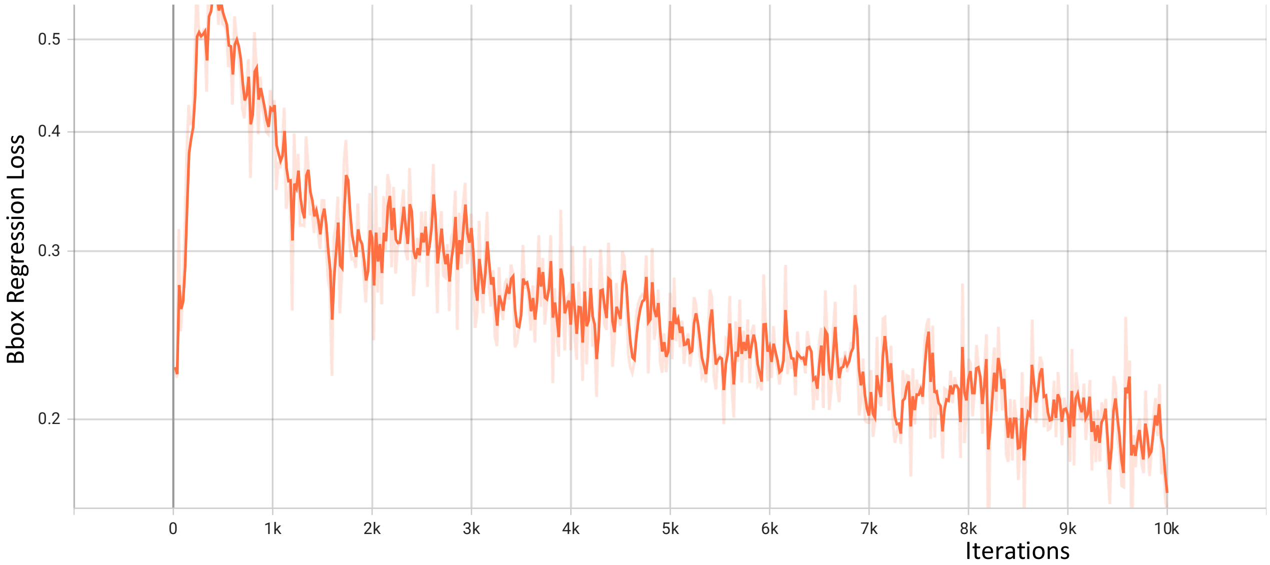

Figure 19: Bounding Box Regression Loss for Faster RCNN (R-50 FPN)

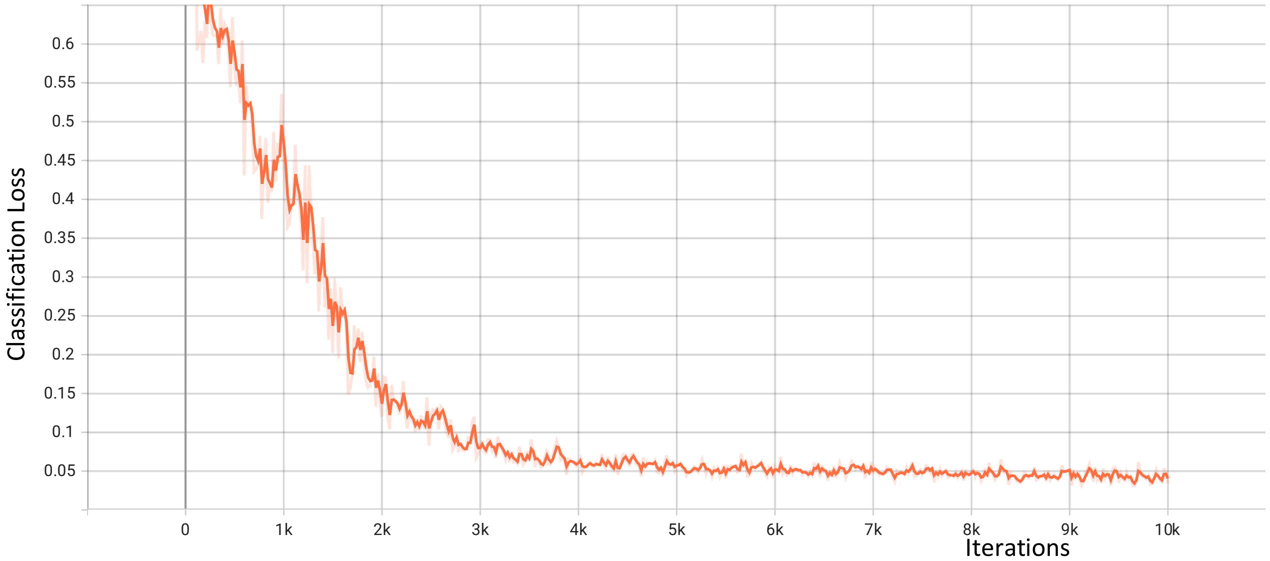

Figure 20: Classification Loss for Faster RCNN (R-50 FPN)

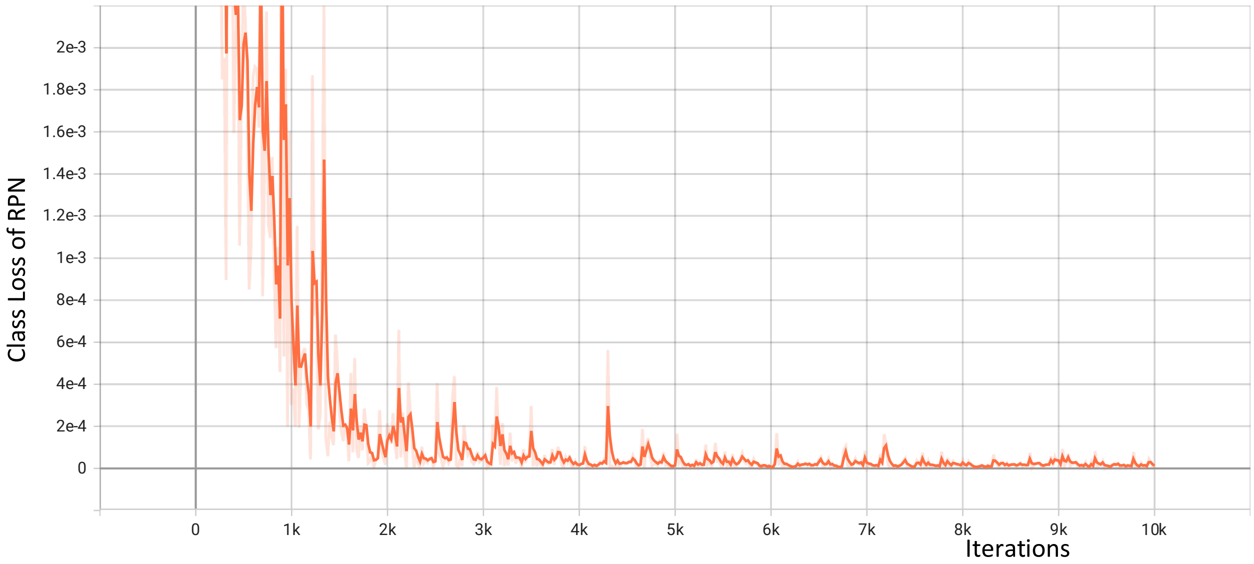

Figure 21: Class Loss of RPN for Faster RCNN (R-50 FPN)

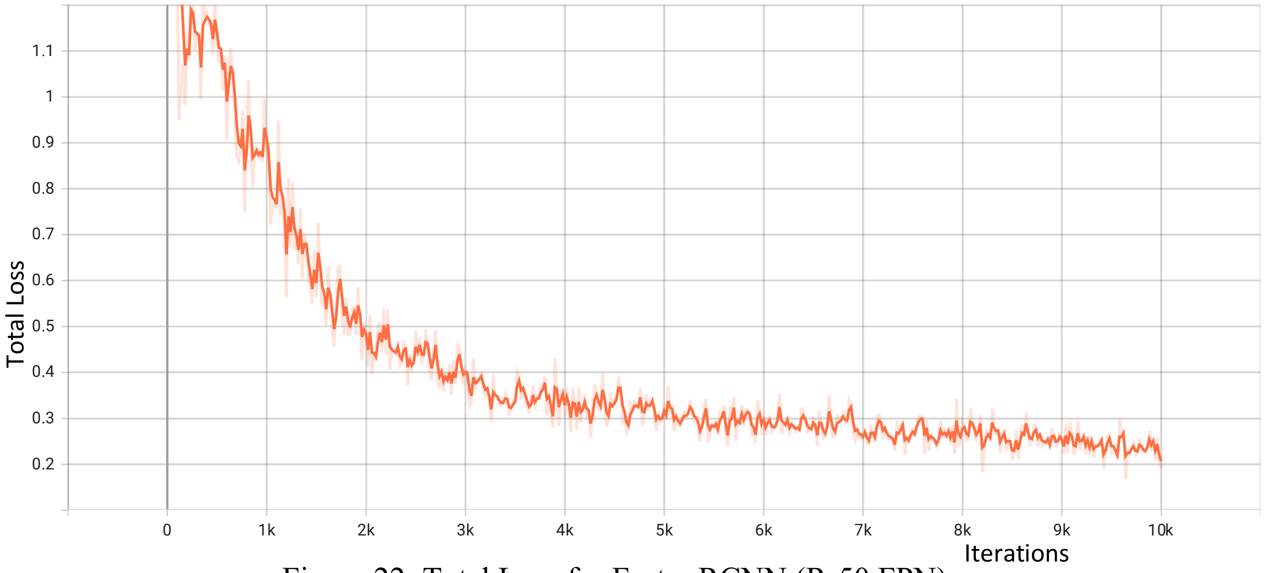

Figure 22: Total Loss for Faster RCNN (R-50 FPN)

4.2.1.3 Faster RCNN (Xception-101 FPN)

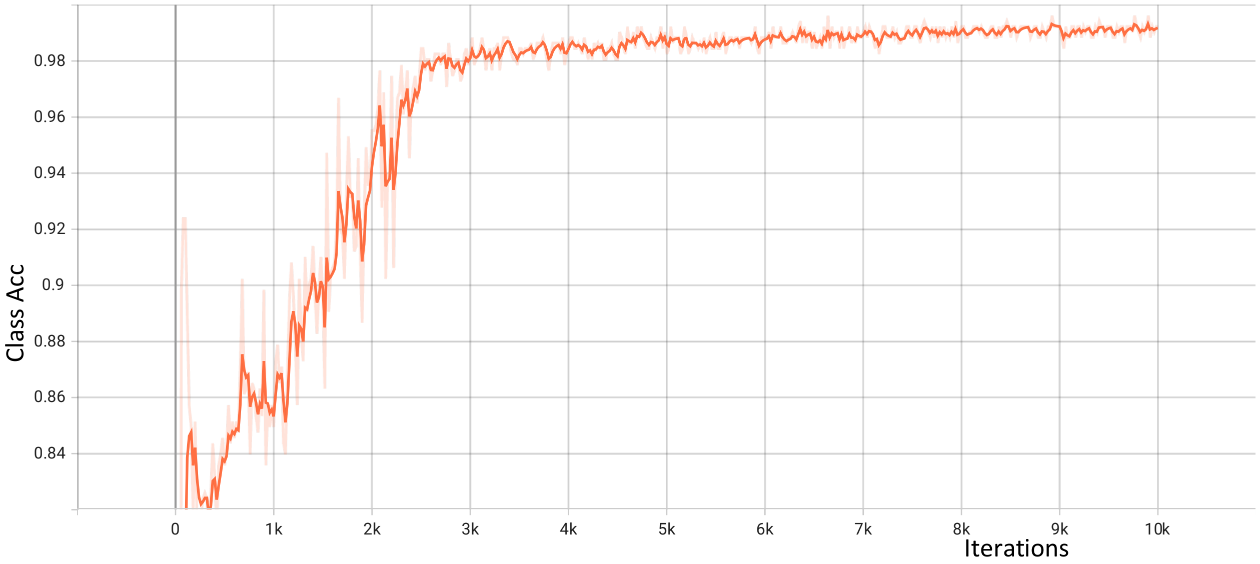

Figure 23: Average Class Accuracy for Faster RCNN (Xception-101 FPN)

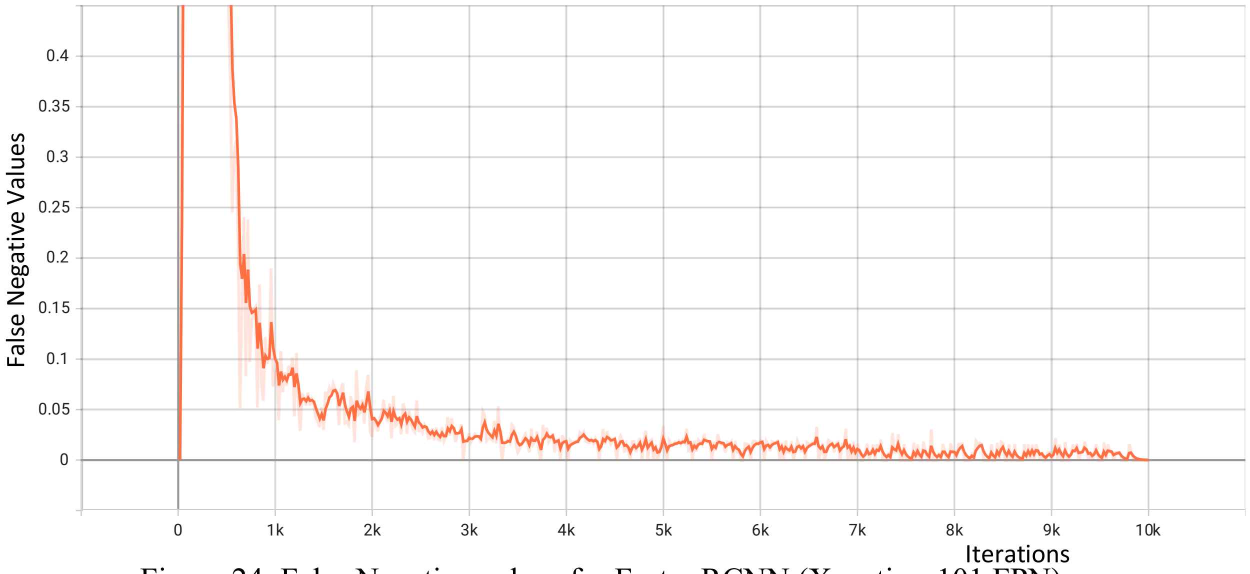

Figure 24: False Negative values for Faster RCNN (Xception-101 FPN)

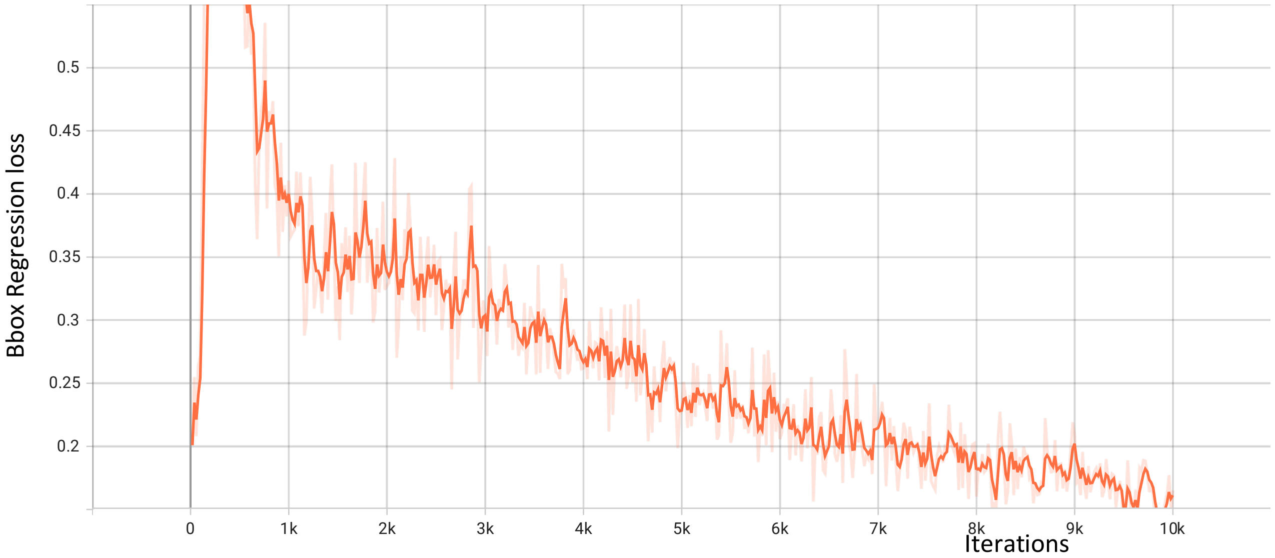

Figure 25: Bounding Box Regression Loss for Faster RCNN (Xception-101 FPN)

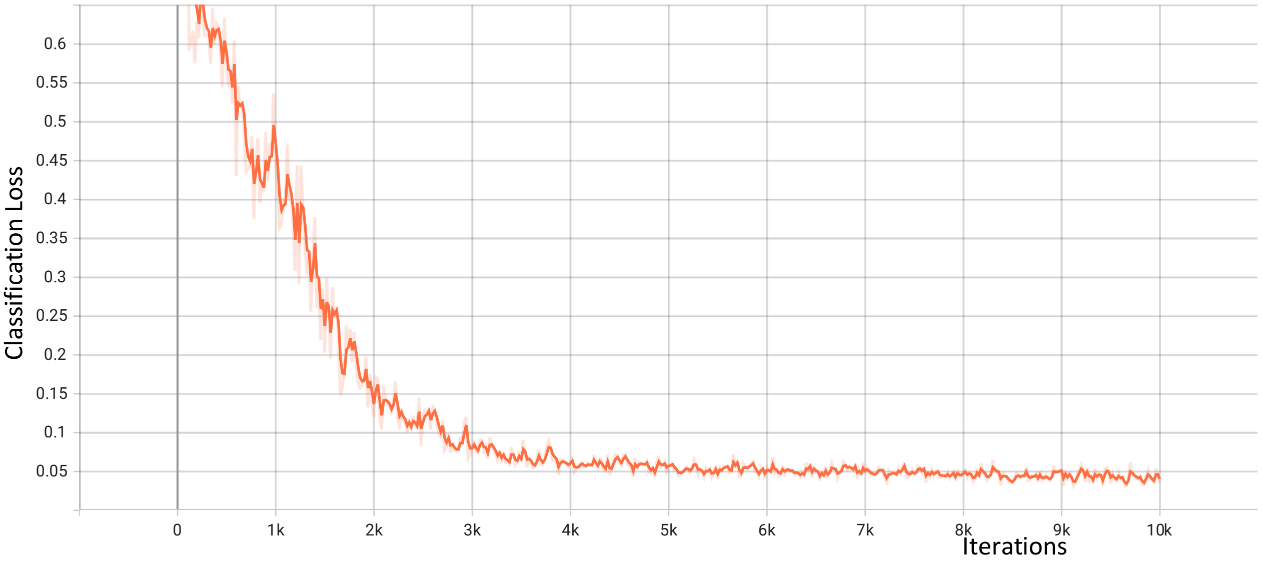

Figure 26: Classification Loss for Faster RCNN (Xception-101 FPN)

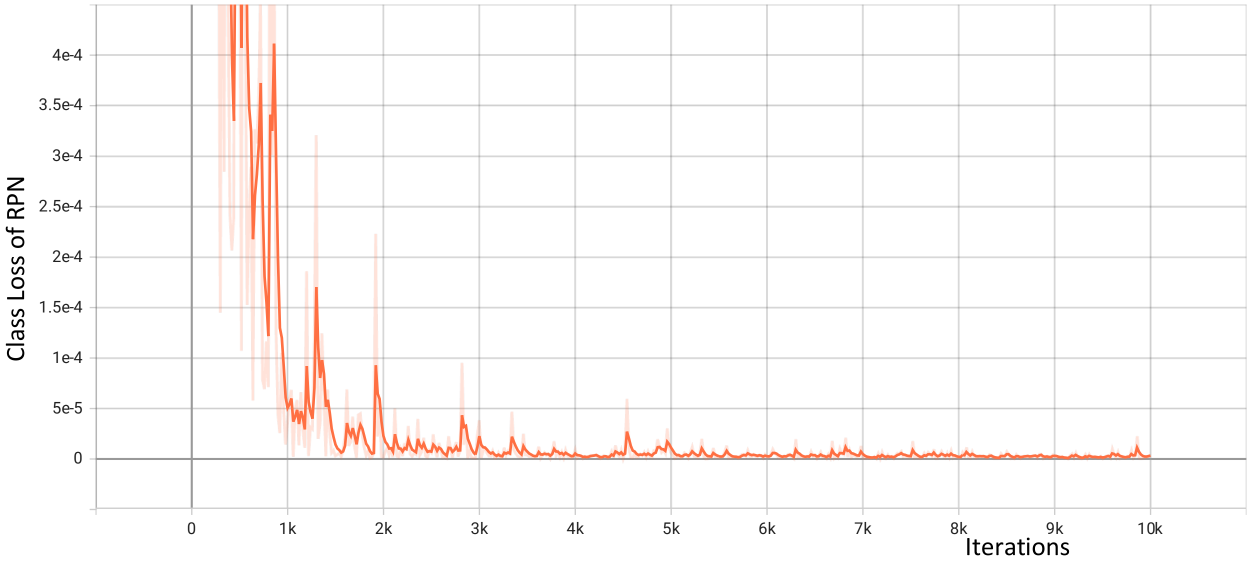

Figure 27: Class Loss of RPN for Faster RCNN (Xception-101 FPN)

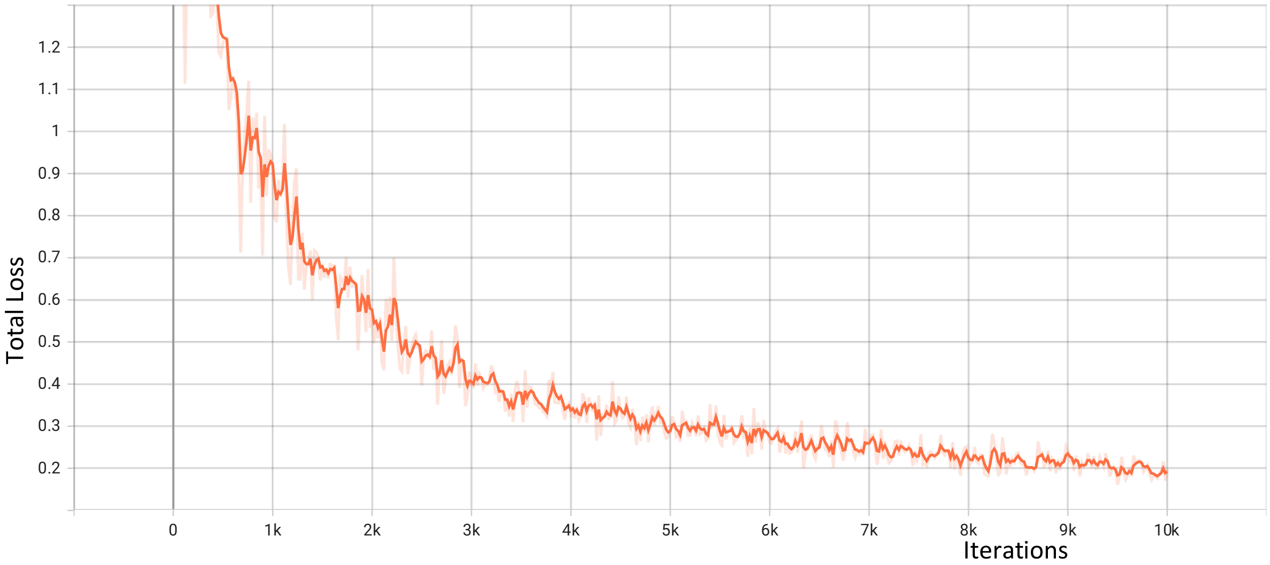

Figure 28: Total Loss for Faster RCNN (Xception-101 FPN)

4.3 Discussion

The novelty of this research lies in the collection of vast amounts of data pertaining to Pakistani traffic signs and types. This data as mentioned in detail above has been collected from several cities while having a diverse range of visual features and artifact. These include images with varying range of exposure, brightness, and occlusion. Furthermore, this dataset is annotated for traffic type and sign recognition and can be used by researchers to develop and improve their models for the roads of Pakistan.

Furthermore, this is pioneering research in the implementation of multiple object detection-based systems to recognize the Pakistani traffic type and signs. It is the first of its kind research in Pakistan which has a model trained on frames from video footage from the streets. It also scores impressively in all performance metrics used internationally to characterize related models.

CHAPTER 5: FUTURE WORK

This research was meant as am initial foray into enabling better traffic conditions in Pakistan. Even though the research resulted in favorable outcomes and performance metrics there is always room for improvement. The performance metrics can be improved by use of data augmentation to help account for class imbalance. More training iterations/epochs can be used in case of availability of better computational resources. Another case of improvement can be use of all frames instead of extracting key frames and using them at a higher resolution to improve results.

Object detection is a hot research topic hence recently published models can be used for higher performance scores. Masked RCNN and image segmentation models to can also be used to obtain pixel-wise detection but would require even more tedious annotation process but could result in better real-world results when deploying the model. The research could be compounded by development of specialized hardware to help turn the research project into a commercially viable product.

CHAPTER 6: CONCLUSION

This motivation behind conducting this research was laying a foundation for a dataset and a model which can be used by self-driving vehicles and existing vehicles to improve the traffic conditions in the country. During this research, a first of its kind dataset was collected from the roads of Pakistan and across various cities including, but not limited to, Islamabad, Quetta, Lahore, and Karachi. The dataset amounted to **5 hours, 42 minutes and 1 second** of video footage, and 109,463 images of keyframes. The footage was annotated using rectangular bounding boxes and **5 distinct classes** which were pedestrians, bikes, LTVs, HTVs, and traffic signs. The traffic sign class was then further divided into 35 subclasses.

Consequently, a deep learning model was trained for the traffic signs. It was Faster-RCNN architecture with ResNet-101 and Feature Pyramid Network as the backbone. It was trained and tested to detect and classify traffic signs at a mAP of **75.636%** and an overall class accuracy of more than **98%**.

REFERENCES

[1] “jochem_todd_1995_2.pdf.” Accessed: Jul. 28, 2022. [Online]. Available: https://www.ri.cmu.edu/pub_files/pub2/jochem_todd_1995_2/jochem_todd_1995_2.pdf

[2] X. J. Zhu, “Semi-supervised learning literature survey,” 2005.

[3] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, “Text classification from labeled and unlabeled documents using EM,” *Mach. Learn.*, vol. 39, no. 2, pp. 103–134, 2000.

[4] “Riding shotgun in Tesla’s fastest car ever,” *Engadget*. https://www.engadget.com/2014-10-09-tesla-d-awd-driver-assist.html (accessed Jul. 28, 2022).

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] X. Yin, J. Han, J. Yang, and P. S. Yu, “Crossmine: Efficient classification across multiple database relations,” in *Constraint-Based mining and inductive databases*, Springer, 2006, pp. 172–195.

[7] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “The German traffic sign recognition benchmark: a multi-class classification competition,” in *The 2011 international joint conference on neural networks*, 2011, pp. 1453–1460.

[8] “Alarming figures of traffic accidents need attention.” https://www.thenews.com.pk/print/910436-alarming-figures-of-traffic-accidents-need-attention (accessed Jul. 29, 2022).

[9] V. K. Kukkala, J. Tunnell, S. Pasricha, and T. Bradley, “Advanced driver-assistance systems: A path toward autonomous vehicles,” *IEEE Consum. Electron. Mag.*, vol. 7, no. 5, pp. 18–25, 2018.

[10] M. Galvani, “History and future of driver assistance,” *IEEE Instrum. Meas. Mag.*, vol. 22, no. 1, pp. 11–16, 2019.

[11] P. Sermanet and Y. LeCun, “Traffic sign recognition with multi-scale convolutional networks,” in *The 2011 international joint conference on neural networks*, 2011,

pp. 2809–2813.

[12] Z. Malik and I. Siddiqi, “Detection and recognition of traffic signs from road scene images,” in *2014 12th International conference on frontiers of information technology*, 2014, pp. 330–335.

[13] K. T. Islam, R. G. Raj, and G. Mujtaba, “Recognition of traffic sign based on bag-of-words and artificial neural network,” *Symmetry*, vol. 9, no. 8, p. 138, 2017.

[14] R. Malik, J. Khurshid, and S. N. Ahmad, “Road sign detection and recognition using colour segmentation, shape analysis and template matching,” in *2007 international conference on machine learning and cybernetics*, 2007, vol. 6, pp. 3556–3560.

[15] J. Kim, S. Lee, T.-H. Oh, and I. S. Kweon, “Co-domain embedding using deep quadruplet networks for unseen traffic sign recognition,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, vol. 32, no. 1.

[16] H.-Y. Lin, J.-M. Dai, L.-T. Wu, and L.-Q. Chen, “A vision-based driver assistance system with forward collision and overtaking detection,” *Sensors*, vol. 20, no. 18, p. 5139, 2020.

[17] D. Fernández Llorca, A. Hernández Martínez, and I. García Daza, “Vision-based vehicle speed estimation: A survey,” *IET Intell. Transp. Syst.*, vol. 15, no. 8, pp. 987–1005, 2021.

[18] T. Joachims, “Transductive inference for text classification using support vector machines,” in *Icml*, 1999, vol. 99, pp. 200–209.

[19] X. Zhu and X. Wu, “Class noise handling for effective cost-sensitive learning by cost-guided iterative classification filtering,” *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1435–1440, 2006.

[20] P. D. Turney, “Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm,” *J. Artif. Intell. Res.*, vol. 2, pp. 369–409, 1994.

[21] “CUDA C++ Programming Guide,” p. 379.

[22] J. Borenstein and Y. Koren, “The vector field histogram-fast obstacle avoidance

for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, 1991.

[23] J. A. Stark, "Adaptive image contrast enhancement using generalizations of histogram equalization," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 889–896, 2000.

[24] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1498–1506, 2012.

[25] "LISA Traffic Light Dataset." https://www.kaggle.com/datasets/mbornoe/lisa-traffic-light-dataset (accessed Jul. 28, 2022).

[26] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 633–647, Apr. 2014, doi: 10.1007/s00138-011-0391-3.

[27] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2110–2118.

[28] C. Lin, L. Li, W. Luo, K. C. Wang, and J. Guo, "Transfer learning based traffic sign recognition using inception-v3 model," *Period. Polytech. Transp. Eng.*, vol. 47, no. 3, pp. 242–250, 2019.

[29] J. Sokipriala and S. Orike, "Traffic sign classification comparison between various convolution neural network models," *Int. J. Sci. Eng. Res.*, vol. 12, no. 7, pp. 165–171, 2021.

[30] A. Wahab, A. Khan, I. Rabbi, K. Khan, and N. Gul, "Audio Augmentation for Traffic Signs: A Case Study of Pakistani Traffic Signs," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 11, 2018.

[31] Z. Nadeem, A. Samad, Z. Abbas, and J. Massod, "A Transfer Learning based approach for Pakistani Traffic-sign Recognition; using ConvNets," in *2018 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, 2018, pp. 1–6.

[32] A. A. Sikander and H. Ali, "Image Classification using CNN for Traffic Signs in Pakistan," *ArXiv Prepr. ArXiv210210130*, 2021.

[33] L. Deng and D. Yu, “Deep learning: methods and applications,” *Found. Trends® Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2014.

[34] R. Dechter, “Learning while searching in constraint-satisfaction problems,” 1986.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.

[36] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European conference on computer vision*, 2014, pp. 818–833.

[37] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[38] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *ArXiv Prepr. ArXiv14091556*, 2014.

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

[40] “Medium,” *Medium*. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional- (accessed Jul. 29, 2022).

[41] J. Murphy, “An overview of convolutional neural network architectures for deep learning,” *Microway Inc*, pp. 1–22, 2016.