Modelling false positive reduction in maritime object detection

A Dissertation

By

Nosiphiwo Nkele

Submitted in fulfillment of the requirements of the Degree of Master of Science in Computer Science

Faculty of Science & Agriculture

University of Fort Hare

Together in Excellence

Supervisor:

Prof K. Sibanda

[June 2019]

Abstract

Target detection has become a very significant research area in computer vision with its applications in military, maritime surveillance, and defense and security. Maritime target detection during critical sea conditions produces a number of false positives when using the existing algorithms due to sea waves, dynamic nature of the ocean, camera motion, sea glint, sensor noise, sea spray, swell and the presence of birds. The main question that has been addressed in this research is how can object detection be improved in maritime environment by reducing false positives and promoting detection rate. Most of Previous work on object detection still fails to address the problem of false positives and false negatives due to background clutter. Most of the researchers tried to reduce false positives by applying filters but filtering degrades the quality of an image leading to more false alarms during detection. As much as radar technology has previously been the most utilized method, it still fails to detect very small objects and it may be applied in special circumstances. In trying to improve the implementation of target detection in maritime, empirical research method was proposed to answer questions about existing target detection algorithms and techniques used to reduce false positives in object detection. Visible images were retrained on a pre-trained Faster R-CNN with inception v2. The pre-trained model was retrained on five different sample data with increasing size, however for the last two samples the data was duplicated to increase size. For testing purposes 20 test images were utilized to evaluate all the models. The results of this study showed that the deep learning method used performed best in detecting maritime vessels and the increase of dataset improved detection performance and false positives were reduced. The duplication of images did not yield the best results; however, the results were promising for the first three models with increasing data.

Keywords

Object detection, Machine Learning, Neural Networks, Convolutional Neural networks, Computer Vision

Declaration

I Nosiphiwo Nkele Hereby confirm that “Modelling false positive reduction in maritime object detection” is my own work and has not been submitted for any requirement at University of Fort Hare and any other university. The dissertation does not contain any published work without a reference or acknowledgment.

Signature .... Date: June 2019

Acknowledgements

First and foremost, I would like to convey my genuine gratitude to the almighty God through Jesus Christ our Lord, he has given me strength, wisdom, inspiration, comfort and hope all my life and through the completion of this study. I am very grateful to my husband Gcinikaya Nkele for his support and for always encouraging me to work even if it was painful at times. I would like to acknowledge my supervisor Professor K. Sibanda for his guidance, constructive criticism and patience in this research. I am thankful to my parents-in-law Mr. M. and Mrs. N. Nkele for their support. A special thanks goes to my mother Mrs. N. Mbandzi for her love, teachings and support she gave me. I acknowledge the financial assistance which came from the Council of Scientific & Industrial Research (CSIR) and Armaments Corporation of South Africa (ARMSCOR). I would also like to thank Mr. M.S. Scott for his help in finding financial assistance from the above departments.

Table of Contents

Table of figures

Tables

Acronyms

CNN	Convolutional Neural Network
R-CNN	Region-based Convolutional Neural Network
HDR	High Dynamic Range
BBS	Background Subtraction Algorithm
MGM	Multiple Gaussian Mixture
SGM	Single Gaussian Model
TRDR	Tracker Detection Rate
FAR	False Alarm Rate
CSTFICA	Complex-Valued Spatiotemporal FastICA
ARPA	Automatic Radar Plotting Aid
ABM	Area-Based Matching
FBM	Feature-Based Matching
TBM	Template Based Matching
DBM	Deep Boltzmann Machines
DBN	Deep Belief Networks
SDAE	Stacked Denoising Auto-Encoder
DCAE	Deep Convolutional Auto-Encoder

RANSAC	Random Sample Consensus
DPM	Deformable Part Model
L-DBRF	Local Directional Background Removal Filter
NSF	Mean Subtraction Filter
SVM	Support Vector Machine
RPN	Region Proposal Network
R-FCN	Region-Based Fully Convolutional Network
YOLO	You Look Only Once
XML	Extensible Markup Language
CSV	Comma-Separated Values
FDR	False Discovery Rate
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative

1 Chapter one: Introduction

The dissertation addresses supervised target detection in a maritime environment. Target detection is a computer technology whereby objects of interest are detected in a specific environment using images and videos. In this study images captured from the maritime environment are utilized for boats and ships detection. This section provides an introduction to the work contained in this dissertation. It covers the research problem, research aim, research questions, research objectives, justification, expected results, research deliverables, research limitations and dissertation development.

1.1 Research problem

The problem undertaken in this work is boats and ships detection in a maritime environment. A consistent and fast algorithm is essential for detecting maritime vessels like boats and ships, to identify suspicious activities occurring in the maritime environment. It is challenging to uncover targets with a naked eye in a complex environment like maritime, therefore a target detection has a suitable outcome in this challenge. Due to critical conditions of the sea, target detection tends to produce a number of false positives. Sea waves, swell, sea-glint, birds, dynamic nature of the ocean, camera motion, sensor noise and sea spray have a negative effect on a detector leading to these false positives [1]. The horizon line caused by miscellaneous background and the boundary cloud scatter can produce a number of false detections [2]. False positives can also be caused by sun-glint which appear like small targets [2]. In trying to resolve the problem a machine learning algorithm is utilized in this research.

1.2 Research aim

The aim of this study is to improve the implementation of target detection in maritime environment.

1.3 Research Questions

1. Which methods are used for target detection in maritime environment?
2. Which techniques are available to reduce false positives during target detection?
3. Can we design a model that can improve the accuracy of target detection in maritime environment?

1.4 Objectives

1. To review and analyze methods for target detection in maritime environment.
2. To explore ways of reducing false positives during target detection.
3. To design and implement a target detection model that minimises the occurrence of false positives.

1.5 Justification

Target detection is very important as it is the key technology in intelligent monitoring systems and is involved in any military and civilian applications [3]. Maritime surveillance domain is becoming important therefore security is a requirement because of increasing threats such as illegal smuggling, oil spills, fishing, immigration and piracy in another part of the world [4]. It is also significant for the military to see and locate their enemies for protection or attack.

1.6 Research outcome

The outcome of this research is a robust target detection algorithm which improves the detection rate and further reduces the occurrence of false positives occurring during the detection.

1.7 Research deliverables

A dissertation report documentation of a research done on modelling false positive reduction in maritime target detection. The research also produced a research paper which is currently under review by SATNAC.

1.8 Research limitations

The work contained in this dissertation is target detection in a maritime environment. Although there are many objects on the ocean and surroundings, this study only focused on detecting boats and ships on the background of the sea. The data used in this research contained only boats and ships.

1.9 Dissertation development

Chapter 2: presents a detailed background on target detection. More literature on algorithms used for target detection was reviewed and the detection methods were compared.

Chapter 3: gives an insight into the methodology used in completing the research. It also encompasses the experimental design followed in this research and explaining how each stage of the experimental design was carried out.

Chapter 4: presents results and analysis. The chapter further discusses the analyses on the performance of the proposed algorithm based on the size increase of training data.

Chapter 5: concludes by discussing the results obtained in this study for each objective, furthermore presenting future work and recommendations.

2 Chapter two: Background and Literature review

This chapter presents a summary of the literature on target detection. It encompasses the background, related work, and conclusion based on the related work. The background gives detailed information about image formation, machine learning, neural networks, image processing, and computer vision. The related work part encompasses detection work using different methods like radar technology, Gaussian mixture models, adaptive background models, and deep learning algorithms. It is concluded that Faster R-CNN will be a suitable method for this research.

2.1 Background

Although object detection is a small field in computer vision and image processing, it has become a necessity for the South African navy. The field of tracking and detection is the center of attraction under the prism program [1]. Target detection in a cluttered maritime environment plays a very important role in civil applications and military [5].

2.1.1 Formation of image

The goal of the research was to intelligently analyze and manipulate images or video frames to detect boats and ships in maritime environments. The image manipulation and analysis follow a certain scene of geometry and image formation processes. The image formation is composed of geometric primitives which are points, lines, and plains to describe three- dimensional shapes [6].

Images are formed out of discrete color and intensity values. The values relate environment lighting surface properties, camera optics and properties of a sensor. The image exists because of point or area light sources [6].

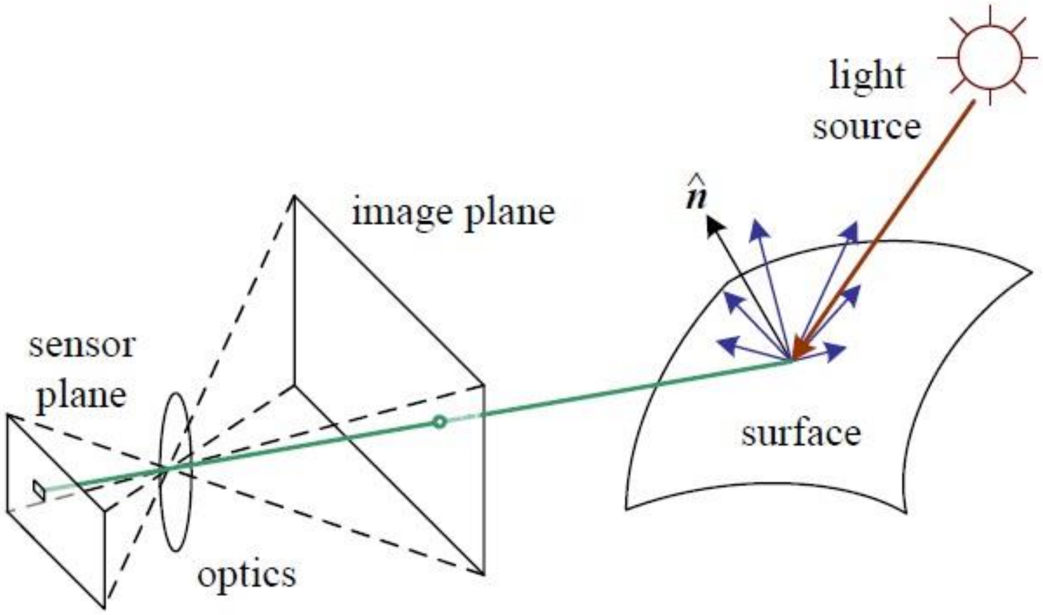


Figure 2.1: *Image formation* [6].

The light hits the surface with an object, scattered, and reflected. This light reaches the camera, pass across the lens and reach the sensor. The photons arriving at the sensor are finally converted into digital Red, Green, Blue (RGB) values which are observed in digital images[6].

2.1.2 Machine Learning

Machine learning is a technology of programming computers to learn from input data and changing the experience into knowledge [7]. The technology of machine learning encourages the development of robust and high performing vision algorithms and the development of best architectures in less development time. Machine learning is very useful in computer vision as it converts the sensed signal into meaningful representation and bridge the gap between representations and information to run the task. The inductive machine learning strategy encompasses supervised and unsupervised learning [8]. Unsupervised learning methods include K-means, Gaussian mixtures, hierarchical clustering, and spectral clustering. Methods for supervised learning include support vector machines, decision trees, K-nearest neighbors and naïve Bayes, neural networks.

2.1.3 Neural Networks

The attempts to find the mathematical representations of processing information in biological systems led to the discovery of the term neural network [9]. Artificial neural networks consist of a number of artificial neurons. The neuron has a number of input ports and an output port. A weight is allocated to each input port and a delta weight can be altered to make the convergence faster. The weighted sum of inputs makes up the neuron's output [10]. In 1991 it was concluded that multilayer feedforward architecture (Figure 2. 2) provides the neural network with the ability to learn efficiently and it is now widely implemented worldwide [11].

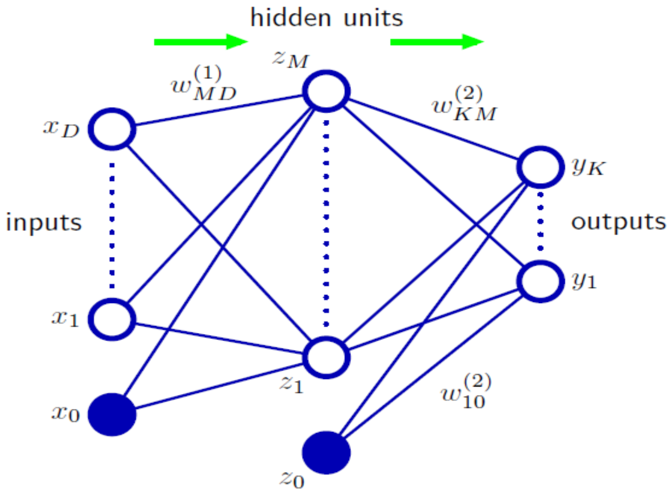


Figure 2.2: *Neural network* [9]

Figure 2. 2 exemplifies a neural network with two layers. The neurons denote input variables, hidden variables, and output variables. The links between nodes represent the weight parameters.

The link from additional input and hidden variable x_0 and z_0 denotes bias parameters. The arrows signify the direction of the information flow during forward propagation [9].

2.1.4 Image processing and Computer vision

Image processing operations are included at the beginning of computer vision methods to process the images for further image analysis. These image processing steps can be exposure correction, color balancing, image noise reduction, increasing sharpness and image rotation. Image transforms can manipulate each pixel independently of neighboring pixels (point operators) and they can also manipulate the pixels depending on the neighbors [12]. Computer vision began in the 1970s to simulate human behavior and upgrade robots with exceptional intelligence. In 1980s people focused more on the mathematical side of image analysis. In the 1990s more of computer vision projects including object recognition became very popular. In 2000s much focus was on vision graphics fields which encompass stitching of images, light field capture, and rendering, high dynamic range (HDR) capturing of images by bracketing the exposure [13].

Although computer vision methods have become the solution to many video processing problems, maritime environment present a lot of challenges for computer vision [14]. The maritime environment is divided into different categories which are the cause of difficulties in computer vision methods. The maritime environment is composed of foreground objects, outliers, and backgrounds. Foreground objects include sear vessels, small boats, kayaks, buoys, debris, ships, etc. Birds, air vehicles, fixed structures in ports are outliers or background. Water speckle, clouds, foams, and wakes can also be categorized as background [15].

Other factors resulting in difficulties in computer vision include loss of information from three dimensions to two dimensions which occur during the capturing of an image with a camera or an eye. When human beings attempt to learn from images, they use prior knowledge to interpret the current images. With humans, the knowledge gathered in the past allow them to reason and solve new problems. In the past decade artificial intelligence attempted to teach computers to learn and understand observations, they progressed tremendously but computer learning ability is still limited. When interpretation is introduced to computer vision, the use of mathematical logic, linguistic as syntax and semantics are allowed. The resulting images in computer vision can be understood as an instance of semantics [12].

Noise is one of the common effects of images. The presence of noise in images requires mathematical tools such as probability theory, to deal with uncertainty. When more complex methods are used, image analysis becomes very complicated as compared with standard tools. Images and video sequences are very huge, leading to difficulties in achieving real-time performance if the formulated processing is difficult. The measured brightness in images is represented by complex image formation behavior. Radiance (Brightness, image intensity) depends on the irradiance (intensity, the type of light source and position), the location of the observer, the surface local geometry and the reflectance properties of the surface [12].

Algorithms for image analysis use particular storage bin in operational memory and its local neighborhood, during this process the computer perceives the image through a keyhole. It becomes more difficult for a computer to understand more global context when perceiving the world through keyhole [12].

2.2 Related work

Object detection techniques can be divided into supervised and unsupervised approaches. Background subtraction, segmentation and point detectors are included as categories for object detection [16]. Chan, Der and Nasrabadi believed that need to improve situational awareness has led to a rapid increase of sensors to help people interpret situations more efficiently [17]. Many researchers did the detection focusing on small targets with small signal-to-noise ratio, large targets, moving targets or stationary targets using scenes captured with ordinary cameras or sensor-based platforms. According to [18], it is substantial to do robust detection of small objects for self-defense during attacks and in an infrared search. The cluttered environment results in unresolved issues with a false alarm rate when using most of the present algorithms [18].

A research was made by comparing adaptive background models. This research gave an insight into performance detecting methods, time of computation and how these methods are used. Due to the greater demand for video surveillance domain for real-time image processing, systems algorithms that are reliable and efficient were recommended for target detection. In this article, five adaptive background differencing techniques were compared and the same public benchmark datasets were used by the detectors for evaluation. A large dataset was used and the results were compared with respect to autonomously hand labeled ground truth. The methods compared in this research were basic background subtraction algorithm (BBS), the W4, multiple Gaussian mixture (MGM), Single Gaussian Model (SGM), and LOTS [19].

Basic background subtraction (BBS) was the easiest algorithm that detected objects by computing the difference between image background for each color channel and the current frame. Furthermore, classifying one by one pixel as foreground a threshold operation was utilized. Objects were segmented from the background using connected component analysis. The second algorithm denoted as W4 was performed in grayscale images. To form the background scene three values were used to represent each pixel. The values used were maximum intensity (Max), minimum intensity (Min), and maximum intensity difference (D) between sequenced frames as a period of training was continuing. In computing foreground objects, four steps were followed: thresholding, region-based noise removal, filtering and object detection [19].

Single Gaussian model (SGM) algorithm was operated in *pfinder* and it assumed that each pixel was a comprehension of a haphazard variable with a Gaussian distribution. Independently estimation for each pixel was done for first and second-order statistics of the distribution. In a mixture of multiple Gaussians (MGM) all of background pixels were modeled using the mixture of Gaussians and to adapt the weights and parameters of the Gaussians frames. This method has been used mostly for modeling complex and time-varying backgrounds. Lehigh omnidirectional tracking system (LOTS) algorithm was applied to grey scale images and uses two image backgrounds with two per-pixel thresholds. Each pixel was treated differently by the per-pixel threshold image to allow the robustness of the detector as to localize noise in image regions with low size. The steps of this algorithm include background and threshold initialization, detection and labeling and backgrounds and threshold adaptation [19].

The performance of the algorithms was evaluated using the same dataset for all the algorithms. To measure the performance of the detectors Tracker Detection Rate (TRDR) and the False Alarm Rate (FAR), calculated as follows,

$$TRDR = \frac{TP}{TP + FP} \qquad FAR = \frac{FP}{TP + FP}$$

Where TP represents true positive, FP signifies false positive and FN represents false negative. The results showed that LOTS and SGM bring off improved results as compared to [19].

Ghahramani et al. tackled the target detection problem in sea clutter on Blind Source Separation (BSS) [20]. They proposed Complex-valued spatiotemporal FastICA (CSTFICA) in the marine environment using fixed monostatic maritime radar for separating valuable signal and sea clutter noise. In the research, radar was used to solve the problem where the target signals were embedded with an undesired signal. Models which represent target signal and clutter signal lead to different detectors. The cluttering of the sea lead to poor detection of the target more especially when the target was not moving or moving slowly. The aim of the study was to detect weak targets which materialized in the presence of strong sea clutter Doppler cell. K- Distribution was used to model the sea clutter and produce synthetic clutter. Furthermore, a method built on CSTFICA was presented and was independent of clutter and target statistical characteristics. A simulation was used for testing the performance of Blind Source Separation (BSS). Experimental results indicated that the method proposed did not need the prior information of radar signals, clutter PDF modeling, and signals of a target. The proposed method out performed block-ANMF detector when it was observed in experiments. Ghahramani et al. therefore concluded that Blind Source Separation (BSS) algorithm was capable of being used to suppress the radar clutter [20].

Tong et al. analyzed image pattern matching for detection and tracking in marine radar images [21]. Automatic Radar Plotting Aid (ARPA) was added as a tracking feature in the radar. It was able to provide some of the functionalities, but radar object detection was one of the pending technical issues. The following steps were followed, image data preprocessing, finding suitable matching methods and effective & efficient application. Area-based matching (ABM) and feature-based matching (FBM) were used as image matching approaches. Three approaches for Feature-based matching (FBM) were used, namely SIFT/ SURF, Harris/ FAST, and Dense. The approaches failed the initial test, failed minimum requirements, and they were regarded as not reliable. Template Based Matching (TBM) approaches, CC, CCOEFF, and SSD were used. CC was not reliable in the initial test and CCOEFF & SSD were suitable but they were not invariant to rotation and scale. In this study based matching approach, SSD was found to be more suitable for radar image detection [21].

Algorithms for moving object video detection were discussed and background for the maritime environment was analyzed and discussed. Jing and Chaojian stated that existing algorithms for moving video detection included optical flow, background subtraction and frame difference [22]. The measuring of optical flow was the problem existing in image sequence processing. Optical flow methods were analyzed in terms of three categories, namely, pre-filtering with low-pass and band-pass filters, basic measurements abstraction, like spatio-temporal derivatives and the integration of generated measurements to return a 2-D field flow. In the background subtraction, moving objects were detected by subtracting the objects from the background. The foreground was made by all the pixels greater than the threshold and the background was made by the pixels smaller than the threshold. In frame differencing, if the object changed its position, the grey values of the image changed leaving the one with no object movement unchanged. Frame differencing took two frames and compared their grey values. The survey concluded that due to the dynamic background of the sea and weather conditions, methods for moving target detection in the maritime environment had limitations, they did not perform in a satisfactory way. It was inadequate to improve the performance of moving target detection methods by using only one detection method. [22].

Wan et al. proposed an adaptive threshold algorithm built on improved Surendra algorithm of background subtraction and a gradient update algorithm to robustness in different videos [23]. The algorithm included five sectors namely, background extraction, the updating of background, background subtraction, target detection, and post-treatment. The original algorithm was changed after studying the factors affecting threshold value which were changes in the environment. The implementation was performed using 3G of memory, and for software environment Windows 7, Microsoft Visual Studio 2010 Video frame rate of 15 frames / s, the frame size was 240 320. The improved Surendra method used to extract the background was compared with OSTU algorithm and the results displayed that the method proposed was more robust and powerful. OSTU algorithm took longer time than the proposed method. The algorithm recommended was also compared with the Gaussian mixture model and the average background algorithm and the proposed algorithm consumed less time than other algorithms [23].

Jadhavn modelled the dynamic threshold method for target detection and target tracking in an intelligent transportation system, security system in airports and in video surveillance applications [24]. The main objective was to present object detection for moving objects using background subtraction. The proposed method was done by removing the current frame from the background image. The pixels from a moving object were determined to restrain the impact of light alterations [24]. The results were shown by displaying figures for moving object detection with reference background subtraction. The results showed that the proposed real-time algorithm was very fast and not complicated [24].

Wang and Sng examined deep learning algorithms to analyse a video. More emphasis was put on research areas like, image classification, object tracking, object detection, face recognition, and scene labelling. Four derived forms of deep learning architecture were stated as Stacked Denoising Auto-Encoders, Convolutional Neural Networks (CNN), Deep Boltzmann Machines (DBM), (SDAE) and Deep Belief Networks (DBN) [25]. The convolutional neural network performed better as compared with other forms of deep learning because it allowed simple training as it had less parameters to evaluate [25].

Another survey on object detection and tracking approaches was done, with the goal of presenting different algorithms on detection, classification, and tracking of objects. In that research, it was found that if is very difficult to detect unidentified objects and objects with the difference in color, shape, and texture, as a result in many computer vision methods it is assumed that the camera is fixed. For object detection, Frame differencing, background subtraction methods and optical flow were explained and the comparative study was done [26]. According to the research, frame differencing can adapt to dynamic environments but there are difficulties in obtaining complete moving object outline, therefore frame differencing is poor in detecting moving objects. The optical flow was not appropriate for real-time challenging occasions because of a large quantity of mathematical processes, sensitivity to noise and poor anti-noise performance. The comparative study of object detection was based on accuracy and computational time [26]. In background subtraction methods Gaussian of Mixture method was moderate in both computational time and accuracy, and Approximate Median was low to moderate in accuracy and moderate in computational time. Optical flow method was moderate in accuracy and high in computational time. Frame differencing was high in accuracy and moderate in computational time. According to the overall results, background subtraction was the simplest method with better results in providing object information [26].

An adaptive method for object detection and tracking video objects with adaptable background was proposed. The writers argued that in moving camera videos background and foreground information keeps on changing, therefore Pseudo motion was estimated and compensated by video frame frequency domain analysis [27]. Phase correlation was employed to calculate comparative translative offset to estimate global displacement between consecutive video frames. In the presence of noise frequency domain analysis was found to be stronger that Spatial-Domain approach. The experimental results produced outstanding moving objects detection, however, the proposed method was inadequate in inputs against complex backgrounds and with many small objects [27].

Guo et al. examined the usefulness of employing deep learning method to suppress sea clutter and perform target detection process. Deep convolutional auto-encoders (DCAEs) were used to filter out sea clutter and Logistic regression classifier was used to carry out target detection [28]. Deep convolutional auto-encoders suppressed all the sea clutter from the target automatically. The step of deep learning algorithm included data normalization, where pre-processing occurs. Figure 2.3 below represents the deep learning based on deep convolutional auto-encoders [28].

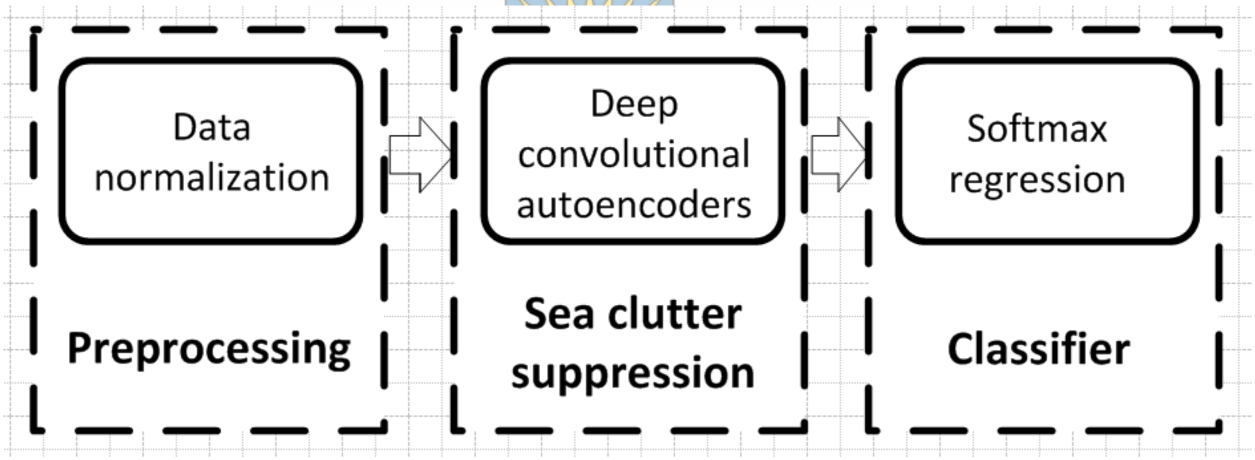


Figure 2.3: *Deep convolutional auto-encoders steps* [28].

The work by Jodalli et al. employed a robust method which recognized and detected objects by feature set and afterward identified different anomalies detected [29]. The methodology followed two steps, detect SURF features and feature set creation and analysis. In the first step, background subtraction was used to remove failures caused by cutter, noise, waves, an unpredictable and dynamic background of the marine environment. Clusters and features were detected and removed using histogram together with background elimination method. The second step dealt with capturing features of objects of interest and their data set was created. Image segmentation was done on those image objects and contour set of objects pertaining to points depicting curves and lines were stored for analysis. Random sample consensus (RANSAC) algorithm was utilized to remove sea clutter. The work was successful with 90% identification [29].

In another research the linear classifier was trained based on deformable part models (DPM) and the average expectancy of foreground likelihood. The average expectancy of foreground likelihood was normalized with respect to the size of a window to deal with its changes. Foreground probability model was build using motion cue and mean expectancy of the pixel level, the foreground likelihood was computed to allocate priori probability into sliding window [30].

Kim and Lee believed that the presence of clutter produces false detection in real world, leading to obstruction of the detection system, so they presented region- adaptive clutter rejection method [2]. Detecting long-range small targets seemed to be challenging due to small and dim signal. To achieve the detection rate, the detection threshold was lowered, but this method produced more false detections due to background clutter. The primary objective of the study was to develop the target detection process by decreasing the number of false positives during the detection [2]. Local directional background removal filter (L-DBRF) and Modified mean subtraction filter (M-MSF) utilized in the study performed well in terms of rejecting horizontal line clutter, producing better detection with few false alarms. False detections were reduced depending on the type of clutter. Cloud clutter false detections were removed by Spatial-Attribute-Based classification, heterogeneous background removal filter was used to remove false detections caused by a horizontal line and temporal consistency filter was used to reject false detection caused by sun-glint. Results showed that region adaptive clutter rebuff method reduced false detection as compared to mean subtraction filter (MSF) with small degradation rate [2].

Frost and Tapamo used level set with shape priors to track moving objects and background subtraction for moving object detection [31]. In the object detection part, the following steps were followed; firstly, the grey scale image frame from the maritime video is inputted into the system, pre-filtering, background subtraction, and post-filtering. After the post-filtering stage when the object shape was known, level set filtering was applied, otherwise, the binary image was converted. Pre-filters, Gaussian filter, and Median were used to remove image noise. Sea motion in a maritime environment created problems when using background subtraction therefore, binary filtering was used after background subtraction to remove white pixels which were not desired. The post-filtering was used for false positive removal in the image. The results produced showed that the Gaussian filter provided the best results, so it was utilized. One risk with filtering algorithms was that removing very small targets was possible [31].

Ross Girshich et al recommended a detection method which combines the region proposal and convolutional neural network (CNN) therefore called R-CNN [32]. The region-based convolutional neural network (R-CNN) improves the performance of ordinary convolutional neural networks (CNN). Compared to HOG-like features, the research was the first to show high detection performance of CNNs in PASCAL VOC. The objects were localized with a deep network and the model was trained with few detection data. The object detection algorithm encompassed generation of region proposals which categorizes the candidate detections which were present. The convolutional neural network removes a feature vector of fixed length from the region thus producing linear SVMs for each class. The network model was configured with seven layers which included five convolutional layers and remaining layers were fully connected layers. The C-NN model was pre-trained on ILSVRC 2012 dataset with bounding box labels. The proposed detection algorithm improved the mean average precision (mAP) related to the previous results which were measured as best [32].

Fast region-based convolutional neural network (Fast R-CNN) detection method was proposed by Ross Girshich [33]. The work made use of a deep convolutional neural network used to categorize object proposals. The previous work on R-CNN had drawbacks which were attempted in Ross Girshich’s work. The R-CNN training process included many steps, it was slow and expensive. Faster R-CNN network produced object proposals for the whole image at once. A set of convolutional layers together with max-pooling layers processed the image thus producing convolutional feature map. For each object proposal feature vector was abstracted using a region of interest (RoI) pooling layer. Feature vectors were further processed to produce the final output. The results showed that Faster R-CNN algorithm produced high-quality detection that R-CNN. The method also used single training with multi-task, captured all network layers and disk storage was not required. It was able to train VGG16 network nine times quicker than R-CNN, 213 times quicker during testing time and improved high mean average precision on PASCAL VOC 2012 [33].

Although Fast R-CNN improved the detection by reducing running time, Faster R-CNN algorithm was proposed to make more improvements on detection. The work introduced region proposal network (RPN). RPN and detection network shared the full image convolutional features. In RPN network, the object bounds and objectness scores were predicted at each spot. Region proposal network took an input image, processed it and produced rectangular proposals with objectness total. The region proposals were created by descending a network over the feature map on preceding convolutional layer. The sliding window was matched into a lower dimensional vector and the vector was given to two fully connected layers. Region proposal and object detection network were trained on one scale images. The method was tested on PASCAL VOC 2007 data for detection. Proposal quality and detection rate were improved in Faster R-CNN [34].

Region-based fully convolutional network (R-FCN) method was presented for object detection. This method attempted to upgrade from costly per region subnetwork by sharing almost all computation on the whole image. The work of R-FCN adopted a two-stage object detection technique which included region proposals and region classification. During the training phase, the region proposals were pre-computed for a simple end-to-end training of R-FCN architecture. The positive examples were defined as a region of interest (RoI) overlap and negative examples otherwise. The experiments were performed on PASCAL VOC with 20 object classes. This work adopted state-of-art image classification supports like ResNets which are fully convolutional. The method was found to be accurate and fast during training and reference phase [35].

YOLO 9000, a state-of-art detection algorithm proposed by Redmon and Farhadi was found to detect over 9000 object classes. The steps carried out in the work included improving the original YOLOv2 detection and trained the improved algorithm with over 9000 object categories from ImageNet and COCO dataset. The dataset was combined using hierarchical classification. YOLO9000 was proved to be faster and detected more than 9000 object classes in real time [36].

2.3 Comparison of detection methods

The maritime environment is a dynamic environment with a lot of clutter. Most of the algorithms for target detection produce many false positives because of the background clutter. The horizon line detection method has been widely used by object detection researchers as part of image pre-processing to improve the robustness of detection algorithms in maritime environments. Horizon line detection mark the targeted frames from frames outside the region of interest. The region based horizon detection and edge maps are mostly utilised for horizon detection in maritime environment for most detection techniques. However, horizon detection has not been used in this study because most of the objects in the images lie exactly in horizon line. When there are objects in horizon line, horizon detection performance can be poor leading to poor object detection results.

Previous works as shown above, presented a rapid use of radar technology for vessel detection in a maritime environment. Radar detection produced a high detection rate but clutter restricts the radar signal, leading to difficulties in detection. Non- metal, small, stationary and slow-moving objects were not detected when using radar technology. Furthermore using radar technology was very costly[37]. Background subtraction methods posed a challenge when using moving camera. BSS utilise ICA to recover independent sources when source signals are not clear. While this method is very sufficient in recovering independent sources, they are not brought in an orderly manner leaning to minimum solution because independent sources are difficult to identify.

Convolutional neural network (CNN) required many regions to be accurate, R-CNN required more computation because it runs three models, Fast R-CNN also required more computation time due to slow selective search method, Faster R-CNN process was also slow because more systems were run but it was faster than Fast R-CNN and YOLO method had difficulties to detect small objects.

In this study, Faster R-CNN was utilized as an approach to detect targets. Although YOLO and other methods were faster than R-CNN, they could not be used due to computing problems, Faster R-CNN was suitable. To retrain the model, small data with two object classes were utilized therefore there was no need for a model capable of training over 9000 object classes. The goal of this research was to produce more detections with reduced false positives and track the performance as training data increases.

2.4 Conclusion

This chapter discussed the background work on the field of this research. The background explained image formation, machine learning, neural networks, images processing, and computer vision. The related work included most of the work in object detection, including classic algorithms and modern algorithms. The last part of this chapter presented the common aspects, differences, and challenges in the related work. It concludes by explaining the significance of using Faster R-CNN for this project.

3 Chapter three: Research methodology & Design

This chapter gives an insight into how the research was conducted, elaborating more on the approach used to answer research questions. It encompasses the research process, which includes formulation of research problem, background and literature review, conducting experiments, evaluation and final results. The experimental design incorporates image labeling, data conversion, training of models and testing. The chapter also presents the description of data collection and format.

3.1 Methodology

Research is the gathering of knowledge through scientific approach in order to find an outstanding solution to a problem. Basically, research can be undertaken in many ways depending on its nature. Research types range from descriptive, analytical, applied, fundamental, quantitative, conceptual and empirical [38]. In this study, empirical research was used. Empirical research incorporates the use of experiments to uncover and clarify facts and revise theories. It can be validated or invalidated based on observations and experiments [39]. In the context of this study empirical research was utilised experimentally and empirical evidence was acquired. This research produced empirical evidence as it involved training of data and testing which generated experimental results. Firstly, the research questions were formed and the knowledge was acquired through experimentation which were in agreement with efficiency of the model. Questions on the existing methods for detecting objects in a maritime background and the performance of these methods in terms of reducing false positives in target detection were answered through experiment. More literature about target detection was revised, furthermore, experiments were carried out and the results were analyzed quantitatively.

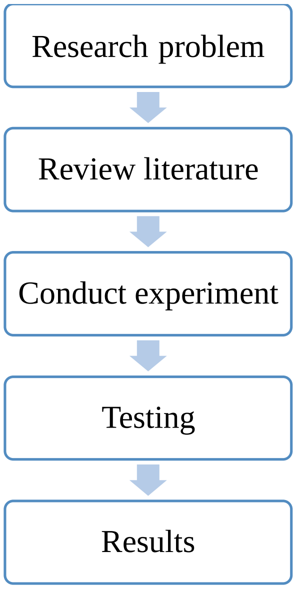


Figure 3.1: *Research process*

Figure 3.1 shows the steps carried out in this study. Firstly the problem that led to this study was formulated and the research questions were outlined. Secondly, a background and review study was completed on machine learning, artificial neural networks, computer vision, convolutional neural networks, existing detection approaches and state of art detection methods. The experimental design was represented and described. Testing phase followed and raw data was collected and analyzed. Through the analysis of data, results were formulated and were then discussed.

3.2 Research Design

This sub-section presents the experimental design. The problem undertaken in this study was a boat and ship detection. The problem was tackled in a supervised manner because it involved supervised learning. Pre-trained faster R-CNN with inception v2 [40] was trained using labeled data containing images with objects labeled as a boat or ship. An open source framework, Tensorflow object detection API [41] was utilized. Tensorflow object detection API is a tensorflow built API to train and evaluate computer vision projects like object detection. This framework contains pre-built architectures and weights for faster R-CNN with inception v2 model. The experimental design contains five stages namely:

- Image labeling
- Converting files from XML to CSV
- Converting CSV to TFRecords
- Training and testing

Figure 3. 3 shows the architecture for experimental design.

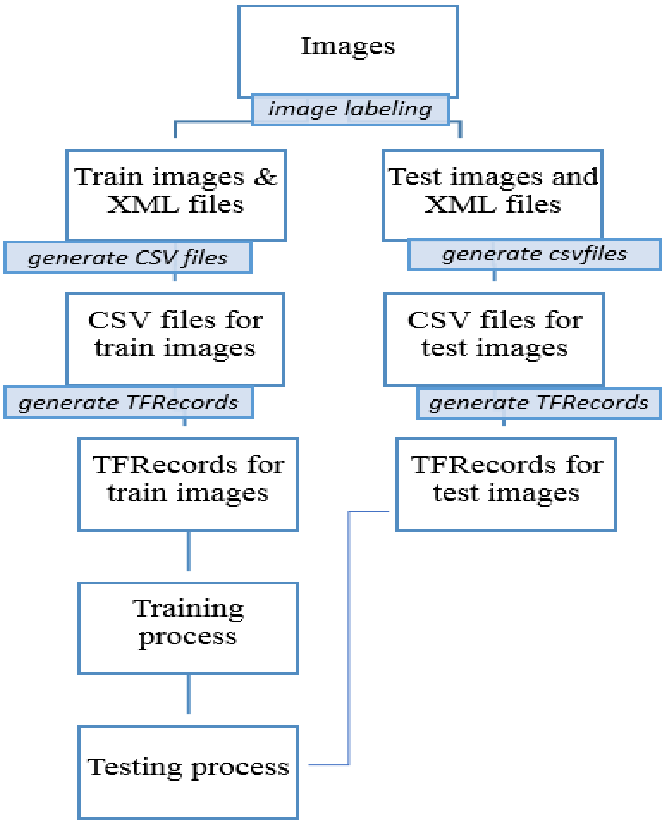
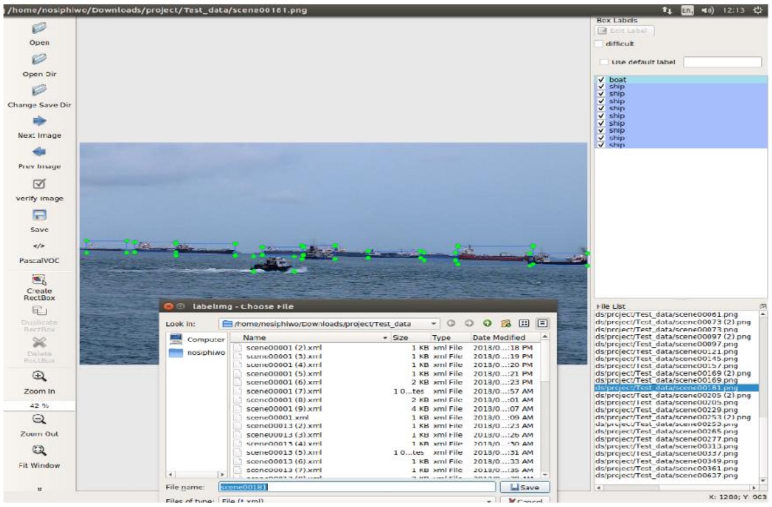


Figure 3.2: *An architecture of an experimental design*

3.2.1 Dataset and image labeling

The dataset was acquired on CVonline image databases. The web page is composed of the computer vision videos and images which researchers can use in evaluating the algorithms. The dataset used in this research was made from included videos from Singapore maritime dataset [14] and images from maritime imagery in the visible and infrared spectrum [42]. The Singapore videos were captured using 70D cameras in Singapore waters. They were captured in the high definition (1080 x 1920). The dataset included videos which were captured by a fixed camera and a moving camera. The videos were taken during different times of the day, like in the morning, midday, afternoon and evening [14]. The images from maritime imagery in the visible spectrum were used mostly for training in this study. ISVI IC-C25 RGB camera was utilized to capture (5056 x5056) pixel images. During the capturing of these images the camera focus and exposer were adjusted to be suitable for all times of the day [42].

Maritime environment videos were converted into frames using VLC. Video frames and images were used as initial sample for training and testing phase. The images were labeled using LablImg. During the labeling, each image was opened and all possible ships and boats present in the image were rounded with rectbox and each image was saved. After the labeling process was through, the directory where these labeled images were stored contained XML files in addition to each labeled image.



/home/nosiphiwo/Downloads/project/Test_data/scene00181.png

☑ boat
☑ ship
☑ ship
☑ ship
☑ ship
☑ ship
☑ ship
☑ ship
☑ ship
☑ ship

labelImg - Choose File

Look in: /home/nosiphiwo/Downloads/project/Test_data

Name	Size	Type	Date Modified
scene00001 (2).xml	1 KB	xml File	2018/0...:18 PM
scene00001 (3).xml	1 KB	xml File	2018/0...:19 PM
scene00001 (4).xml	1 KB	xml File	2018/0...:20 PM
scene00001 (5).xml	1 KB	xml File	2018/0...:21 PM
scene00001 (6).xml	2 KB	xml File	2018/0...:23 PM
scene00001 (7).xml	1 0...tes	xml File	2018/0...:57 AM
scene00001 (8).xml	2 KB	xml File	2018/0...:01 AM
scene00001 (9).xml	4 KB	xml File	2018/0...:07 AM
scene00001.xml	1 KB	xml File	2018/0...:09 AM
scene00013 (2).xml	1 KB	xml File	2018/0...:23 AM
scene00013 (3).xml	1 KB	xml File	2018/0...:26 AM
scene00013 (4).xml	1 KB	xml File	2018/0...:30 AM
scene00013 (5).xml	1 0...tes	xml File	2018/0...:31 AM
scene00013 (6).xml	1 KB	xml File	2018/0...:33 AM
scene00013 (7).xml	1 KB	xml File	2018/0...:35 AM

File name: scene00181

Files of type: File (*.xml)

File List

ds/project/Test_data/scene00061.png
ds/project/Test_data/scene00073 (2).png
ds/project/Test_data/scene00073.png
ds/project/Test_data/scene00097 (2).png
ds/project/Test_data/scene00097.png
ds/project/Test_data/scene00121.png
ds/project/Test_data/scene00145.png
ds/project/Test_data/scene00157.png
ds/project/Test_data/scene00169 (2).png
ds/project/Test_data/scene00169.png
ds/project/Test_data/scene00181.png
ds/project/Test_data/scene00205 (2).png
ds/project/Test_data/scene00205.png
ds/project/Test_data/scene00229.png
ds/project/Test_data/scene00253 (2).png
ds/project/Test_data/scene00253.png
ds/project/Test_data/scene00265.png
ds/project/Test_data/scene00277.png
ds/project/Test_data/scene00313.png
ds/project/Test_data/scene00337.png
ds/project/Test_data/scene00349.png
ds/project/Test_data/scene00361.png
ds/project/Test_data/scene00637.png

X: 1280; Y: 903

Figure 3.3: *LablImg Interface*

Figure 3.3 represents the LablImg interface. The left column items show the labeling tools which are used to open the directory containing the data, browse through the directory using next and previous icon, creating a rectbox, zoom in and out and save. The list of labels for each image is represented at the right corner of the interface. The list of files at the bottom right corner represents the images contained in the working directory. When clicking in each image on the file list, it appears in the center of the LabelImg interface and allows for labeling. When saving the labeled image, the labeling information is saved as an XML file into the same directory containing the original images, figure 3.4 shows the directory with both images and XML files.

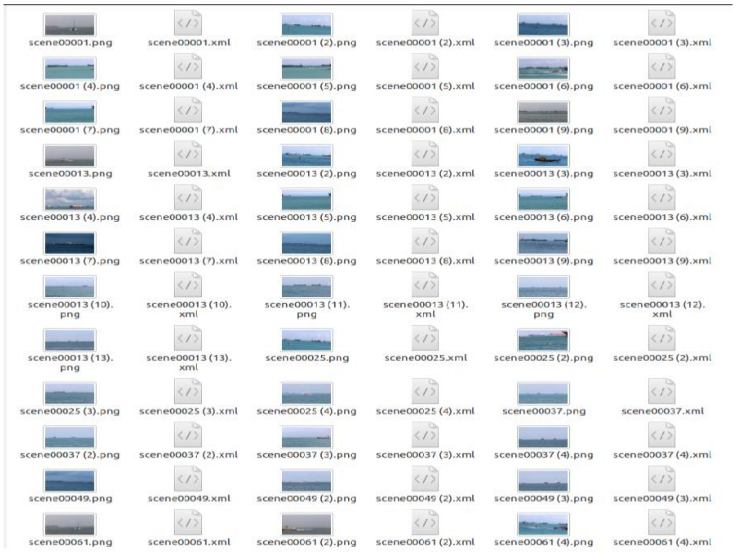


Figure 3.4: *Sample images with XML files*

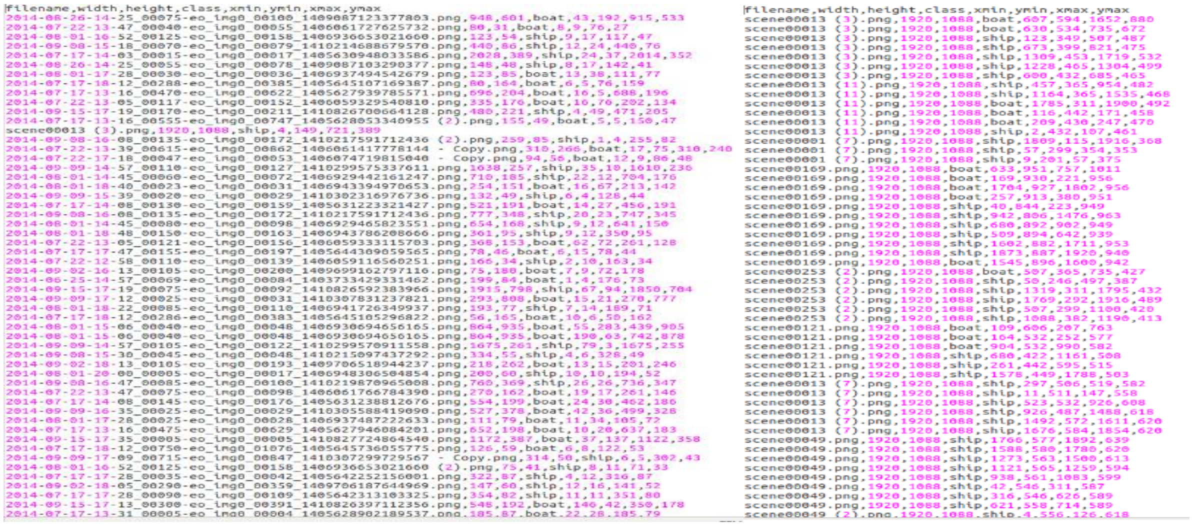
Figure 3.4 shows the sample images with XML files. After the labeling all the images, each image was represented by its XML file. Each XML file contained information about the image it represents. It contained the name of the folder containing the image, image name, image path, image source, image size in terms of width, height and depth, and the name of all labeled objects together with their sizes.

There were 560 labeled images in the first labeling stage, 500 images were reserved for training and 60 images were reserved for testing. The first sample contained 100 labeled images,for the second training phase, the first sample was incresed by 200 labelled images to make 300 labeled images, and these 300 train images were further increased by 200 to make up the third sample which contained 500 labeled images. There were challenges in proceeding with adding 200 new labelled images to the previous 500 train images, so the previous 500 train images were duplicated with 200 train images to form the fourth sample which contained 700 images and for fifth sample

500 train images which were used as third sample were duplicated with 400 train images to make 900 train images.

3.2.2 Converting XML files to CSV files

Tensorflow requires data in TFRecord format for training, therefore XML data was converted into CSV files which was further converted into TFRecords. The conversions were accomplished for each sample. CSV files for train labels and test labels were stored in a new directory. The conversion of XML files to CSV file was computed using python3. Figure 3.5 represents the train CSV file and test CSV file for one sample.



```
filename,width,height,class,xmin,ymin,xmax,ymax
2014-08-26-14-25_00075-eo_img0_00100_1409087123377803.png,948,601,boat,43,192,915,533
2014-07-22-13-47_00040-eo_img0_00055_1406061727625732.png,80,31,boat,8,9,76,27
2014-08-01-16-52_00125-eo_img0_00158_1406936653021660.png,123,54,ship,9,17,117,47
2014-09-08-15-18_00070-eo_img0_00079_1410214688679570.png,440,86,ship,12,24,440,76
2014-07-17-14-03_00015-eo_img0_00017_1405630948033586.png,2028,389,ship,24,37,2014,352
2014-08-26-14-25_00055-eo_img0_00078_1409087103290377.png,148,48,ship,8,17,142,41
2014-08-01-17-28_00030-eo_img0_00036_1406937494542679.png,123,85,boat,13,38,111,77
2014-07-17-18-12_00288-eo_img0_00385_1405645107169387.png,80,164,boat,6,5,76,159
2014-07-17-13-16_00470-eo_img0_00622_1405627939785571.png,696,204,boat,16,5,688,196
2014-07-22-13-05_00117-eo_img0_00152_1406059329540810.png,335,176,boat,16,76,202,134
2014-09-15-17-19_00170-eo_img0_00211_1410826700664128.png,480,221,ship,4,49,471,205
2014-07-17-13-16_00555-eo_img0_00747_1405628053340955 (2).png,155,49,boat,5,5,150,47
scene00013 (3).png,1920,1088,ship,4,149,721,389
2014-09-08-16-08_00135-eo_img0_00172_1410217591712436 (2).png,259,85,ship,1,4,255,82
2014-07-22-13-39_00615-eo_img0_00862_1406061417778144 - Copy.png,310,266,boat,17,75,310,240
2014-07-22-17-18_00047-eo_img0_00053_1406074719815040 - Copy.png,94,56,boat,12,9,86,48
2014-09-09-14-57_00110-eo_img0_00127_1410299575337611.png,1638,257,ship,35,10,1610,236
2014-08-01-14-45_00060-eo_img0_00072_1406929442161247.png,710,185,ship,22,12,704,176
2014-08-01-18-40_00023-eo_img0_00031_1406943394970653.png,254,151,boat,16,67,213,142
2014-09-09-15-39_00020-eo_img0_00029_1410302316976736.png,132,49,ship,6,4,128,44
2014-07-17-14-08_00130-eo_img0_00159_1405631223321427.png,521,191,boat,14,27,456,191
2014-09-08-16-08_00135-eo_img0_00172_1410217591712436.png,777,348,ship,20,23,747,345
2014-08-01-14-45_00080-eo_img0_00098_1406929465823551.png,654,168,ship,9,12,641,150
2014-08-01-18-48_00150-eo_img0_00163_1406943786208666.png,361,95,ship,9,12,350,95
2014-07-22-13-05_00121-eo_img0_00156_1406059333115703.png,368,153,boat,62,72,261,128
2014-07-17-17-47_00155-eo_img0_00197_1405644309059565.png,78,46,boat,6,15,78,44
2014-07-22-12-58_00110-eo_img0_00139_1406059116560251.png,166,34,ship,2,10,163,34
2014-09-02-16-13_00105-eo_img0_00200_1409699162797116.png,75,180,boat,7,9,72,178
2014-06-25-14-57_00069-eo_img0_00084_1403733429331462.png,199,84,boat,1,4,176,73
2014-09-15-17-19_00075-eo_img0_00092_1410826592383966.png,1915,798,ship,67,94,1850,704
2014-09-09-17-12_00025-eo_img0_00031_1410307831237821.png,293,808,boat,15,21,270,777
2014-08-01-18-22_00085-eo_img0_00110_1406941726349937.png,193,77,ship,7,14,189,71
2014-07-17-18-12_00286-eo_img0_00383_1405645105296822.png,56,165,boat,10,6,50,162
2014-08-01-15-06_00040-eo_img0_00048_1406930694656165.png,864,935,boat,55,283,439,905
2014-08-01-15-06_00040-eo_img0_00048_1406930694656165.png,864,935,boat,190,63,742,878
2014-09-09-14-57_00105-eo_img0_00122_1410299570911558.png,1675,261,ship,79,3,1675,255
2014-09-08-15-30_00045-eo_img0_00048_1410215097437292.png,334,55,ship,4,6,328,49
2014-09-02-18-13_00105-eo_img0_00193_1409706518944237.png,218,262,boat,13,15,201,246
2014-08-01-20-00_00005-eo_img0_00017_1406948306504854.png,200,60,ship,10,10,194,52
2014-09-08-16-47_00085-eo_img0_00100_1410219870965008.png,760,369,ship,26,26,736,347
2014-07-22-13-47_00075-eo_img0_00098_1406061766784390.png,270,162,boat,19,17,261,146
2014-07-17-14-08_00145-eo_img0_00176_1405631238812676.png,554,199,boat,24,30,462,186
2014-09-09-16-35_00025-eo_img0_00029_1410305588419090.png,527,378,boat,42,36,499,328
2014-08-01-17-28_00025-eo_img0_00028_1406937487222633.png,111,79,boat,11,34,105,72
2014-07-17-13-16_00475-eo_img0_00629_1405627946084201.png,652,198,boat,10,20,637,183
2014-09-15-17-35_00005-eo_img0_00005_1410827724864540.png,1172,387,boat,37,137,1122,358
2014-07-17-18-12_00750-eo_img0_01076_1405645736055775.png,126,59,boat,6,8,122,53
2014-09-09-17-09_00715-eo_img0_00847_1410307299729567 - Copy.png,314,50,ship,6,5,302,43
2014-08-01-16-52_00125-eo_img0_00158_1406936653021660 (2).png,75,41,ship,8,11,71,33
2014-07-17-17-28_00035-eo_img0_00042_1405642252156001.png,322,87,ship,4,12,316,87
2014-09-02-18-05_00290-eo_img0_00359_1409706187644969.png,147,60,ship,12,16,141,52
2014-07-17-17-28_00090-eo_img0_00109_1405642313103325.png,354,82,ship,11,11,351,80
2014-09-15-17-13_00300-eo_img0_00391_1410826397112356.png,548,192,boat,146,42,350,178
2014-07-17-13-31_00005-eo_img0_00004_1405628902189537.png,185,87,boat,22,28,185,79
```

```
filename,width,height,class,xmin,ymin,xmax,ymax
scene00013 (3).png,1920,1088,boat,607,594,1652,880
scene00013 (3).png,1920,1088,boat,630,534,735,672
scene00013 (3).png,1920,1088,ship,123,349,507,487
scene00013 (3).png,1920,1088,ship,673,399,821,475
scene00013 (3).png,1920,1088,ship,1309,453,1719,532
scene00013 (3).png,1920,1088,ship,1228,465,1304,499
scene00013 (3).png,1920,1088,ship,600,432,685,465
scene00013 (11).png,1920,1088,ship,457,365,954,482
scene00013 (11).png,1920,1088,ship,1164,365,1535,468
scene00013 (11).png,1920,1088,boat,1785,311,1900,492
scene00013 (11).png,1920,1088,boat,116,442,171,458
scene00013 (11).png,1920,1088,boat,209,430,247,470
scene00013 (11).png,1920,1088,ship,2,432,107,461
scene00001 (7).png,1920,1088,ship,1809,115,1916,368
scene00001 (7).png,1920,1088,ship,57,299,354,353
scene00001 (7).png,1920,1088,ship,9,201,57,375
scene00169.png,1920,1088,boat,633,951,757,1011
scene00169.png,1920,1088,boat,169,930,221,956
scene00169.png,1920,1088,boat,1704,927,1802,956
scene00169.png,1920,1088,boat,257,913,380,951
scene00169.png,1920,1088,ship,40,844,223,949
scene00169.png,1920,1088,ship,942,806,1476,963
scene00169.png,1920,1088,ship,680,892,902,949
scene00169.png,1920,1088,ship,509,894,642,939
scene00169.png,1920,1088,ship,1602,882,1711,953
scene00169.png,1920,1088,ship,1873,887,1920,940
scene00169.png,1920,1088,boat,1545,896,1600,942
scene00253 (2).png,1920,1088,boat,507,365,735,427
scene00253 (2).png,1920,1088,ship,50,246,497,387
scene00253 (2).png,1920,1088,ship,1319,311,1795,432
scene00253 (2).png,1920,1088,ship,1769,292,1916,489
scene00253 (2).png,1920,1088,ship,507,299,1100,420
scene00253 (2).png,1920,1088,ship,1088,382,1190,413
scene00121.png,1920,1088,boat,109,606,207,763
scene00121.png,1920,1088,boat,164,532,252,577
scene00121.png,1920,1088,boat,904,532,990,582
scene00121.png,1920,1088,ship,680,422,1161,508
scene00121.png,1920,1088,ship,261,442,595,515
scene00121.png,1920,1088,ship,1578,449,1788,503
scene00013 (7).png,1920,1088,ship,297,506,519,582
scene00013 (7).png,1920,1088,ship,11,511,147,558
scene00013 (7).png,1920,1088,ship,523,532,926,608
scene00013 (7).png,1920,1088,ship,926,487,1488,618
scene00013 (7).png,1920,1088,ship,1492,572,1611,620
scene00013 (7).png,1920,1088,ship,1676,584,1854,620
scene00049.png,1920,1088,ship,1766,577,1892,639
scene00049.png,1920,1088,ship,1588,580,1780,620
scene00049.png,1920,1088,ship,1273,563,1500,613
scene00049.png,1920,1088,ship,1121,565,1259,594
scene00049.png,1920,1088,ship,938,561,1083,599
scene00049.png,1920,1088,ship,42,546,311,587
scene00049.png,1920,1088,ship,316,546,626,589
scene00049.png,1920,1088,ship,621,558,714,589
scene00049 (2).png,1920,1088,ship,4,556,126,618
```

Figure 3.5: *Train and Test CSV files*

Figure 3.5 displays the CSV files which were created for train labeled data and test labeled data. Each CSV file contains information for all labeled images. It encompassed filename, width, height, class, xminimum, yminimum, xmaximum and ymaximum.

3.2.3 Converting CSV files to TFRecords

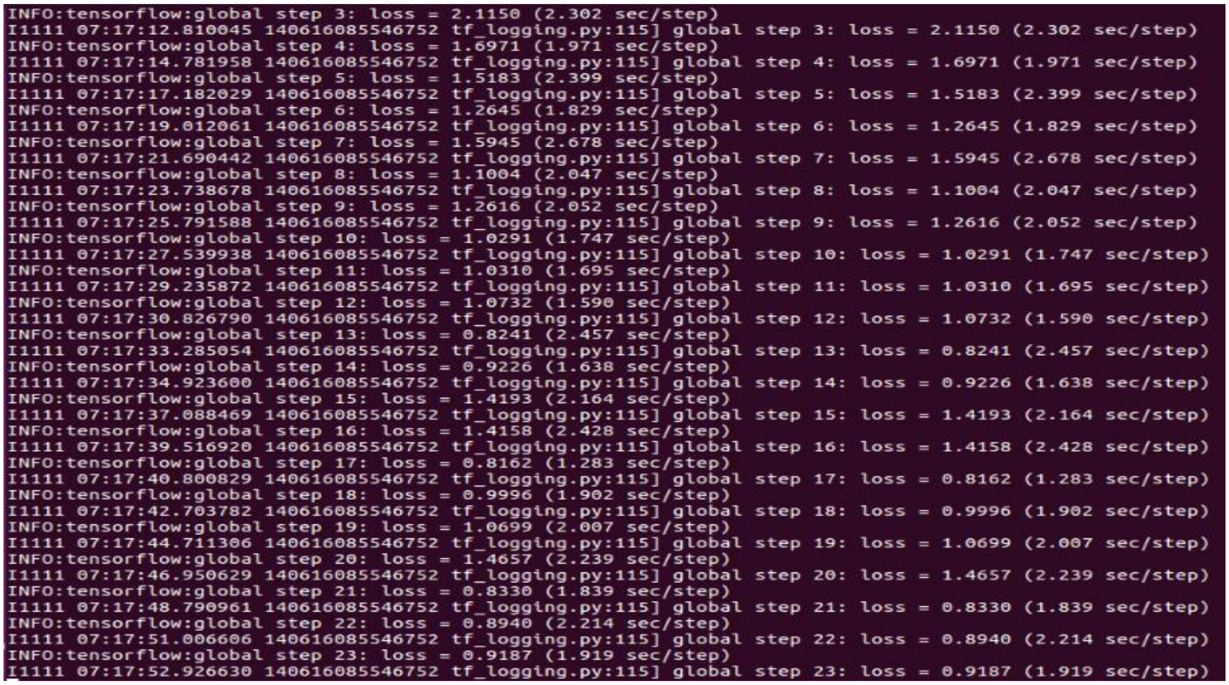
Tesnsorflow records were generated using python3 and tensorflow. During the conversion from CSV files to TFRecords, train labels were taken as input to generate train TFRecords and test label was taken as input to generate test TFRecords. Image width, image height, filename, image source id, image format, image bounding box xminimum, image bounding box xmaximum, image bounding box yminimum, image bounding box ymaximum, and image class label were considered during conversion. The train TFRecords and test TFRecords were generated as output and were put in the same folder as train and test labels.

3.2.4 Training Process

In this study supervised learning is utilized as a machine learning method for object detection. Faster R-CNN with inception v2, configured in MSCOCO dataset was utilized to detect two object classes, boat, and ship. The image resizer was set to keep the aspect ratio resizer at a minimum dimension of 600 and 900 for maximum dimension to accommodate all images with varying sizes. The first stage features stride, it was set to 16. First stage maximum proposals were set to 300, localization loss weight was set to 2.0 for the first stage, objectness loss weight was set to 1.0 for the first stage, initial crop size was set to 14, max-pool kernel size was set to 2 and the max-pool stride was set to 2. Maximum detections per class were set to 100 and total detections were set to 300.

SOFTMAX was utilized as a score converter. Localization loss weight for the first stage was set to 2.0 and classification loss weight for the second stage was 1.0. The batch size was set to 1 because the CPU used for this research was incapable of taking a large batch size at a time. Initial learning rate was set to 0.0002 and momentum optimizer value to 0.9. The model was set to train until the training process reached global step 200000. It was believed that at this stage the model would have fully learned. The TFRecords and label map were utilized as input. Each sample data was trained using the above configuration settings.

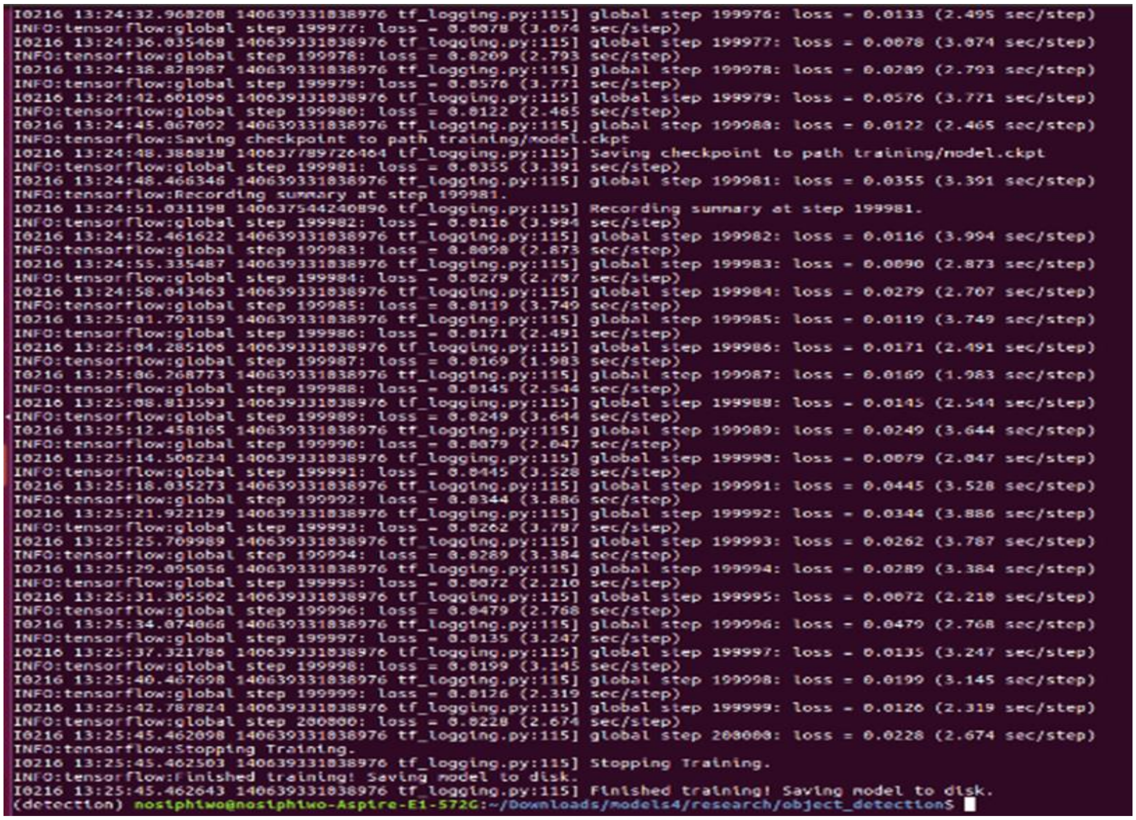
A virtual environment called Detection was created ubuntu 16 machine terminal. Tensorflow was installed in this environment and all the training process were run in the same environment using python3. Figure 3.6 represents an overview of the training process.



```
INFO:tensorflow:global step 3: loss = 2.1150 (2.302 sec/step)
I1111 07:17:12.810045 140616085546752 tf_logging.py:115] global step 3: loss = 2.1150 (2.302 sec/step)
INFO:tensorflow:global step 4: loss = 1.6971 (1.971 sec/step)
I1111 07:17:14.781958 140616085546752 tf_logging.py:115] global step 4: loss = 1.6971 (1.971 sec/step)
INFO:tensorflow:global step 5: loss = 1.5183 (2.399 sec/step)
I1111 07:17:17.182029 140616085546752 tf_logging.py:115] global step 5: loss = 1.5183 (2.399 sec/step)
INFO:tensorflow:global step 6: loss = 1.2645 (1.829 sec/step)
I1111 07:17:19.012061 140616085546752 tf_logging.py:115] global step 6: loss = 1.2645 (1.829 sec/step)
INFO:tensorflow:global step 7: loss = 1.5945 (2.678 sec/step)
I1111 07:17:21.690442 140616085546752 tf_logging.py:115] global step 7: loss = 1.5945 (2.678 sec/step)
INFO:tensorflow:global step 8: loss = 1.1004 (2.047 sec/step)
I1111 07:17:23.738678 140616085546752 tf_logging.py:115] global step 8: loss = 1.1004 (2.047 sec/step)
INFO:tensorflow:global step 9: loss = 1.2616 (2.052 sec/step)
I1111 07:17:25.791588 140616085546752 tf_logging.py:115] global step 9: loss = 1.2616 (2.052 sec/step)
INFO:tensorflow:global step 10: loss = 1.0291 (1.747 sec/step)
I1111 07:17:27.539938 140616085546752 tf_logging.py:115] global step 10: loss = 1.0291 (1.747 sec/step)
INFO:tensorflow:global step 11: loss = 1.0310 (1.695 sec/step)
I1111 07:17:29.235872 140616085546752 tf_logging.py:115] global step 11: loss = 1.0310 (1.695 sec/step)
INFO:tensorflow:global step 12: loss = 1.0732 (1.590 sec/step)
I1111 07:17:30.826790 140616085546752 tf_logging.py:115] global step 12: loss = 1.0732 (1.590 sec/step)
INFO:tensorflow:global step 13: loss = 0.8241 (2.457 sec/step)
I1111 07:17:33.285054 140616085546752 tf_logging.py:115] global step 13: loss = 0.8241 (2.457 sec/step)
INFO:tensorflow:global step 14: loss = 0.9226 (1.638 sec/step)
I1111 07:17:34.923600 140616085546752 tf_logging.py:115] global step 14: loss = 0.9226 (1.638 sec/step)
INFO:tensorflow:global step 15: loss = 1.4193 (2.164 sec/step)
I1111 07:17:37.088469 140616085546752 tf_logging.py:115] global step 15: loss = 1.4193 (2.164 sec/step)
INFO:tensorflow:global step 16: loss = 1.4158 (2.428 sec/step)
I1111 07:17:39.516920 140616085546752 tf_logging.py:115] global step 16: loss = 1.4158 (2.428 sec/step)
INFO:tensorflow:global step 17: loss = 0.8162 (1.283 sec/step)
I1111 07:17:40.800829 140616085546752 tf_logging.py:115] global step 17: loss = 0.8162 (1.283 sec/step)
INFO:tensorflow:global step 18: loss = 0.9996 (1.902 sec/step)
I1111 07:17:42.703782 140616085546752 tf_logging.py:115] global step 18: loss = 0.9996 (1.902 sec/step)
INFO:tensorflow:global step 19: loss = 1.0699 (2.007 sec/step)
I1111 07:17:44.711306 140616085546752 tf_logging.py:115] global step 19: loss = 1.0699 (2.007 sec/step)
INFO:tensorflow:global step 20: loss = 1.4657 (2.239 sec/step)
I1111 07:17:46.950629 140616085546752 tf_logging.py:115] global step 20: loss = 1.4657 (2.239 sec/step)
INFO:tensorflow:global step 21: loss = 0.8330 (1.839 sec/step)
I1111 07:17:48.790961 140616085546752 tf_logging.py:115] global step 21: loss = 0.8330 (1.839 sec/step)
INFO:tensorflow:global step 22: loss = 0.8940 (2.214 sec/step)
I1111 07:17:51.006606 140616085546752 tf_logging.py:115] global step 22: loss = 0.8940 (2.214 sec/step)
INFO:tensorflow:global step 23: loss = 0.9187 (1.919 sec/step)
I1111 07:17:52.926630 140616085546752 tf_logging.py:115] global step 23: loss = 0.9187 (1.919 sec/step)
```

Figure 3.6: *training process at lower steps*

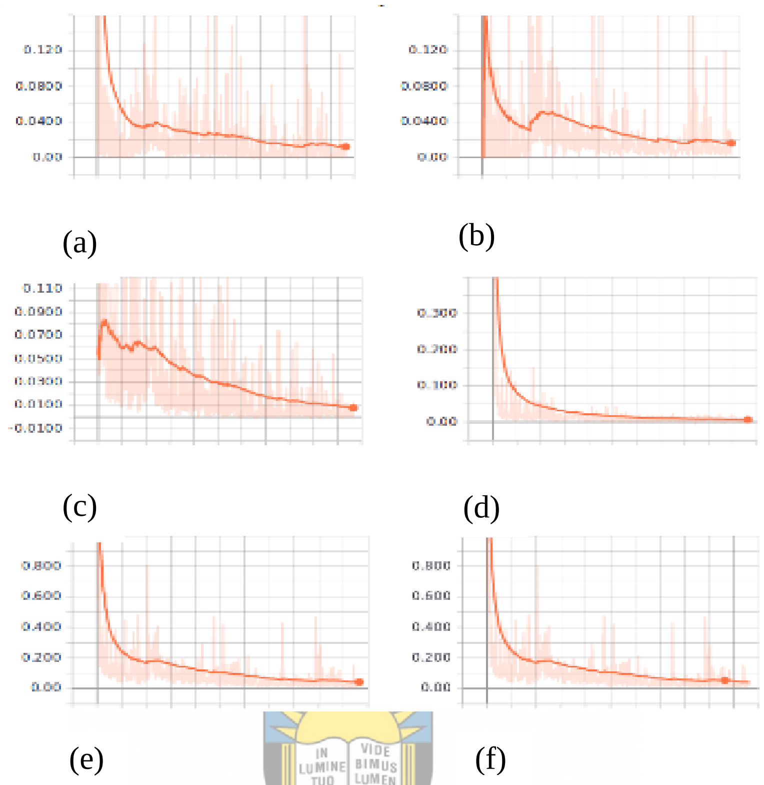
Figure 3.6 represents the training process during the first few minutes. The training process started at global step 1 to global step 200000. While the training process was running, it frecorded the time, global step number, the loss and the number of seconds it took for each step to run. Figure 3.7 displays the training process at 200000 global steps.



```
I0216 13:24:32.968208 140639331838976 tf_logging.py:115] global step 199976: loss = 0.0133 (2.495 sec/step)
INFO:tensorflow:global step 199977: loss = 0.0078 (3.074 sec/step)
I0216 13:24:36.035468 140639331838976 tf_logging.py:115] global step 199977: loss = 0.0078 (3.074 sec/step)
INFO:tensorflow:global step 199978: loss = 0.0289 (2.793 sec/step)
I0216 13:24:38.828987 140639331838976 tf_logging.py:115] global step 199978: loss = 0.0289 (2.793 sec/step)
INFO:tensorflow:global step 199979: loss = 0.0576 (3.771 sec/step)
I0216 13:24:42.601096 140639331838976 tf_logging.py:115] global step 199979: loss = 0.0576 (3.771 sec/step)
INFO:tensorflow:global step 199980: loss = 0.0122 (2.465 sec/step)
I0216 13:24:45.067092 140639331838976 tf_logging.py:115] global step 199980: loss = 0.0122 (2.465 sec/step)
INFO:tensorflow:Saving checkpoint to path training/model.ckpt
I0216 13:24:48.386838 140637789726464 tf_logging.py:115] Saving checkpoint to path training/model.ckpt
INFO:tensorflow:global step 199981: loss = 0.0355 (3.391 sec/step)
I0216 13:24:48.466346 140639331838976 tf_logging.py:115] global step 199981: loss = 0.0355 (3.391 sec/step)
INFO:tensorflow:Recording summary at step 199981.
I0216 13:24:51.031198 140637544240896 tf_logging.py:115] Recording summary at step 199981.
INFO:tensorflow:global step 199982: loss = 0.0116 (3.994 sec/step)
I0216 13:24:52.461622 140639331838976 tf_logging.py:115] global step 199982: loss = 0.0116 (3.994 sec/step)
INFO:tensorflow:global step 199983: loss = 0.0090 (2.873 sec/step)
I0216 13:24:55.335487 140639331838976 tf_logging.py:115] global step 199983: loss = 0.0090 (2.873 sec/step)
INFO:tensorflow:global step 199984: loss = 0.0279 (2.707 sec/step)
I0216 13:24:58.043463 140639331838976 tf_logging.py:115] global step 199984: loss = 0.0279 (2.707 sec/step)
INFO:tensorflow:global step 199985: loss = 0.0119 (3.749 sec/step)
I0216 13:25:01.793159 140639331838976 tf_logging.py:115] global step 199985: loss = 0.0119 (3.749 sec/step)
INFO:tensorflow:global step 199986: loss = 0.0171 (2.491 sec/step)
I0216 13:25:04.285106 140639331838976 tf_logging.py:115] global step 199986: loss = 0.0171 (2.491 sec/step)
INFO:tensorflow:global step 199987: loss = 0.0169 (1.983 sec/step)
I0216 13:25:06.268773 140639331838976 tf_logging.py:115] global step 199987: loss = 0.0169 (1.983 sec/step)
INFO:tensorflow:global step 199988: loss = 0.0145 (2.544 sec/step)
I0216 13:25:08.813593 140639331838976 tf_logging.py:115] global step 199988: loss = 0.0145 (2.544 sec/step)
INFO:tensorflow:global step 199989: loss = 0.0249 (3.644 sec/step)
I0216 13:25:12.458165 140639331838976 tf_logging.py:115] global step 199989: loss = 0.0249 (3.644 sec/step)
INFO:tensorflow:global step 199990: loss = 0.0079 (2.047 sec/step)
I0216 13:25:14.506234 140639331838976 tf_logging.py:115] global step 199990: loss = 0.0079 (2.047 sec/step)
INFO:tensorflow:global step 199991: loss = 0.0445 (3.528 sec/step)
I0216 13:25:18.035273 140639331838976 tf_logging.py:115] global step 199991: loss = 0.0445 (3.528 sec/step)
INFO:tensorflow:global step 199992: loss = 0.0344 (3.886 sec/step)
I0216 13:25:21.922129 140639331838976 tf_logging.py:115] global step 199992: loss = 0.0344 (3.886 sec/step)
INFO:tensorflow:global step 199993: loss = 0.0262 (3.787 sec/step)
I0216 13:25:25.709989 140639331838976 tf_logging.py:115] global step 199993: loss = 0.0262 (3.787 sec/step)
INFO:tensorflow:global step 199994: loss = 0.0289 (3.384 sec/step)
I0216 13:25:29.095056 140639331838976 tf_logging.py:115] global step 199994: loss = 0.0289 (3.384 sec/step)
INFO:tensorflow:global step 199995: loss = 0.0072 (2.210 sec/step)
I0216 13:25:31.305502 140639331838976 tf_logging.py:115] global step 199995: loss = 0.0072 (2.210 sec/step)
INFO:tensorflow:global step 199996: loss = 0.0479 (2.768 sec/step)
I0216 13:25:34.074066 140639331838976 tf_logging.py:115] global step 199996: loss = 0.0479 (2.768 sec/step)
INFO:tensorflow:global step 199997: loss = 0.0135 (3.247 sec/step)
I0216 13:25:37.321786 140639331838976 tf_logging.py:115] global step 199997: loss = 0.0135 (3.247 sec/step)
INFO:tensorflow:global step 199998: loss = 0.0199 (3.145 sec/step)
I0216 13:25:40.467698 140639331838976 tf_logging.py:115] global step 199998: loss = 0.0199 (3.145 sec/step)
INFO:tensorflow:global step 199999: loss = 0.0126 (2.319 sec/step)
I0216 13:25:42.787824 140639331838976 tf_logging.py:115] global step 199999: loss = 0.0126 (2.319 sec/step)
INFO:tensorflow:global step 200000: loss = 0.0228 (2.674 sec/step)
I0216 13:25:45.462098 140639331838976 tf_logging.py:115] global step 200000: loss = 0.0228 (2.674 sec/step)
INFO:tensorflow:Stopping Training.
I0216 13:25:45.462503 140639331838976 tf_logging.py:115] Stopping Training.
INFO:tensorflow:Finished training! Saving model to disk.
I0216 13:25:45.462643 140639331838976 tf_logging.py:115] Finished training! Saving model to disk.
(detection) nosiphiwo@nosiphiwo-Aspire-E1-572G:~/Downloads/models4/research/object_detection$
```

Figure 3.7: *Training process at 200000 global step*

The training process took seven days until it reached global step 200000. It was finished after 200000 global steps and the model was saved to disk. During and after the training phase, the performance of the training was recorded in the tensorboard. The following figures illustrates the losses for each model. Figure 3.8 shows losses for a model trained with 100 labeled images, figure 3.9 presents losses for a model trained with 300 labeled images, figure 3.10 shows losses for a model trained with 500 labeled images, figure 3.11 presents losses for a model trained with 700 labelled images and figure 3.12 shows losses for a model trained with 900 trained images.

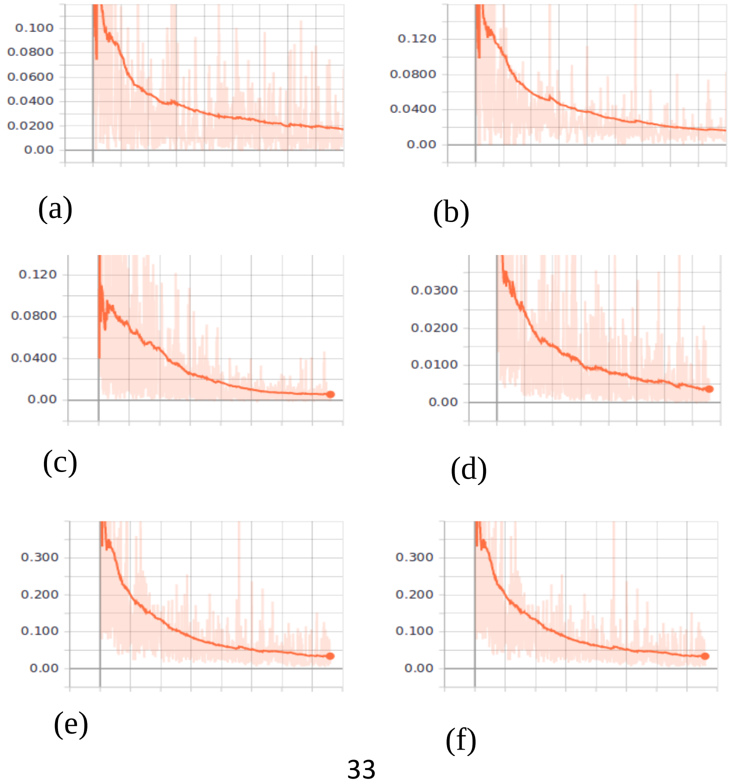


(a) (b)

(c) (d)

(e) (f)

Figure 3.8: *Losses for sample 1: (a) Classification loss, (b) Localization loss, (c) Localization loss (d) Objectness loss, (e) Total loss and (f) Clone loss*

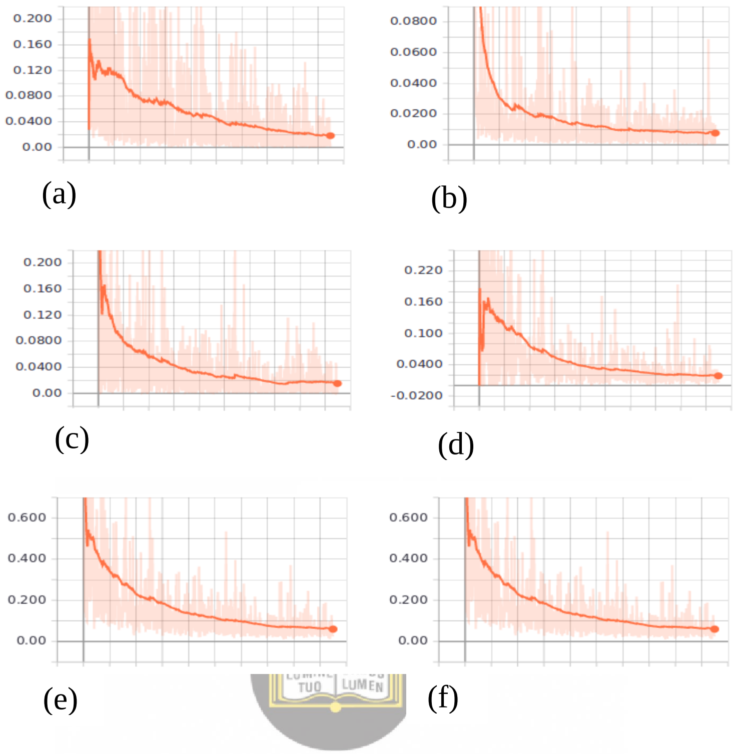


(a) (b)

(c) (d)

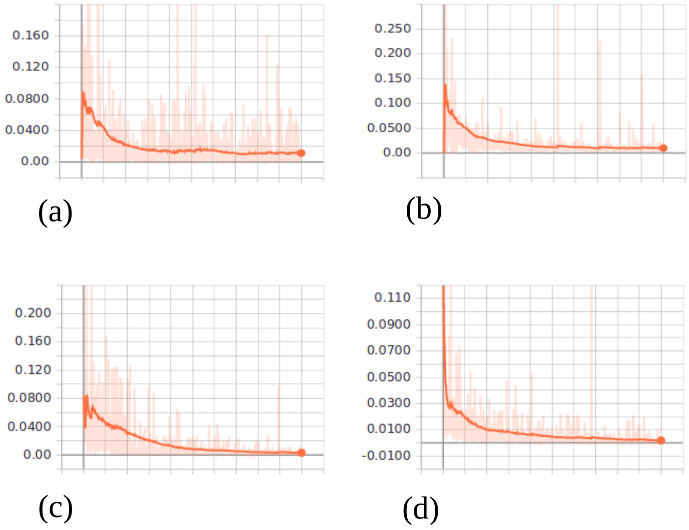
(e) (f)

Figure 3.9: *Losses for sample 2: (a) Classification loss, (b) Localization loss, (c) Localization loss (d) Objectness loss, (e) Total loss and (f) Clone loss*

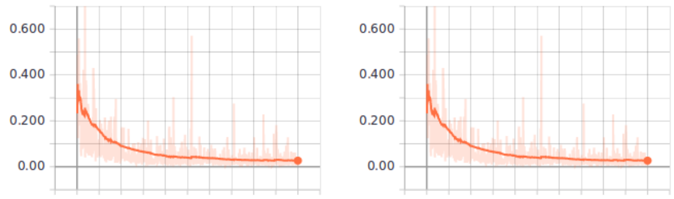


(a) (b) (c) (d) (e) (f)

Figure 3.10: *Losses for sample 3: (a) Classification loss, (b) Localization loss, (c) Localization loss (d) Objectness loss, (e) Total loss and (f) Clone loss*

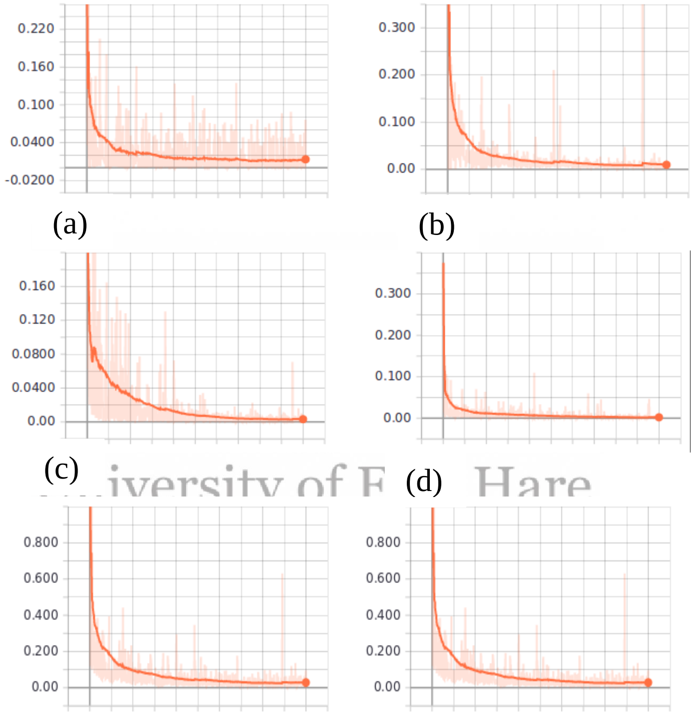


(a) (b) (c) (d)



(e) (f)

Figure 3.11*: Losses for sample 4: (a) Classification loss, (b) Localization loss, (c) Localization loss (d) Objectness loss, (e) Total loss and (f) Clone loss*



(a) (b)

(c) (d)

(e) (f)

Figure 3.12: *Losses for sample 5: (a) Classification loss, (b) Localization loss, (c) Localization loss (d) Objectness loss, (e) Total loss and (f) Clone loss*

A loss function is a positive value that determines the irregularity between the detected value and the true value. Figure 3.8 to 3.12 shows classification loss, localization loss, objectness loss, clone loss and total loss after the training process using increasing sample size. The graphs show that all losses were decreasing as the model continued the training process. When the loss is decreasing or small, the robustness of the model is improved. When the total loss is less than 1, it means that the model has fully trained, but in this research it was trained up to global step 200000 to further improve its accuracy.

3.2.5 Testing process

The saved models were tested using python3 on a notebook. Each model was tested using the same dataset to track the performance of the models based on sample size. The test data contained 20 images and the output was recorded as results. The results were analyzed and the information was recorded in the form of tables and graphs.

3.3 Conclusion

This chapter has explained the methodology of the study. It outlined how data was collected, converted and utilized to train faster R-CNN with inception v2 model. The chapter has also encompassed experimental design, explaining more on the approach followed in computing the experiments. The performance of the training process in terms of losses was also presented graphically from sample 1 to sample 5 and the evaluation process was explained. The results will be represented in the next chapter.

4 Chapter four: Findings

In this study transfer learning was conducted on faster R-CNN model with inception v2 which was pre-trained on COCO dataset. This model was trained on five different samples of data varying in size. the results of the experiments of the experiments were presented in this chapter. It further presents how detection performance was evaluated. Results for each train sample size are described using tables and mapped in graphs and false discovery rate (FDR), precision and recall are computed and presented in a graph.

4.1 Sample size

As stated in chapter 3, the training process was conducted for five sample data. sample one contained 100 train images, sample two contained 300 train images, sample three contained 500 train images, sample four contained 700 train images and sample five contained 900 train images. During testing phase, the same set of data was used to test these models which were trained with different sample data to track their performance. The data set contained 20 images denoted as image1 up to image20. The following subsections present the results based on the five sample data.

Sample one to sample five present the results obtained after evaluating the faster R-CNN model with inception v2 using different sample size. The data was increased for each testing phase. Figure 5.1 to figure 5.5 from the samples display 20 images with bounding boxes. Each Bounding box from each image represents the detection instance. The green bounding box denotes the ship instances and the blue bounding box denotes the boat instances. The subsections below further present the interpretation of results in the form of tables.

Table 5.1 to table 5.5 illustrate the detection performance in each image. Each image was observed to identify the number of detections, false negatives, false positives and true positives. Image no depicts the order of images from image one to image twenty, total objects depicts the number of objects present in each image, total detections portrays the number of detections in each image, false positives denotes false detection instances, true positives denotes correct detection instances and false negative depicts the number of objects not detected.

4.1.1 Sample one

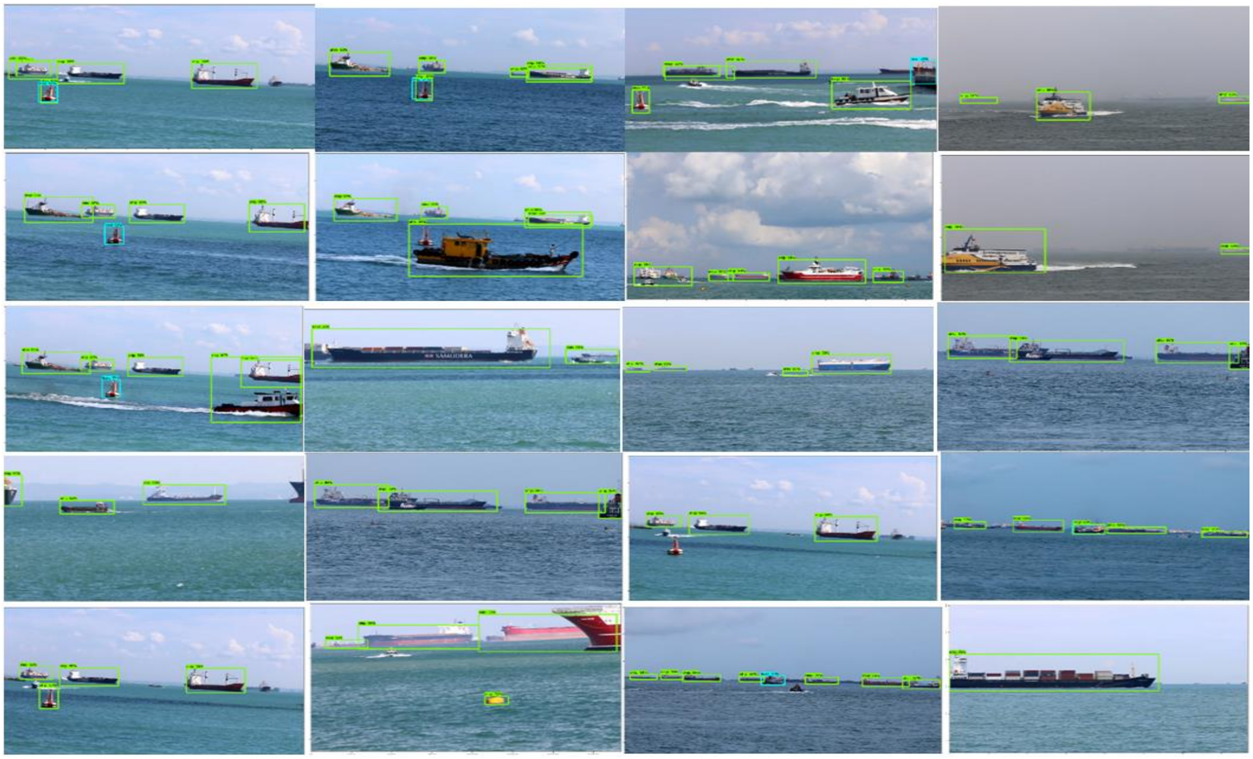


Figure 4.1: *Results of the model trained with 100 images.*

Table 4.1: *Detection performance on the model trained with 100 train images*

Image no	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*
Objects	8	7	8	4	5	5	8	4	6	4	4	5	4	5	8	7	8	5	11	5
Detections	5	6	5	3	5	4	5	2	6	2	4	4	3	4	3	6	4	4	8	1
FP	1	1	3	0	0	1	1	0	1	1	1	0	1	0	0	1	1	2	1	0
TP	4	5	2	3	5	3	4	2	5	1	3	4	2	4	3	5	3	2	7	1
FN	3	2	3	1	0	1	4	2	0	2	1	1	1	1	5	2	4	1	3	4

4.1.2 Sample two

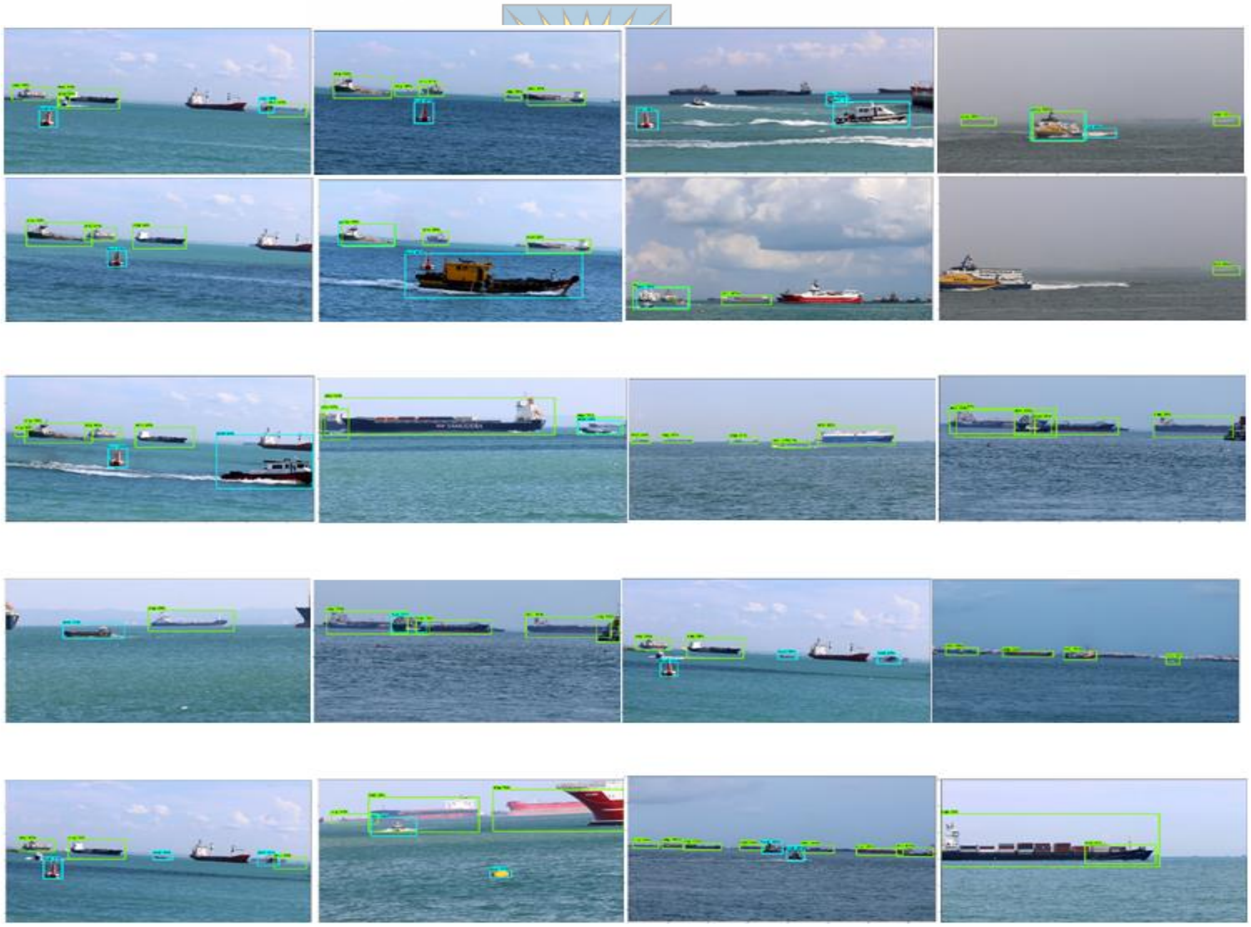


Figure 4.2: *Results of the model trained with 300 train images.*

Table 4.2: *Detection performance on the model trained with 300 train images*

Image no	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*
Objects	8	7	8	4	5	5	8	4	6	4	4	5	4	5	8	7	8	5	11	5
Detections	6	6	3	5	4	5	3	1	6	4	7	4	2	4	5	4	6	5	9	1
FP	2	0	0	2	0	1	2	0	0	1	3	0	0	0	1	1	1	0	1	0
TP	4	6	3	3	4	4	1	1	6	3	4	4	2	4	4	3	5	5	8	1
FN	2	1	5	1	1	1	5	3	1	1	0	1	2	0	2	3	3	1	2	3

4.1.3 Sample three

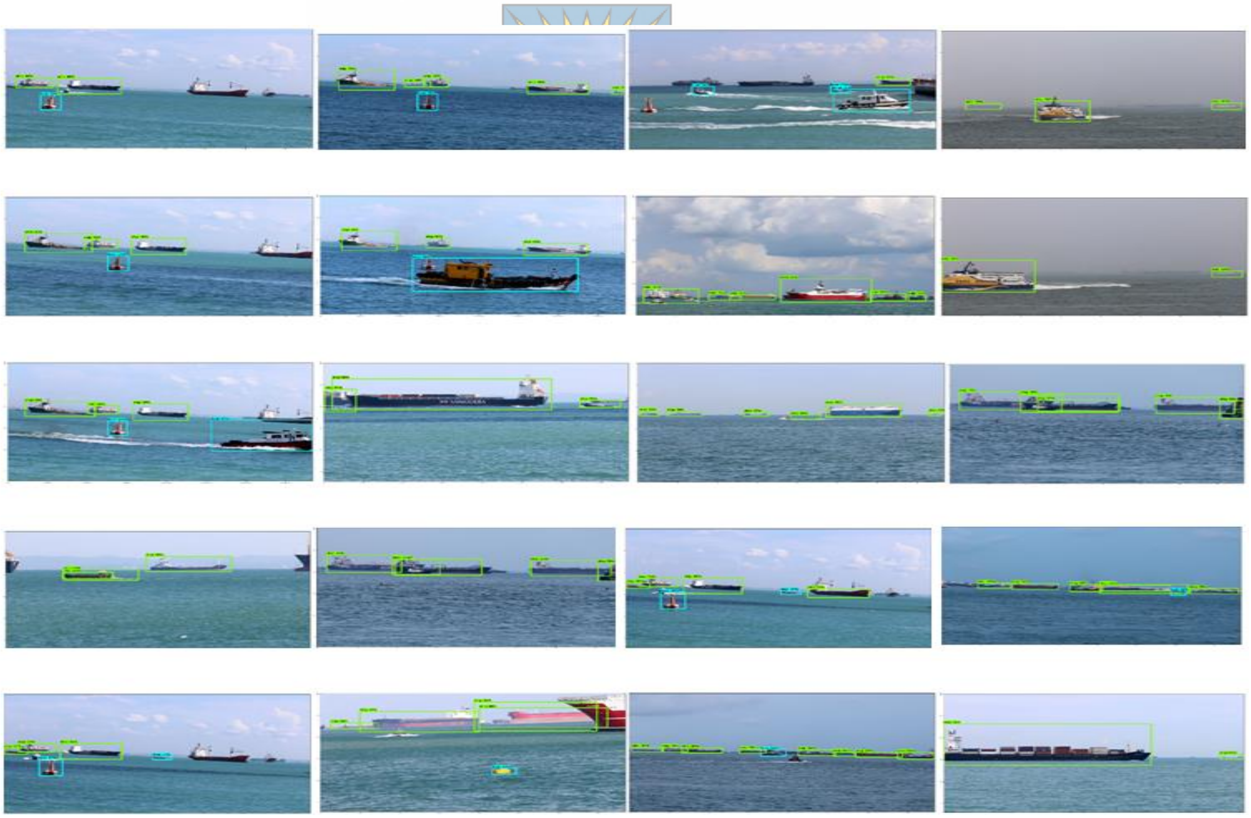


Figure 4.3: *Results of the model trained with 500 train images.*

Table 4.3: *Detection performance on the model trained with 500 train images*

Image no	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*
Objects	8	7	8	4	5	5	8	4	6	4	4	5	4	5	8	7	8	5	11	5
Detections	3	6	4	3	4	4	6	1	5	3	6	4	2	5	6	6	5	4	9	2
FP	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1	1	0
TP	3	6	4	3	4	4	5	1	5	2	5	4	1	5	6	5	5	3	8	2
FN	5	1	4	1	1	1	2	2	1	1	1	0	2	0	2	1	3	2	1	3

4.1.4 Sample four

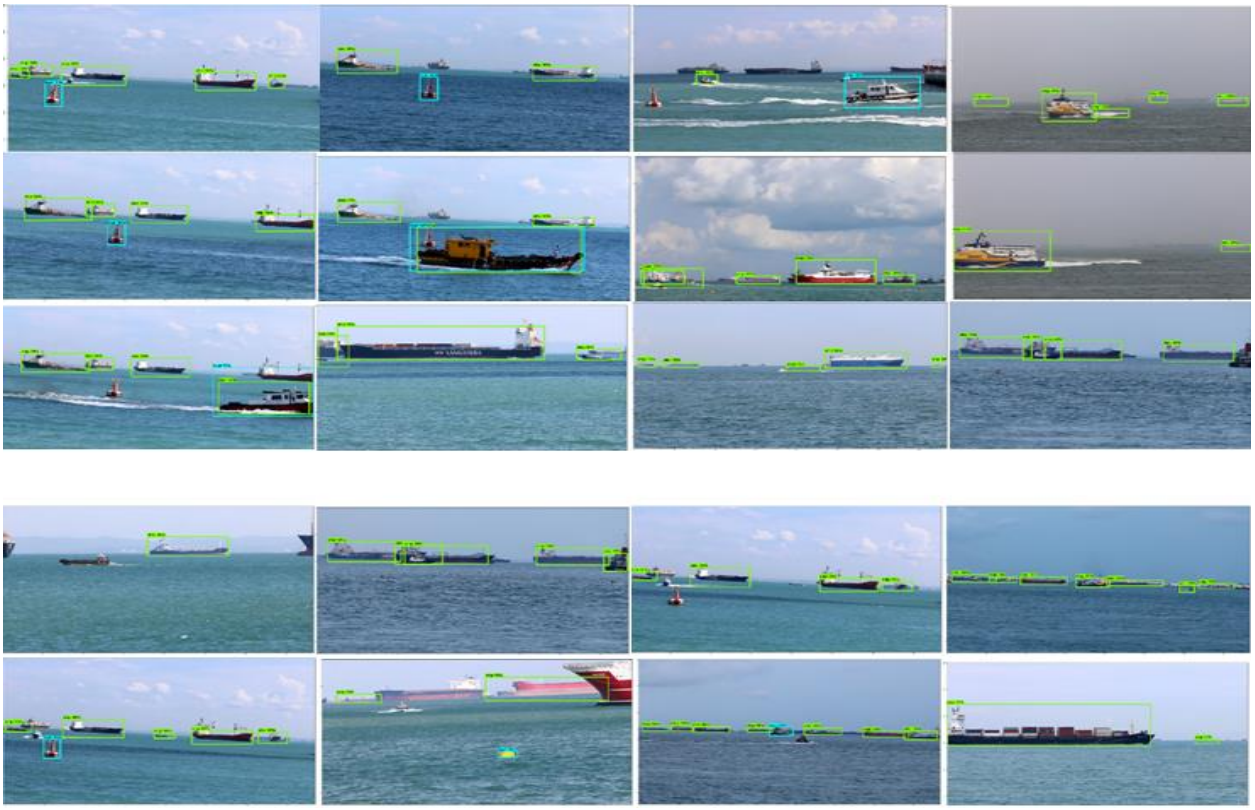


Figure 4.4: *Results of the model trained with 700 train images.*

Table 4.4: *Detection performance on the model trained with 700 train images*

Image no	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*
Objects	8	7	8	4	5	5	8	4	6	4	4	5	4	5	8	7	8	5	11	5
Detections	6	3	2	5	5	4	4	2	5	3	5	4	1	5	4	6	6	3	8	2
FP	0	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	1	1	0
TP	6	3	1	4	5	3	3	2	4	2	4	4	1	5	4	5	5	2	7	2
FN	2	4	6	1	0	2	4	2	2	1	2	1	3	0	4	0	2	4	1	3

4.1.5 Sample five

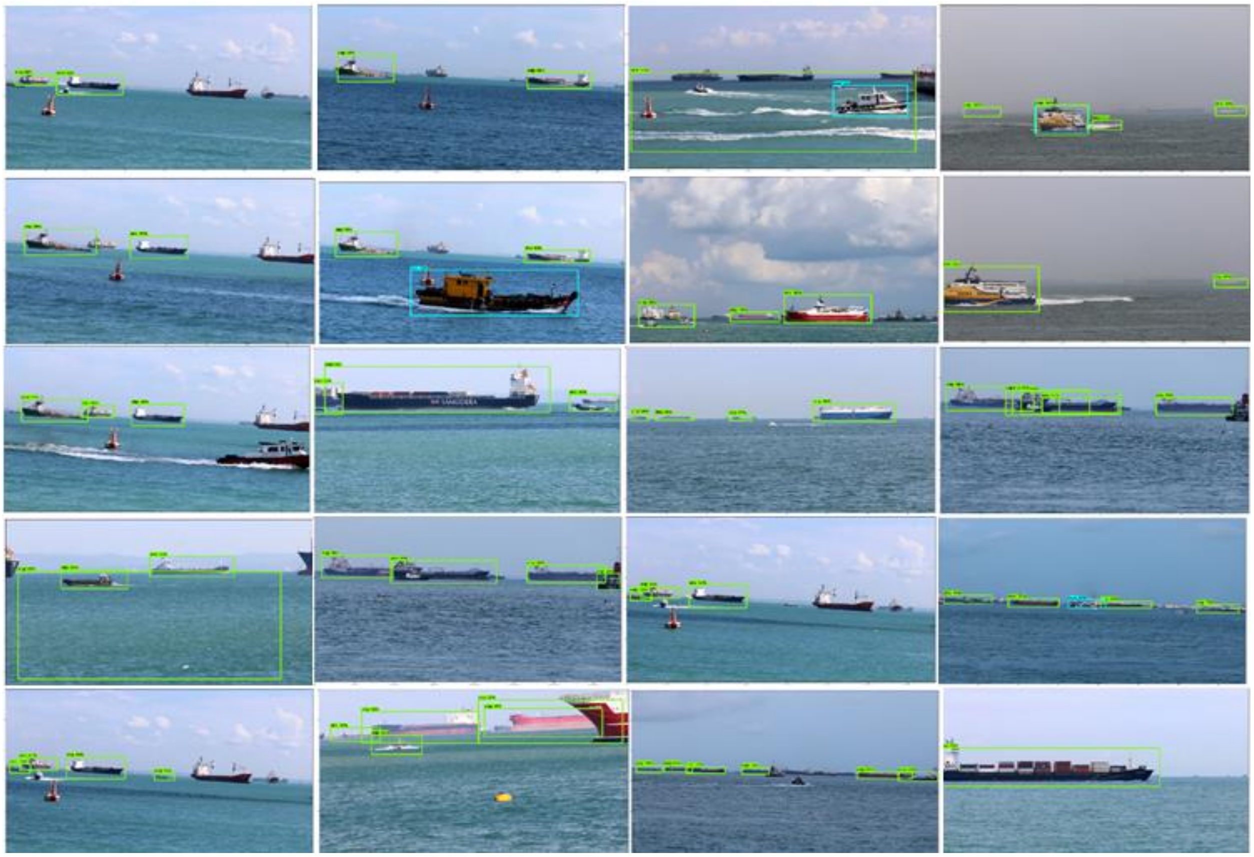


Figure 4.5*: Results of the model trained with 900 images.*

Table 4.5: *Detection performance on the model trained with 900 images*

Image no	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*
Objects	8	7	8	4	5	5	8	4	6	4	4	5	4	5	8	7	8	5	11	5
Detections	2	2	1	4	2	3	3	2	3	4	4	4	4	4	3	5	4	4	6	1
FP	0	0	1	2	0	0	1	0	0	1	0	0	2	0	0	0	1	1	0	0
TP	2	2	1	2	2	3	2	2	3	3	4	4	2	4	3	5	3	3	0	1
FN	6	4	6	1	3	3	5	3	3	1	1	1	2	1	5	2	4	1	5	4

4.2 Performance evaluation

The detection performance of the proposed faster R-CNN was evaluated in testing phase. The test data contained 20 images with 121 objects in total. The same test data was used to test five models trained with different sample size. This study made use of binary classification method. The Confusion matrix was utilized to present the binary classification of results. The confusion matrix contains the possible outcomes which describe the detection performance. It depicts actual instances and predicted instances. The following table shows the confusion matrix.

Table 4.6: *Confusion matrix*

Actual class	Predicted class	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

- True Positive(TP): a boat instance was correctly detected and classified as a boat or a ship instance was correctly detected as a ship.

- True Negative(TN): a non-boat instance was correctly classified as non-boat instance or a non-ship instance was correctly classified as not ship instance.
- False Positive(FP): a non-boat instance was incorrectly detected and classified as a boat or a non-ship instance was incorrectly detected and classified as a ship.
- False Negative(FN): a boat instance was incorrectly classified as non-boat or a ship instance was classified as non-ship.

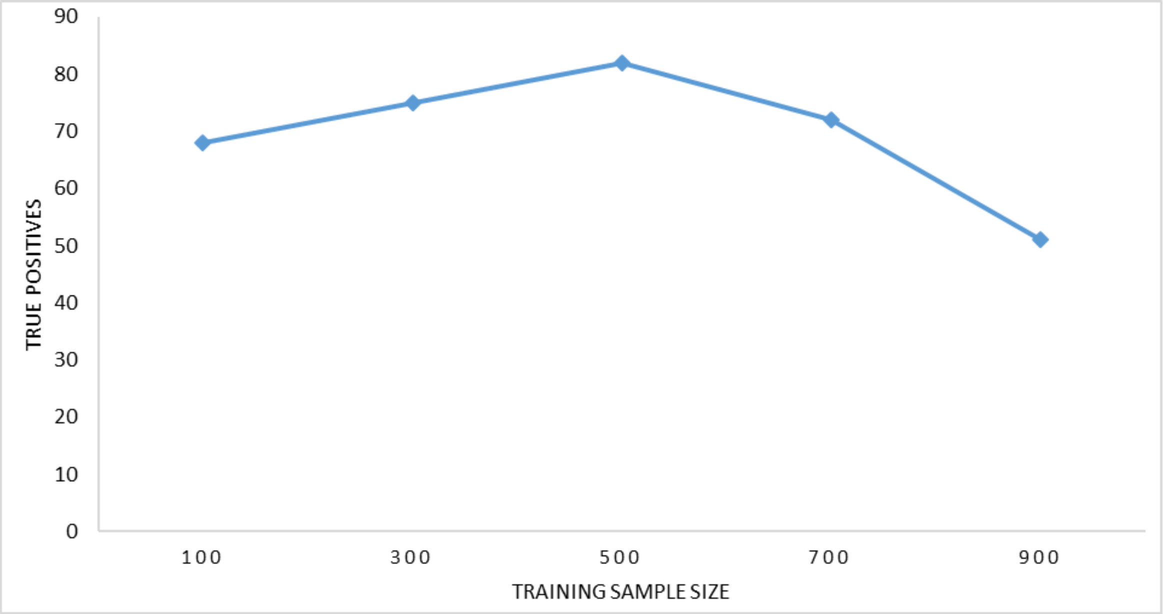


Figure 4.6: *True positives based on train sample size*

Figure 4.6 shows the true positives based on increasing train sample size. The line graph shows that when the train sample size was 100 the number of true detections were 68, the number of true detections was 75 when the sample size was increased to 300 and when the data was further increased to 500 train images it jumped to 82 true detections. After 500 sample size graphs started to decrease in true detections due to the duplication of data. This means that by increasing sample size the number of true detections also increases.

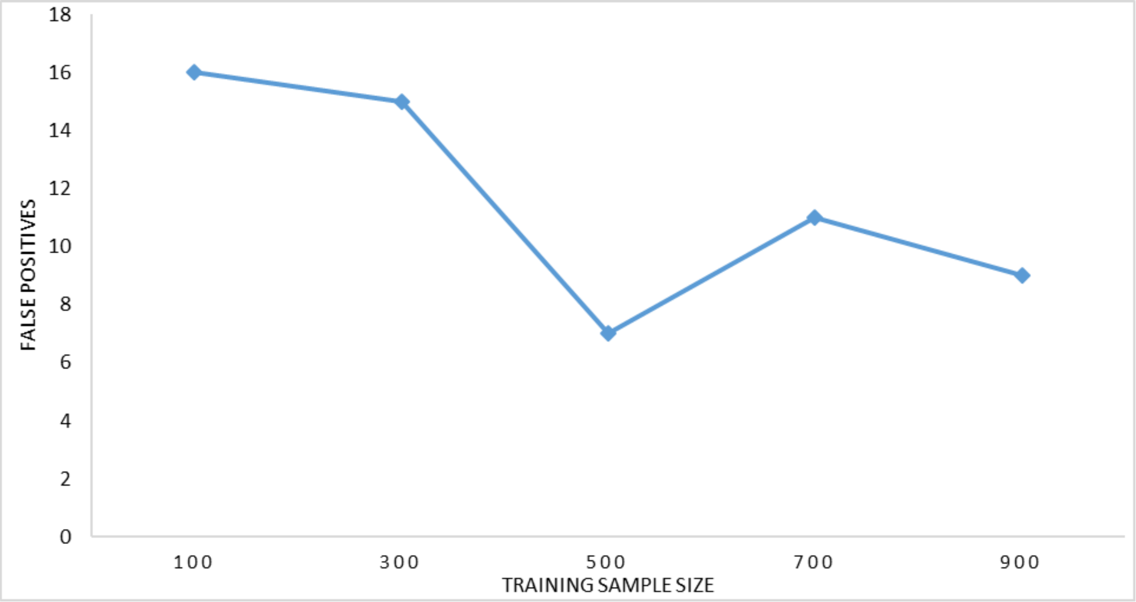


Figure 4.7: *False positives based on train sample size*

Figure 4.7 shows false positives based on increasing train sample size. Considering the increased sample size, the number of false positives started from 16, but as the sample was increased to 500, false positives decreased to 7. The last two samples were duplicated so the false positives were affected.

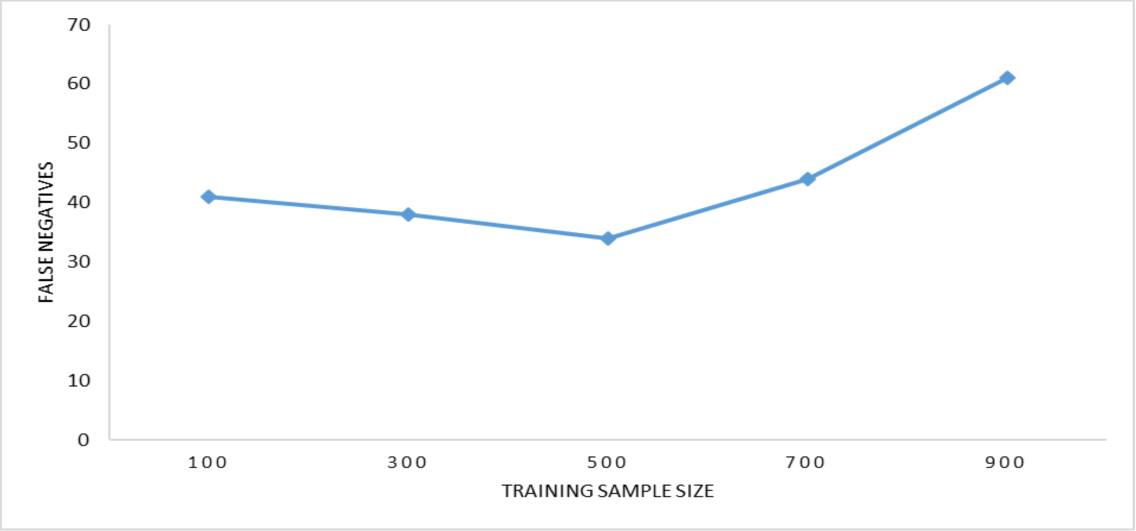


Figure 4.8: *False negatives based on train sample size*

Figure 4.8 shows false negatives based on the increasing sample size. As the data was increased from 100 to 500 train images, the graph shows that in 100 sample size false negatives were 41, at 300 false negatives were 38 and at 500 train images false negatives were 34. This means that increasing data reduced the number of false negatives. However, data duplication increased the number of false negatives.

The results obtained from faster R-CNN models trained with different sample size were added together for each entity. The 20 test images contained 121 objects and detected objects for each train sample were added together. All false positives, true positives, false negatives were added together for a model trained with 100 train images, a model 300 train images, a model trained with 500 images, a model trained with 700 images and a model trained with 900 images. False discovery rate (FDR), precision and recall were computed for results obtained from each model. The following formulas were utilized to determine false discovery rate, precision and recall.

- False discovery rate is the correlation of false positive compared to total detected objects.

$$FDR = \frac{fp}{tp + fp}$$

- Precision is the proportion of all true positives compared to all objects detected.

$$PR = \frac{tp}{tp + fp}$$

- Recall is the proportion of all true positives compared to all possible detections.

$$RC = \frac{tp}{tp + fn}$$

Table 4.7: *Compiled results for different train sample size*

Trained images	*Total objects*	*Detected objects*	*False Positives*	*True Positives*	*False negatives*	*FDR*	*Precision*	*Recall*
100	121	84	16	68	41	0,190476	0,809524	0,623853
300	121	90	15	75	38	0,166667	0,833333	0,663717
500	121	89	7	82	34	0,078652	0,921348	0,706897
700	121	83	11	72	44	0,13253	0,86747	0,62069
900	121	66	9	51	61	0,15	0,85	0,455357

This model yields best results when observing the outcome of each sample size as they increase. When looking at all of the results based on increasing sample sizes the number of false positives decreased as train sample increased and it was always lower than the number of true positives. Although the train sample size was very low, transfer learning improved the learning capability. Boats and ships were detected and the number of false positives reduced as training data was increased.

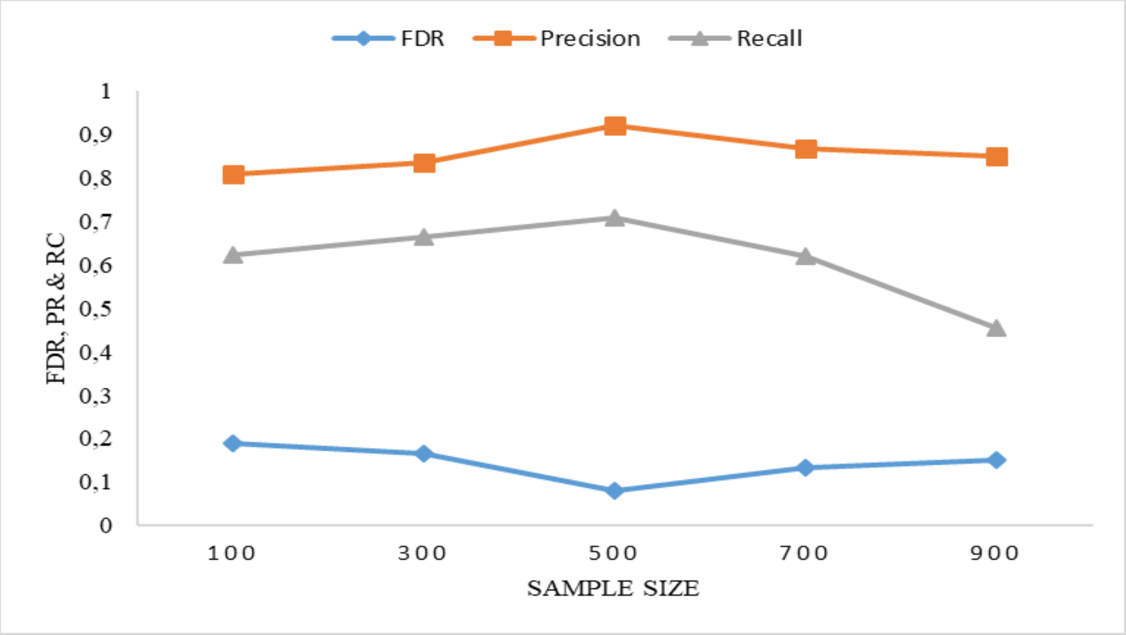


Figure 4.9: *Detection performance compared to training size*

Figure 4.9 shows false discovery rate (FDR), precision and recall compared to the train sample size. False discovery rate (FDR) was utilized to predict the occurrence of false positives and compare the results with the results when the model was further re-trained with increasing sample size. From 100 to 500 train sample FDR ranges from 0.19 to 0.07, this means that it was decreased by increasing sample size. In this line graph the precision ranges from 0.81 to 0.92 with the first increasing sample and from 0.92 to 0.85 when the data was duplicated. This means that the precision rate increased with increasing sample size but was disturbed when the data was duplicated. The recall also increased with increasing data, ranging from 0.6 to 0.7, but decreased when data was duplicated.

4.3 Comparison of findings

Chapter two of this dissertation discussed related work whereby classic and modern algorithms for object detection were discussed and analysed. It was found out that supervised deep learning object detection algorithms were best suited for maritime object detection. When a model is fed with many examples of the targeted object, the model accurately identifies objects. Background subtraction object detection methods may lead to a lot of false positives because sea background is dynamic and cluttered with a lot of objects which may not be desired targets hence separating the background from the foreground is a challenge.

Secondly, discussions on supervised deep learning algorithms like R-CNN, Fast R-CNN, Faster R-CNN, and YOLO were made and it was concluded that according to literature, Faster R-CNN model was the best method for this research since it performs better and faster than R-CNN and Fast R-CNN. YOLO 9000 also performs better but it is designed to accommodate up to 9000 images, therefore it was not utilized in this research because it requires more computation power and the data used was smaller. The results produced in this study showed that the proposed algorithm performed well in detecting objects and the false positives were reduced as data was increased.

4.4 Conclusion

This chapter encompasses the results obtained from this study. Firstly, it displayed the results obtained from training faster R-CNN inception v2 using 100 train images, 300 train images, 500 train images, 700 train images and 900 train images. The results obtained were represented in tables, where number of false positives plus the amount of false negatives and the amount of true positives for each image were mapped. Furthermore, graphs of false positives, false negatives and true positives were represented for each sample size. The detection performance was evaluated using confusion matrix. Lastly the false discovery rate(FDR), precision and recall were computed and mapped in a graph. The results will be discussed in chapter five.

5 Chapter Five: Conclusion and future work

Chapter one of this research addressed the tenacity and implication of the research. It specified the research problem, research aim, objectives and outline of the dissertation. Background study of object detection was discussed in chapter two. Chapter two also reviewed and analyzed literature on work published on object detection to answer objective one and objective two. Chapter three encompassed the methodology followed and techniques used to solve the problem introduced in chapter one. Experiments were also carried out in methodology and design chapter. Chapter four encompassed the results obtained from experiments. This chapter will discuss the results presented in chapter four, furthermore present conclusion and future work.

5.1 Dissertation summary

The main aim of the research was to improve the implementation of target detection in maritime environment by minimizing the occurrence of false positives. This was done through reviewing and exploring literature review on methods that are used for target detection and false positive reduction. A modified neural network method was utilized in this study. As discussed in chapter two, neural network is one of the supervised machine learning methods. Supervised machine learning methods are very powerful as they can learn from real life examples and apply that knowledge in solving challenging problems.

The results of this research depended upon answers to research questions relatively to the research objectives. The first two objectives were met through literature review and their solution led to the completion of objective three through experiments. In trying to answer the research question as to which methods are used for target detection in a maritime environment the focus was not only put on methods used in a maritime environment, but the research included all detection methods. The literature also reviewed methods used in videos since videos are made up of frames.

This model yields best results when observing the outcome of each sample size as they increase. When looking at all of the results based on increasing sample sizes the number of false positives decreased as train sample increased and it was always lower than the number of true positives. Although the train sample size was very low, transfer learning improved the learning capability. Boats and ships were detected and the number of false positives reduced as training data was increased. Eventually the problem of object detection can be tackled better with supervised learning algorithms.

5.2 Empirical findings vs objectives

Objective one: To review and analyze methods for target detection in a maritime environment

When literature was reviewed it was observed that background subtraction is a commonly used method for video detection object detection. Background subtraction detection method is very simple, inexpensive and is best in detecting moving objects as it detects the difference between foreground and background frame. The problem with background subtraction is that the foreground contains many objects than background which leads to difficulties in discriminating foreground objects from a background especially when the objects are stationary or moving slowly. Adaptive background models aided as an improvement to background subtraction, but false positives persisted due to background clutter. Ray and Chakraborty tried to solve the problem of difference and changes in background and foreground by employing phase correlation to compute comparative translation offset for estimating the global displacement between video frames [27]. As discussed in the literature review, blind source separation seems to suppress sea clutter in radar technology which was widely used in maritime detection. In this research visible images were utilized, therefore radar technology was not an option. Deep learning algorithms are real-time algorithms which showed great performance with less computation time as compared to other methods. As much as they were implemented as an improvement of previous detection algorithms they still require more computation power which can be expensive. R-CNN, Fast R-CNN, Faster R-CNN, YOLO and R-FCN shows a great improvement in object detection. This objective was met because it was concluded that faster region-based convolutional (R-CNN) model is the best method for detecting targets in a maritime environment because it is able to learn real-life examples and use it to enhance its background knowledge for better detection.

Objective two: To explore ways of reducing false positives during target detection.

In related work it was found that the main challenge in object detection is the occurrence of false positives. To address this problem many researchers, used additional algorithms to the detection methods. Region adaptive clutter rejection methods like local directional background removal filter (L-DBRF), M-MSF, attribute-based classification temporal consistency filter together with heterogeneous background removal filter are utilized by researchers to suppress background clutter before detection. The post-filtering process is a common method for minimizing false positives but it degrades the quality of an image leading to the loss of significant features. In this research, a real-time deep leaning method was utilized to detection boats and ships while reducing the number of false positives. Deep learning methods improve detection because it involves many layers with different weights which promotes learning on many examples. Different data sample sizes were used to track the performance of the model as the data increases. The results show that false positives were reduced as data increased.

Objective three: To design and implement a target detection model that minimises the occurrence of false positives.

As it has already been stated, the research made use of faster R-CNN with inception v2 pre-trained in COCO dataset. Computer restrains and few datasets led to the use of transfer learning. The model was re-trained with the available data which contained two classes, boat, and ship. Transfer learning is a method whereby upper layers are fine-tuned with existing data to solve a new problem at hand. COCO dataset is an 80 class large data for object detection, segmentation and captioning.

The results were presented in chapter four. They were tabulated to identify total detections, false positives, true positives, and false negatives for each image. The image number denoted the position of the image for image one to ten of the test data. The total detections included true and false positives for each image. For each of the five samples of train data, the outputs were tabulated in the same format but on different tables. The tables were used to make the results easier to read and compare true detections against false detections.

5.3 Recommendations and future work

This study managed to answer the questions which were proposed, however, there are many questions still not answered in this field. The detection was made for two different classes in a maritime environment, though there are many objects that are present in a maritime environment which can be a treat to people and security. The research only detected using images. A study can be done whereby all moving objects in a maritime video are detected and tracked. The data used in this research contained images with only the objects to be detected so the model was not familiar with negative examples, detection results can be improved with very large and relevant training data. It is recommended that the same research must be done using more than one algorithms.

5.4 Conclusion

The aim of this research was to improve the implementation of target detection in maritime environment. Object detection in maritime environment is a challenging problem more especially in maritime videos due to the occurrence of false positives. The sea background is very dynamic and contains many objects which can complicate the detector. In this research, a pre-trained faster region based convolutional neural network (R-CNN) was employed for detecting boats and ships in maritime environment. The proposed model was re-trained with different sample size to track the performance as the data was increased. In the results, the detection performance improved every time the sample size increased, reducing false positives and false negatives and increasing true positives. The last two train data which contained 700 and 900 train images were duplicated, the duplication had an undesirable outcome on the performance of the detection algorithm as it decreased true positives and increased false positives.

References

[1] A. Bachoo, B. Duvenhage, and J. De Villiers, “PRISM Project List,” 2015.

[2] S. Kim and J. Lee, “Small infrared target detection by region-adaptive clutter rejection for sea-based infrared search and track,” *Sensors (Switzerland)*, vol. 14, no. 7, pp. 13210–13242, 2014.

[3] D. Manolakis, D. Marden, and G. a Shaw, “Hyperspectral Image Processing for Automatic Target Detection Applications,” *Lincoln Lab. J.*, vol. 14, no. 1, pp. 79–116, 2003.

[4] D. Bloisi, L. Iocchi, D. Nardi, M. Fiorini, and M. Engineering, “Integrated Visual Information for Maritime Surveillance,” *Clean Mobil. Intell. Transp. Syst.*, pp. 237–264, 2015.

[5] F. Xu, J. Liu, M. Sun, D. Zeng, and X. Wang, “A hierarchical maritime target detection method for optical remote sensing imagery,” *Remote Sens.*, vol. 9, no. 3, p. 280, 2017.

[6] R. Szeliski, “Computer Vision : Algorithms and Applications,” *Computer (Long. Beach. Calif).*, vol. 5, p. 979, 2010.

[7] S. Ben-david, *Understanding Machine Learning : From Theory to Algorithms*. 2014.

[8] N. Sebe, I. Cohen, A. Garg, and T. S. Huang, *Machine Learning in Computer Vision*. New York, 2005.

[9] M. Jordan, J. Kleinberg, and B. Scho, *Pattern Recognition and Machine Learning*. New York, 2006.

[10] L. N. Long and A. Gupta, “Scalable Massively Parallel Artificial Neural Networks,” *J. Aerosp. Comput. Information, Commun.*, vol. 5, no. 1, pp. 3–15, 2008.

[11] K. Hornik, “Approximation Capabilities of Multilayer,” *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[12] M. Sonka, V. Hlavac, and R. Boyle, *Imaga Processing, Analysis, and Machina Vision*. 2008.

[13] R. Szeliski, “Computer Vision : Algorithms and Applications,” pp. 15–18, 2010.

[14] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, “Video Processing From Electro-Optical Sensors for Object Detection and Tracking in a Maritime Environment: A Survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 1–23, 2017.

[15] D. K. Prasad, C. K. Prasath, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, “Maritime Scenario Using Computer Vision,” *arXiv Prepr. arXiv1608.01079*, vol. 11, no. 1, pp. 31–36, 2016.

[16] D. P. Frost, J.-R. Tapamo, R. Peplow, and A. K. Bachoo, “Maritime Tracking Using Level Sets with Shape Priors,” *Univ. KwaZulu-Natal*, 2012.

[17] A. L. Chan, S. Z. Der, and N. M. Nasrabadi, *Encyclopedia of Optical Engineering*. 2003.

[18] Z. Cui, J. Yang, S. Jiang, and C. Wei, “Target detection algorithm based on two layers human visual system,” *Algorithms*, vol. 8, no. 3, pp. 541–551, 2015.

[19] D. Hall *et al.*, “Comparison of target detection algorithms using adaptive background models,” *IEEE Int. Work. Vis. Surveill. Perform. Eval. Track. Surveill.*, no. 1, pp. 113–120, 2005.

[20] H. Ghahramani, M. Barari, M. H. Bastani, H. Ghahramani, M. Barari, and M. H. Bastani, “Maritime Radar Target Detection in Presence of Strong Sea Clutter Based on Blind Source Separation,” *IETE J. Res.*, vol. 60, no. 5, pp. 331–344, 2017.

[21] L. Tong, F. Heymann, and T. Noack, “Radar target detection based on methods of image pattern matching,” *Zesz. Nauk. Morska w Szczecinie*, vol. 36, no. 108, pp. 162–167, 2013.

[22] G. Jing and S. Chaojian, “Survey on Methods of Moving Object Video Detection in Marine Environment,” *Int. Conf. Comput. Sci. Inf. Technol. (ICCSIT).*, vol. 51, pp. 437–440, 2012.

[23] S. Wan, H. Liang, T. Wang, and Z. Cui, “Moving Target Detection Algorithm Research Based on Background Subtraction Method,” *3rd Int. Conf. Multimed. Technol.*, pp. 1179–1186, 2013.

[24] M. . Jadhav and J. Jyoti, “Moving Object Detection and Tracking for Video Survelliance,” *Int. J. Eng. Res. Gen. Sci.*, vol. 2, no. 4, pp. 372–378, 2014.

[25] L. Wang and D. Sng, “Deep Learning Algorithms with Applications to Video Analytics for A Smart City: A Survey,” pp. 1–8, 2015.

[26] H. . Parekh, D. . Thakore, and U. . Jaliya, “A Survey on Object Detection and Tracking Methods,” *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2, no. 2, pp. 2970–2978, 2014.

[27] K. S. Ray and S. Chakraborty, “An Efficient Approach for Object Detection and Tracking of Objects in a Video with Variable Background,” *arXiv Prepr. arXiv1706.02672*, pp. 1–11, 2017.

[28] S. A. I. Guo, Q. I. Zhang, Y. Shao, and W. Chen, “Sea Clutter and Target Detection with Deep Neural Networks,” *DEStech Trans. Comput. Sci. Eng. aiea*, pp. 316–326, 2017.

[29] S. . Jodalli, V. Simon, M. Vijayalakshmi, and Na. Dharani, “Identification and Classification of Objects in Marine Image Data Set for Coastal Surveillance,” pp. 24–26, 2015.

[30] L. Wang, X. Zhao, and Y. Liu, “Reduce false positives for human detection by a priori probability in videos,” *2015 3rd IAPR Asian Conf. Pattern Recognit.*, vol. 208, pp. 584–588, 2016.

[31] D. Frost and J.-R. Tapamo, “Detection and tracking of moving objects in a maritime environment using level set with shape priors,” *EURASIP J. Image Video Process.*, no. 1, pp. 1–16, 2013.

[32] R. Girshick, J. Donahue, T. Darrell, U. C. Berkeley, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, pp. 580–587, 2014.

[33] R. Girshick, “Fast R-CNN,” *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1440–1448, 2015.

[34] S. Ren, K. He, and R. Girshick, “Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks,” *Adv. neural Inf. Process. Syst.*, pp. 91–99, 2015.

[35] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN : Object Detection via Region-based Fully Convolutional Networks,” *Adv. neural Inf. Process. Syst.*, pp. 379–387, 2016.

[36] J. Redmon and A. Farhadi, “Better , Faster , Stronger,” *Proc. IEEE Conf. Comput. Vis.*

pattern Recognit., pp. 7263–7271, 2017.

[37] R. Wijnhoven, K. van Rens, E. Jaspers, and P. de With, “Online Learning for Ship Detection in Maritime Surveillance,” *Proc. 31th Symp. Inf. Theory Benelux*, pp. 73–80, 2010.

[38] C. Kothari, *Research methodology: methods and techniques*. 2004.

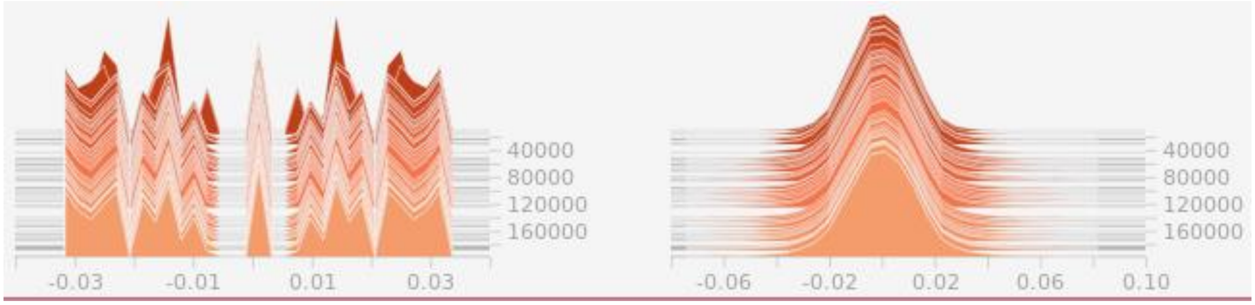
[39] I. S. MacKenzie, S. J. Castellucci, and Sigchi, “Empirical research methods for human-computer interaction,” *32nd Annu. ACM Conf. Hum. Factors Comput. Syst. CHI EA 2014*, pp. 1–6, 2014.

[40] “Tensorflow detection model zoo,” 2018. [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md%22%3Ehttps://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md. [Accessed: 07-Mar-2019].

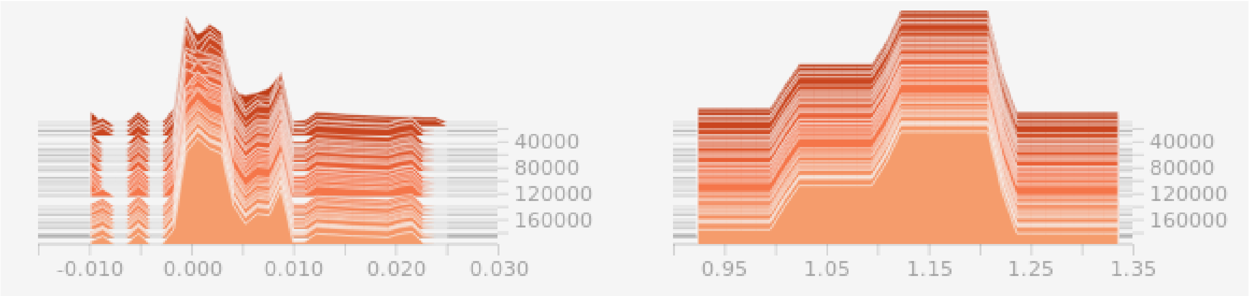
[41] “tensorflow/models,” 2018. [Online]. Available: https://github.com/tensorflow/models/tree/master/research/object_detection%22%3Ehttps://github.com/tensorflow/models/tree/master/research/object_detection. [Accessed: 24-Apr-2019].

[42] M. M. Zhang, J. Choi, K. Daniilidis, M. T. Wolf, and C. Kanan, “VAIS : A Dataset for Recognizing Maritime Imagery in the Visible and Infrared Spectrums,” *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 10–16, 1972.

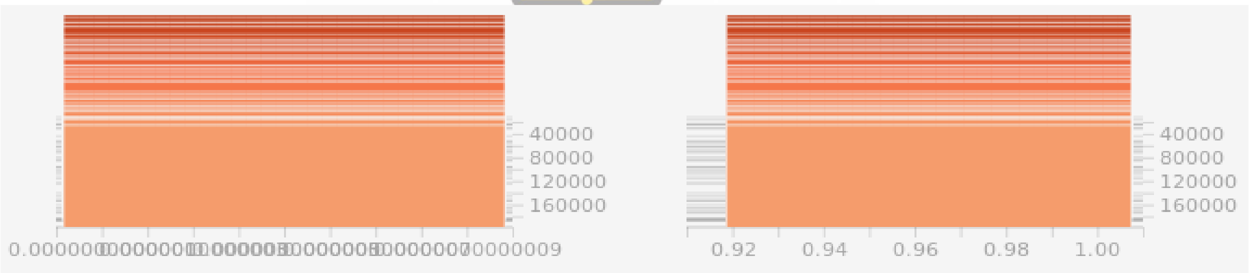
Appendix A: Histogram for training performance of 100 train images



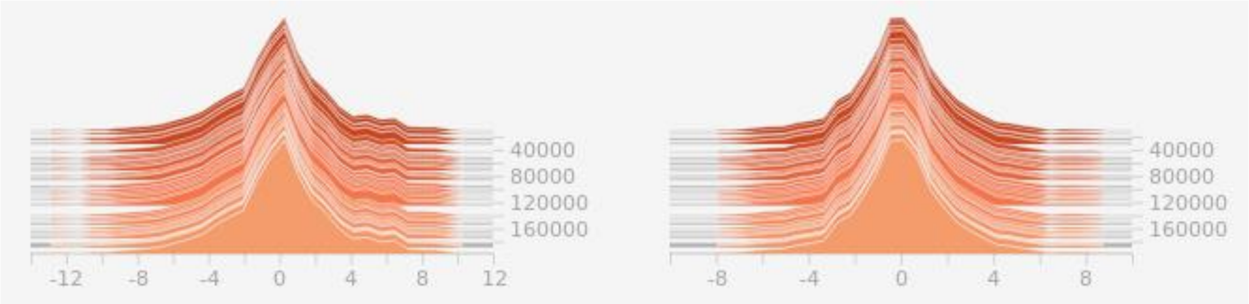
Biases and weights



Beta and gamma

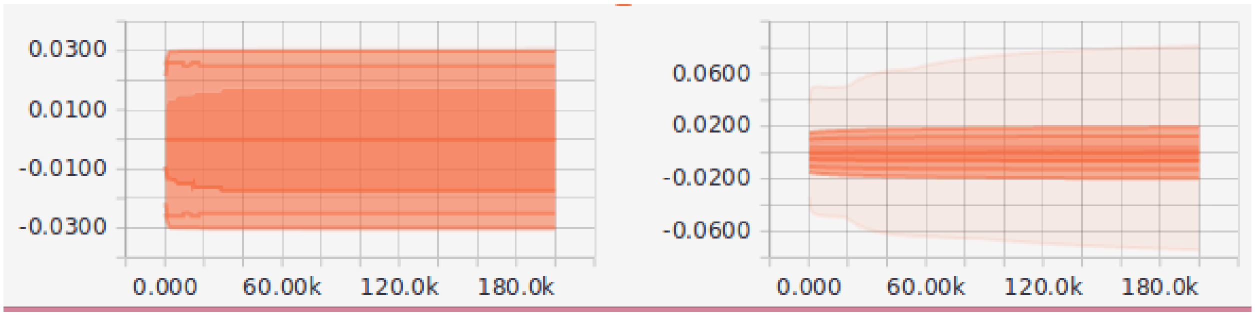


Moving mean and moving variance

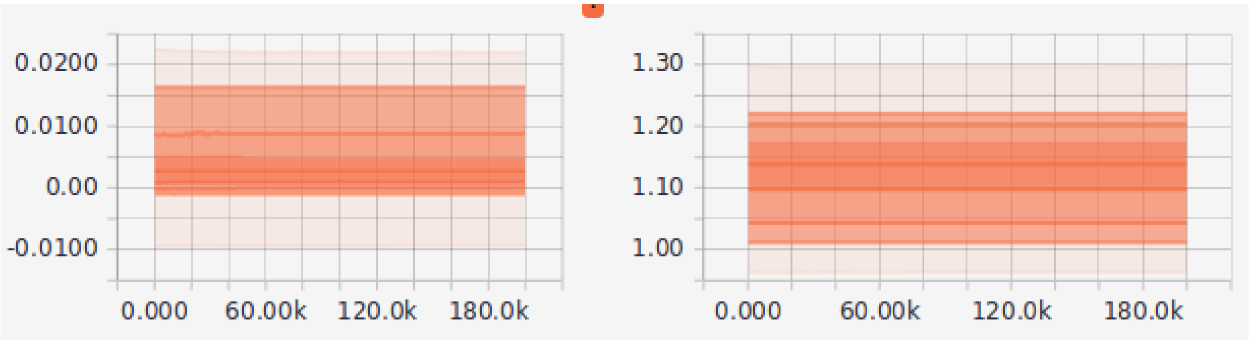


Depth wise weights and pointwise weights

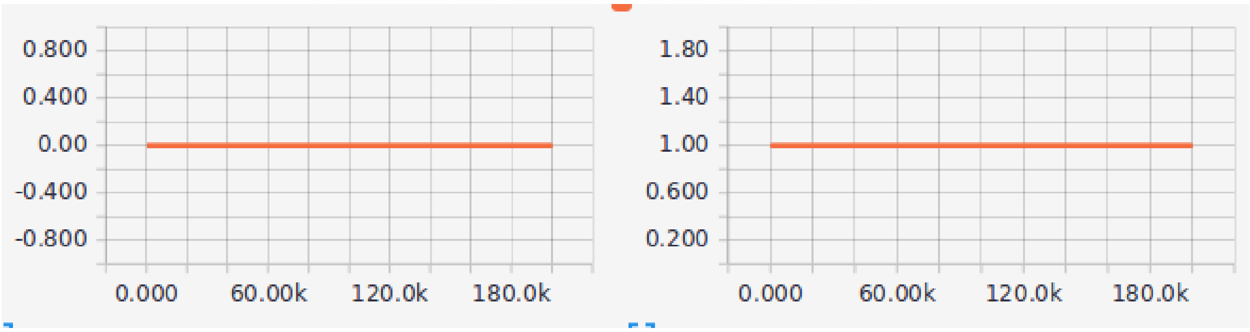
Appendix B: Distribution for training performance of 100 images



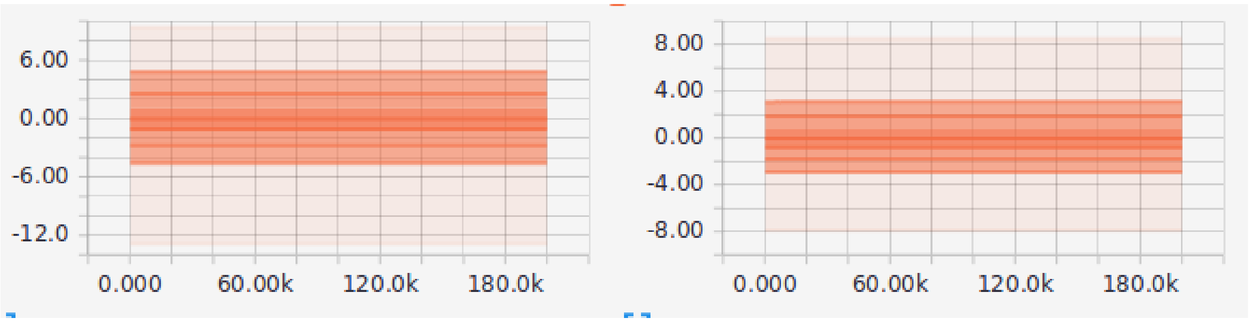
Biases and weights



Beta and gamma

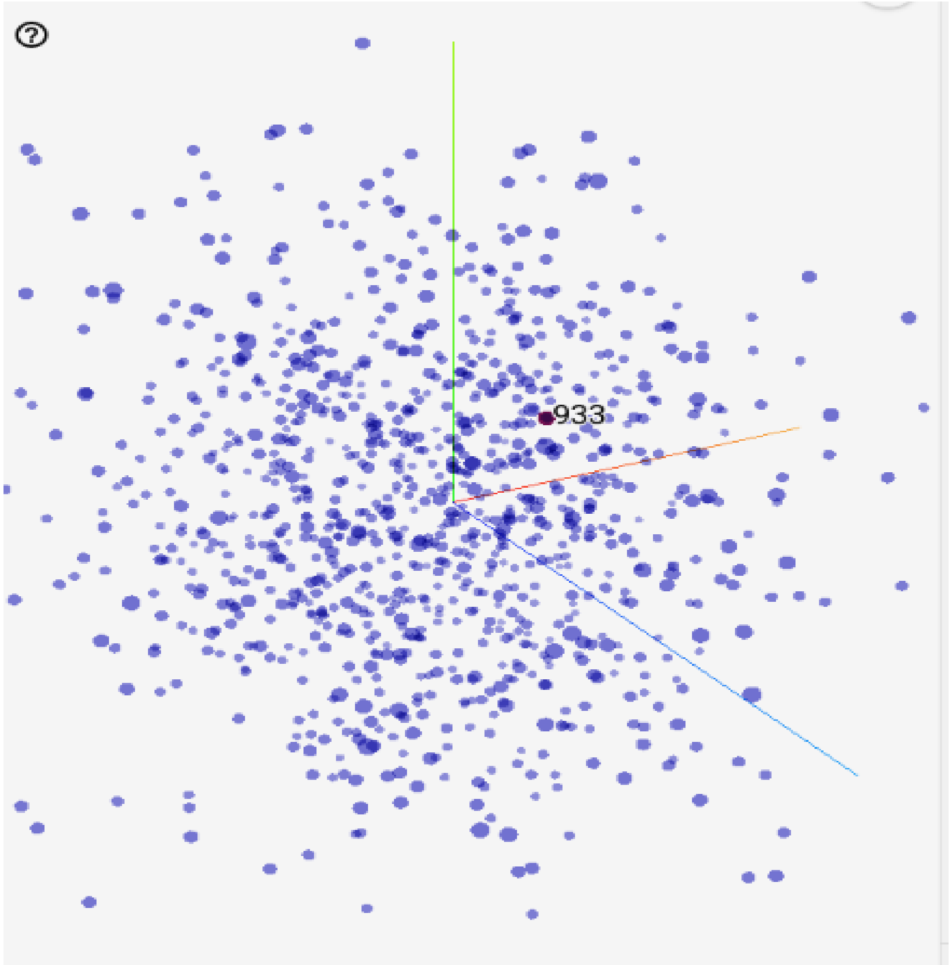


Moving mean and moving variance

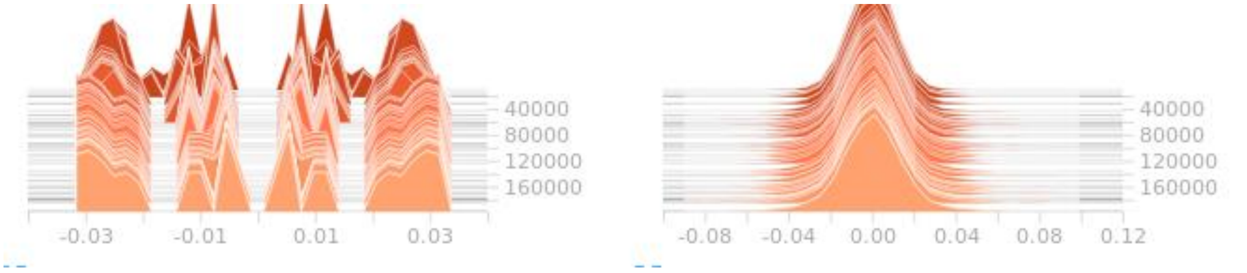


Depth wise weights and pointwise weights

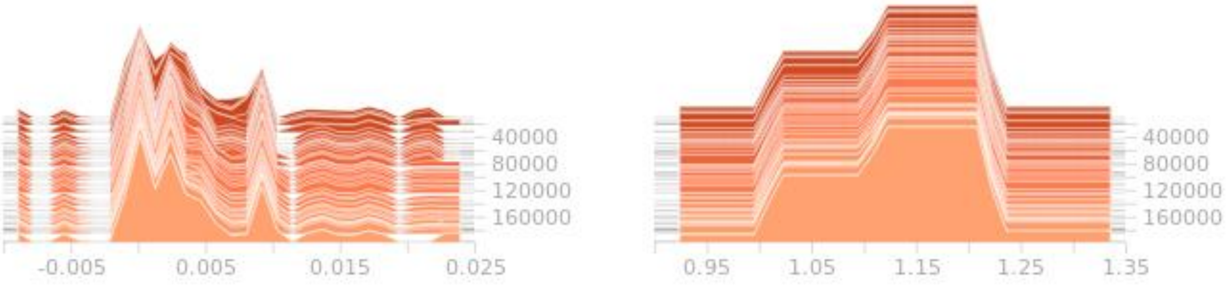
Appendix C: Projector for training performance of 100 train images



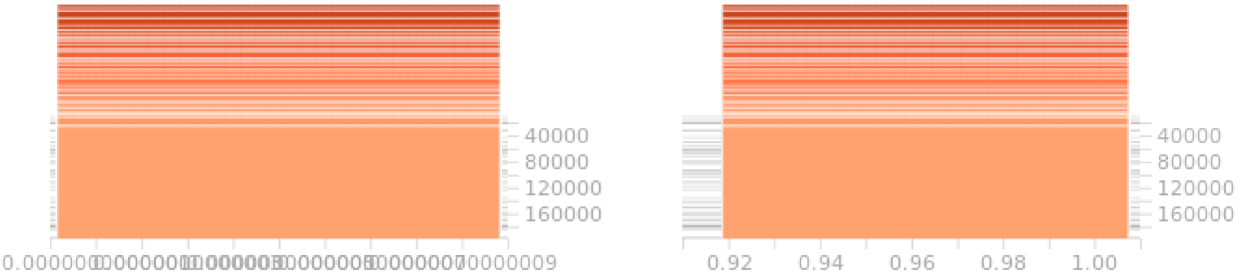
Appendix D: Histogram for training performance of 300 train images



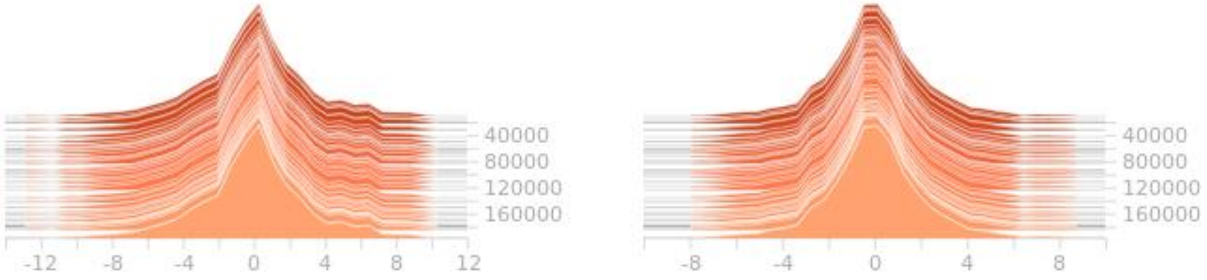
Biases and weights



Beta and gamma

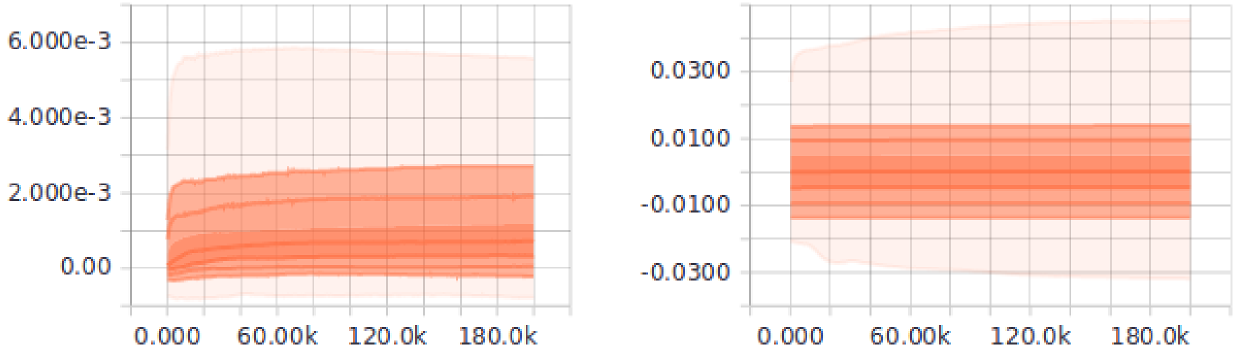


Moving mean and moving variance

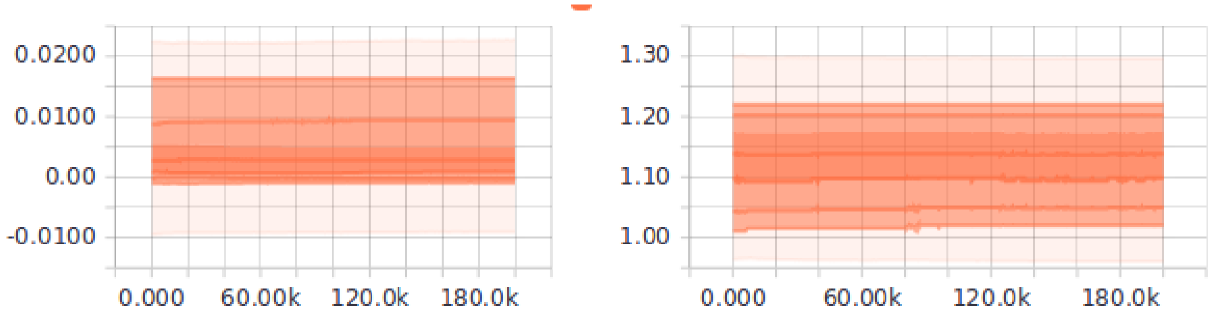


Depth wise and pointwise weights

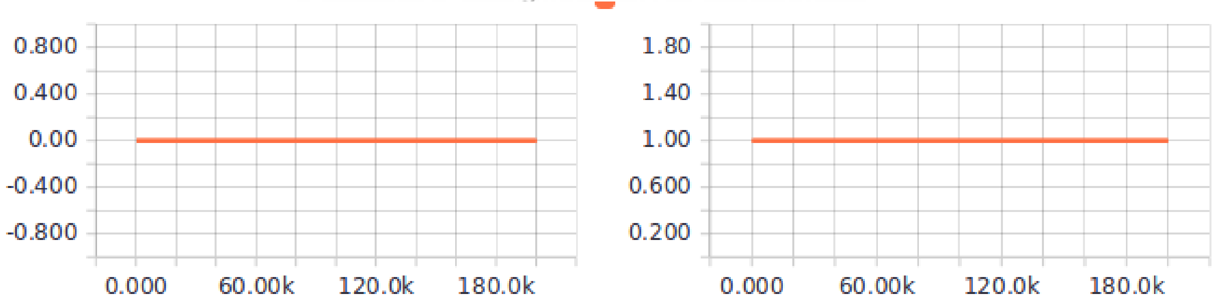
Appendix E: Distribution for training performance of 300 train images



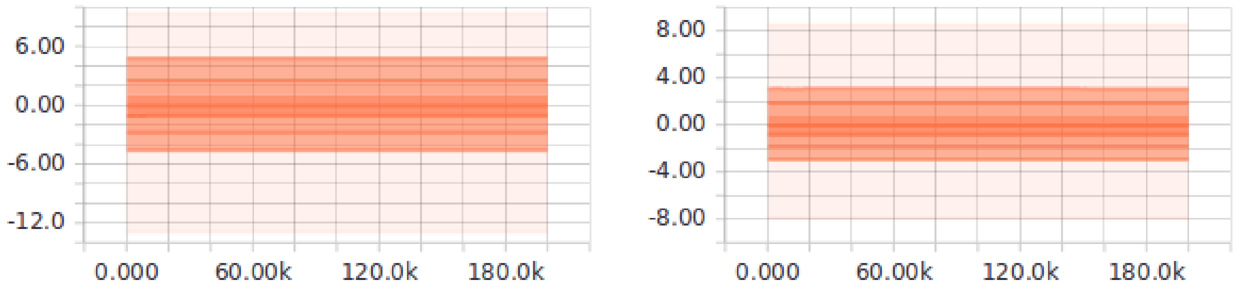
Biases and weights



Beta and gamma

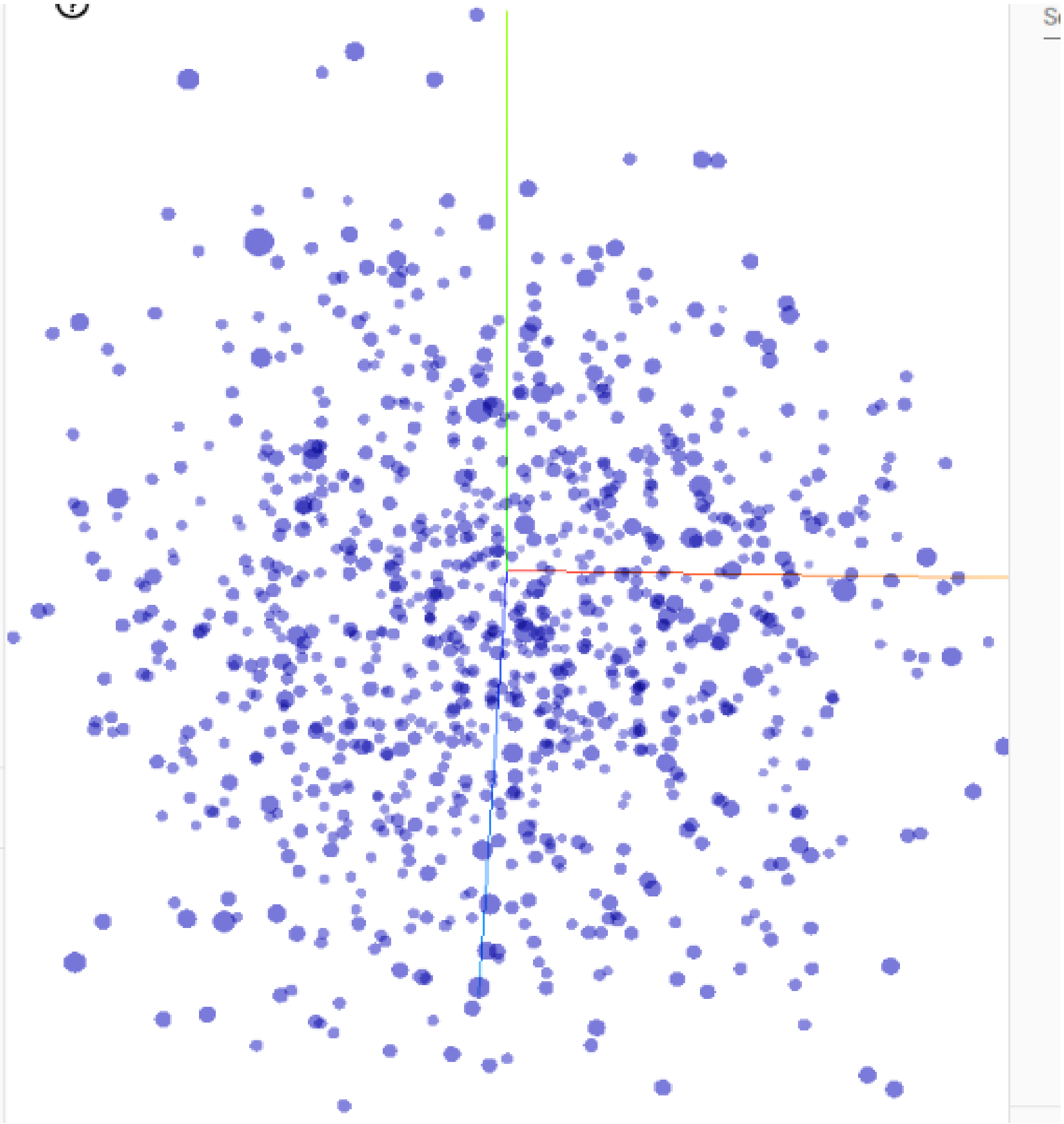


Moving mean and moving variance

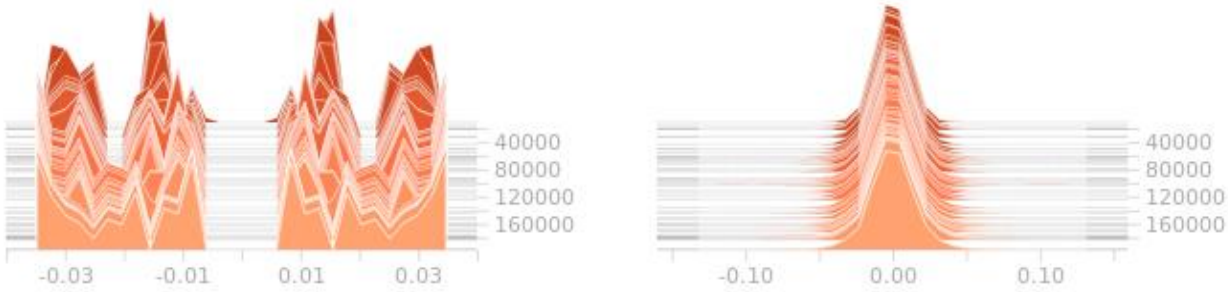


Depth wise and pointwise weights

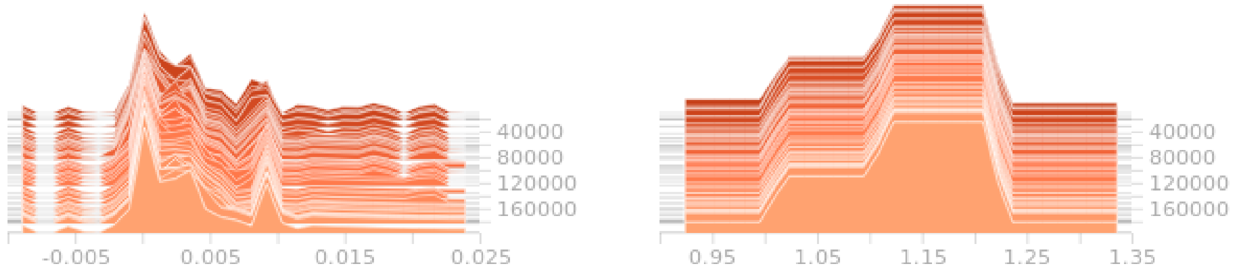
Appendix F: Projector for training performance of 300 train images



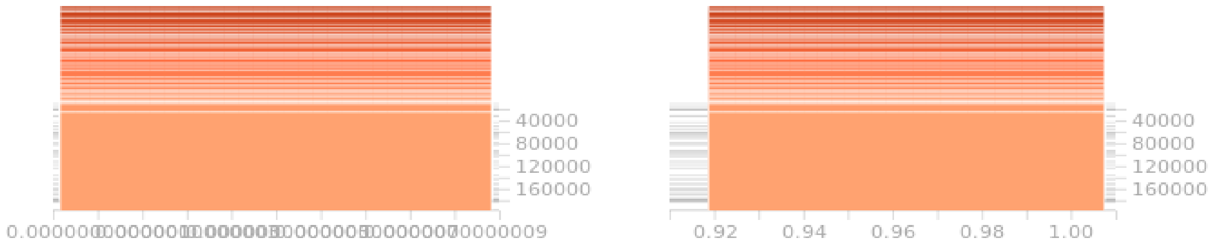
Appendix G: Histogram for training performance of 500 train images



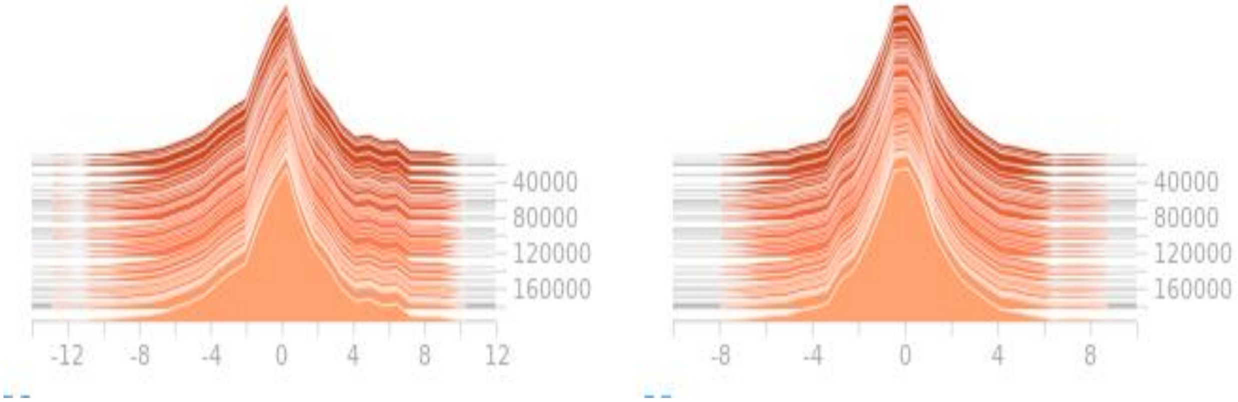
Biases and weights



Beta and gamma

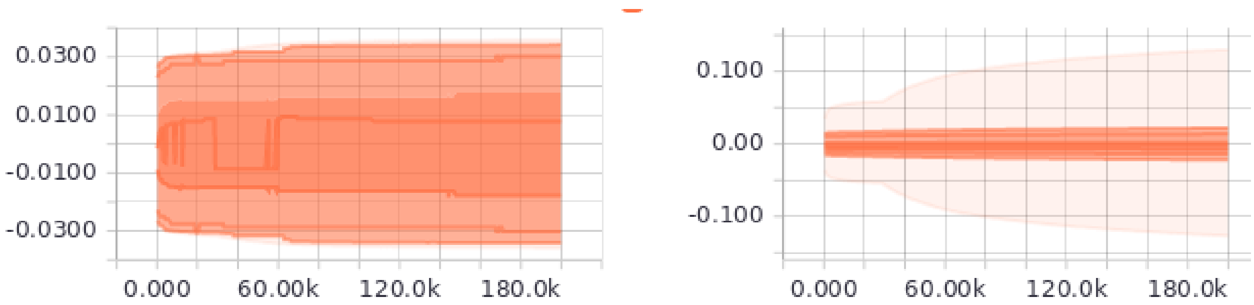


Moving mean and moving variance

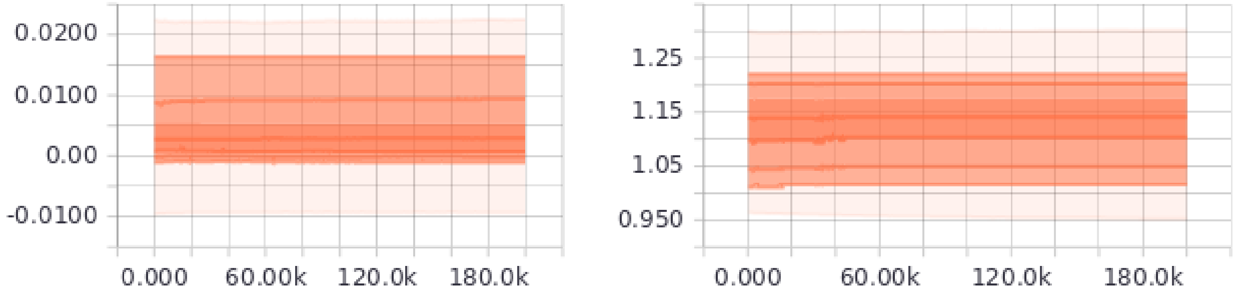


Depth wise weights and pointwise weights

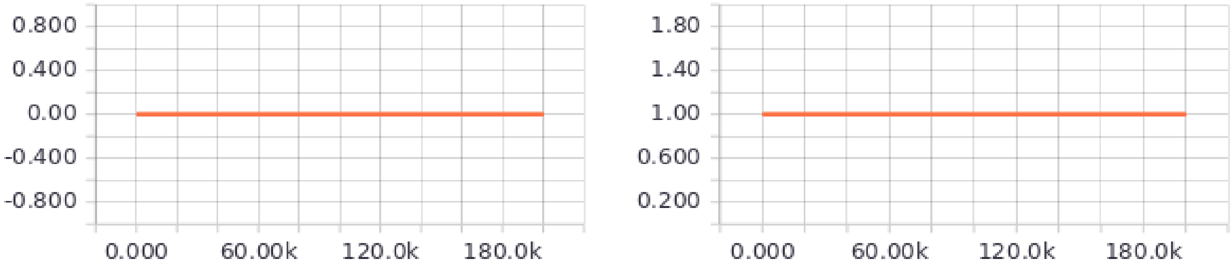
Appendix H: Distribution for training performance of 500 images



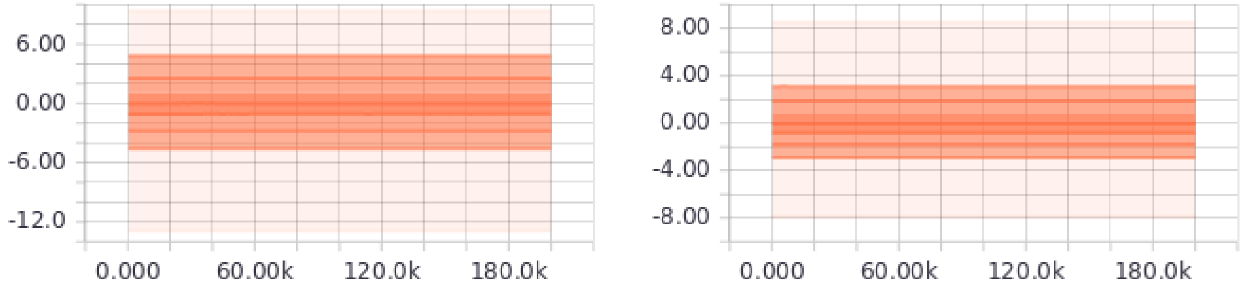
Biases and weights



Beta and Gamma

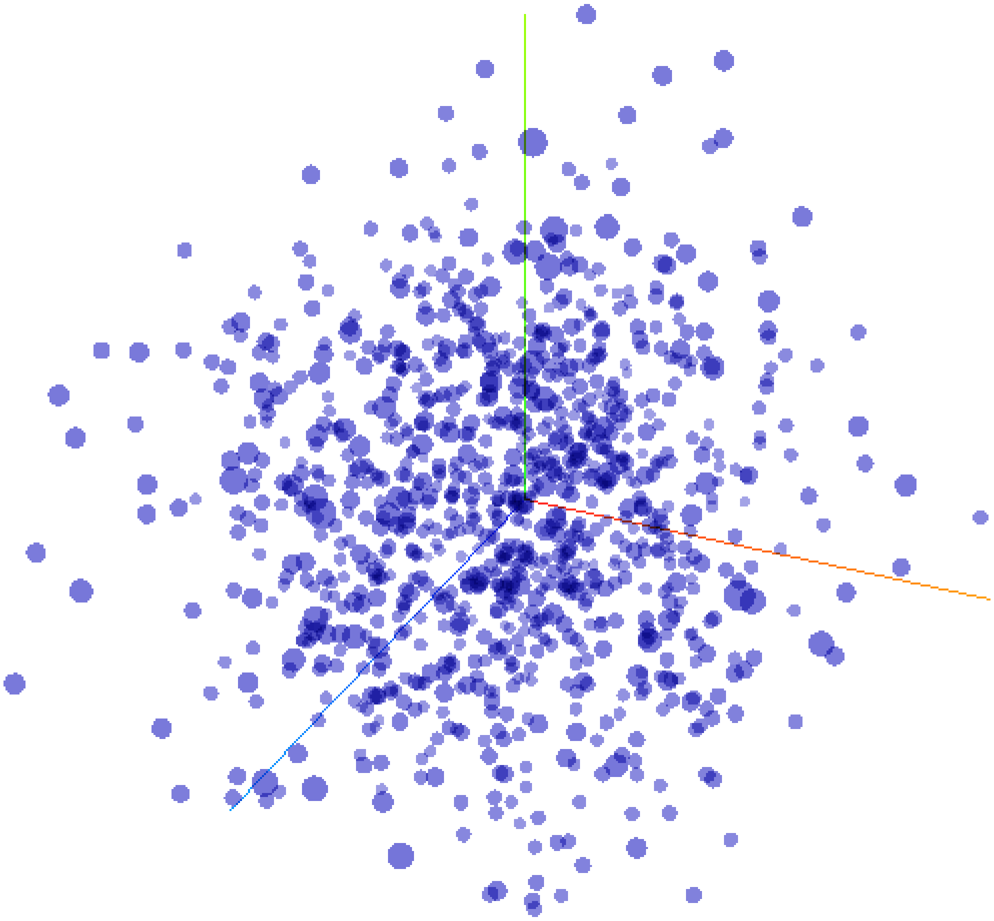


Moving mean and moving variance

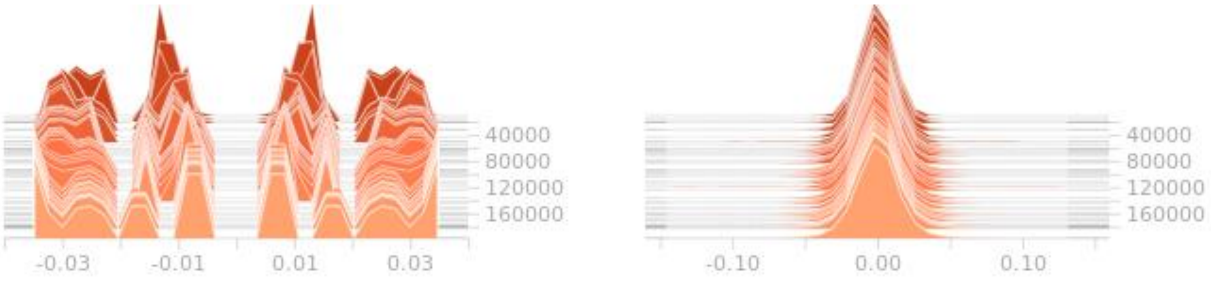


Depth wise and pointwise weights

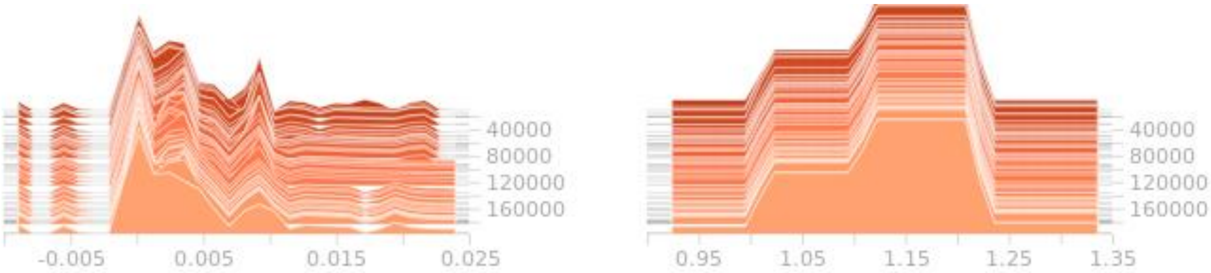
Appendix I: Projector for training performance of 500 images



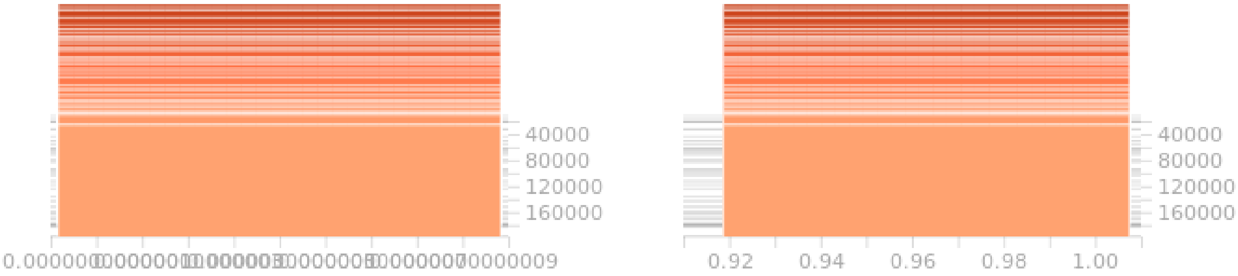
Appendix J: Histogram for training performance of 700 duplicated train images



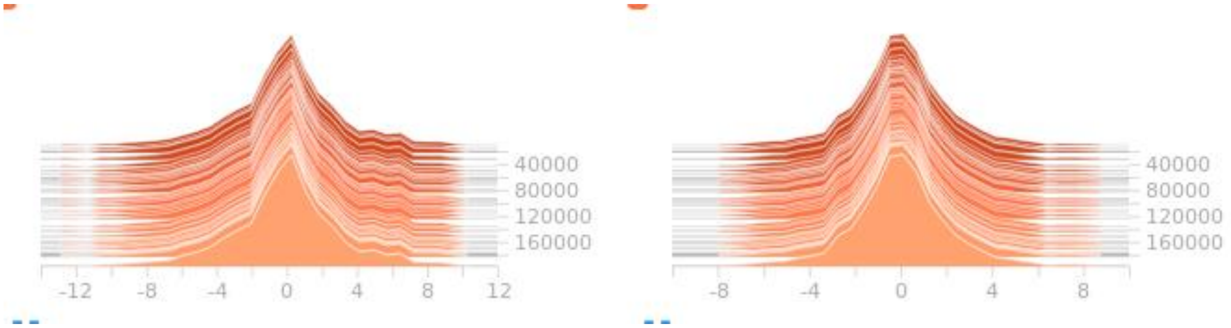
Biases and weights



Beta and gamma

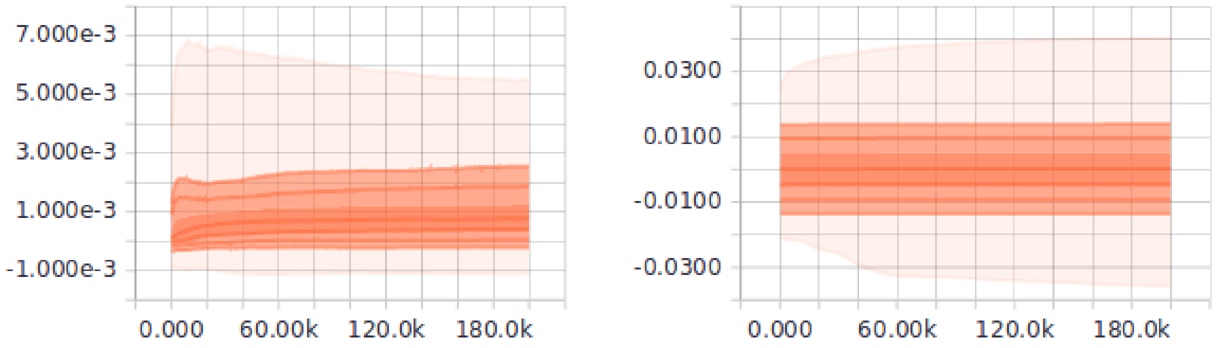


Moving mean and moving variance

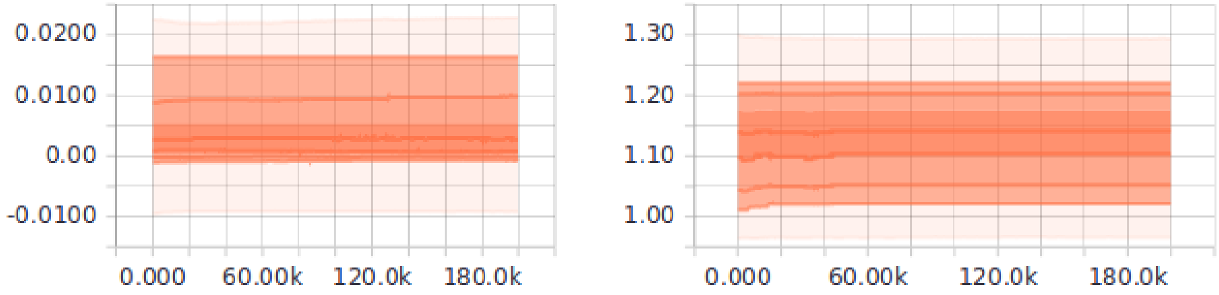


Depth wise and pointwise weights

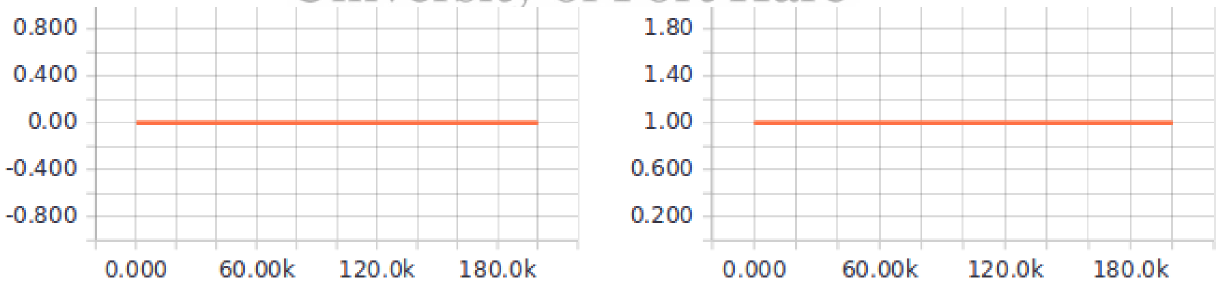
Appendix K: Distribution for training performance of 700 duplicated train images



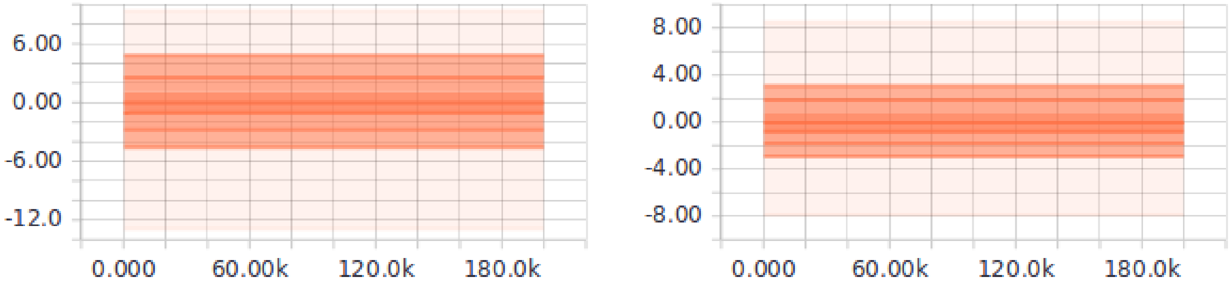
Biases and weights



Beta and gamma

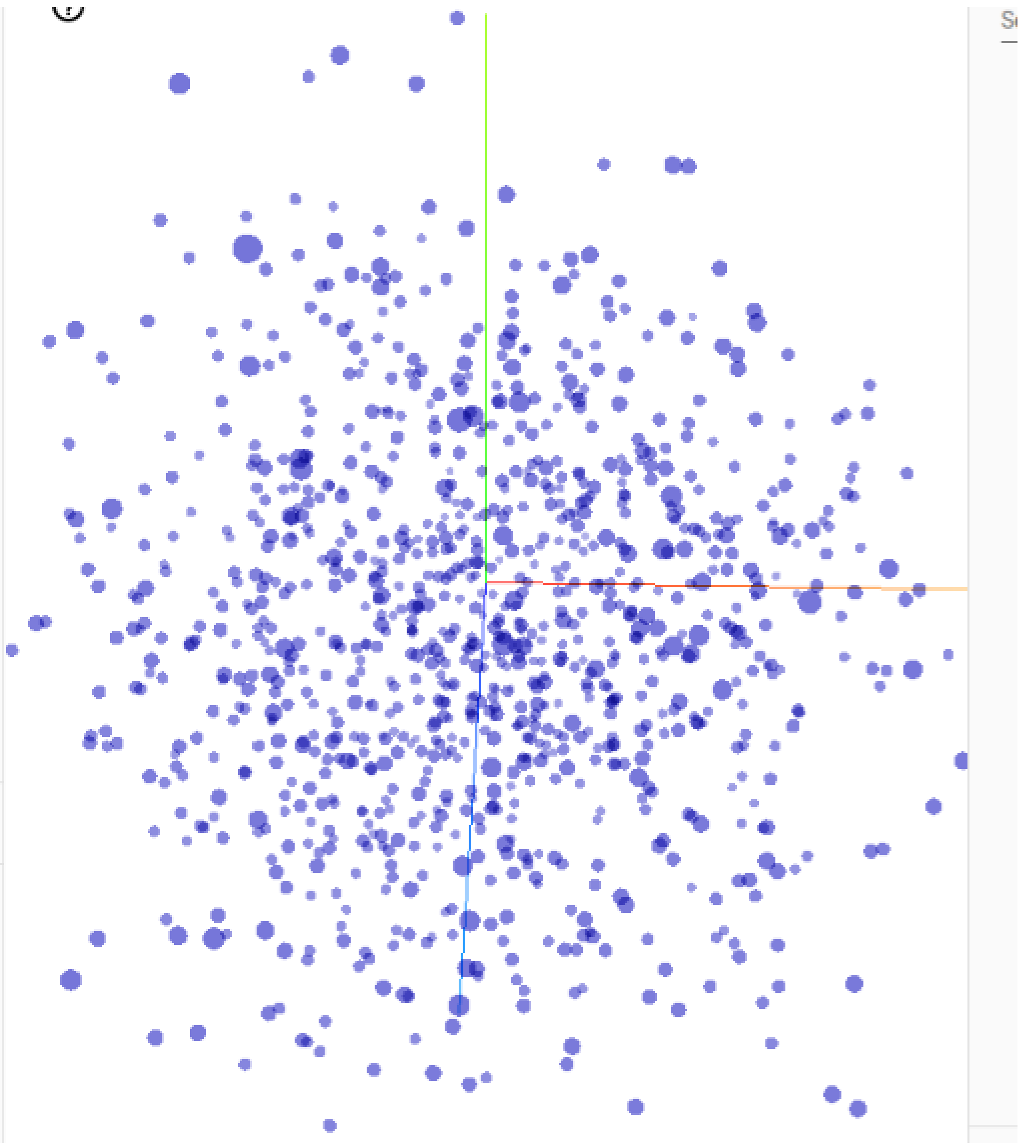


Moving mean and moving variance

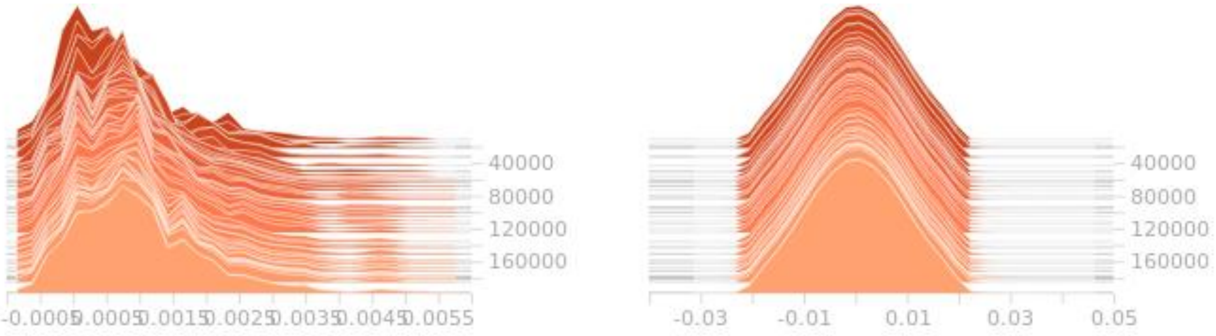


Depth wise and pointwise weights

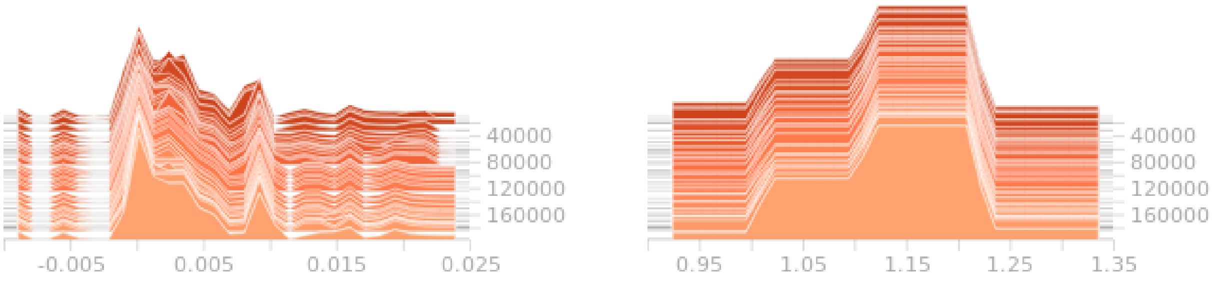
Appendix L: Projector for training performance of 700 train images



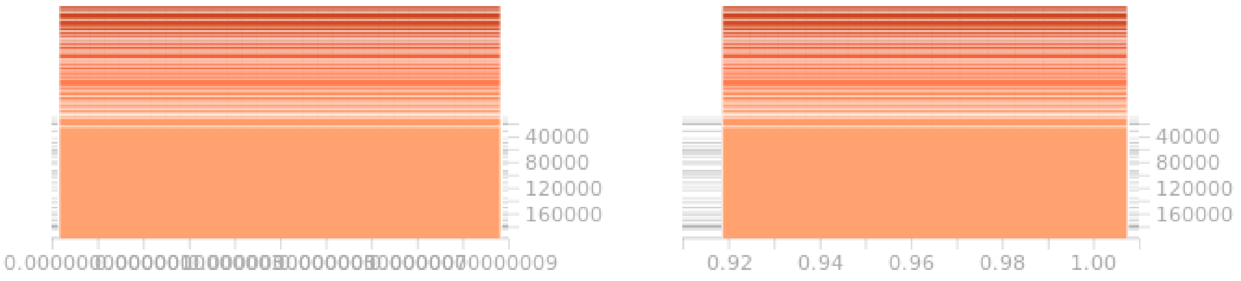
Appendix M: Histogram for training performance of 900 duplicated train images



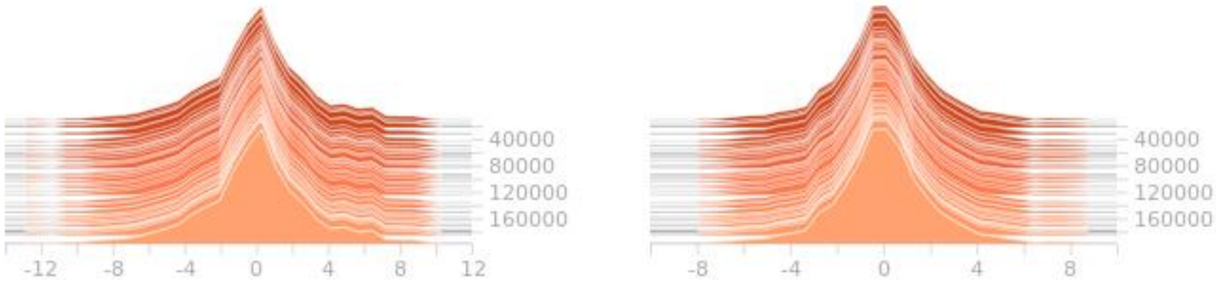
Biases and weights



Beta and gamma

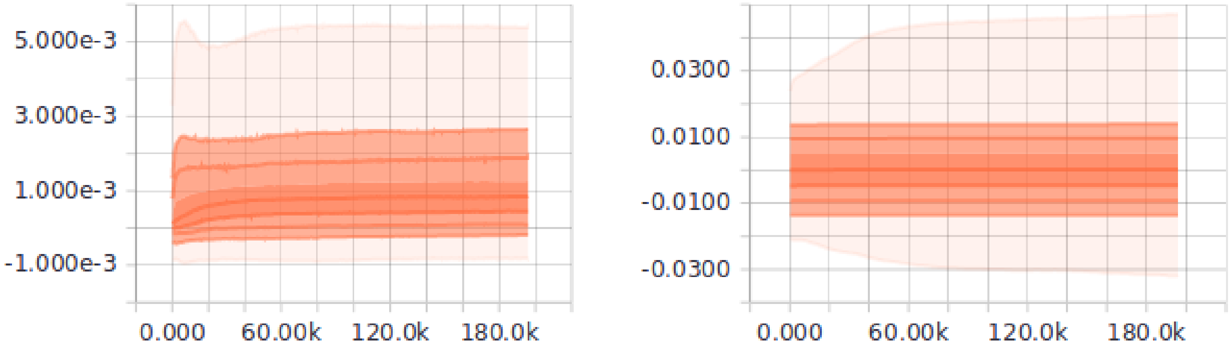


Moving mean and moving variance

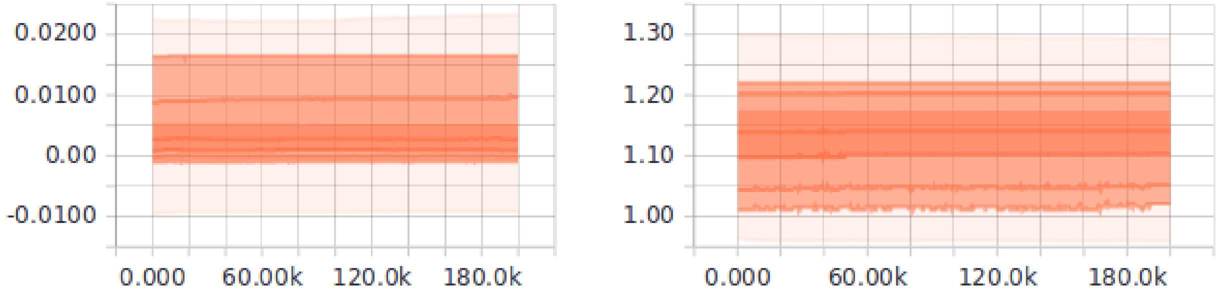


Depth wise and pointwise weights

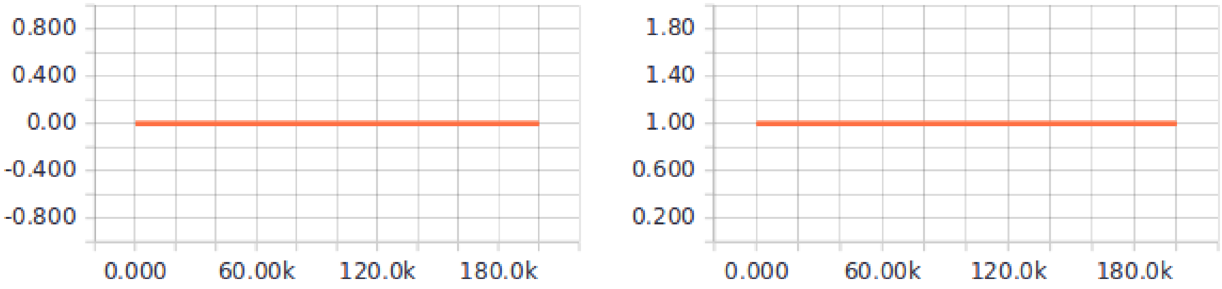
Appendix N: Distribution for training performance of 900 duplicated train images



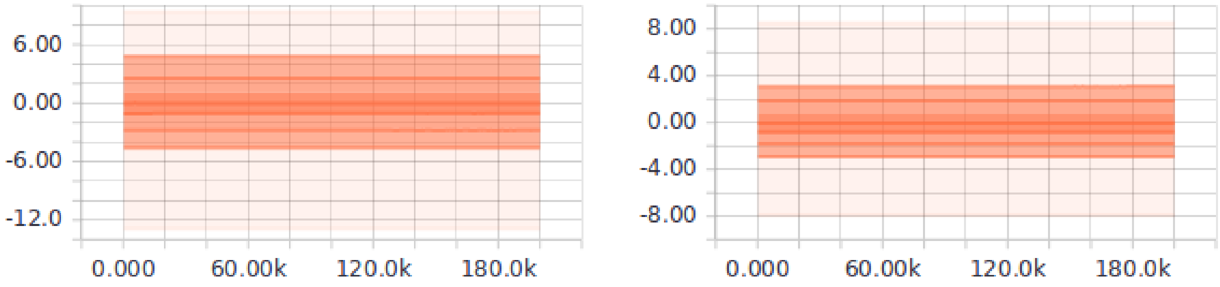
Biases and weights



Beta and gamma



Moving mean and moving variance



Depth wise and pointwise weights

Appendix O: Projector for training performance of 900 duplicated train images