ROBUST OBJECT DETECTION IN THE WILD VIA CASCADED DCGAN

Submitted

By

Ranjith Kumar Dinakaran

The Thesis Submitted in Partial Fulfilment of the Requirement of the University of Northumbria at Newcastle for the Degree

Doctor of Philosophy

Research is undertaken in the Faculty of Engineering and Environment

ABSTRACT

This research deals with the challenges of object detection at a distance or low resolution in the wild. The main intention of this research is to exploit and cascade state-of-the-art models and propose a new framework for enabling successful deployment for diverse applications. Specifically, the proposed deep learning framework uses state-of-the-art deep networks, such as Deep Convolutional Generative Adversarial Network (DCGAN) and Single Shot Detector (SSD). It combines the above two deep learning models to generate a new framework, namely DCGAN-SSD. The proposed model can deal with object detection and recognition in the wild with various image resolutions and scaling differences. To deal with multiple object detection tasks, the training of this network model in this research has been conducted using different cross-domain datasets for various applications. The efficiency of the proposed model can further be determined by the validation of diverse applications such as visual surveillance in the wild in intelligent cities, underwater object detection for crewless underwater vehicles, and on-street in-vehicle object detection for driverless vehicle technologies. The results produced by DCGAN-SSD indicate that the proposed method in this research, along with Particle Swarm Optimization (PSO), outperforms every other application concerning object detection and demonstrates its great superiority in improving object detection performance in diverse testing cases. The DCGAN-SSD model is equipped with PSO, which helps select the hyperparameter for the object detector. Most object detectors struggle in this regard, as they require manual effort in selecting the hyperparameters to obtain better object detection. This research encountered the problem of hyperparameter selection through the integration of PSO with SSD. The main reason the research conducted with deep learning models was the traditional machine learning models lag in accuracy and performance. The advantage of this research and it is achieved with the integration of DCGAN-SSD has been accommodated under a single pipeline.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

Instead of taking Artificial Intelligence (AI) as an advanced model or hypothesis, AI should be viewed as making everyone's life reliable and more organized. In many deployments, AI and Convolutional Neural networks (CNNs) have proven to be very reliable and highly effective, as well as efficient in obtaining results with the feature extractors in recent computer vision research. Machine Learning (ML) and Artificial Intelligence concepts were initially proposed by [Turing 1950]. Since the invention of CNNs, they have experienced a significant revival in recognition, principally for complex computer vision problems involving effective feature extraction. The competition conducted by The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [Krizhevsky, 2012] since 2010 has used the ImageNet dataset, which has over 14 million images; the data are associated with several metadata, which is used to develop the applications related to computer vision problems.

1.1 Motivations

In the future, AI and computer vision will help humans to lead sophisticated lives. This research focuses on object detection in smart cities, autonomous vehicles, and underwater. The application is not limited to these areas. It can be extended further to industrial applications in the oil and gas industries where AI can play a significant role in many areas, e.g., document classification and object detection in Piping and Instrument Drawings, so-called PID.

We live in the era of information technology, where science and machinery underpin everyone's day-to-day lives, and technologies are evolving in different ways. However, the foundations for the concepts behind this modern era are ML and AI. The field of object detection is vast, and research faces numerous challenges whenever it takes new turns since solutions for one problem are not always applicable to a similar research problem in the same Framework. There is always a solution for every situation, but its all about how the chosen solution can make difference when compared to others. This research provides one general solution for the problems most object detection frameworks face today.

1.2 The Proposed Methodology

The combination of two different models. This research comprises a specific combination of models, layers, and parameters and the optimization of layers using Particle Swarm Optimization (PSO) that make up the structure of a CNN as the "parameters" of the Network. The initial stage of research was based on the fundamentals of computer vision which deals with Histogram of Oriented Gradient (HOG) and Support Vector machines (SVM), which support object detection, but the results and efficiency were not stable based on the HOG-SVM, which is also one of the reasons the research interest moved towards deep learning. DCGAN [Radford, 2016 and SSD, with its layout and architecture optimization using Particle Swarm Optimization (PSO), becomes a robust and more extensive network with in-depth

layers and build. The architectural design and the methodology is hand-crafted no such kind of cascaded model or architecture has existed before. The research-based in this thesis with DCGAN-SSD is the first novel finding published as the first paper with the combination of DCGAN–SSD for object detection. Other articles speak about different applications; these papers are referred to within chapters in this thesis. This research covers design and architecture, including the PSO for specific parameters in the object detection model. This architecture structure has been built to achieve the best possible results and to tweak hyperparameters. The main disadvantage of the architecture is handling anyone who runs or tests this architecture, other than those who created it, will require an intensive knowledge of the design and architecture. To make this simple in the Network, this research includes the PSO for parameter optimization, which helps the user or researcher choose the best training parameters without spending too much time training their Network.

This research provides an effective way to achieve better object detection with CNNs with image classification through network optimization and optimized training with Visual Geometry Group (VGG-19) [Simonyan & Zisserman, 2015] architecture for the SSD and the DCGAN cascaded with SSD which forms the DCGAN-SSD model. To optimize the network training and learning rate, PSO has been used. Through this approach this research establishes an efficient way to perform object detection through CNN. This research uses the outline of the VGG [Simonyan & Zisserman, 2015] architecture.

Whenever there is a discussion about object detection, ML, AI, and computer vision, the most essential factor to consider is Convolutional Neural networks (CNN) [LeCun et al., 1998], one of the best features extractors in research and solutions for real-world computer vision. Also important is choosing the suitable architecture for the object detector and its application. These neural nets sometimes face concussion problems due to the addition of new classes or objects to be detected. When deep diving inside, the most appropriate choice should be made for efficient architecture, and a neural network scheme is used to build the architecture. In this research, the CNN is used to create the architecture, and the built architecture is embedded in the SSD [Liu et al., 2016]. This SSD is the state-of-the-art object detector invented by Liu et al [2016]. The reason behind choosing this SSD for this research is that it is flexible in changing the architecture inside the SSD.

Deep networks require significant training efforts to reach competitive performance [Liu et al., 2016]. Training is mandatory in neural networks: just as a child cannot differentiate colours without proper training, the same applies to the AI framework. Without formal training, detecting different objects for any framework is impossible. To provide this training, the architecture needs to be built to learn the model. The learned model will be able to differentiate objects, and it can be customized for different categories of objects, depending on the need. For the experiments in this research, the architecture used is VGG-19 [Simonyan & Zisserman, 2015], which the Visual Geometry Group invented; the

architecture is used in SSD, and also VGG-16 architecture is used by training the architecture from scratch for specific applications. VGG-19 architecture also needs the training to differentiate the object from a different class. This VGG-19 architecture is working with SSD for the first time. Therefore, it requires training and testing when it comes to class: other objects, such as humans, animals, cars, furniture etc., each carry different class names. Therefore, each object is considered a separate class.

1.3 Research Challenges

The object detector is not capable of cleaning the objects, therefore, there is a necessity to train in all kinds of data related to the situation. The challenge arises only in resolution clarity, where the quality of object detection varies according to environmental conditions and is reflected in the video frames as discussed below because of specific environmental conditions and differences in camera quality. In this research, to avoid scaling the dataset and lengthy training time and enhance image resolution, Cascaded the DCGAN [Radford et al., 2016], which can be usedd to enhance the image resolution. Also, the generator part in DCGAN can generate realistic images of any objects that can then be used as datasets to train the object detectors; however, this technique is not implemented in this research, as well-established datasets are used to train the object detectors. For the research study, DCGAN was cascaded with an object detector to enhance the video frames' image resolutions, which can also be used to assemble more detectors. DCGAN is a state-of-the-art model that can generate the dataset, as it is discussed above, where the generator part of the DCGAN produces realistic images that can be used to train the detector for example, on paper [Kim. D et al, 2019].

The main motivation for combining two models DCGAN and SSD together is to create a DCGAN-SSD model to form a robust new model in this research and a solution for the object detection framework. The model DCGAN-SSD can be used for object detection applications and can be extended for different applications, such as self-driven cars, smart cities, and underwater object detection. The architecture is shown in Figure 1.1, combining the two models. Also, several test results in the future chapters show that the object has a very robust output, even in the most blurred images in underwater scenarios, and in different environmental conditions. Although it takes a considerable amount of time for the models to be trained, the use of supercomputers can speed this up. When output comes into the picture produced by the model, the time needed for this training is justified to get such an accurate output.

The challenge faced during the research was combining the models, as they needed to be trained separately, and then the models had to be extracted from the training. No preprocessing is done to the training dataset images, other than the scaling with a tanh activation function. The activation function is to work with the model's output; when a new class has been added, the trained model will be used in SoftMax layers. All weights are initialized to zero-centred distribution, with Standard Deviation (SD) 0.2. The Leaky Relu is used in DCGAN and SSD instead of the Relu activation function, and in the Leaky Relu the slope is set to 0.2. Also, when an application changes, the new set of training is

conducted for the particular application, and the model extracted from training. This problem is most common in Artificial Neural networks (ANN); therefore, when in research, these kinds of issues are most common time consumption, for instance.

1.4 Research Contribution

Modern-day technology primarily relies on Artificial Intelligence, Machine Learning, and Deep Learning, which is a subset of one another, in that object detection plays a vital role in many fields in public and private life, such as self-driven cars, track-and -trace, document classification, and traffic management in smart cities. The shortfall of object detection is in resolution: if the resolution is good, and if there is no problem in object detection due to resolution, there will not be any question about efficiency in object detection; but where resolution is poor, object detection becomes problematic. Chapter 2 shows that why most researchers are focused on the resolution issue in object detection and that legacy models are unable to perform well with multiple classes due to the stability and resolution issues. Therefore, this research mainly focused on the resolution issues. The following chapters in this thesis summarize the issues and how this research worked to neutralize the resolution problem in object detection. The research contribution is as follows.

Chapter 3 provides a detailed description of how object detection works using the traditional method HOG-SVM. Here, HOG has been used for image processing and SVM for classification. The chapter discusses the problem faced in object detection due to the resolution issues and the resolution impact in object detection. Experiments were conducted in different environmental conditions, and the results are presented in the chapter. These experiments were very helpful in identifying the problem of resolution which most of the object detectors are facing. Furthermore, the limitations of object detection which arose do not ensure the efficiency of the legacy object detectors. In later chapters, the research moved on to Deep Learning in response to the analysis based on the experiments and evolution of different object detection frameworks.

Chapter 4 discusses the adopted system models that set the stage for the research work presented in this thesis. Due to the legacy models' lack of efficiency, this research moved to Deep Learning. This thesis adopted the object enhancement system DCGAN and the object detector SSD as the DCGAN-SSD model. Since there was initially considerable complexity using this portion of the model on an exclusive basis, two system models, each representing systems that fall under each of the broad categorizations of the complex areas in this research work, are presented. Apart from the different scenarios adopted, the chapter also presents the necessity of combining these two models, when used under realistic, practical conditions. The system models consist of two parts, DCGAN and SSD, which will act as separate system models in operations DCGAN is used to enhance the quality of images, the enhanced images are then moved to object detector SSD where they are treated for object detection, and both the models are trained separately, thereby keeping the resolution issues to a minimum with the help of

DCGAN. The object detection task were evaluated on the CALTECH, ImageNet, CIFAR-10 and CIFAR-100 image datasets, and the proposed model achieved good performance in computation time with better performance, The proposed DCGAN-SSD model allows for fast, effective object detection using VGG-19 architecture with stochastic assessment. The research intends to proceed this way with VGG architecture's help, which made it possible.

Using this combinational model is very effective in different environmental conditions. This research compared the results obtained only from SSD object detector with the results obtained only from the combination of DCGAN-SSD. Results achieved by DCGAN-SSD outperformed SSD. This research tries to prove that, due to resolution issues, standard object detectors will fail to detect objects. But instead of training the detectors with all the possible resolutions, it is enough to train with one type, and then for variable resolutions, DCGAN will do the job in enhancing the images. Therefore, the message of is that, instead of inventing different object detectors for various resolution problems, a combinational model will work effectively with any object detectors. Various applications in later chapters will demonstrate this based on a DCGAN-SSD test case.

Chapter 5 tests the adopted DCGAN-SSD system using moving camera videos and several test cases, including a Uber accident video and a London taxi video both night and day. Taking into account the different testing environments in the moving camera, the time to process the frame per second (fps) is deficient, based on the low fps the system needed to perform at a high-level speed, In this research initially took the test case where a Uber self-driven car failed and killed a pedestrian, also took the same video and tested with DCGAN-SSD, where it was to detect the object with stopping distance. The detection result is shown in Fig. 5.2. Another challenging test case is with London taxi videos, which comprise footage from both nighttime and daytime, and the video is blurry. Both SSD and DCGAN-SSD were used, to perform this experiment. The DCGAN-SSD model has been trained with a CIFAR-100 dataset, with training data of 50000 training images and 10000 test images. Testing the model in the wild using London taxi videos and Uber failed test video, the DCGAN-SSD model gave good results.

Chapter 6 discusses underwater object detection where the test environment is entirely different, and the objects are different. In this chapter, the test environment is an underwater environment. The similarity between the previous chapter (Chapter 5) and Chapter 6 is that both use a moving camera. To test the underwater environment monochrome video camera was used. The environment underwater is utterly different from the above ground; the shapes of the objects are also different. Humans underwater will appear different compared with out of the water, and fish and marine vertebrates will keep changing shape from time to time. Another main challenge in underwater object detection is the contrast in lighting, as shown in Figure 6.1. Many videos shot underwater in the wild underwater suffer from resolution problems, and resolution issues can be found in all the footage reviewed in this chapter.

Similarly, simulation results show more than a four-fold improvement in detection results when DCGAN-SSD is used. Furthermore, the results show the efficiency of DCGAN-SSD detection, both for objects fully submerged and those half-submerged underwater. The people in the images are dressed differently, with oxygen cylinders and goggles, yet even so the DCGAN–SSD can manage this challenge and perform the detection perfectly. Still, the efficiency is reasonable when compared with only SSD. To accomplish this experiment, the different underwater datasets used, along with CIFAR-100, were CADDY and Roboflow for the objects, fish, plants, and marine vertebrates. The results chart for various underwater classes is also shown in this chapter.

Chapter 7 discusses the hyperparameter problem which causes object detectors, including DCGAN-SSD, to fail to perform and obtain efficiency. To solve this parameter problem manually would take days of training the model and gauging the performance to find the best hyperparameter to counter this issue. This research used PSO for object detector SSD to optimize the hyperparameters. Three hyperparameters have been optimized for learning rate, momentum and decay using PSO, which gave excellent results in object detection. The combinational model DCGAN-SSD already gave a good result; with PSO added to DCGAN-SSD, there was increased efficiency. The test were run with different videos in different environments, and the results are shown in the chapter with comparison results in Table 7.1.

1.5 STRUCTURE OF THE THESIS

The research work presented in this thesis focuses on finding how resolution affects the performance of object detectors and demonstrates a solution for addressing the resolution issue one of the main influencing factors in object detection which object detectors are facing. This research finds the optimal solution for handling low-resolution videos and enhances the frames using DCGAN; these feature-rich frames are very helpful in object detection with SSD. The thesis is organized into eight chapters: Chapter 2 is about the need for object detection and gives background about DCGAN and object detectors in general; Chapter 3 explains how resolution affects object detection and the problem of the traditional object detector HOG-SVM has dealing with different resolutions. Chapter 4 addresses the means of overcoming resolution issues with experiments in various environments using the combination of DCGAN-SSD. Chapter 5 describes a test experiment with a moving camera for self-driven cars and shows the performance where Uber failed. Chapter 6 deals with object detection in an underwater environment. Chapter 7 is about object detector optimization using PSO for optimizing object detection.

Chapter 2 describes the background and related work, shares views from related research literature and shows how researchers have tried to solve the problem in their own ways, where they have failed, and how the problem persists. This chapter helps the research to identify the problem facing the object

detection industry in the present scenario and establishes the motive for the research, which is to move object detection forward.

Chapter 3 shows the initiation taken in this research to analyse the effects of resolutions in object detection and how to encounter the problem using one of the legacy methods to understand the depth of the problem from scratch using the Histogram of Oriented Gradient (HOG) Support Vector Machine (SVM). The detected results obtained using the method are shown in the Figures.

Chapter 4 discusses the combinational method, where two state-of-the-art models, DCGAN and SSD, combined as DCGAN-SSD to solve the problem of resolution faced most commonly by object detectors in all the environment structures. Exciting results are shown in the form of Figures and Tables, which show the detection accuracy comparisons between using SSD only for object detection and when the combinational model DCGAN-SSD is used.

Chapter 5 shows the effectiveness of DCGAN-SSD in a moving camera and demonstrates that this combinational model can be used for self-driven cars. To check how safe it is, this chapter took a real-world scenario of UBER's test fail that resulted in killing a pedestrian and used the same video to analyse the detection quality. Those results are given in the chapter, which also took another scenario of a London taxis video to analyze performance during both night and day-time. Those results are also presented with Figures, Tables and Charts based on the detection accuracy under different classes.

In **Chapter 6**, an experiment is conducted in a complicated underwater environment, where the objects vary from their normal forms. The problem with resolution is more extreme. The video is captured using a monochrome camera to test the underwater scenario with two test cases, SSD and DCGAN-SSD. The results obtained are shown in the chapter.

In **Chapter 7**, a different problem (that of hyperparameter tuning) is addressed, and a solution is presented. This chapter utilizes PSO to tackle the hyperparameter selection. The research compares the results of DCGAN-SSD and DCGAN-SSD with PSO; they indicate that PSO improves the efficiency of object detection.

Chapter 8 presents this research work's conclusion by reviewing the proposed methods' contributions and discussing the practical implications for an efficient implementation of the DCGAN-SSD for real-time problem-solving. Future research directions on the implementation of DCGAN-SSD commercially for solving real-time problems are mentioned in this research, with some suggested industry collaborations.

CHAPTER 2. BACKGROUND & RELATED WORK

2.1 Object Detection

2.1.1 The requirement of Object Detection in various Applications

Video analysis and image understanding are closely related to object detection, and most recent research has paid attention to that [Zhao et al., 2015]. Object detection requires a method for deciding whether or not the targeted object within the class is present in the frame. Apart from the problem faced by object detectors and computer vision, object detection is a mandatory tool for most applications. Object detection is an emerging technology with the capability of identifying the class of moving objects in a video sequence; automated visual surveillance is a process of surveillance that includes a wide area of applications, such as human object detection at a distance, monitoring the congestion in traffic etc [Manjula et al., 2016]. It is used in surveillance, for robotic tasks, digital image databases, etc. [Kachouane et al., 2012]. The challenge of detection in computer vision has been a subject of research for several decades [Liu et al., 2020]. Object detection in the defence field is very crucial; automatic target detection is one of the areas where several algorithms have been designed and developed to detect and identify targets, but object detection can fail if the object's size is reduced, and the deep convolutional neural Network's (DCNN) performance can be affected by image blurring, additive noise, image contrast and loss in compression. A limited amount of research has been done to determine the limits in the performance of DCNN-based object detection performance when there is a variation in object size [Donohue et al., 2019]. [Manjula et al., 2016].

In the present framework approaches to object detection, performance efficiency is low for the detection of smaller objects and sometimes fails to detect objects with various geometric transformations [Cao et al., 2020]. Recently, substantial progress has been made in the case of facial detection [Schneiderman & Kanade, 2000; Viola et al., 2003]. Present systems achieve a decent percentage in detection rate, but when it comes to real-time, there is a performance backlog in the system and variation in efficiency. Most of detection systems require a huge training database to achieve good results due to variations in resolutions and environmental conditions. Therefore, the real-time search for a good object detector continues [Levi & Weiss et al., 2004]. Computer vision is a continuously growing field, and it is very hard for anyone to keep up-to-date, especially in the field of object detection. When considering self-driven vehicles, it is self-evident, due to all weather conditions that have to be considered (just to mention one parameter), that a massive amount of training is required [Janai et al., 2020]. Object detection and facial recognition can also be used for security purposes and are widely used in the public domain. While some privacy concerns are involved in scanning someone's face without permission, this is considered a minor concern when using object detection [Jiang. R et al., 2016].

Humans can detect an object at a far distance and instantly know what the image is. If the algorithms could do the same much faster, there would be no need for sensors in self-driving cars. The cars could be driven without any dedicated sensors [Du. S et al., 2019]. Generic object detection, which works in real-time, is much less based on Deep Learning. Deep Learning is characterized by many features and a robust representation of abilities (these can also be combined with hand-crafted features) [He et al., 2019]. The advantages of recent developments in technologies like autonomous vehicles have shown that precision and accuracy are not the only factors to be considered in addition, it is mandatory to consider the model to accommodate a realistic environment. Earlier methods, like YOLO [Redmon et al., 2016] and RCNN [Girshick et al., 2014], achieved high average precision (mAP), but only by using Graphical Processing Units (GPU's) [Pedoeem & Huang et al., 2018]. There is a problem with SSD: small objects are not correctly detected; in most cases, it misses the small objects. Also, SSD only looks for one layer for each scale. To overcome this problem, it is necessary from time to time to change the architecture in SSD from VGG to Resnet or to some other architecture, which is time-consuming as each architecture needs retraining. The scale-insensitive convolutional neural Network (SINet) [Hu et al., 2018], intended for ROI pooling, could not handle the structure of small objects, and also a single detection network cannot handle the large intra-class distance for a large variation in scales [Chen Z et al., 2019]. A trainable system for object detection in clustered and cluttered scenes is a highly desirable technique in the field of object detection [Papageorgiou & Poggio, 2004].

2.1.2 History of Various Object Detection Algorithms

CNN is a high trending technology in the field of computer vision, especially in object detection. Without CNN, object detection would not have developed in fields such as self-driving cars, smart city pedestrian detection, or underwater object detection. The very early publication of CNN is based on the self-organizing neural Network without affecting the change in positions [Fukushima, et al 1980]. In this research, SSD [Liu et al., 2016] is used as an object detector. By cascading two models together, SSD resolves the complexity with its neural layers in object detection when it gets high-resolution images. YOLO cascading with DCGAN could be carried out as another research topic. This research focused on SSD analysis as well as the integration of SSD and DCGAN for object detection. This may also allow identifying how the proposed model enhances the performance of a comparatively less efficient object detector. The development of another cascading architecture based on YOLO cascaded with DCGAN may require substantial work in cascading, training, and hyper-parameter tuning to achieve competitive performance. Therefore, it is out of the scope of this research study. SSD is a Deep Learning-based object detector that can be used in real-time object detection [Liu et al., 2016]. [Zou et al. 2019] proposed an exact and effective detection method called a Single Shot Object Detection with Feature Enhancement and Fusion (FEFSSD) to enhance and manipulate the surface and deep features in the feature pyramid structure of the SSD algorithm. To attain the performance level, the authors introduced the Feature Fusion Module and two Feature Enhancement Modules and integrated them into

the standard structure of the SSD [Zou et al., 2019]. A standard integration technique with SSD and DCGAN can be considered to perform feature enhancement and object detector combinations. Researchers are working to enhance object detection features such as real-time object detectors [Shi et al., 2019]. The problem of object detection has been worked on since the beginning of computer vision research. A recent breakthrough was the structured cascade detector proposed by Viola and Jones [2001]. Pedestrian detection is a vital task in modern driver assistance systems or self-driven cars. The deployment of lightweight feature-classification setups is possible to achieve through HOG-SVM [Bilal & Hanif, 2019]. Detecting vehicles and humans is highly challenging due to differences in appearance, background, contrast, and illumination [Chen Y et al., 2008; Xie et al., 2009; Jin et al., 2007]. Complexity in the computation prevents SVM in implementing real-time vehicle detection. The feature vector dimension is the influential factor in computing time for SVMs [Lee et al., 2015]. The shadows influence and create problems in detecting the objects if there is are shadows, those areas will not be detected by background subtraction with HOG-SVM [Ahmed et al., 2017].

Spatial pyramid pooling Net(SPP Net) [He et al., 2014] was introduced to overcome the fixed size problem, where most of the detectors are struggling due to the size variations, Their proposed SPP layer enables the Network to engender fixed length irrespective of the image size. As such, the SPP Net is comparatively faster than R-CNN [Zou et al., 2019]. Fast region based Convolutional Neural Network (R-CNN) [Girshick et al., 2014] is an updated version of R-CNN the model can train a detector and apparently enable a bounding box regressor under the same Network. The operation and the detection speed of the detector is accurate, and faster than R-CNN. Faster RCNN [Ren et al., 2015] proposed the model shortly after Girshick et al [2014], Fast RCNN which is a Deep Learning model. Most object detection blocks are integrated in a single pipeline as an end-to-end framework. When YOLO [Redmon et al., 2016] was introduced it was the first one stage detector in the world of Deep Learning and out-performed all previous object detection models. The Network divides the image into regions and predicts bounding boxes. Later, many different versions of YOLO were introduced [Zou et al., 2019]. SSD [Liu et al., 2015], considered the second single stage detector after YOLO, is considered a competitor for YOLO. Most object detection research is conducted based on these object detectors.

2.1.3 Architecture for Object Detector

A few object detectors can be used for real-time object detections, and several architectures can time object detections. Several architectures can fit the object detectors with convolution layers and activation functions. When there is a need for a suitable architecture for object detection, there needs to be a deep analysis of application and implementation, and an examination of R-CNN [Girshick et al., 2014], Fast R-CNN [Girshick et al., 2014], Mask R-CNN [He.W et al., 2019] which is an extension of Fast R-CNN, SSD [Liu.W, et al., 2016], and You Only Look Once (YOLO) [Redmon, et al., 2016]. These are the current object detectors; however, only two can be used for real-time object detection,

which is SSD and YOLO. SSD is more compatible for cascading with DCGAN and the deblurring model.

A number of different architectures that can be used in detectors include: LeNet-5 [LeCun, et al., 1998], AlexNet [Krizhevsky et al., 2012], VGG-16 [Simonyan & Zisserman, 2015], Inception-V1 [Szegedy et al., 2015], Inception-V3 [Szegedy et al., 2016], ResNet-50 [He et al., 2016], Xception [Chollet, 2017], Inception ResNet-V2 [Szegedy et al., 2017], ResNeXt-50 [Xie et al., 2017], and VGG-19 [Simonyan & Zisserman, 2015]. This research uses the VGG 19 architecture for the SSD mainly because of the architecture depth. The effect of the convolutional network depth is based on accuracy in large-scale image recognition, which is mainly needed for the feed forward Network where VGG 19 offers depth in the network and feature extraction. VCG 19 also has advantages in object detection.

2.2 Resolution Issues in Image Analysis

Object Detection techniques have been widely researched in recent years, and many methods have been recommended to solve the object detection challenge effectively. Most of the techniques deal with object detectors using SSD [Liu et al., 2016], Inception v3 [Szegedy et al., 2016] and many other detectors which follow the traditional object detection method of training models and updating datasets for new samples. Szegedy et al. [2015] used DCGAN for colourisation techniques for monochromatic problems in between grayscale and colour [Szegedy et al., 2015]. This chapter will examine the background literature on object detection, the different architectures involved in object detection, and what makes the research into the field of object detection different. The chapters below also present an overview of object detection under different circumstances and applications.

When considering object detection and unsupervised learning (which is related to computer vision and object detection), DCGAN [Radford et al., 2016] is the working model which is used in this research, which supports the work in both the context of computer vision and object detection in the form of vectors. In the case of object detection, images are classified under a group of patches [Coates & Ng, 2012] to powerfully represent images. Other applications can solve the challenges more complicatedly by training auto-encoders, using stacked convolution networks [Vincent & Larochelle, 2010]. This works by separating the object components in the code [Zhao et al., 2015]. The methods applied by DCGAN represent better learning features from image pixels than from auto encoders. There are two types of generative natural image parametric and non-parametric [Radford et al., 2016]. In this research, The research used non-parametric generative images with DCGAN, as vectors also use datasets such as patches of images, and the non-parametric approach offers super-resolution. This work has already been cited by other authors, such as Mv & Khan [2020], who discuss this research problem and solution in object detection and Kim. B et al. [2020] and Ayachi et al. [2020], who focus on the advanced driver assistance systems.

DCGAN is used to enhance the data and features from the objects [Bian et al., 2019] to make them suitable for detection. Another main intention of using DCGAN is image retrieval, as mentioned by [Zhang et al. 2018], with emerging hyperspectral images generated from different imaging sensors. The processing and analysis of images require effective image retrieval techniques through feature enhancement so that the images obtained from different resources with different resolutions are optimized for object detection; in this case, the images need to be retrieved by enhancing the features with the help of DCGAN [Dinakaran et al., 2020]. A novel small ship detection using GAN and YOLO v2 method was proposed by Chen Z et al. [2019], Specifically, a modified Wasserstein Generative Adversarial Network(WGAN) [Ren et al., 2015] with gradient penalty. According to work presented, WGAN with gradient penalty is first used to generate small ship training samples and enhance data. The enhanced feature is sent to YOLOv2 for object detection based on the augmented training samples [Wu et al., 2020]. To solve the issues in synthetic images for traffic sign detection, complicated images are created using DCGAN, with small amount of data stored as a solution for the synthetic image. DCGAN performs excellent image generation performance [Dewi et al., 2021]. Dewi et al used DCGAN synthetic data to obtain image data for traffic sign detection. The low-light environment is very close to each other daily life, and it is evident in many cases of research that low level lighting will affect object detection. Therefore, a preprocessed night image is used as the input signal in DCGAN network; the DCGAN system then generates virtual images similar to daytime scenes and the images are used with advanced detectors to complete the detection tasks, where the performance is good in night object detection [Wang C et al., 2020]. Chen G et al [2018] used GAN as a semi-supervised learning method to deal with labeled and unlabelled data. In this research, semi-supervised learning extracts useful information from labeled and unlabelled data to achieve a reasonable classifier, which can be taken forward for object detection [Chen Z (b) et al., 2020].

2.2.1 Data Enhancement from Low Resolution

With the development of huge volumes of high-resolution hyperspectral images produced by all sorts of imaging sensors, the processing and analysis of these images require effective retrieval techniques. This research used fewer samples than usual for training DCGAN, as the model has the ability to produce more samples for training, which can be used for training the object detector. Fang et al. [2019] used DCGAN for their experiment's gesture recognition to train the model with fewer samples, making strong representations for the images by extracting the deep features [Mahmoud & Guo, 2019]. For this reason, DCGAN was used for deep feature extraction in this research, as it allows for deep penetration of the object. DCGAN was also used to verify the distributions of the mean and standard deviation of fake fingerprints generated by DCGAN with those of actual fake fingerprints. It is the main area in the research to employ DCGAN [Radford et al., 2016] in order to generate enhanced training data from samples with problems such as decolorisation or low-resolution. In a second method, the mean Hamming distance, which is a method of evaluating the similarity of images, is used for measuring the

similarity between the generated fake fingerprints and the actual fake fingerprints [Choi & Jung, 2019]. To identify the correct features on which the detection should be conducted, DCGAN was utilized to increase the features and to identify the right images from the dataset used for the detector training. Technically, this involves enhancing the image features. Objects that suffered from high contrast or low resolution and images that were captured in different environmental conditions were preprocessed through the DCGAN and pipelined to the SSD training set. From this data, the object detector (in this case SSD) was trained to allow for high-quality and efficient detection.

2.2.2 Enhancing the Colouring Features

Another main reason why DCGAN is used in this research is to handle the unbalanced dataset [Du. Y et al, 2019]. It is not often straightforward to identify the number of samples suffering from fault conditions, but, comparatively, this is expected to be less than the number of samples under normal conditions in applications such as object detection, which is called an unbalanced dataset. To address this problem, in the research, research utilized DCGAN. In particular, this research constructed a framework using Deep Convolution Generative Adversarial Network (DCGAN) as a generator to generate images to balance the imbalanced data and used the Convolutional Neural network (CNN) model as a classifier to verify the classification results [Li et al., 2018]. Although this research did not delve into this intensively, also an attempt been initiated to create a pipeline between the generator and the dataset designated for the object detector. Experimental results show that the DCGAN Framework is able to synthesize real pedestrian images with diversity [Kim et al., 2019]. The model is used to add colorisation in places where it is lacking, and the DCGAN model learns to colorise the lagging areas [Suárez et al., 2017]. Enhancing the colours makes it possible to better identify the main features within an image, as when there are lagging colours there are missing features, creating the possibility of misdetection. As an example, DCGAN has been used to enhance the data in images of tomato leaf disease, with data augmentation being used to recreate the actual data instead of relying on unreal data. Here, the model is helpful in enhancing the real data from a given vector. [Wu et al. 2020] discuss the augmented real data as real outputs, which the DCGAN model achieved success in all the above experiments and in this research.

2.2.3 Under Water Object Detection

The most challenging aspect of this research is object detection underwater, mainly due to complicated objects, the variety of species, and the noisy underwater environment. Underwater object detection is used primarily by unmanned underwater vehicles and in research such as marine biology and marine ecology studies. For example, [Strachan et al. 1993] used object detection to detect marine life with different colours and shapes. In this research, the attempt made in underwater object detection to detect humans underwater with different costumes, such as carrying an oxygen cylinder or wearing a mask and glasses. The real challenge faced in underwater object detection is a scarcity of underwater datasets

involving humans. To rectify the problems faced in chapters and related research publications, the CIFAR-100 [Krizhevsky, et al & Hinton, et al 2009] dataset was used to train, and to test the research using random data available through different sources. This is necessary as issues persist, such as focal loss [Dinakaran et al., 2017], which can make it difficult to view such objects with the naked eye (due to resolution problems), and contrast and environmental conditions. Dinakaran et al.'s [2017] research highlight the losses that may occur with solid samples in training and validation. In contrast, in this research DCGAN does improve the factors, making it a better fit for self-placed learning [Kumar et al., 2010].

2.3 Hyperparameter fine-tuning via Particle Swarm Optimization (PSO)

Neural networks have been widely adopted in many fields with respect to object detection, speech recognition, and image description generation. However, these neural networks still depend on their hyperparameters, which control the activities of the process although the hyperparameters themselves do not directly control the process, to allow for maximum performance. In this research, PSO is used to fetch the hyperparameters for training and validation from the dataset the reason being, PSO is used to select the network configuration based on the particles in three dimensions. This research has optimized three parameters: the learning rate, momentum, and time decay. PSO is used widely in the field of data science to handle large datasets. It is also used in map-reduce to handle large datasets for parallel programming in distributed computing. When the CIFAR-100 dataset is used (which is also considered a large dataset consisting of 100 classes and with around 600000 images) to deal with large datasets, PSO helps train and validate by obtaining the correct hyperparameters from the training and validation sets. After the hyperparameters have been fetched from PSO, the entire dataset is trained for the model. The process with the particular hyperparameter is vast, but the efficiency is high. Figure 7.2 shows the results of video obtained from PSO hyperparameter. The results clearly show that the PSO-obtained training set gives accurate results. In most of the research, optimizing the hyperparameters is an obstacle, especially when it is needed to optimize Deep Neural Network (DNN) parameters with a large dataset [Lorenzo et al., 2017]. A similar problem is faced in this research, as at each occasion of training, there are different factors of efficiency in object detection when not using PSO. Variation with efficiency is only to be expected the research expects both steady efficiency and variation in efficiency, it is clear hypothesis to choose or find the best solution, and it is hard to make decisions. PSO solves this problem with efficiency by optimizing the parameters, and the efficiency is shown in the object detection.

2.4 Summary

In this chapter, the findings from various research and approach towards object detection, and resolution issues, affected other research and the background study about them is very helpful to find the way in

this research by considering a problem faced by other researchers and their experiments. The researchers used GAN and DCGAN for super resolution also to generate datasets with high resolutions, which is very helpful findings for this research. The findings give focus mainly on resolution issues faced by object detector and parameter optimization issues. The hyperparameter optimizations used in other research for different purpose is also a useful study for hyperparameter optimization in this research, all the important studies are addressed with experiments in this research.

CHAPTER 3. IMAGE RESOLUTION ISSUES IN OBJECT DETECTION

This research chapter examines the fundamentals of object detection and the factors influencing it. This is the most important aspect of this research and paved the way toward identifying the problem in object detection. In other words, the findings set the scene for this research. The research took the paradigm of object detection in video surveillance applications for smart cities, and automated traffic control and extended to object detection for driverless cars and underwater object detection in future chapters of this thesis. As object detection becomes a widely applied technology to help improve the quality of human life in the era of digital living, pedestrian detection is a key component for people-centered smart city applications and is cited as beneficial for wellbeing, security, traffic guiding, and unmanned vehicles. Currently, most surveillance cameras have low-quality resolution to save costs, and this means that image resolution (or lack of it) plays a major role in the detection, where accuracy becomes a primary concern. While lower resolution can save cost and processing time, it has so far not been adequately reported how resolution can impact detection accuracy. This chapter investigates the limit of low-resolution images regarding pedestrian detection accuracy and experimentally demonstrates its impact on the most widely applied HOG-SVM pedestrian detector. From the experiments, it is found that there is an optimal resolution to balance between speed and accuracy, while the show in this chapter that resolution obviously influences accuracy and computing time.

3.1 Pedestrian Detection in Smart Cities

Pedestrian detection is becoming a necessary technology in the modern world for many smart city applications [Zheng et al., 2016]. For example, pedestrian detection has been exploited in the automobile industry for driverless vehicles and in the identification of anomalous activities. While much work has been carried out in this area, there is still a big technical gap to bridge to meet the needs of practical applications [Zheng et al., 2016; Masoud & Papanikolopoulos, 2001]. Pedestrian detection systems often generate region-of-interests (ROIs) using a sliding windows algorithm [Papageorgiou & Poggio, 2004; Dalal & Triggs, 2005] to thoroughly examine the entire image. So far, most pedestrian detectors [Zhang. Y et al., 2016; Dalal & Triggs, 2005] are based on the Histogram of Oriented Gradient (HOG) that performs better pedestrian detection. Figure 3.1 shows an example of a HOG-based pedestrian detector. Using HOG for pedestrian detection in many experiments has been a success. However, when it comes to practical applications, pedestrian detection needs to attain both high accuracy and real-time performance. Particularly, most surveillance cameras are of very low quality and take images from a great distance, making pedestrian detection or object detection a more difficult task to achieve.

While the video resolution may play an important role in detection accuracy [Enzweiler & Gavrila, 2011], it is unclear how image resolution will impact a pedestrian detector, especially in experiments.

This makes it hard for engineers designing smart city applications to estimate the performance of their systems with respect to the camera resolution they choose. Many systems have an emerging need to save processing time, which prefers a lower resolution to attain better speed, but accuracy can be drastically undermined by lowering resolution. Hence, finding a way to balance resolution with computational time and detection accuracy is particularly important. This question has so far not been well answered according to the review of the academic literature. Though a recent theoretical analysis on Resolution [Enzweiler & Gravila, 2011] has been conducted, there is still a need for validated results to answer this open question. In this chapter, the research carries out a number of experiments on a HOG-SVM pedestrian detector and offers a clear answer to the above question through experimentation for example, Figure 3.1 shows the object detected inside the region of interest, and the video was taken inside Northumbria University campus to hopefully facilitate the engineering problem of visual surveillance in smart cities [Zheng et al., 2016].

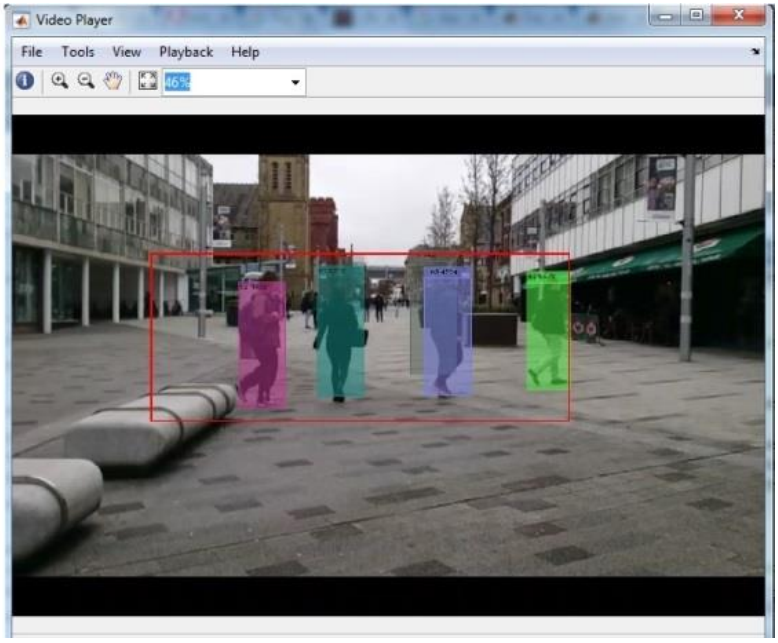


Figure 3.1. Pedestrian detection via the HOG-SVM detector.

3.2 PEDESTRIAN DETECTION USING HOG-SVM

3.2.1 Histogram of Orientated Gradient (HOG)

HOG has been successfully applied to the challenge of pedestrian detection, and it has a history which has been shown to work for other object types, such as vehicles. HOG data output (descriptor) is used to train classifiers. Different sets of training data, including a variety of sizes and views of people, are processed within a bounding box. In the standard Framework [Dalal & Triggs, 2005], the HOG descriptors are often extracted from a known dataset (such as the Caltech Pedestrian dataset, ETH, EPFL [Enzweiler & Gravila, 2011]), Where ETH is a pedestrian dataset. The dataset often includes samples of various poses and viewing angles to cover a wide range of pedestrians. The dataset was used

to generate both positive and negative images i.e., those with pedestrians and without pedestrians. Support Vector Machine was subsequently used to classify the HOG descriptors generated from the image dataset.

To estimate HOG, the orientation of gradients is first computed, as shown in Figure 3.2. Following this, the objects are treated separately, and the features are extracted from the images and divided into cells. The objects or images have a dense grid, and in each grid there will be the presence of a cell, and in each cell a pixel gradient is picked and the orientation and magnitude figured out. The histogram is created in the cell and is rated concerning the magnitude of gradients. For an image, the bins are allocated with different colours from 0 to 180 degrees, and the image bin for every pixel in the cell will be examined to determine the orientation as per the gradient magnitude. The next step is to combine several cells to create a block, allowing the normalizing of the images or objects and providing contrast normalization. This normalization is used to surmount the problem that might arise due to light intensity and to provide accurate data. This will create the blocks and collect HOGs over the detection window,

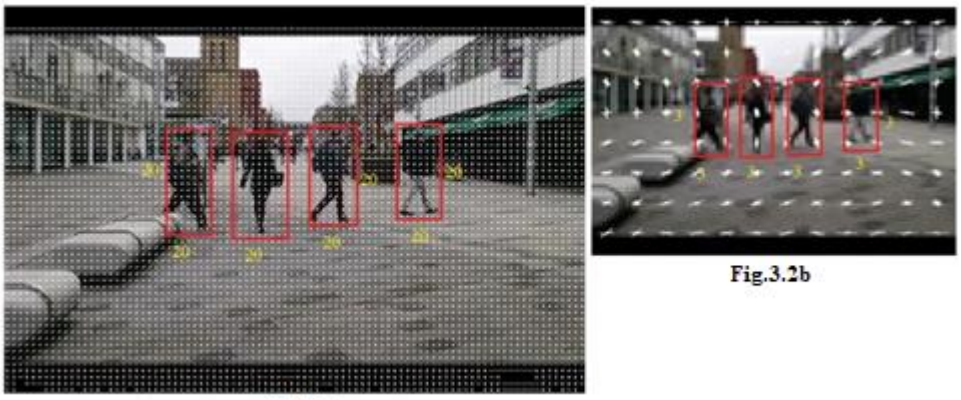


Fig.3.2a

Fig.3.2b

Figure 3.2 (a) The HOG visualisation of a high-resolution image in 301x202 pixels, where 6167 cells, 9.85 pixels in each cell, and an average of 59.41 blocks are used to create the histogram. (b) The HOG visualisation of a low-resolution image in 191x135 pixels, where 149 cells 0, 1.27 pixels in each cell, and an average of 55.5 blocks are used to create the histogram. All these measured uses Image J tool.

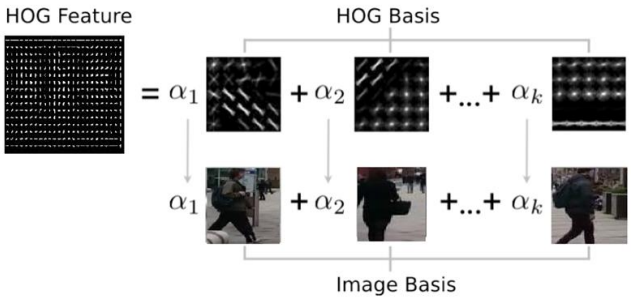


Figure 3.3. HOG [Dalal & Triggs, 2005] vector projected initially. By jointly learning a coupled basis of HOG features and natural images, it can transfer the coefficients to the image basis to recover the natural image.

as shown in Figure 3.3. HOG feature descriptor used for pedestrian detection is calculated on a 64×128 patch of an image. The image may be of any size. Typically patches at multiple scales are analyzed at many image locations. The only constraint is that the patches being analyzed have a fixed aspect ratio. In this research case, the patches need to have an aspect ratio of 1:2. For example, they can be 100×200, 128×256, or 1000×2000 but not 101×205.

To illustrate this point, shown in Figure 3.3 a large image of size 720×475. The part of the image selected a patch of size 100×200 for calculating HOG feature descriptor. This patch is cropped out of an image and resized to 64×128. Now this will make it ready to calculate the HOG descriptor for this image patch. First the horizontal and vertical gradients need to be calculated, after the histogram of gradients is calculated, this is achieved by using the filter in the code. From Figure 3.3, k denotes infinity.

3.2.2 Support Vector Machine (SVM)

SVM is used in this research chapter to separate both positive and negative samples with a hyperplane in between both samples. In the view of SVM sample separation, the positive samples and negative samples are differentiated with 1 and -1 labels, as given in decision equations 3.1 – 3.3, respectively, and it is evident that pedestrians are expected to be detected in the window, so all the computed images with pedestrians are labeled 1 while the negative images are labeled -1. When applied to pedestrian detection, HOG features from the search window of each frame are fed into SVM for classification, and the result will decide if there is a pedestrian in the scanned window. If a window is asserted positively, the window is then labeled out, as shown in Figure 3.4.

Mathematically, the samples in SVM are separated with the decision rule:

$$\vec{x}.\vec{w} = c \text{ (the point lies on the decision boundary} \quad (3.1)$$

3.4a. High-res image

3.4b. Low-res image

Figure 3.4. Object detection with different image resolutions. Difference in resolution from the same image with different pixel quality and the alignment in cells are illustrated.

$$\vec{x}.\vec{w} > c\ (Pedestrian) \qquad (3.2)$$

$$\vec{x}.\vec{w} < c\ (Non\ Pedestrian) \qquad (3.3)$$

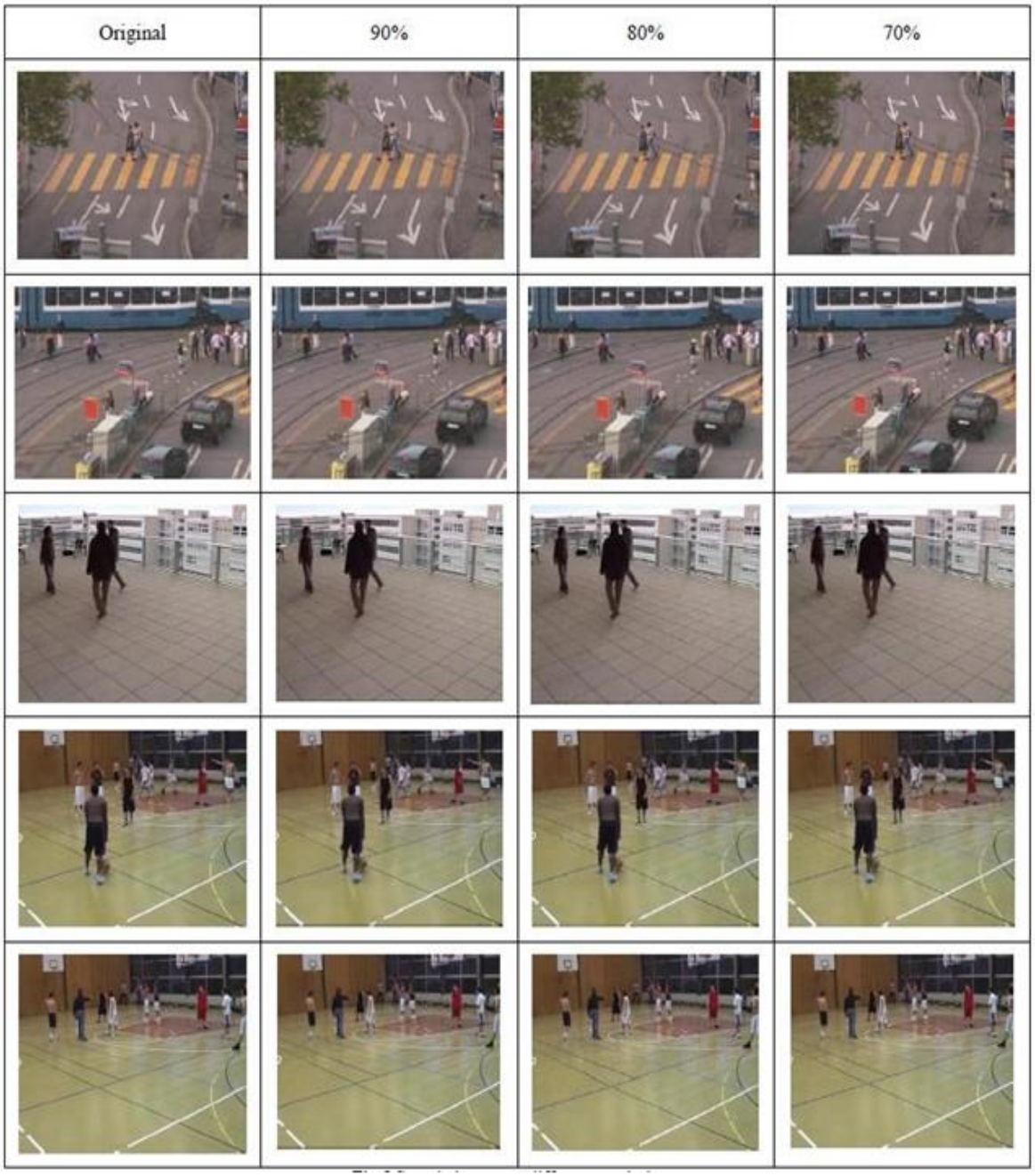


Figure 3.5 Sample images at different resolutions ETH, EPFL

In common, the projection of any vector or another vector is called dot-product. Hence, the dot product of x and w vectors are taken into consideration. If the dot product is greater than 'c' then it can be said that the point lies on the right side. If the dot product is less than 'c' then the point is on the left side, and if the dot product is equal to 'c' then the point lies on the decision boundary. The bias term is, indeed, a special parameter in SVM. Without it, the classifier will always go through the origin. So, SVM does

not give the separating hyperplane with the maximum margin if it does not happen to pass through the origin, unless you have a biased term. Therefore, the bias term is necessary for SVM.

3.3 HOG-SVM based Pedestrian Detection with Various Resolutions

The HOG descriptors discussed so far are the most widely applied features for pedestrian detections, even on low resolution video frames [Enzweiler & Gavrila, 2011]. Numerous studies [Ma et al., 2011; Wang et al 2015] have introduced the idea of combining diversified descriptors to enhance detection performance. In this work, the HOG-SVM was used as the first step research target and investigated how the resolution will impact the HOG-SVM detector's performance. This is particularly important for smart cities' surveillance engineering or traffic control. While thousands of cameras will be installed in smart cities, the chosen camera resolution is relevant to the data amount generated from these cameras for processing. There is a vast difference in cost; hence,
engineers need to know the best acceptable resolution for pedestrian detection, weighing up the gains in processing time against the cost of cameras.

3.4 RESOLUTION ISSUES IN PEDESTRIAN DETECTION

Even though HOG features are considered for machine learning, to be considered on the various applications, it depends on how well objects fit in HOG space this is an added challenge to face in HOG methods. If the model could enumerate human vision on the HOG feature space, it could get vision into the concert of HOG with a perfect learning algorithm. The built-in interface is given to look at HOG visualizations of window patches at the exact resolution as deformable part models. This chapter's work is based on classifying a HOG visualization as a positive example.

In this research project, the main finding is that resolution highly impacts the efficiency of a descriptor due to the reduction in cells from different resolution images. As shown in Figure 3.7. the same images are considered under different resolutions to check the efficiency in object detection. The same set of images with different resolutions shown in Figure 3.7 with detection difference, in the detection shown it is clear the variation in pixels will also affect the efficiency in detection results. This challenge is the main motivation for this research to carry forward with Deep Learning models such as DCGAN and SSD. These results suggest that some performance space is still to be squeezed depending on the resolution of images. It may be that focusing effort on using better features that capture more refined details and higher-level information will lead to substantial performance improvements in pedestrian detection of cells. The resolution plays a vital role in pedestrian detection. As it can be seen in Figure 3.6 mAp with recall rate, also in Figure 3.7, it can be witnessed when There is higher resolution, the greater the potential detection, with precision and recall graph. A higher resolution will also increase the maximum speed of detection. The resolution refers to the capability of the eyes to examine or

determine the object clearly with definite boundaries. A pixel is a unit of the digital image. Resolution depends upon the number of pixels, usually within any human eye or lens setting.

The resolution of the pixel is proportional to their size of them. If the size of the pixel is smaller, the object in the image will be clearly presented. Images with smaller pixel sizes might consist of more pixels. The number of pixels correlates to the amount of information within the image. For example, in Figure 3.5 the difference between the same image with different resolution and variation in pixels is clearly shown. Figure 3.4 shows the difference in the image quality between the two pictures. The Figure with broken pixels is the kind of image that is produced by closed circuit television (CCTV) cameras. The main problem with CCTV cameras is that they do not produce perfect quality outputs. Also, surveillance camera operators have little understanding of the effect of having and using low-quality surveillance cameras for security tasks. This makes it challenging for anyone hoping to detect people in low-quality, low-resolution video footage, especially if the person hides in a shaded region, making it doubly difficult to find or detect them.

3.5 EXPERIMENTS AND DISCUSSIONS

3.5.1 Datasets

To validate the impact of resolution on pedestrian detectors in this research, prepared a dataset of pedestrian images, and also collected 20 images at a resolution of 512×500 pixels from the ETH dataset; the images were then resized to 90%, 80%, and 70%. As a result, got four sets of images. Figure 3.5 shows the sample images for the experiments conducted. The HOG-SVM pedestrian detector was implemented in Matlab. With the above datasets, it is possible to run the tests on the dataset at different resolutions to see how the HOG-SVM detector can perform at different resolutions.

3.5.2 Evaluation Methods

The experiments were carried out on the datasets at different resolutions. The evaluation was done with three criteria: detection accuracy, recall rate, and F-measure. The test was conducted by running codes on the test videos in the experiments. Pedestrians detected were counted manually, as were how many missed and how many detections were false positive. With these figures, the results were

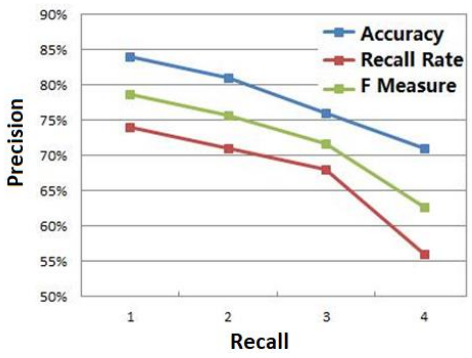


Figure 3.6 Mean Average Precision Table

Summarised into the above three measures with the graph shown in Figure 3.6.

Considering N pedestrians in the test images, and with K pedestrians detected and L false positive detections, the accuracy was calculated by the following formula:

$$A = \frac{K}{K + l} \qquad (4)$$

And the recall rate was calculated as:

$$R = \frac{K}{N} \qquad (5)$$

Finally, the F- measure was used to measure the overall performance

$$F = 2 * \frac{A * R}{A + R} \qquad (6)$$

Usually, accuracy may be contradicted by a recall rate. For example, the detection could be 100% correct, but only 1 pedestrian among 100 may be detected, leading to a recall rate of 1%. Hence, the accuracy or recall rate itself cannot fully demonstrate the detector's performance. More convincingly, the *F* measure could be better used for evaluation.

3.6 Experimental Results

The experiments were run on a laptop with 8GB memory and 3.0GHz. The code was run on all images, and the number of detected pedestrians (K) and the number of false detected regions (L) at each resolution were computed. Finally, the accuracy, recall rate, and F-measure based on each set of images (by resolution) were obtained. Figure 3.7 shows the detection results of the sample images from Figure 3.5. It is observed that more pedestrians were missed in detection with a decreased resolution, along with more false positive detections. For example, in the 20th image of Figure 3.7, it can be clearly seen that there is a miss in detection, while at 80% resolution, the detection is with occlusion, proving a resolution impact. Figure 3.5 contains 20 images with different resolutions, with the images presented before the code is run on the images. Figure 3.7 contains 20 images with different resolutions processed

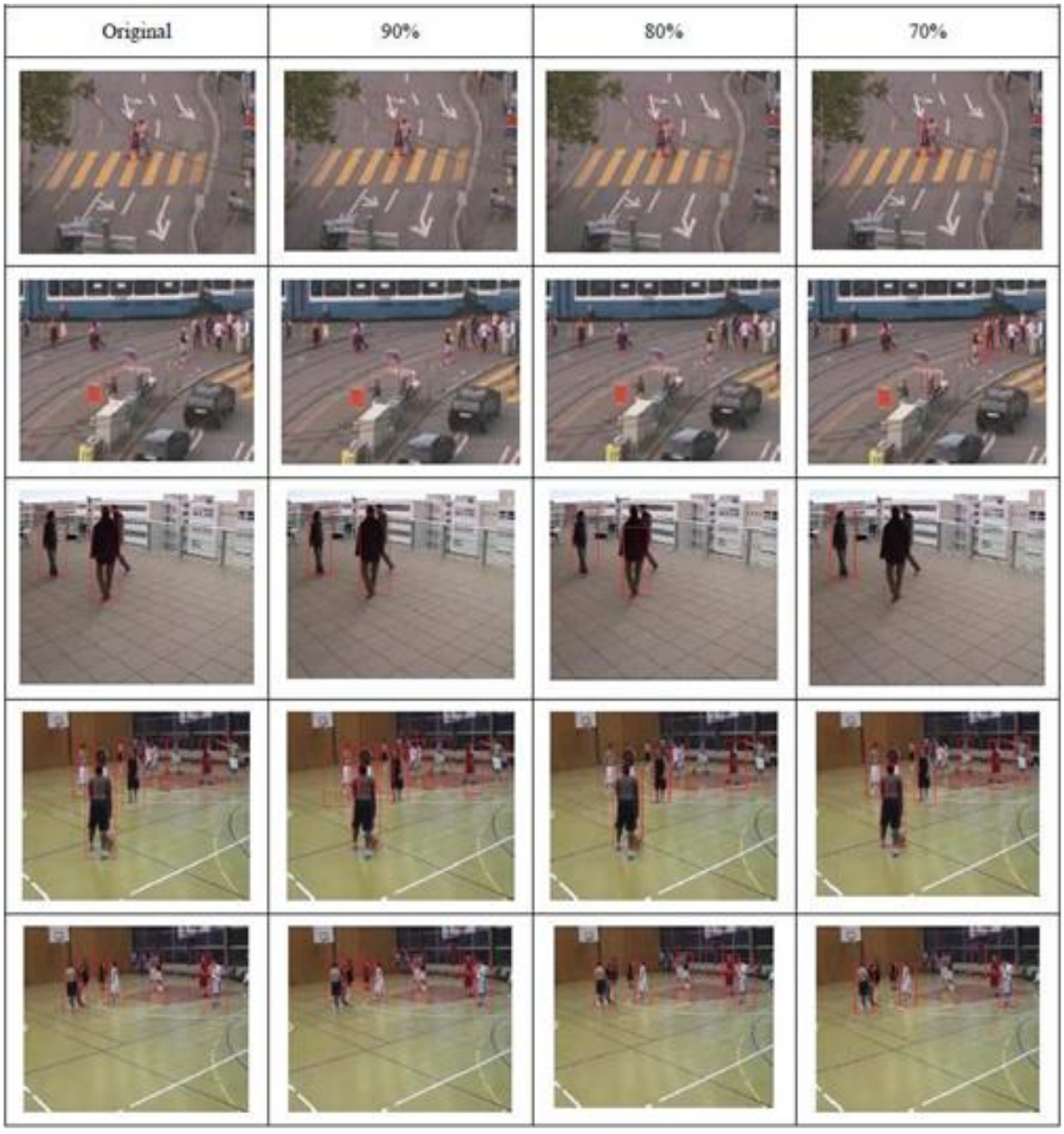


Figure 3.7 Detection results on sample images at different resolutions, Images from ETH and EPFL datasets[ETH, EPFL].

with the code on the images, where the images with 100% resolution offer good detection. There is clear variation in the object detection due to variation in the resolution. The miss rates per image were also recorded and computed, and the evaluation measures were computed as shown in Eq. (3)-(4). Figure 3.6 shows the results of the evaluation measures. It is identified that both accuracy and recall rates decrease following a reduction in resolution. As a result, the F-measure also decreases, meaning that resolution plays an important role in the HOG-SVM pedestrian detector. When deploying

surveillance cameras in smart cities, it is essential to consider these results and try to balance the cost of cameras and the detection performance of software applications. By considering these problems, the research moves forward in the following chapters to apply Deep Learning to object detection with low-resolution images.

This research may need further theoretical analysis to explain why resolution is so important in the HOG-SVM detector, which will form part of future work with Deep Learning models. It is assumed that when resolution is lowered, the features from images are blurred, and as a result, HOG features may become less discriminable. Hence, constructing resolution-invariant features or descriptors and moving forward with Deep Learning-based pedestrian detection with a robust architecture like VGG, Inception and SSD, Café or Zoo models with Python could be a potential solution instead of traditional solutions with MATLAB for more robust pedestrian detection.

3.7 Summary

While lower resolution can save computation time, it has not yet been verified how the resolution can impact accuracy. In this chapter, the research investigated the limits of low-resolution regarding accuracy in pedestrian detection. The research experimentally demonstrated its effects on the well-applied HOG-SVM pedestrian detector (a combination of HOG and SVM for pedestrian detection). The experiments conducted in this research found that there is an optimal resolution to balance speed and accuracy. At the same time, it was revealed that the resolution apparently influences both accuracy and computing time, presenting a question for practical applications that will be further addressed in the following chapters in this thesis. In smart cities, video surveillance is becoming a widely applied technology to help improve the quality of life in the era of digital living, and also pedestrian detection is a key component for people-centered smart city applications, including wellbeing, security, traffic guiding, and unmanned vehicles. So far, most surveillance cameras have low-quality Resolution for cost-saving reasons, and the impact of image resolution on detection accuracy is a major concern. This research provides an experimental answer on how image resolution could impact pedestrian detection for engineering smart city applications.

CHAPTER 4. RESOLUTION ENHANCED OBJECT DETECTION VIA DCGAN-SSD FRAMEWORK

Generative Adversarial Networks (GANs) have provided a promising Deep Learning model for many computer vision problems due to their powerful capabilities to enhance data for training and testing. In this part of the research chapter, the work involves leveraging a DCGAN model and proposing a new architecture with a cascaded SSD for pedestrian detection at a distance, which is a challenge due to the various sizes of pedestrians in videos at a distance. To overcome the low-resolution issues in pedestrian detection at a distance, DCGAN is employed to improve the resolution and to reconstruct more discriminative features for an SSD to detect objects in images or videos. A crucial advantage of this method is that it learns a multi-scale metric to distinguish multiple objects at different distances under one image, while DCGAN serves as an encoder-decoder platform to generate parts of an image that contain better discriminative information. To measure the effectiveness of the proposed method, experiments were carried out on the Canadian Institute for Advanced Research (CIFAR) dataset, and it was demonstrated that the proposed new architecture achieved a much better detection rate, particularly on vehicles and pedestrians at a distance, making it highly suitable for smart city applications that need to discover key objects or pedestrians at a distance.

4.1 Introduction

Locating pedestrians on streets has become an important task in many smart city applications, such as careless cars using car cameras, forensic surveillance in smart cities, hospital surveillance on patients, etc. Recently, deep CNNs have shown promising results for automated object detection in videos and images, particularly in detecting deformable objects such as pedestrians or faces [Dinakaran et al., 2017; Ren et al., 2015; Dosovitskiy et al., 2016; Liu et al., 2016; Storey et al., 2019].

However, it is still a challenging issue, as real-world applications demand very high detection rates under critical conditions, such as detecting pedestrians at a distance. A missed detection, for example, in driverless vehicles, can result in an unrecoverable disaster. This research reports a combinational architecture designed for object detection at a distance to address this challenge. While objects at a distance are usually blurred due to low resolutions in images or videos, DCGANs [Radford et al., 2016] have shown an extraordinary power to improve the quality of images or videos from low resolution [Goodfellow et al., 2014; Wang .K et al., 2020; Chen Z(a) et al., 2019; Finn et al., 2016] due to the "generative" nature of DCGANs. By learning from realistic data, DCGANs can produce faked images or videos that are very close to real data. Taking advantage of DCGAN makes it easy to cope with pedestrian detection at a distance, which is usually associated with challenges due to small sizes and vague visual features.

Based on these assumptions, this research developed a cascading architecture for pedestrian detection based on cascading a DCGAN with an SSD [Liu et al., 2016]. The DCGAN plays as a feature enhancer to rebuild the more realistic features of objects that appear at a distance and achieve a higher resolution for the detection under all stages in object detection. As a wider context for the work, combining DCGANs cascaded with SSD can extract the right amount of data in each frame, where objects appear at a distance in a video. The research contributions in this work may include:

- Presenting a framework for object detection at a distance by combining DCGANs with SSDs.
- Experimentally testing if DCGAN-based enhancement on features can help improve the detection rates for SSD.
- Developing a practical method for pedestrian detection at a distance is highly in demand by real-world smart city applications such as driverless cars and density-based traffic signals.

The evaluation is carried out via the quantified cross-validation on the Canadian Institute for Advanced Research (CIFAR) dataset and illustrated by real-world examples of street videos for qualitative comparison.

4.2 Proposed Cascaded DCGAN-SSD

Object detection is considered to be a major challenge in computer vision, although there has been some success in recent years thanks to advances in Deep Learning technologies. Among different Deep Learning architectures, DCGANs are one of the most interesting. As shown in Figure 4.1, the standard architectures of DCGANs often consist of two parts: a generator that can produce high-resolution images from low-resolution inputs, and the other is a discriminator that copes with the capability of verifying that the quality of generated high-resolution images against the real images.

Within various Deep Learning-based object detectors, You Only Look Once (YOLO) [Redmon et al., 2016] and SSD [Liu et al., 2016] are the two state-of-the-art methods that quickly capture the object regions. This research has chosen SSD as object detector and cascaded it with a DCGAN generator. The proposed combination aims to improve the detection rates of distant objects in images or videos by simply taking advantage of the DCGAN generator to produce higher-resolution features of the objects. The pipeline of the whole architecture in Figure 4.1 can be depicted as follows. The SSD, which is employed in this work, was first trained on ImageNet. As such, in this research, the SSD is trained with CIFAR-100. The high-quality features reconstructed from DCGAN are then fed into the trained SSD. The data propagates through the examiner's convolutional features from all layers, max pooling each layer's representation to produce a 4×4 spatial grid. These features

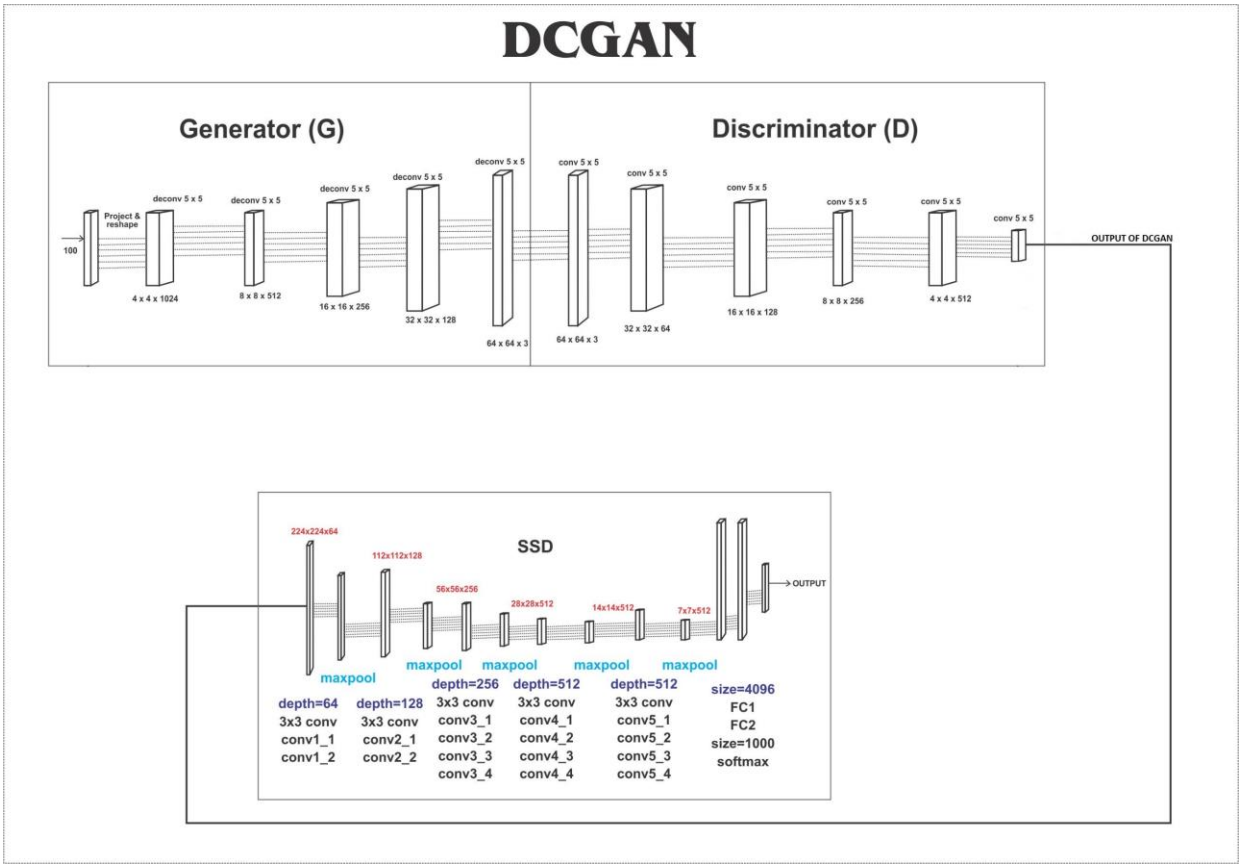


Figure 4.1. The proposed architecture of DCGAN-SSD where DCGAN [Radford et al., 2016] is integrated with SSD [[Liu et al., 2016] as an end-to-end system.

are then flattened and concatenated to form a 28672-dimensional vector, and a regularised linear classifier is trained on top of them. Notably, the SSD detector performance lags in the scale factor, where the DCGAN can improvise the scale factor in SSD by providing the detector with super-resolution images. Each feature layer can produce a fixed set of predictions for detection using convolutional filters in the convolutional detector, as shown at the top of the SSD architecture of Figure 4.1. The SSD is associated with a set of bounding boxes having default sizes relating to each cell from each feature map. The default size of bounding boxes is extracted in a convolutional manner with the feature map. The position of each bounding box is related to its size in the cell, and the class indicates the presence of the feature cell in the cell bounding boxes.

When the input size is considered, the DCGAN model requires an input image resolution of 32x32, which is hardcoded based on the 32x32 input image size given to the discriminator in this research. Therefore it obtains the output image resolution of 32x32. However, it is possible to get the different output resolutions for various image sizes, creating instability between the generator and the discriminator. To avoid the instability, some changes can be made in the code to achieve a variable output image, but this is harder to achieve. The model has several parameters, and in this research, the main hyperparameters associated with the final output include the learning rate, training epoch, batch

size, momentum, and weight decay. In this chapter, no hyperparameter optimization is conducted s for SSD. The optimization is in this chapter is performed by a trial-and-error method with training and validation sets. Since the backbone layer VGG is trained in SSD, the fully connected layer has been removed.

4.3 Experiment Results

In this initial stage of Deep Learning research, the research work used the CIFAR-100 dataset to carry out the validation and examined if the proposed DCGAN-SSD architecture outperforms the single SSD, particularly on distant objects or for pedestrian detection.

In this research, DCGAN was implemented in two steps. In the first stage of this experiment, the Fig 4.1 architecture is just a prediction of the architecture structure, with the actual architecture designated in Python code using PyTorch and Tensorflow, as illustrated by [Simonyan & Zisserman, [2015]; Zisserman et al., [2015]; Denton et al, [2015]; Ledig et al., [2017], which recreates the state-of-the-art DCGAN results using multiple object encoding. In this research, DCGAN was trained on the CIFAR-100 dataset, with all images sized 32×32, the batch size of 72, and 25 epochs across 60,000 images. The generator, G, is designed to map the latent space vector (z) to dataspace. Since the given input data are images, converting z to the data-space process eventually creates an RGB image of the same size as the training images (i.e., 3x32x32). This is accomplished through a series of strided convolutional transpose layers, each paired batch norm layer and a LeakyReLU activation. The output of the generator is fed through a tanh function to obtain the output data range of [-1,1][−1,1]. After the convolutional transpose layer, the batch normalization function works as a contributor as these layers help with the flow of gradients during training.

The discriminator, D, is a binary classification network that takes an image as input and outputs a scalar probability that the input image is real. Here, D takes a 3x32x32 input image, processes it through a series of Conv2d, BatchNorm2d, and LeakyReLU layers, and outputs the final probability through a Sigmoid activation function. This architecture can be extended with more layers if necessary based on use case, but there is significance to the use of the strided convolution, BatchNorm, and LeakyReLUs. Also, batch norm and LeakyReLU functions promote healthy gradient flow, which is critical for the learning process of both G and D. With respect to the model training, a total of 25 epochs are used, considering the same result obtained even after 25 epochs, therefore 25 epochs used to obtain the desirable performance.

Figure 4.2 shows the detection results on the CIFAR-100 dataset, using DCGAN-SSD and SSD only, respectively. Figure 4.2a is the results from SSD only, where many objects were missed in the detection. Figure 4.2b shows the results from DCGAN-SSD, where the number of missed objects in Figure 4.2a

were detected successfully. Mainly, objects at a distance were detected in these images, which were mostly missed by the SSD only method.

a) SSD only b) DCGAN-SSD

Figure 4.2 Detection results for the comparison between a) detection performed only with SSD, b) detection performed by DCGAN-SSD.

Table 4.1 shows the detection rates on 100 images from CIFAR-100 with distant objects in them. The statistical results clearly show that a new research combination of the proposed DCGAN-SSD framework can easily outperform the standard SSD, simply due to the improved detection rates on tiny objects at a distance. The object detection rate was improved from 35.5% with only SSD to 80.7% with DCGAN-SSD, which is an average calculated from Table 4.1. Specifically, the individual results for different classes in DCGAN-SSD and only SSD for object detection are calculated manually in each frame for 100 frames. When the model was evaluated with 30 epochs and 25 epochs, the result was the same, as the model received the same maximum output efficiency. Considering increasing the epochs will lead to an increase in computation time, the model is limited to 25 epochs, with this concrete reason, the model is trained only with 25 epochs. Figure 4.3 shows the overall detected frames from DCGAN-

SSD, and Table 4.1 and Figure 4.4 show the difference in the number of objects detected by SSD and DCGAN-SSD. Figure 4.5 shows the accuracy difference between SSD and DCGAN-SSD.

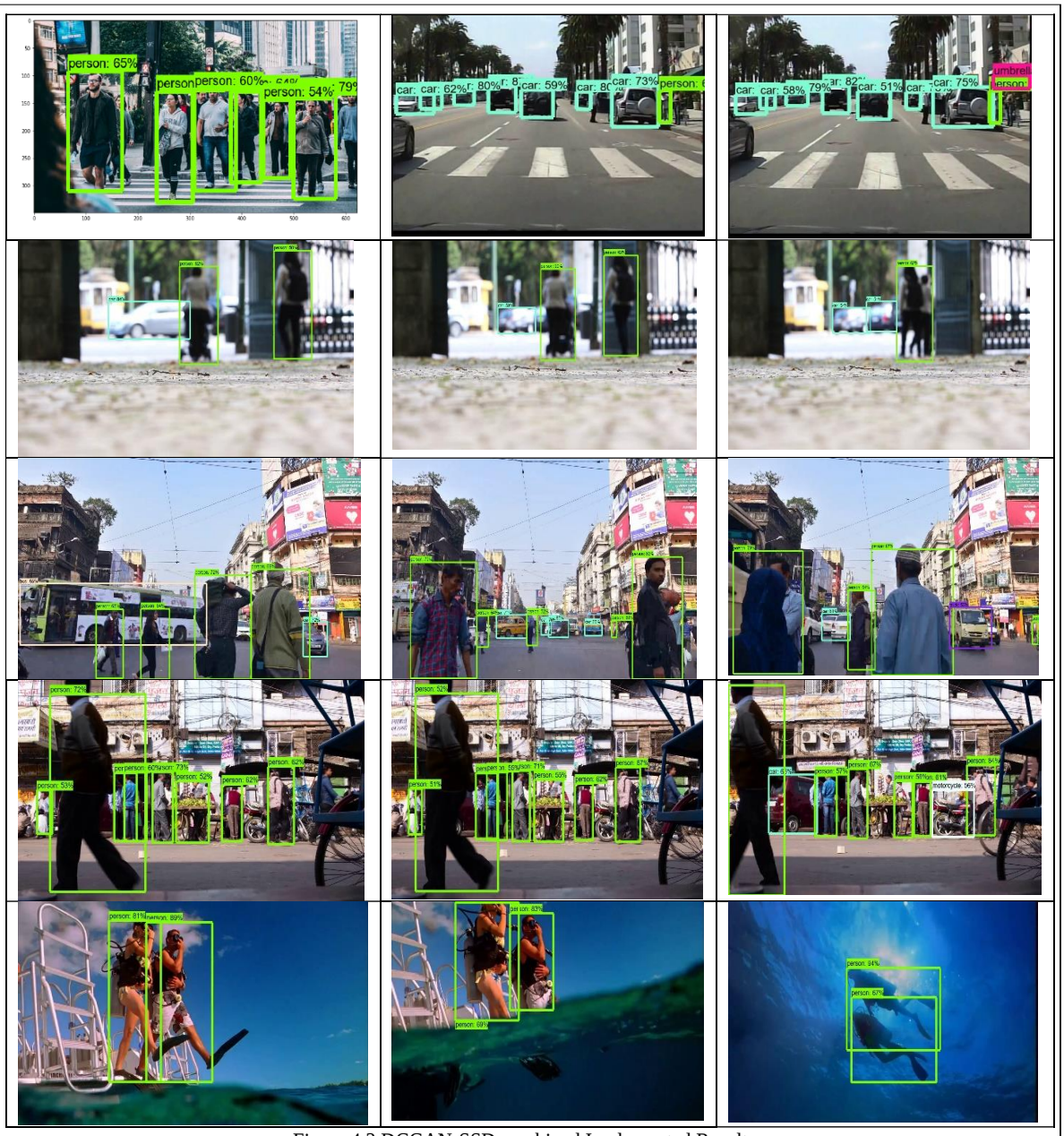


Figure 4.3 DCGAN-SSD combined Implemented Results

	Detection of objects		Percentage of Detection/100 Images		Detection Instances
Categories	SSD	DCGAN-SSD	SSD	DCGAN-SSD	
Human	333	596	46%	82%	721
Vehicles	62	206	26%	89%	230
Cyclists	12	34	29%	82%	41
Children's	5	13	36%	68%	19
Animals	0	9	0	75%	12
Traffic Lights	25	32	69%	88%	36

Table 4.1 The comparison results between two experiments performed by SSD and DCGAN-SSD, for different object categories respectively

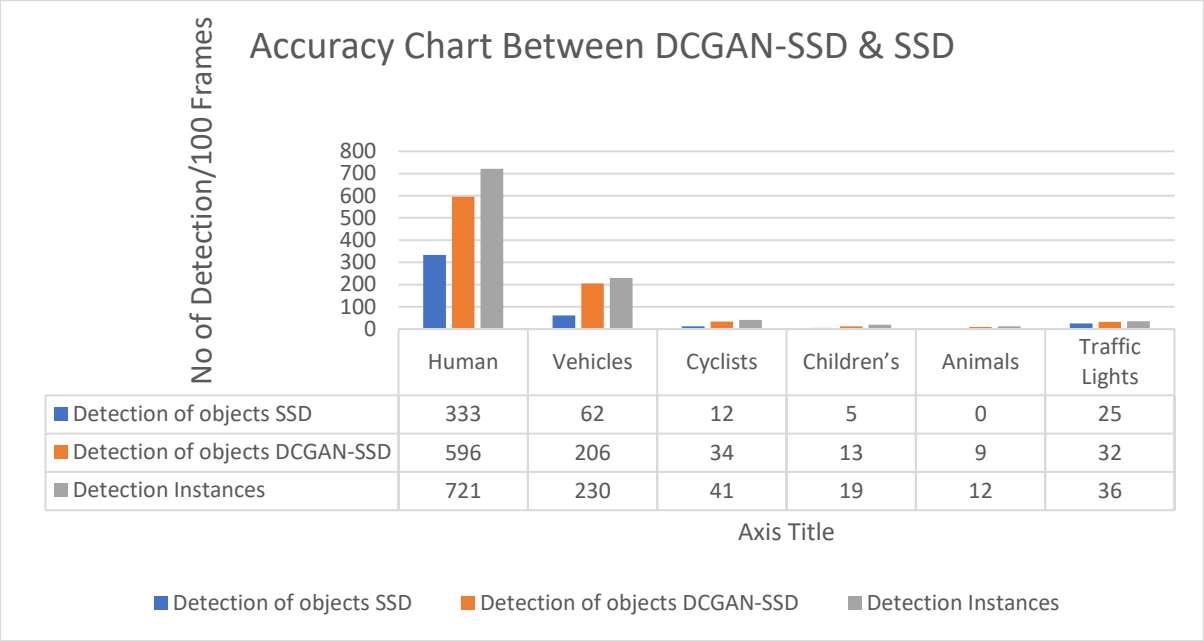


Figure 4.4 Shows the number of detections between SSD and DCGAN-SSD under most important categories

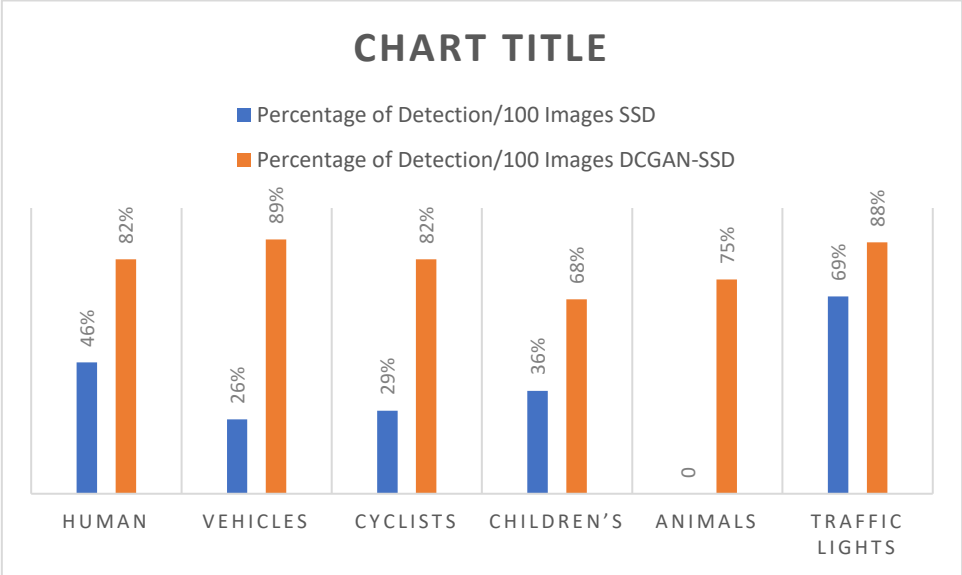


Figure 4.5 Shows the percentage difference in detection and misdetection accuracy between SSD and DCGAN-SSD

In figure 4.3 the results output achieved from DCGAN-SSD only shown, whereas in Figure 4.2 the comparison between the output achieved from DCGAN-SSD and SSD only shows the output calculated from the accuracy output calculated by comparing SSD only video frames and DCGAN-SSD video frames, the accuracy calculated by considering the 100 video frames processed by SSD only and the same video frames processed by DCGAN-SSD, both the video frames are counted manually against the instances and detection achieved, where Table 4.1 shows the calculations and the detection percentage

for each class, with standard deviation between all classes SSD only achieves 34.3 % and DCGAN-SSD achieves 80.7%

The results from Fig 4.4 and 4.5 show that, in all the aspects and classes, DCGAN-SSD performed better in detection with better accuracy rates when compared to those obtained by SSD. The reason behind this high performance with DCGAN-SSD is that the frames sent over to the SSD for detection are already pre-processed and the image resolution has been enhanced using DCGAN, making detection easier with SSD. In contrast, when purely using SSD, the option for enhancement of the frame is not available; therefore, it detects based on the training conducted for the model. Also, in the original work of SSD [Liu et al., 2016], the performance of SSD was 74.3%, but such a result was achieved in in ideal conditions, whereas in this research the SSD and DCGAN-SSD were tested in wild conditions. That is the reason there is a variation in output results for SSD between the original paper and those presented in this research.

4.4 Summary

In this chapter, the new method of cascading two state of art models been proposed, which is a new initiative not done previously for object detection with DCGANs with SSD to detect pedestrians and objects at a distance, mainly for smart city applications. Using deep convolutional generative adversarial networks by implementing the DCGAN [Nguyen et al., 2017; Radford et al., 2016], more robust discriminative features are extracted around tiny objects and hence, as a consequence, the detection rate is drastically improved. Based on this encouraging evidence demonstrating the new method's advantages, future research furthering the approach will be valuable for smart city applications that need to detect objects in the real world, particularly tiny objects at a distance, to secure life and avoid accidents. While DCGANs have been very successful in many other applications, this research can also be successfully implemented in robust solutions to many real-world challenges in smart cities.

CHAPTER 5. IN-VEHICLE OBJECT DETECTION IN THE WILD FOR AUTOMATED VEHICLES

In-vehicle human object identification is essential in vision-based automated vehicle driving systems. Pedestrians and other vehicles on the roads or streets are the primary targets to protect from driverless vehicles. A challenge is a difficulty of detecting objects moving in real-world conditions, while illumination and image quality can vary a great deal. To address this challenge, the research work exploits DCGANs [Radford et al., 2016] with SSD to handle real-world conditions. This chapter describe how DCGAN was trained with low-quality images to handle the challenges arising from the standard conditions in smart cities. At the same time, a cascaded SSD is employed as the object detector to perform object detection with DCGAN. This chapter's application was tested under real-world conditions using taxi driver videos taken in London streets in both daylight and at night, and the tests from the in-vehicle videos demonstrate that this strategy can achieve a greatly improved detection rate under real-world conditions.

5.1 Object Detection with Moving Camera

Vision-based object tracking and detection play essential roles in emerging self-driving vehicle systems [Hussain & Zeadally, 2018; Heylen et al., 2018; Sallab et al., 2017; Zhang. X et al., 2016; Du. S et al., 2019; Maqueda et al., 2018]. Following the accident in which an Uber self-driving test car killed a pedestrian [Hussain & Zeadally., 2018], the issue of reliable tracking and detection has been raised as a major barrier to securing self-driving vehicles in real-world conditions. If a self-driving system cannot overcome this challenge, driverless vehicles will not be a boon but a curse to smart cities with potential road killers on the loose everywhere. To tackle in-vehicle object detection under real-world conditions, the research aims to leverage DCGAN [Radford et al., 2016; Goodfellow et al., 2014], a powerful method to enrich the data coverage and consequently improve the performance of DNNs. This research work proposes by combining DCGAN with SSD [Dinakaran et al., 2019], which is used in multiple applications, one of which is for in-vehicle object detection in real-world conditions. Such a DCGAN-SSD framework has been shown to be very efficient in image-based object detection. In this chapter, the work is focused on night-vision vehicle guidance. While this application is targeted at in-vehicle object detection for self-driving vehicles, SSD as the object detector is favoured due to its performance to guarantee the speed of object detection. In comparison [Dinakaran et al., 2019] with RCNN, Fast RCNN and Faster RCNN, SSD is several orders of magnitude faster in terms of computing time, which makes it suitable for real-time applications such as self-driving vehicles.

The investigation is based on the feasibility of applying DCGANs to datasets for in-vehicle object detection. In this practical examination, the main focus is to train DCGAN with a set of visual objects from the CIFAR-100 dataset as a first step. The discriminator will cope with discriminating the real

image containing the visual objects provided in the dataset from the generated image, with the generated background pixels between the visual objects. The work starts with model training using the training data for object detection [Goodfellow et al. 2014; Radford, et al, 2016] and then focuses on the use of DCGANs, to improve the performance of the existing SSD object detector.

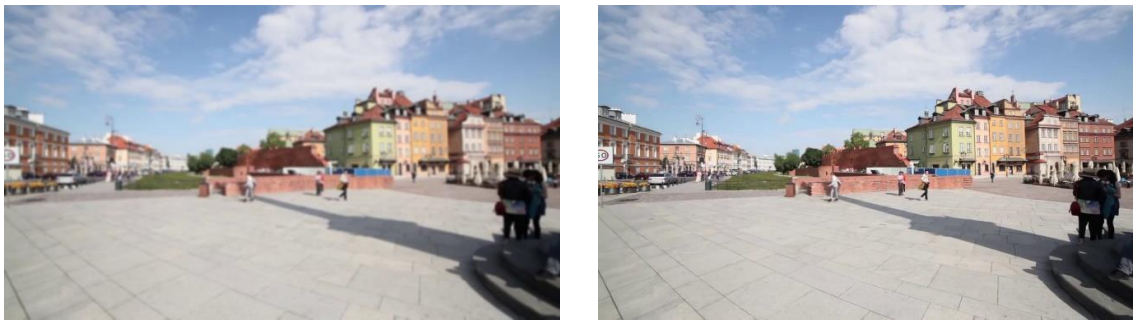
5.2 Background to Score Object Detection in Self-driven cars

The most popular generative model approach are GANs [Goodfellow et al., 2014]. GANs [Goodfellow et al., 2014] outline for learning generative models. GANs have been applied to improve image resolution and image quality [Dinakaran et al., 2019; Radford et al., 2016]. It has also been attempted to accommodate GANs in the object detection task to address small-scale problems by generating super-resolved representations for small objects [Dinakaran et al., 2019]. In this research, DCGAN is used to enhance the feature and as a supporting character for SSD object detector to provide a clear image for object detection, as shown in Figure 5.1b, without losing any data in the image. The reason behind using DCGAN is that, recently, several attempts have been made to improve image generation using generative models. For example, their variants, for example, conditional GANs, reciprocated conditional GANs, and DCGANs [Radford et al., 2016] use a Conv-Deconv GAN architecture to learn good image representation for several image synthesis tasks. Denton et al. [2015] use a Laplacian pyramid of generators and discriminators to synthesize multi-scale high-resolution images. Reed et al. [2017] use a DCGAN conditioned on text features encoded by a hybrid character-level convolutional RNN. The encoder with a conditional GAN (cGAN), to inverse the mapping of a cGAN for complex image editing is named as the result Invertible cGANs. Makhzani and Frey [2014] and Larsen et al. [2016] used GANs to improve the realism of the generated images.

5.3 Object Detection from inside the vehicle

5.3.1 Using DCGAN to improve the frames

In this proposed strategy, the work aim to improve the video frames from real-world settings using DCGAN. Figure 5.1 shows a test example. Figure 5.1a shows the raw image obtained from the video,



a) Before treated by DCGAN b) After treated by DCGAN

Figure 5.1. The generated high-res images from low-res ones

with a low pixel ratio of 320x100 and a size of 27kb. Figure 5.1b is the high-resolution image in which

the pixels and the image ratio have been enhanced. After enhancement, the resolution is 1920x1080 and the size 451kb. DCGAN is unable to process the video files without lag. Therefore, the pre-processing of video files is performed first, and then frames are given to the DCGAN for object enhancement before being sent to SSD for object detection. Also, batch norm and LeakyReLU functions promote healthy gradient flow, which is critical for the learning process of both G and D. Moreover, the reported results are obtained using 25 training epochs.

5.4 Multi-model cascaded Object detector DCGAN-SSD

Object detection is considered to be a major challenge of the image classification task, where the main goal is to classify and localize every object from the input image. The object detection problem is considered a major challenge in computer vision. However, there has been some progress in recent years because of advanced machine learning tools, such as Deep Learning and GANs [Goodfellow et al., 2014]. There are two main region proposal methods: You Only Look Once (YOLO) and SSD. When the research utilizes and cascades DCGAN-SSD, the main advantage is that the combination reduces the scale factors in the images, and SSD convolutional feature layers are also integrated to the end of the curtailed base network. These layers in DCGAN and SSD progressively reduce the size at different scales. The prediction model varies for each feature layer that operates in SSD.

The most communal technique for validating the eminence of unsupervised illustration learning algorithms is to apply them as a feature extractor on supervised datasets and validate the performance of linear models tailored on top of these features. The convolutional layers are leveraged to resize the images, and there is no room for enhancement of images or to fill the feature space. However, Figure 5.1 demonstrates the changes in the image pixels with the changes in the convolutional layers. These layers in the DCGAN are built to enhance the image quality for any size. The feature space is filled with feature maps for better detection quality. A pre-trained SSD network (which has already been trained with the material used in previous chapters to save computational time) is used for this application. The discriminator's convolutional features from all layers were used, max-pooling each layer representation, to produce a 4×4 spatial grid for 32x32 size images. These features are then flattened and concatenated to form a dimensional vector and a regularized linear classifier is trained on top of them. This experiment achieves better accuracy than by using SSD on its own.

5.5 Validating the Network in the real world with in-vehicle videos

In this experiment, DCGAN was implemented in two steps. First, the DCGAN codes are implemented based on PyTorch and Tensorflow, as illustrated in Dinakaran et al. [2019]. Trained DCGAN on the CIFAR datasets, with all images sized at 32×32, along with a batch size of 72 and 25 epochs, across a total of 60,000 images, then used the CIFAR datasets to train the DCGAN-SSD detector. The main aim is to validate if the proposed DCGAN-SSD architecture outperforms the single SSD, particularly with

long-distance objects or pedestrians in detection. Following this, the research involves running the trained DCGAN-SSD detector on real-world videos taken from in-vehicle moving cameras and comparing their performance against an SSD-only detector. Figure 5.2 shows the test result performed on the Uber self-driven car testing videos taken from in-vehicle cameras. The Uber self-driven car failed to stop when the pedestrian crossed the road with a bicycle, but when the same video was processed with DCGAN-SSD, the pedestrian was captured at a distance. Also, Figure 5.3 shows the London Taxi video frames tested with DCGAN-SSD. The videos consist of daytime on-the-street videos and night-vision videos from a moving camera. The test samples show that the test results are overwhelmingly better when DCGAN is attached to improve SSD performance. Mainly, the DCGAN-SSD detector can detect pedestrians in the dark shadowing regions, which is critical and challenging for vision-based driverless vehicle systems.

As reported, Uber suspended its driverless vehicle project due to the accident of its driverless vehicle in night, when the object detection system ignored a pedestrian crossing the road and, tragically, was hit by Uber's driverless vehicle (in test) [New York times media, article, 2018]. Optimistically, this research experiment provides a reliable object detection system that can be used at night and hence could be extremely useful for a vision-based driverless system. The difference between Uber's in-vehicle object detection and this research's object detection is shown in Figure 5.2. Notably, the SSD detector performance lags in the scale factor, which DCGAN compromises in this experiment, in which the DCGAN can improvise the scale factor in SSD by providing the detector with super-resolution

a) Frame Extracted from Uber video

b) Uber Frame after processed by DCGAN-SSD

Figure 5.2 Frame Processed by Uber, is 5.2.a), the same processed by DCGAN-SSD 5.2.b)

	Detection Rate-Night-time		Instances
Categories	SSD	DCGAN-SSD	
Human	8	422	462
Vehicles	68	580	592
Cyclists	0	4	4

Table 5.1 The comparison results between two experiments performed by SSD and DCGAN-SSD, for different object categories respectively during night-time

images. It results in a larger total feature vector size due to the highest layers for feature vectors of 4×4 spatial locations. Further improvements could be made by fine-tuning the discriminator's representations, but this will be addressed in future research. Also to be considered are issues of how to make the detector robust to noises [Jiang. R et al & Crooks et al 2019] and how to include 3D CNN as an alternative solution for video object detection [Storey et al., 2019].

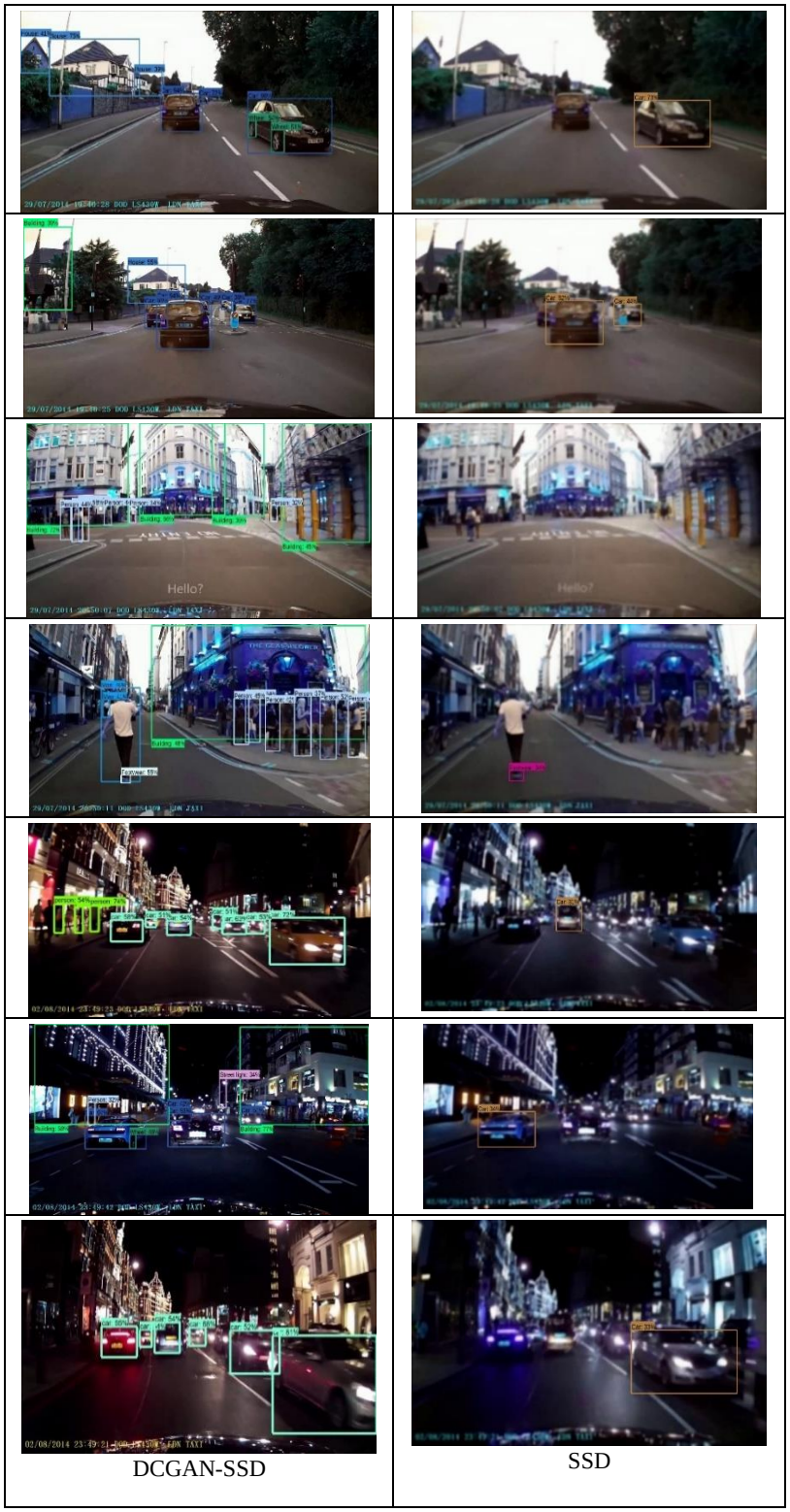


Figure 5.3 Experimental Results performed on Wild Videos and experiments based on SSD vs DCGAN-SSD

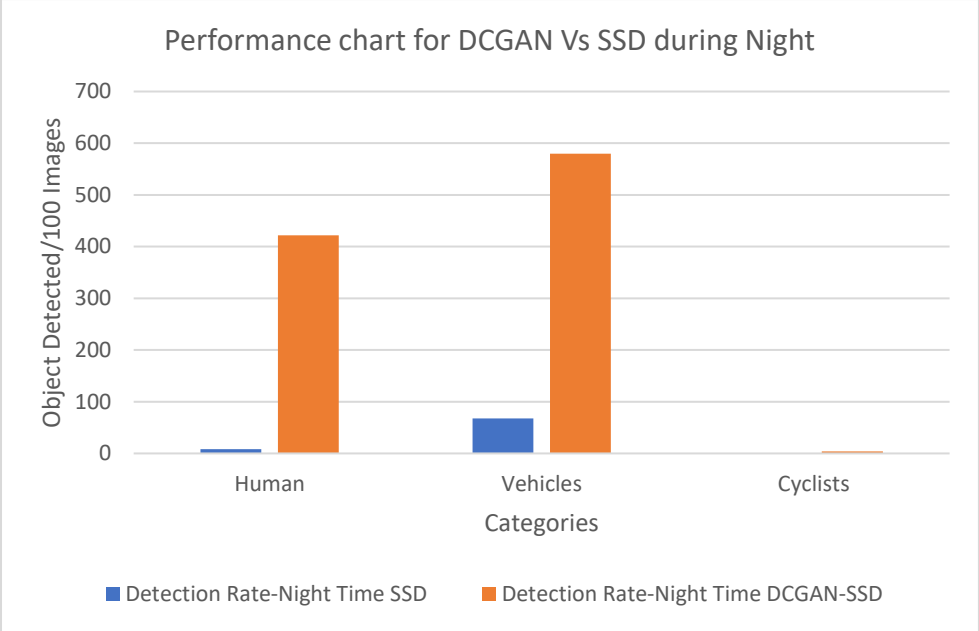


Figure 5.4. Performance Difference in Object Detection between DCGAN-SSD vs SSD during night-time in London taxis

	Detection Rate-Daytime		Instances
Categories	SSD	DCGAN-SSD	
Human	3	182	201
Vehicles	89	726	763
Cyclists	0	2	3

Table 5.2 The comparison results between two experiments performed by SSD and DCGAN-SSD, for different object categories respectively during daytime

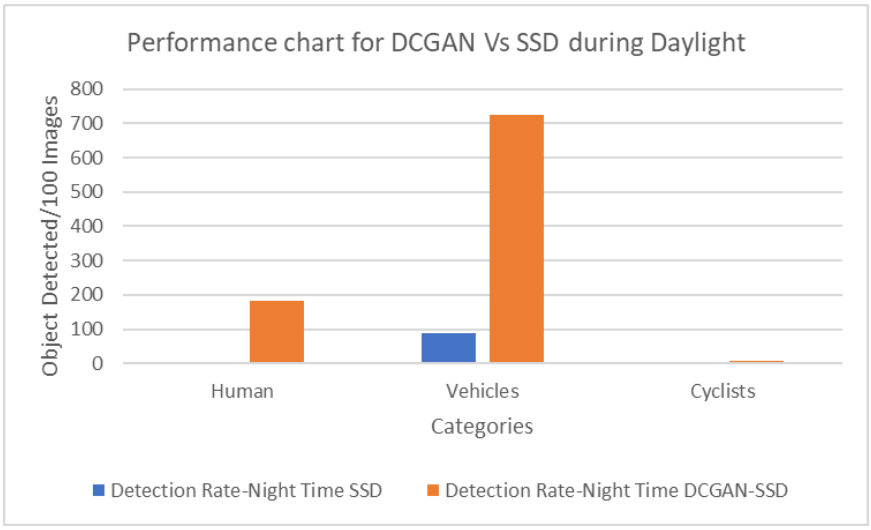


Figure 5.5. Performance Difference in Object Detection between DCGAN-SSD Vs SSD during daytime in London taxis

From the experiment results, the graph charts in Figure 5.4 and Figure 5.5 show the quality in detection between DCGAN-SSD and only with SSD, based on the data in Table 5.1 and 5.2, where DCGAN-SSD performs better in both the scenarios during night-time object detection, and in daytime object detection respectively; and also the detection is from moving cameras. During night-time it is also able to detect black cars very precisely, as is shown in Figure 5.3.

5.6 Summary

Object Detection plays a significant role in vision-based autonomous driving vehicles. Objects like humans, animals, traffic lights, cars' rear lights, poles on the road, and dividers are considered primary targets for protection from driverless vehicles and to protect the vehicle itself in smart cities. The challenge is to detect objects moving in real-world conditions when illumination and image quality can vary greatly. This chapter examined the feasibility of applying DCGANs with a Single Shot Detector (SSD) to handle real-world conditions to manipulate the challenges. The DCGAN was trained with low-quality images to handle the challenges arising from the real-world conditions in smart cities, and a cascaded SSD is employed as the object detector to perform with the DCGAN. The datasets from the Canadian Institute for Advanced Research (CIFAR), Caltech, and KITTI were used for training. Further in-vehicle tests on London streets demonstrate that this strategy can drastically improve detection rates in real-world conditions.

CHAPTER 6: UNDERSEA TARGET DETECTION FROM UNMANNED UNDERWATER VEHICLES

The primary tasks that the challenging of underwater systems encounter are the computing time and the high cost. This is a complicated task for object detection by the emerging trend and the evolution of Deep Learning technology and image processing algorithms. In consideration of this, DCGAN-SSD tested with the underwater environment. The proposed research in this chapter has also been extensively evaluated using underwater scenarios with computer vision, a very different field with increasing research interest in recent years. The videos used in this chapter are taken in the wild in underwater environments.

6.1 Underwater object detection by encountering blurred resolution

The installations and initial setup for underwater object detection are very complex, owing to the fabrication required for human pose estimation and the light dissemination presented by the underwater environment. Under certain conditions, such as high contrast lighting environment, objects can be blurred in the video, or the video itself can be blurred in the underwater atmosphere where the image contrast also changes. Therefore, object detection will be challenging, and this is also one of the fields in which this research can implement the Deep Learning models to solve challenges such as tackling low lighting conditions and diverse distracting factors. Nevertheless, computer vision with neural networks in the underwater environment has already been in use for watching or inspecting submarines, monitoring the borders of the ocean, mapping, and terrain segmentation. Technically, the chapter deals with the two primary state of art models in a combination DCGAN-SSD method, for object DCGAN [Radford et al., 2016] and SSD [Liu et al, 2016], cascaded together to generate DCGAN-SSD. The two models work together for a complicated underwater object detection task. There is a need for more precise and stronger object detection, which can be used for accomplishing complex computer vision tasks. A robust object detection approach can also classify different objects under various conditions, including environmental, weather-related, and time of day, such as morning or evening.

This chapter presents a combinational framework that conducts object detection in underwater atmospheres. The model performs based on the images captured by the camera and is comprehensively evaluated using different datasets, including various effects for underwater image processing. In this research, the initial contribution stage is based on multi-feature object detection to find underwater objects that can be humans, fishes, plants, and marine vertebrates. The model can also be transformed and trained with parameters to detect the different object's underwater environments. The underwater scenarios can be anything like sea, river, or even swimming pools, where the framework advances and simplifies the work in object detection. The presented technique explores destination in the object rendering with feature matchlike prominent feature consistency, which compromises accurate bounding

among the images. The images are managed initially and handled to remove the environmental distortions like blur, illumination, and dissimilarity penetrating the water environment. The present feature vector comprises the values of the Intersection over Union (IOU) in the latent space in DCGAN, as well as the IOU between the images in SSD and its response to the gradient between the bounding boxes of the objects. It is also stretched out to comprise pattern-related in DCGAN to fill the latent space by using the feature fed to the Generator part with other features. The second stage of work is with the associated sections of every clustered feature, which are classified rendering to be compatible with Histogram of oriented gradient(HOG) features as seen in Chapter 3. While the HOG related to objects is focused on the frequency of hits related to time domain, the HOG matching with features removed from the everyday underwater environments is typically scattered more than expected.

The other contribution in the work is to authenticate the projected method with diverse datasets used for object detection in underwater environments. CIFAR-100, CADDY, and Roboflow fish datasets have been used for training and validation. For testing, various videos with random data is used as a test set, since it was hard to find a blurred video to test, which is another challenge in this research. These datasets are created with different underwater conditions, and there are also few other obtainable datasets dedicated to underwater scenarios. These datasets are selected based on the camera quality and other experimental circumstances, including in-depth and different classes present in natural and artificial lighting conditions, sensor guided, etc. Despite these changes, the projected algorithm attains exact and dependable detection based on the training with the datasets. When it is presented with computer vision, other models typically cannot give a solid 2D portrayal of the frame, allowing detection under different circumstances. Be that as it may, in valuation, the focused object's posture can be dependably projected just by controlling and handling the mono-camera output.

6.2 Proposed Method

Various methods present ideas behind the vision-based object detection with the raw images captured without using multiple cameras, with complex textures or pose corresponding to the algorithms based on computer vision processing. The novelty-based on different frames like blur, high contrast, and dark area for object detection under different scenarios the novelty based tests are conducted, especially under the water. And now it is possible to see how the different framework processes, like DCGAN-SSD, can be cascaded together into a pipeline. The technique adopted for image processing can be reliant and complex in creating a pipeline for consistent and robust features attainable for the object, depending on particular underwater scenarios. Early research concentrates on detecting or projecting regions relevant to the background on behalf of participating objects, probably with no previous data available about the object. The subsequent step may be a broad area that is even more challenging and a strong candidate to reduce the potential places in the consecutive segments, with the plan of the areas as a destination object. On the one hand, no further action about the presence of the objects filtered from

the same frame or the consecutive regions in the frames is required (those kinds of suggestions or actions being surrogate to a more actual performance using DCGAN-SSD or are mutually achieved). On the other hand, for inside water detection of objects and their component consists of DCGAN-SSD oriented with the bounding box, which precisely helps to detect the objects in the image. The point cloud is used to work towards the two models trained and then uploaded in the cloud and evaluated for instances of a previously detected object using the backup to store the images detected.

6.3. Image Processing

Underwater object recognition needs the computer vision framework to adapt to troublesome underwater lighting conditions, because t lighting conditions change underwater under every different depth conditions. Specifically, light decreases underwater and creates indistinct objects in the frames concerning features, limited dissimilarity, and increased contrast in the learned images. Object detection becomes even more complex in the case of divers with diving costumes and oxygen cylinders or uneven and flexible object detection. Hence, particular concentrations should be given to improving image quality for observing object detection underwater.

Early-stage of the pipelines is considered to generate images from the features by DCGAN recompense. The disturbances happen based on the environment, including the lighting conditions and the propagation in water through image augmentation. The generated images are fed to the adversarial part of the DCGAN. After feeding the features to the generator, it generates the images from the features. Then, the enriched features enhance the generator-generated images [Ancuti et al., 2012]. The method adopted in this chapter focuses on processing the images against the contrast to recover blurry underwater images. The challenges faced for underwater object detection, such as dark or fading lights, low contrast, and indistinguishable objects, are removed when the generator generates the images out of features to encounter the contrast, blur, and other noise distortions limit detection quality. This is achieved by not feeding the features that create distortion. When the objects are fed, it is expected in the detection. Rich features have been appended to enhance image quality and avoid image distortions, The DCGAN methods been used to enhance the quality of images with rich features. The features are first applied to the generator latent space G. In particular, the component $G_{in,i}$ of each pixel i is extracted, a median filter is applied to the generator G for the image to obtain a new blurred value $G_{blur,i}$, and the new value is computed as $G_{out,i}$ $1.5G_{in,i}$,$0.5G_{blur,i}$. The features given to the generator part and the generated images are sharpened with super-resolution.

6.4.Processing Images Obtained from Monocular camera

The ultimate aim of an image processing model may include taking the position of a region of interest ROI encompassing destination in objects and object detection in the underwater environment. The images obtained from the camera and the image obtained through image processing can extensively

develop the efficiency of the application, even in underwater backgrounds or on the seabed. Since the proposed model only needs the features of the objects to be captured while the quality enhancement is done using DCGAN, the bounding boxes of objects are divergent in the image with an object detector. The target object can be detected in a first image or at least its detection can be simplified by restricting the search region to be analyzed in later frames or images, leading to progressively advancing the experiment. Object detection is complex and, when considered inside the water, even more complicated. But, employing such systems again would lead to more expensive computational costs depending on camera quality and the images the camera attained by computer vision produces. If the object has just recognized pictures, the 2D information relating to the sectioned ROI is utilized to gauge the object's posture. In the following, two quick methods for distinguishing proof of an ROI conceivably containing the target object (ROI_{area} and $ROI_{feature}$) and an increasingly exact procedure for target object detection (ROI_{shape})[Prats et al., 2012] are illustrated. Furthermore, a technique for the posture estimation of rounded objects is proposed [Prats et al., 2012].

Henceforward, at the initial stage, DCGAN undertakes that the unknown object never lodges more than a given percentage of the image pixels. The hidden technology lies in the features fed to the generator G, where the generator generates only the images with the added features. When it has a uniform colour, there is a need to generate multiple features and images, just because the uniformly coloured features may belong to any class of objects. The region equivalent to a given hue level is also an essential feature to generate. Only a region that is less than 50% of the image is designated as part of the area in the images for the discriminator D. The second stage obtains data from DCGAN particularly based on features obtained from the DCGAN, which are to be detected based on colour features of the target and the area. When the object feature is known, even in very low resolution a more specific or random colour mask by DCGAN can be applied to make the object more robust from fragile view to detect the object with SSD accurately by approximating the object's bounding box. Hence, SSD is attained by comprising the regions where the feature is close to the threshold and the predictable target features. The region calculated by ROI based on area or ROI based on the features is made accessible for additional processing.

6.5. Object Detection

The projected ROI_{shape} algorithm to attain a bounding box based on the shape of the features in the frames achieves object detection in a two-step process in image processing and bounding shape authentication. The goal is to identify a related ROI region based on the Intersection over Union (IOU). When compared to the previous chapter, the IOU parameters are changed for the IOU to achieve the best performance. The detected objects can be human beings, fishes, submarines or even underwater vehicles. ROI_{shape} depends on the relative feature reliability of the object for separation based on IOU and on its regular bounding shape for verification. The detection attains these two salient and

comparatively overall features of artefacts in an underwater atmosphere. In a divergence from former research findings with the former, this research found that uneven segmentation methods (ROI_{area} and $ROI_{feature}$), i.e. ROI_{shape} can identify whether the target object belongs to the image before executing object detection.

The image separation step categorizes each pixel based on its features and colours, rendering its consistent vector of local features. The input image features can be minimized to a smaller size in the further layers of DCGAN or it can also be done in SSD based on the image quality, which can be helpful for detection and rich features to bring out the best results from the generator G to match the images from the discriminator D output to minimize the computational complexity of the object classification with SSD. The above process is conducted using the new framework, DCGAN-SSD, is presented in this research. The underlying grouping step dependent on every pixel is free of the characterization of different pixels. The feature space and the latent space embraced in this part are larger than that utilized straightforwardly to SSD and could be expanded for better image classification. The anticipated technique is effectively functional for different underwater datasets and is consistent with distinctive situations in this experiment. The shape authentication is applied to each vector allocated in each filter and strides. The algorithm processes the boundings of features in each binary image consistent with the feature cluster features. Each closed outline signifies a region and shaping and matches the shape among outlines. The outline of human objects in specific when their estimation in the image plane is about a rectangle mostly containing parallel edges [Goodfellow et al, 2014].

6.6.Experimental Results

6.6.1.Discussion on Experimental Outcomes from DCGAN-SSD

The experimental assessment was performed using CADDY, ROBOFLOW, and HAR-DAISY datasets which are based on different images captured at different depths (from 2m to 10m depth), and different light conditions and objects. When it is associated with object detection from land and underwater, background variation is one of the main advantages of underwater object detection. The background varies in land object detection, and underwater the background variation is limited. However, the main drawback in underwater object detection is the equipment used which is high cost and shows limitations for the experimentation for most research studies. The most welcome approach in object detection is the feature extraction method. However, it is less reliable for underwater object detection. This is one of the reasons that two state of art models cascaded to generate DCGAN-SSD for underwater detection.

DCGAN enhances the frames as shown in Figure 6.1 to ensure reliable and robust underwater object detection.

Figure. 6.1. The difference in the images obtained from the video frames on the left and the images generated by DCGAN

The third stage is the image obtained from SSD with detection after it is processed and enhanced from DCGAN. The features extracted from the images have been used to evaluate the object poses for the respective datasets. In this research, the visual geometry group, the VGG-19 model, is used, in the single shot detector, as in Figure 6.2, which makes it easier compared with manual annotations. When the accurate region of interest can be obtainable, an adequate target can be fixed based on the point features. The projected standards of the object's magnitudes can be matched to the closest features. Figure 6.3 shows the comparison in the detection between SSD and DCGAN-SSD. It is evident that SSD, combined with DCGAN, can clean the blurred image and provide a good detection quality, even with inferior camera quality in the underwater environment. Table 6.1 and Figure 6.4 shows the accuracy chart from different classes based on their detection rate. Even with different postures of the humans under water, the impressive detection results proved the advantage of using DCGAN-SSD together instead of SSD alone. The essential factor here is when the SSD alone tries to detect the objects, it fails because of poor quality images, but when SSD is combined with DCGAN, it is able to outperform on the same images due to the enhancement done by DCGAN. Therefore, resolution affects the performance of any detectors. In this research findings, there is another advantage: there is no need for detectors to be trained on all the data like blur or contrast, whereas DCGAN will help in normalizing the images.

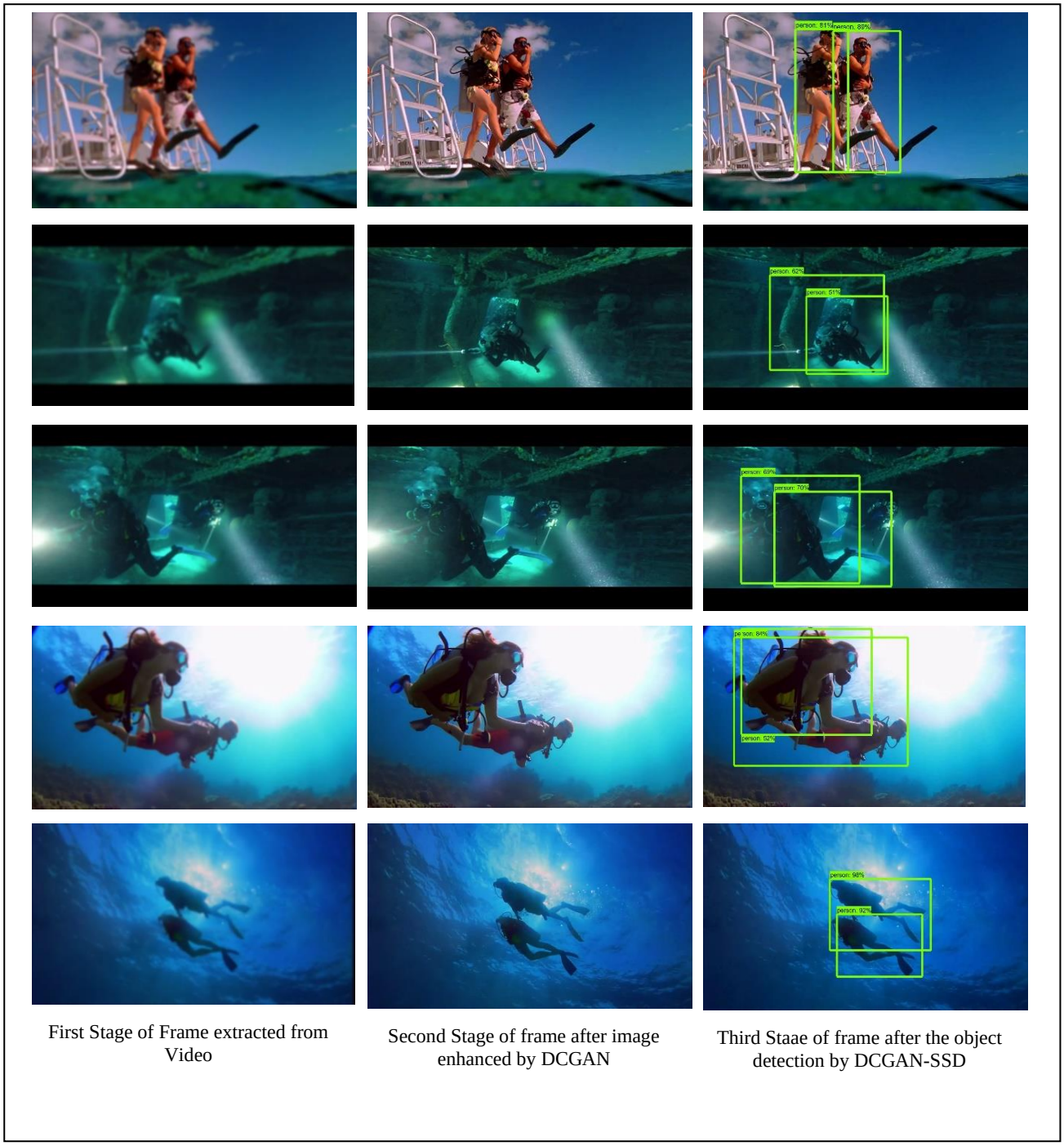
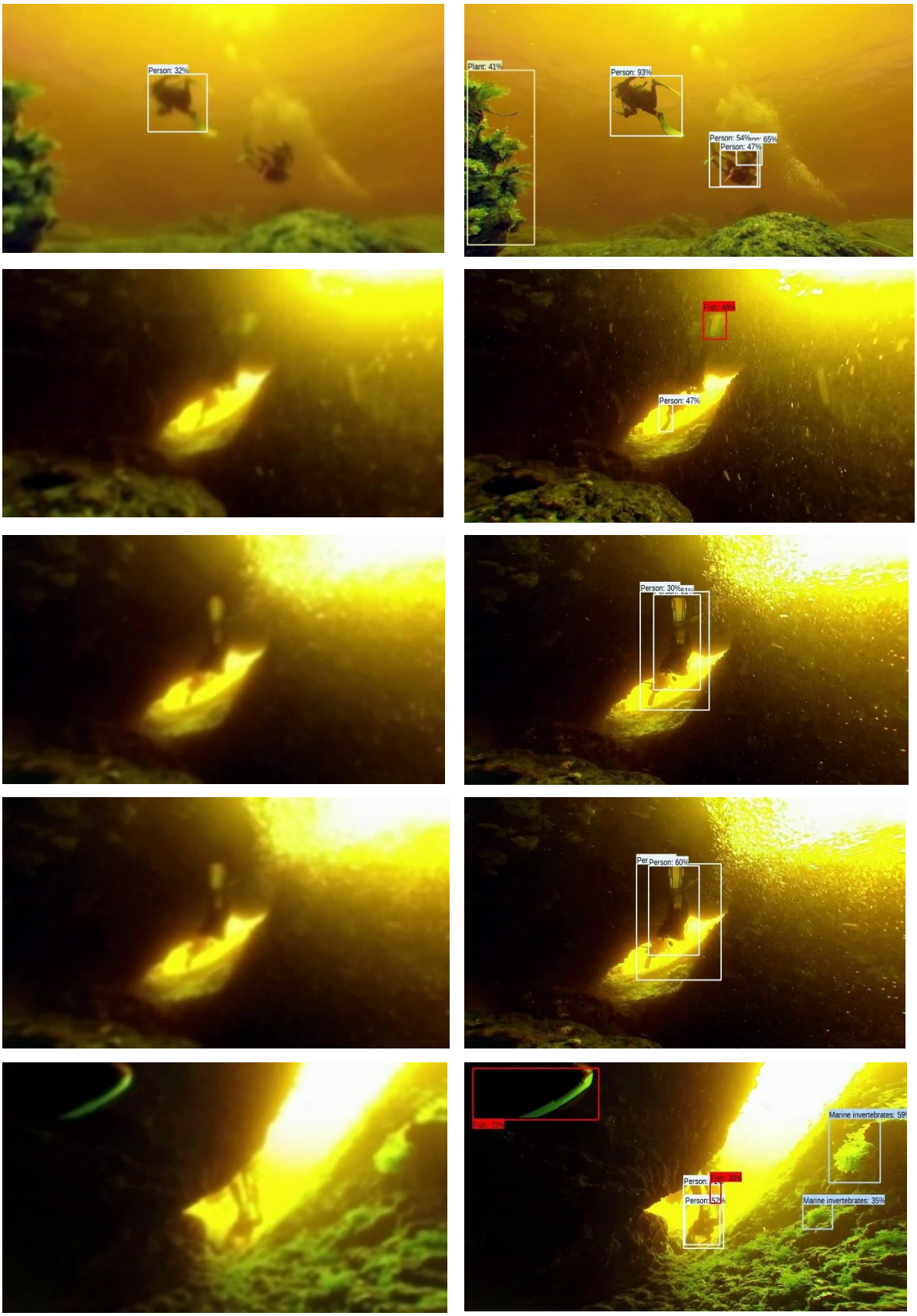


Fig 6.2. The three stages of the objects and the frames. The first stage is to extract the actual image from the video and the second stage shows the images obtained after the enhancement using DCGAN



SSD **DCGAN-SSD**

Figure 6.3 Shows the difference in the detection ratio between SSD and DCGAN-SSD. SSD failed to detect any objects due to resolution of the image is very low

Categories	Detection Rate Using		Object Instances	Detection Efficiency		False Positives	
	SSD	DCGAN-SSD		SSD	DCGAN-SSD	SSD	DCGAN-SSD
Human	2	114	126	1%	90%	1	13
Fish	0	51	58	0	87%	0	4
Plants	0	23	31	0	74%	0	0
Animals	0	2	0	0	0	0	2

Table 6.1 Table of accuracy comparison: under different categories underwater with 100 image

The difference shown in Figure 6.4 accuracy chart is clear DCGAN-SSD works better in under water

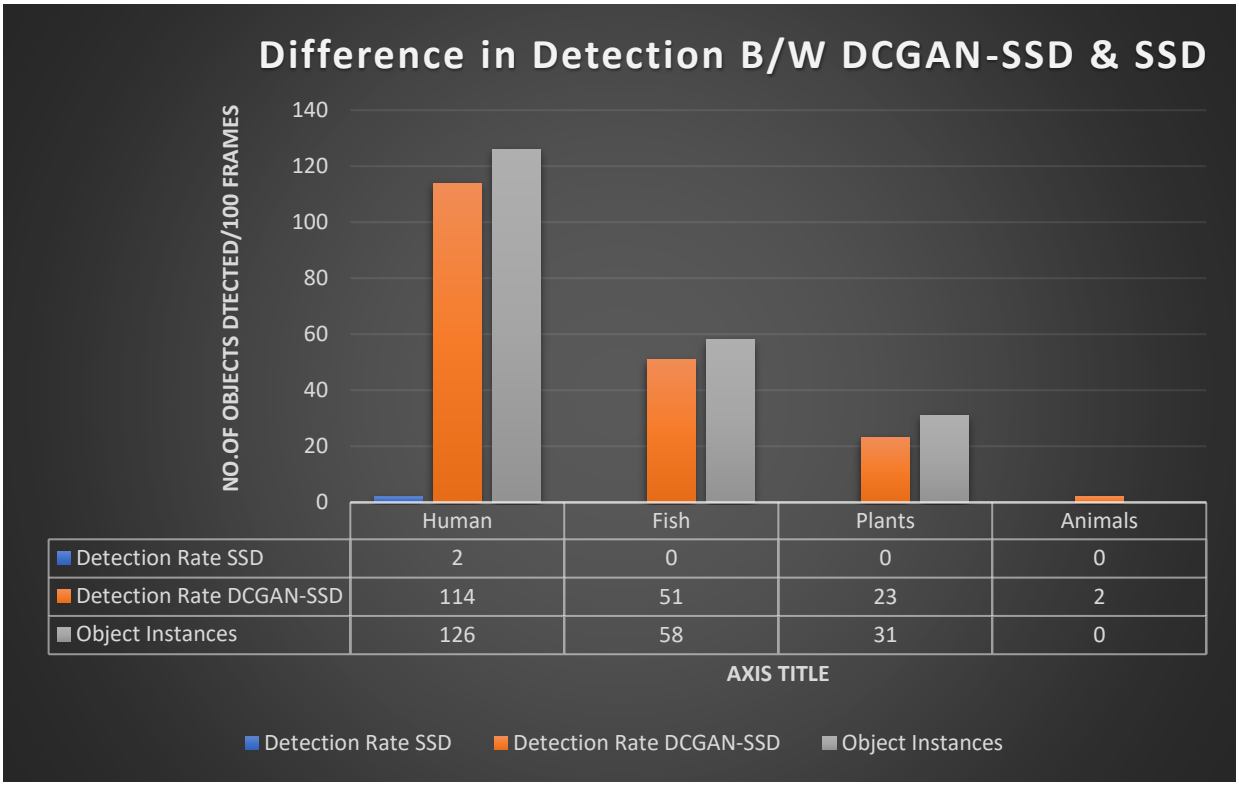


Figure 6.4 Accuracy chart for different classes from Underwater Environment

environment and it is also capable of working efficiently in any challenging environment DCGAN-SSD detected humans 114 out of 126 instances which is 90%, fish 51 out of 58 instances which is 88%, plants 23 out of 32 which is 74%, and animals 2 out of 0 instances where the detection score is based on false positive detection.

6.7.Summary

This chapter examined object detection methods in underwater conditions based on computer vision with added artificial intelligence with neural networks. The research and this chapter showed how the cascaded DCGAN-SSD model is used to detect human objects under rescue conditions, which is very

challenging in the underwater environment. Once the target is detected, the pose estimation can be carried out with legacy techniques with the help of HOG. The efficiency of the proposed method was tested using three significant underwater datasets. It should be born in mind that the appearance of human objects underwater can be entirely different from how they seem in standard daily settings and environments. Underwater, human objects take on different shapes and wear additional equipment like goggles, oxygen cylinders, and different costumes. The water colour changes according to the algae present in the water. These are the significant challenges in underwater object detection. The design of underwater object handling and the detection accuracy in parallel leads to reliability and unreliability based on the detection quality in object detection techniques, which measures the importance of the implementation.

CHAPTER 7 PARTICLE SWARM OPTIMIZATION-BASED HYPERPARAMETER FINE TUNING IN OBJECT DETECTION

The most common challenge everyone faces in neural networks is optimization. It is hard and time-consuming to analyze the hyperparameters manually. This is particularly hard in large networks, which have many layers and a cascaded structure in the case of multiple models. This research work integrates PSO to optimize the SSD, and DCGAN-SSD for the object detection task. DCGAN-SSD performance depends on its parameters, requiring a highly skilled or knowledgeable person to select the high-performance parameters to fine-tune the model. The method is to pick the high performance automatically and train the model with those particular parameters to obtain results with higher accuracy, which is reliable and not time-consuming. This chapter mainly aims for better performance in the existing network with little or no alteration to the DCGAN-SSD network. The results prove that DCGAN-SSD can perform better with image description tasks when it works together with PSO hyperparameter optimization.

7.1 Importance of Using PSO in Object Detection

Neural networks have seen enormous growth and are now called Deep Neural Networks or DNNs. DNNs are used very widely for multiple applications. However, there are a few limitations and hurdles for DNNs one is performance, which is addressed in this chapter, and initiating a massive DNN like DCGAN-SSD. The performance of DCGAN-SSD relies on its hyperparameters it takes time to estimate and select the suitable parameters. The challenge of optimizing the parameters remains in DNN and this research with DCGAN. This chapter proposes a powerful way to optimize these challenging parameters with PSO. The main contribution of using PSO with a large-scale network like DCGAN-SSD is challenging but works very efficiently for the hyperparameters like initial learning rate, decay, and momentum. By using PSO, the performance of the DCGAN-SSD architecture is improved.

Hyperparameter algorithms show better performance than human efforts in trial and error experiments [Bergstra et al., 2011]. This research work encountered more challenges during the implementation due to the integration in a large network with DCGAN-SSD, which has very high computational complexity. Few algorithms have been very efficient in achieving the target in optimizing problems. This chapter validates PSO for the optimization problem with DCGAN-SSD, performed in dataset CIFAR-100. The output results prove that PSO gives improved output with the existing architectures. They also prove that PSO is efficient in selecting the hyperparameters, which is demonstrated practically in this chapter. In this work, the problem was handled in selecting hyperparameters using the PSO algorithm. The PSO has shown robust object detection applications for DCGAN-SSD at a considerable scale and for a scalable network.

Many existing studies have been proposed for automatic hyperparameter identification for machine learning and Deep Learning methods [Fielding & Zhang, 2020; Fielding et al., 2018], demonstrating great efficiency in solving diverse classification and regression problems. Evolutionary algorithms show significant performance in image classification and solving diverse optimization problems. They are widely adopted in deep network generation and hyperparameter fine-tuning applications [Tan et al., 2019]. Motivated by the significant performance of such metaheuristic algorithms and their simplicity in terms of implementation, This research adopted such algorithms for hyper-parameter identification for the proposed deep network, i.e., DCGAN-SSD, for object detection. Moreover, owing to the high computational complexity, This research mainly optimized the hyperparameters of the SSD model within the hybrid architecture in this research. The empirical results indicate the efficiency of the proposed PSO-enhanced DCGAN-SSD object detection model. Then employed the obtained class labels and their corresponding regional proposals as inputs to an LSTM network for the generation of the natural sounding description of each image frame. Evaluated using a number of the well-known image (e.g., CIFAR-100) and video data sets, the results indicate the efficiency of the PSO-enhanced object detection and image captioning networks.

7.2 Contribution

This chapter demonstrates PSO-based hyperparameter selection for DCGAN-SSD. As previously mentioned, DCGAN-SSD is a large network for training and handling and can be handled easily in parallel and independently from DCGAN-SSD. Here, PSO is used with the wrapper DCGAN-SSD in the training process to get the hyperparameters to avoid classification errors for consistent results in object detection. Since this chapter in the research uses a highly complicated and scalable network, PSO exposes scalability by using GPUs.

The wide experimental study, including various environments with DCGAN-SSD with CIFAR-100 datasets, leads to different contributions listed below.

- Firstly, the work proposes a hybrid end-to-end model, i.e., DCGAN-SSD, for object detection, which enables the parallel training of both DCGAN and SSD.
- The PSO algorithm is used to identify the most optimal learning hyperparameters for the cascaded SSD network with VGG-like architectures as the backbone for object detection.

This chapter is based on and influenced by the clarity of the results obtained from the experiment. The emphasized results clarify the novelty involved in this research, the PSO-based object detection and hyperparameter task, and the capability of the DCGAN-SSD architecture, which shows the output's performance. The work provides insights and the capacity of the proposed algorithm.

7.3 Method of Approach

In this chapter, a different estimation approach gives an improved hyperparameter selection for the combination DCGAN-SSD to improvise the classification process that leads to object detection. This hyperparameter selection process is vital in training SSD for object detection applications. It is not limited to one network. It can also be applied to more than one network. This chapter involves using DCGAN-SSD, where both the networks can be trained together, but PSO can be used only on SSD to reduce the complexity. In this research, the DCGAN-SSD network with PSO is customized to fit together in a single nutshell. There is some complexity in the integration of the networks like parallel processing. Also, there is a limitation in processing large networks in standard computers and processing high dimensional dataset, and also had to manage a few stability issues in training.

To resolve the problems and increase the performance of DCGAN-SSD for object detection tasks, this chapter's approach mainly addresses the problem and solution in finding the best hyperparameters and resolution with an efficient method using PSO. Through PSO-based hyperparameter optimization, the main advantages are i) finding the best hyperparameters for the object detection model and auto-tuning under a single pipeline; ii) its training and evaluation, where the heuristic PSO algorithm identifies the best parameter based on the output provided by the algorithm the parameters being taken for training the model and involved in tuning being learning rate, decay, and momentum; and iii) the dropout rate for the selected parameters are determined based on the number of particles N, each representing a possible solution. The swarm particles then change their positions using an evolutionary process guided by the personal and global best solutions. The main advantage of PSO is that it can traverse through a large, many-dimensional search space in a simple yet efficient procedure. While it cannot guarantee that it finds the global optimum solution, it is likely to find a close-to-optimum solution with sufficient iterations.

7.4 Overview of PSO and a simple Structure of PSO coding structure for Hyperparameter selection

Below shown is the code structure in the form of pseudo code used to code PSO:

Initialize the swarm and the search parameters
While (K<the maximum number of iterations)
For(i =1 to number of particles N))
{
Evaluate particle i;
If the fitness of x_i^t is greater than the fitness of p_{i_best}
Then update $p_{i\_best} = x_i^t$
If the fitness of x_i^t is greater than that of the global best which is g_{best} then,
Update $g_{i\_best} = x_i^t$
For (each dimension, i.e. d_1, d_2, d_3 dimension for the learning rate, momentum, weight decay respectively)
Then update velocity vector v_i^{t+1} using defined PSO velocity updating equation

Update particle position x_i^{t+1} using defined PSO position updating equation
End (for dimensions)
}
End For Loop (for particles)
End While Loop

This chapter presents an overview of PSO and a simplistic code layout depiction used for hyperparameter optimization configuration. Various intelligent algorithms have been presented in recent years to solve a range of difficult problems and implementations for real-time applications, including the DCGAN-SSD network. Also, these algorithms solve real-time problems in medicine, finance, engineering, etc., but for all these algorithms, there is one common problem in finding hyperparameters. Swarm intelligence is considered for this purpose, inspired by real-world examples of birds searching for their food by continuously changing their position until they attain their target. PSO is the algorithm used to find the swarm position, as introduced by Eberhart and Kennedy [1995]. PSO is also one of the heuristic optimization techniques, where it populates and then updates the state of individuals from the population through an advancement process. Compared to the other swarm intelligence algorithms, PSO is simple and easy to handle very efficiently, and the solution provided by the PSO algorithm is optimal and tangible. In PSO, the results depend on the number of particles N, and PSO performs the best with 20 particles (the highest tested value) in this experiment. Experiments with population sizes between 2 and 20 were performed in this research study, but a population size of 20 shows impressive model diversity and performance. PSO is a stochastic optimization technique established on the movement and intelligence of swarms. It uses a number of particles(search agents) that constitute a swarm, which move around in the search space and search for global optimality. The PSO solution is not repeatable; it is used to find the optimal solution without repeating the process. If no changes have been made to any of the data, however often the PSO runs, it will produce identical results. Also, based on the use case, if the output is unsatisfactory, it can be repeated by adding more training data, as in this research. The PSO process was repeated three times for the case studies in this research.

Learning rate	Momentum	Weight decay	Fitness (in loss)
0.000285	0.007951	0.44710	231.5625
0.006902	**0.00908**	**0.82245**	**9.58052**
0.000521	0.001392	0.2512	198.371
0.001138	0.000235	0.4467	132.3029
0.005217	0.001104	0.92989	16.11632
0.000445	0.003728	0.7033	156.3364
0.003516	0.003442	0.1698	77.77496
0.001108	0.009621	0.5448	122.4384
0.002458	0.007613	0.7595	31.23957
0.001261	0.002016	0.1152	163.3016

Table 7.1. The hyperparameters recommended by each particle for the DCGAN-SSD model

PSO practically uses particles as a population, where the m-dimensional vector articulates each particle. Each particle of PSO is an interpretation of the solution in m-dimensional search space, where m represents the coordinates of the particle's position. Table 7.1 shows the results for the experiments conducted with the hyperparameter search using a set of four runs where the mean results are used for performance comparison. The best-optimized result has been highlighted. The optimized DCGAN-SSD model shows enhanced performance for object detection. PSO is considered a group of particles, and it is a group of particles initialized randomly and populated in a search space. PSO direction is influenced by two factors position and velocity, which is based on an individual best previous experience p_{i_best} and based on all individual best of the previous best or swarm particles, which is g_{i_best} , where the t and t+1 denote the iterations and generations of the new position, d denotess the particle dimension, x_i^t denotes the particle at generation t of the position in the dimension i, v_i^{t+1} particle generation t+1, and the velocity at the i dimension.

7.5 Experimental Results

The experiment was performed on different wild videos in the wild to compare and differentiate the result of the PSO-optimized DCGAN-SSD and DCGAN-SSD networks. The number of videos used for the analysis was 23. The number of images extracted from the video averaged 850 a video, which is 19550. Two videos are presented in this chapter for comparison. The model was trained on a CIFAR-100 dataset consisting of a total of 60,000 images. Out of 100 classes, where 50,000 is considered for training and for testing, the research work used videos from the wild, some by capturing through mobile phones and some from online sources, to train the model with PSO number epoch given is 100 and 100000 iterations. Table 7.2 shows the difference in the evaluation results while using only DCGAN-SSD and DCGAN-SSD used with PSO by optimizing the hyperparameters.

Figure 7.1 and Figure 7.2 show the results of the two videos processed, and Figure 7.3 and Figure 7.4 show the accuracy chart based on the detection rate which is done manually the data is also presented in Table 7.2 and Table 7.3, respectively, for the data shown. The videos used are a busy Indian street video and a London taxi video, respectively, with DCGAN-SSD and PSO DCGAN-SSD. The video result from a busy Indian street showed the improvement in detection quality when PSO DCGAN-SSD was used. The video length is 10 seconds, and the frames are contained in video 602. Therefore, approximately 60.25 frames per sec(fps) this amount of fps is considered best for testing when there is speed involved, such as self-driven cars. Also, London taxi video was used to test cars in cities under night and daylight conditions.

Models	Human in total 1068	Vehicles In total 286	Overall Efficiency
DCGAN-SSD	894	240	83%
PSO DCGAN-SSD	943	276	87%

Table 7.2: Shows the comparison chart for the result obtained through 2 different models, which is DCGAN-SSD and DCGAN-SSD with PSO hyperparameter optimization for busy Indian street video technique.

Fig 7.1. The difference in the experiment result is based on DCGAN-SSD and DCGAN-SSD, PSO optimized from the video in busy Indian street.

DCGAN-SSD DCGAN-SSD, PSO Optimized

Figure 7.2 . Shows the difference in the experiment result based on DCGAN-SSD and DCGAN-SSD, PSO optimized from the video London Taxi in Daytime and night-time.

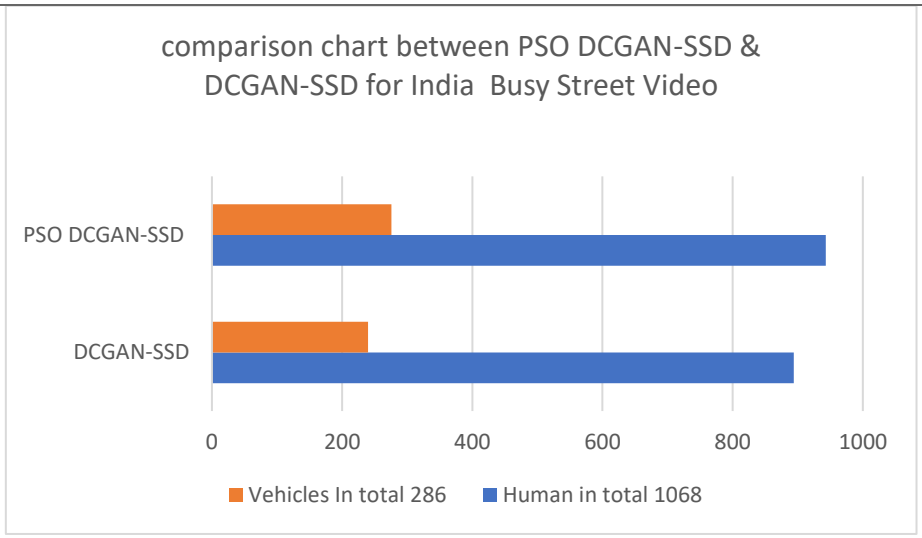


Figure 7.3 Comparison Chart for PSO DCGAN-SSD & DCGAN-SSD for India Busy Street Video

Models	Human Total - 401	Vehicles total - 820	Cyclists total - 9	Traffic Lights total - 16	Overall Efficiency
	Detected	Detected	Detected	Detected	
DCGAN-SSD	212	672	4	12	83%
PSO DCGAN-SSD	380	786	6	15	89%

Table 7.3: Shows the comparison chart for the result obtained through 2 different models, which is DCGAN-SSD and DCGAN-SSD with PSO hyperparameter optimization technique for London Taxi vid

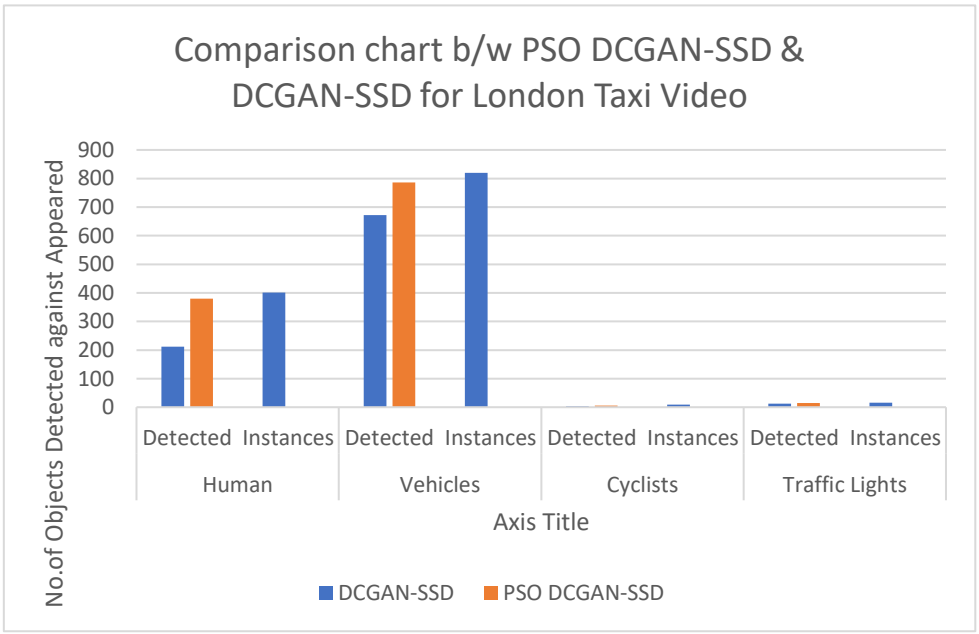


Figure 7.4 Comparison Chart for PSO DCGAN-SSD & DCGAN-SSD for London Taxi Video

From Figure 7.3 and Figure 7.4 the difference is visible. It is evident how the hyperparameters play an important role in taming any models. Also the, Figures 7.1 and Figure 7.2 can see clear difference between DCGAN-SSD and PSO DCGAN-SSD, the PSO optimized DCGAN-SSD performs better than DCGAN-SSD this is because of hyperparameter optimization. The optimized parameters take less time to get trained. It is very clever to compare the training outputs and take the best parameters, and the outputs are shown in the charts in Figures 7.3 and 7.4.

7.6 Summary

This research conducts PSO-based hyperparameter fine-tuning for the proposed DCGAN-SSD object detector. The detected class labels and salient regional features are subsequently used to generate image content using an LSTM network. The empirical results pertaining to object detection indicate the superiority of the optimized DCGAN-SSD network compared to the model with default manually assigned settings. The efficiency of the proposed optimized model can also be demonstrated in the image captions, which yield more enriched class information and labels for use as inputs for the LSTM network to generate more detailed descriptions for future work in future directions. The aim for future work should also be based on conducting hyperparameter fine-tuning for the LSTM model further to enhance performance to better tackle complex image captioning tasks. A variety of spatial-temporal graph neural networks [Yan et al., 2018; Peng et al., 2020] will also be explored for model development.

CHAPTER 8. CONCLUSION AND FUTURE WORK

8.1 SUMMARY OF THESIS

This research has identified a problem with object detection and proposes a solution for overcoming the main issue object detectors face. The thesis began by providing a brief overview and background on the fundamental problem facing object detection and how the quality of object detection is affected due to resolution, which has consistently been the main challenge since object detection was first introduced and persists to the present day. In the past decade, all types of images with different resolutions have been used to train the object detector. The main drawbacks in this technique are training time and the enormity of the dataset required. Even top-performing object detectors will lose efficiency due to the poor resolution of the images. Based on these limitations in the object detector, DCGAN-SSD is used together to overcome these problems, and the framework DCGAN-SSD is employed efficiently with various applications, as shown in different chapters. In the final chapter of this research, hyperparameter optimization was also discussed to increase the model's efficiency.

Chapter 2 provided a detailed outlook on the related work by other researchers who recognize the importance of object detectors in this modern era. It also covered the work already undertaken on object detection, feature enhancement using DCGAN, the history of object detectors, and various architectures used in object detectors. It explored how different resolutions and postures affect object detection and how DCGAN is used to generate and enhance the images from low resolution to a detectable resolution. Finally, it mentioned underwater object detection and the need for optimizing parameters using PSO for training with hyperparameters.

Chapter 3 presented a brief working of object detection using the traditional method, i.e., HOG-SVM. The major achievement of this chapter rested on the necessity of object detection, the roots of object detection, and the factors affecting the object detection framework, such as low image resolutions, low contrast, and different lighting, weather, and environmental conditions. This chapter became the stimulus for carrying forward this research into Deep Learning, and the findings of this chapter laid a great foundation for subsequent research studies.

Chapter 4 sets the stage for the adopted system and discusses the main innovation in this research. The major contribution of this chapter is to denote how efficient is the proposed DCGAN-SSD under different image resolutions and the performance by reducing the resolution issues to a remarkable percentage, which cannot be done by using SSD only. The outputs show the performance of the DCGAN-SSD model and compare the model with the system using SSD only. The comparison results between DCGAN-SSD and SSD were shown in Table 4.1 and in the comparison frames in Figure 4.1. Also, the overall performance of DCGAN-SSD was presented in Figure 4.2.

Chapter 5 dealt with the DCGAN-SSD test in moving cameras. The CIFAR-100 dataset was used as a training set and the trained model is tested in the wild, with moving videos where the challenge is higher than in static ones. In this part of the research, a few videos are considered, i.e. a London taxi video and an Uber self-driving failed case video. The performance of DCGAN-SSD is outstanding in all the test cases. Figure 5.2 shows the London Taxi video, where DCGAN-SSD performance is better in both night and day videos. In previous studies, the Uber test case for self-driving cars failed by killing a pedestrian [Wakabayashi et al.,2018]. The failure test case video was taken and processed with DCGAN-SSD. The cascaded model DCGAN-SSD detected the pedestrian in a distance where the self-driving car would have applied enough braking to avoid the incidents. The results in both Figure 5.2 and Table 5.1 provide evidence that the performance of the DCGAN-SSD model is outstanding. The output results are framed in Table 5.1; the output is also noted as a comparison of results between DCGAN-SSD and SSD only.

Chapter 6 discussed underwater scenarios, where the test environment is entirely different and even more challenging. CADDY and ROBOFLOW are the datasets used to train the model for underwater environments. This chapter's research area is completely different from the previous chapters. Object detection was performed in this part of the research using static and moving cameras. The detection objects included both underwater creatures and human divers. The underwater creatures included fish, turtles, sea mammals, and marine vertebrates. The major challenge is in the training part, since the shapes of all the creatures which come under the detection capability are not varied, therefore, it is necessary to include creatures in various shapes and aspects for training to avoid misdetection. Figure 6.1 shows the resolution problem in video frames in the wild. Resolution issues can be found in all the images, but simulation results show a huge improvement in detection results when DCGAN-SSD is used. The results also show that DCGAN-SSD is better compared with the experiment using SSD only.

Chapter 7 discussed the main issue where object detectors struggle to perform and obtain efficiency due to a lack of optimized hyperparameters. Even though when the performance of the DCGAN-SSD is outstanding as described in all other chapters, due to the sophisticated training process and model settings, the network performance can be further improved by optimizing learning hyperparameters. To solve this parameter identification problem manually would take days of training the model and gauging the performance to find the best hyperparameter. In this research, PSO has been integrated for object detector SSD to optimize the hyperparameters, where the PSO can optimize three major hyperparameters, namely, learning rate, momentum and weight decay. Using PSO worked very well and gave a boosted output in all videos used in other chapters. In this chapter, The performance comparison between DCGAN-SSD against those of DCGAN-SSD-PSO. The comparison results have been tabulated in Table 7.1. The results from this chapter are auspicious and lay the foundation for obtaining optimal solutions using PSO-based hyperparameter selection to avoid high retrains of the manual selection process.

8.2 LIMITATIONS

This research has shown the possibility of overcoming the resolution problem in a very effective way. The main limitations are the hardware requirement to train for various applications and the expert technical support requirements. Also, the initial setup of the DCGAN-SSD configuration can only be done by experts in the field. And, when running a project for testing or demonstration purposes, it is essential to have a very good technical expert with in-depth knowledge in Deep Learning for testing DCGAN-SSD under various environments where the project needs implementation.

8.3 FUTURE WORK

The object detection framework is used widely and in diverse technologies. It is a backbone for various applications where the safety of human life is involved and in situations in which human life would be at risk if something were to go wrong. It is present in applications as wide-ranging as autonomous self-driving vehicles, object detection in cancer treatment, industrial automation, and intelligent sensors. In fact, contemporary human life in the developed world brings us all into close contact with artificial intelligence and machine learning techniques. It will be interesting for future research to analyze the co-existence of other object detection frameworks under the emerging DCGAN-SSD model. A prominent feature of most of these frameworks should be detection quality. This research showed how it was possible to increase detection quality (Chapter 4). The results show the reliability of the proposed model. Also, the training parameters are optimized using PSO, which is another added feature to increase the efficiency of object detection. Future work development of this research will mostly be directed to improvising factors for different applications and making the negligible amount of false negatives positives.

The considered combinational system in this research adopted the VGG-19 architecture. The results obtained in sections 4.3, 5.5, and in Chapter 6 and 7, show that the impact of DCGAN-SSD in above the ground, underwater, during night-time, and during daytime is significantly higher when compared to using the object detectors alone; in addition to this, using PSO with DCGAN-SSD as described in Chapter 7, enhances the object detector performance when compared to using DCGAN-SSD alone. Future researchers can also adopt the DCGAN in combination with object detectors and PSO to make their work easier. The future work of this research will also include testing the model in the natural world environment, with guidelines when humans are involved in the testing. The road map ahead includes the testing and implementation of the technology in agriculture —for example, calculating or predicting the harvesting of crops, rainfall prediction, animal tracking, and vehicle tracking. There are many scopes, and I hope to improve the model to make it useful for different applications. I am also

free to share the research with other researchers who are interested in taking it forward in their direction for the betterment of this world and human life.

REFERENCES

Ahmed, A.H., Kpalma K. & Guedi, A.O. (2017). Human Detection Using HOG-SVM, Mixture of Gaussian and Background Contours Subtraction. *13th International Conference on Signal-Image Technology & Internet-Based Systems* (SITIS), 334-338,.

Ancuti, C., Ancuti, C.O., Haber, T., & Bekaert, P. (2012). Enhancing underwater images and videos by fusion. 2012 IEEE Conference on Computer Vision and Pattern Recognition, 81-88.

Ayachi, R., Said, Y. & Ben Abdelaali, A. (2020). Pedestrian Detection Based on Light-Weighted Separable Convolution for Advanced Driver Assistance Systems. *Neural Processing Letters*. 52/3, 2655-2668

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. NIPS.

Bian, Y., Wang, J., Jun, J. J., & Xie, X. Q. (2019). Deep Convolutional Generative Adversarial Network (dcGAN) Models for Screening and Design of Small Molecules Targeting Cannabinoid Receptors. *Molecular pharmaceutics*, 16(11), 4451–4460. https://doi.org/10.1021/acs.molpharmaceut.9b00500.

Bilal, M. & Hanif, M.S. (2019). Benchmark Revision for HOG-SVM Pedestrian Detector Through Reinvigorated Training and Evaluation Methodologies. *IEEE Transactions on Intelligent Transportation Systems*1-11. 10.1109/TITS.2019.2906132.

Cao, D., Chen, Z. & Gao, L. (2020). An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks. *Hum. Cent. Comput. Inf. Sci.10, 14* .https://doi.org/10.1186/s13673-020-00219-9

Coates, A. and Ng, A. Y. (2012). "Learning Feature Representations with K-means". In Neural Networks, Tricks of the Trade, pp. 561–580. Springer.

Chen, Y., & Chen, C. (2008). Fast Human Detection Using a Novel Boosted Cascading Structure With Meta Stages. IEEE Transactions on Image Processing, 17, 1452-1464.

Chen, Z., Wu, K., Li, Y., Wang, M. & Li, W. (2019) SSD-MSN: An Improved Multi-Scale Object Detection Network Based on SSD. *IEEE Access, vol. 7*, 80622-80632, 2019, doi: 10.1109/ACCESS.2019.2923016.

Chen, Z., Chen, D., Zhang, Y., Cheng, X., Zhang, M., & Wu, C. (2020). Deep learning for autonomous ship-oriented small ship detection. Safety Science, 130, 104812.

Chen, G., Liu, L., Hu W. & Pan, Z. (2018). Semi-Supervised Object Detection in Remote Sensing Images Using Generative Adversarial Networks. IGARSS 2018 - 2018 *IEEE International Geoscience and Remote Sensing Symposium*, 2018, 2503-2506.

Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1800-1807.

Choi, S. & Jung, S.H. (2019). Similarity Analysis of Actual Fake Fingerprints and Generated Fake Fingerprints by DCGAN. *International Journal of Fuzzy Logic and Intelligent Systems* 19(1):40-47.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'05), 1, 886-893 vol. 1.

Denton, E.L., Chintala, S., Szlam, A.D., & Fergus, R. (2015). Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. NIPS.

Dewi, C., Chen, R.C., Liu, Y.T. & Tai, S-K. (2021) Synthetic Data generation using DCGAN for improved traffic sign recognition. *Neural Comput & Applic* (2021).

Dinakaran, R., Sexton, G., Şeker, H., Bouridane, A. & Jiang R. (2017). Image resolution impact analysis on pedestrian detection in smart cities surveillance. *Proceedings of the 1st International Conference on Internet of Things and Machine Learning October 2017* Article No.: 36 1–8 https://doi.org/10.1145/3109761.3109797.

Dinakaran, R., Bouridane, A., Zhang, L., Mehboob, F., Rauf, A., & Jiang, R.M. (2019). Deep Learning based Pedestrian Detection at Distance in Smart Cities. ArXiv, abs/1812.00876.

Dinakaran, R.; Easom, P.; Zhang, L.; Bouridane, A.; Jiang, R.; Edirisinghe, E. (2019). Distant Pedestrian Detection in the Wild using Single Shot Detector with Deep Convolutional Generative Adversarial Networks. *2019 Intl Joint Conf. Neural networks (IJCNN). IEEE*, 2019.

Donohue, C. & Young, S. (2019). Image quality and super resolution effects on object recognition using deep neural networks. Proc. SPIE 11006, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, 110061M.

Dosovitskiy, A., Fischer, P., Springenberg, J.T., Riedmiller, M.A., & Brox, T. (2016). Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38, 1734-1747.

Du, Y., Zhang, W., Wang J. & Wu, H. (2019). DCGAN Based Data Generation for Process Monitoring. *IEEE 8th Data Driven Control and Learning Systems Conference* (DDCLS), Dali, China, 2019, pp. 410-415, doi: 10.1109/DDCLS.2019.8908922.

Du, S., Guo, H., & Simpson, A. (2019). Self-Driving Car Steering Angle Prediction Based on Image Recognition. ArXiv, abs/1912.05440.

Eberhart, R.C. & Kennedy, J. (1995). A new optimizer using particle swarm theory. MHS'95. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39-43.

Enzweiler, M., & Gavrila, D. M. (2011). A multilevel Mixture-of-Experts framework for pedestrian classification. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 20(10), 2967–2979.

Fang, W., Ding, Y., Zhang F. & Sheng, J. (2019). Gesture Recognition Based on CNN and DCGAN for Calculation and Text Output. *IEEE Access*, vol. 7, pp. 28230-28237, 2019, doi: 10.1109/ACCESS.2019.2901930.

Fielding B. & Zhang, L. (2018). Evolving Image Classification Architectures with Enhanced Particle Swarm Optimisation IEEE Access, 6. 68560-68575. ISSN 2169- 3536 .

Fielding, B. & Zhang, L. (2020). Evolving Deep DenseBlock Architecture Ensembles for Image Classification. *Electronics.* 9, 11, 1880. 2020.

Finn, C., Goodfellow, I. & Levine, S. (2016). Unsupervised Learning for Physical Interaction through Video Prediction. *CoRR*, Vol abs/1605.075157, 2016, http://arxiv.org/abs/1605.07157

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for mechanism of pattern recognition unaffected by shift in position. *Biol. Cybernetics* 36, 193–202

Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 580-587.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., & Bengio, Y. (2014). Generative Adversarial Nets. NIPS.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.

He, K., Zhang, X., Ren, S. & Sun, J, (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence. 3*7. 10.1109/TPAMI.2015.2389824.

He, W., Huang, Z., Wei, Z., Li, C. & Guo, B. (2019). TF-YOLO: An improved incremental Netwrok for Real-Time Object Detection. *Advanced Intelligent Imaging Technology Special Issue*, MDPI, Appl. Sci. 2019, 9(16), 3225.

Hu, X., Xu, X., Xiao, Y., Chen, H., He, S., Qin, J. & Heng, P. (2018). SINet: A Scale-Insensitive Convolutional Neural Network for Fast Vehicle Detection. *IEEE Transactions on Intelligent Transportation System*s. PP. 10.1109/TITS.2018.2838132.

Heylen, J., Iven, S., Brabandere, B.D., Oramas, J., Gool, L.V., & Tuytelaars, T. (2018). From Pixels to Actions: Learning to Drive a Car with Deep Neural Networks. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 606-615.

Hussain, R. & Zeadally, S. (2018). Autonomous cars: Research results, issues and future challenges. *IEEE Communications Surveys & Tutorials*, DOI: 10.1109/COMST.2018.2869360.

Janai, J., Güney, F., Behl, A. & Geiger, A. (2020). Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art. *Foundations and Trends® in Computer Graphics and Vision:* Vol. 12: No. 1–3, 1-308.

Jiang, R & Crookes, D. (2019). Shallow Unorganized Neural Networks Using Smart Neuron Model for Visual Perception. *IEEE Access, vol. 7*,152701-152714.

Jiang, R., Bouridane, A., Crookes, D. & Celebi, M.E. (2016). Privacy-protected facial biometric verification via fuzzy forest learning. IEEE Trans. On Fuzzy Systems, Vol.24, Aug 2016.

Jin, Z., Lou, Z.,Yang, J. & Sun, Q. (2007). Face detection using template matching and skin-color information *Neurocomputing*, *70* (4–6) (2007), 794-800.

Kachouane, M., Sahki, S., Lakrouf, M. & Ouadah, N. (2012) HOG based fast Human Detection, International Conference on Microelectronics.

Kim, Daeun & Shahid, Muhammad & Kim, Yunseong & Lee, Won & Song, Hyun Chul & Piccialli, Francesco & Choi, Kwang. (2019). Generating Pedestrian Training Dataset using DCGAN. 1-4. 10.1145/3373419.3373458.

Kim, B., Yuvaraj, N., Sri Preethaa, K.R, Santhosh, R. & Sabari, A. (2020). Enhanced pedestrian detection using optimized deep convolution neural network for smart building surveillance. *Soft Comput 24*, 17081–17092.

Krizhevsky A, Sutskever I, Hinton G.E, (2012) ImageNet Classification with Deep Convolutional Neural Networks , Advances I Neural Information Processing Systems 25, NIPS 2012.

Krizhevsky. A and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kumar, M. P., Packer, B., & Koller, D. (2010). Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems* 1189-1197.

Larsen, A.B.L., Sønderby, S.K., Larochelle, H. & Winther, O. (2016). Autoencoding beyond pixels using a learned similarity metric. *Proceedings of The 33rd International Conference on Machine Learning, in Proceedings of Machine Learning Research* 48:1558-1566.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2323. https://doi.org/10.1109/5.726791.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., & Shi, W. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 105-114.

Lee, S., Bang, M., Jung K. & Yi, K. (2015). An efficient selection of HOG feature for SVM classification of vehicle. *International Symposium on Consumer Electronics* (ISCE), 2015, 1-2, doi: 10.1109/ISCE.2015.7177766.

Levi, K., & Weiss, Y. (2004). Learning object detection from a small number of examples: the importance of good features. CVPR 2004.

Li, Z., Jin, Y., Li, Y., Lin, Z., & Wang, S. (2018). Imbalanced Adversarial Learning for Weather Image Generation and Classification. 2018 14th IEEE International Conference on Signal Processing (ICSP), 1093-1097.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., & Berg, A.C. (2016). SSD: Single Shot MultiBox Detector. ECCV.

Liu, L., Ouyang, W., Wang, X. et al. (2020). Deep Learning for Generic Object Detection: A Survey. Int J Comput Vis 128, 261–318 (2020). https://doi.org/10.1007/s11263-019-01247-4

Lorenzo, P.R., Nalepa, J., Kawulok, M., Ramos, L.S., & Pastor, J.R. (2017). Particle swarm optimization for hyper-parameter selection in deep neural networks. *Proceedings of the Genetic and Evolutionary Computation Conference July 2017* 481–488.

Mahmoud, M. & Guo, P. (2019). A Novel Method for Traffic Sign Recognition Based on DCGAN and MLP With PILAE Algorithm. *IEEE Access*, vol. 7, pp. 74602-74611.

Makhzani, A., & Frey, B.J. (2014). A Winner-Take-All Method for Training Sparse Convolutional Autoencoders. ArXiv, abs/1409.2752.

Ma, Y., Chen, X., & Chen, G. (2011). Pedestrian Detection and Tracking Using HOG and Oriented-LBP Features. NPC.

Manjula S., Tamilselvan, L. & Ravichandran, M. (2016). A study on Object Detection. *IJPT conference*, ISSN: 0975-766X.

Masoud, O., & Papanikolopoulos, N. (2001). A novel method for tracking and counting pedestrians in real-time using a single camera. *IEEE Trans. Veh. Technol*., 50, 1267-1278.

Maqueda, A.I., Loquercio, A., Gallego, G., García, N., & Scaramuzza, D. (2018). Event-Based Vision Meets Deep Learning on Steering Prediction for Self-Driving Cars. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5419-5427.

Mv, A., & Khan, D.M. (2020). Recent Trends on Object Detection and Image Classification: A Review. *2020 International Conference on Computational Performance Evaluation* (ComPE), 427-435.

Nguyen, A.M., Clune, J., Bengio, Y., Dosovitskiy, A., & Yosinski, J. (2017). Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3510-3520.

Papageorgiou, C., & Poggio, T.A. (2004). A Trainable System for Object Detection. International Journal of Computer Vision, 38, 15-33.

Pedoeem, J., & Huang, R. (2018). YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. IEEE International Conference on Big Data (Big Data), 2503-2510.

Peng, H., Wang, H., Du, B., Bhuiyan, Z.A., Ma, H., Liu, J., Wang, L., Yang, Z., Du, L., Wang, S., & Yu, P.S. (2020). Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. Inf. Sci., 521, 277-290.

Prats, M., Fernández, J.J. & Sanz, P.J. (2012). Combining template tracking and laser peak detection for 3D reconstruction and grasping in underwater environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, 106-112.

Radford, A. Metz, L. & Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *Proc. Int. Conf. Learn. Represent*., pp. 1-16.

Reed, S., van den Oord, A., Kalchbrenner, N., Colmenarejo, S.G., Wang, Z., Chen, Y., Belov D. & de Freitas, N. (2017). Parallel Multiscale Autoregressive Density Estimation. *Proceedings of the 34th International Conference on Machine Learning*, 2017, 2912--2921, Vol 70.

Redmon, J., Divvala, S.K., Girshick, R.B., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.

Ren, S., He, K., Girshick, R.B., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, 1137-1149.

Sallab, A.E., Abdou, M., Perot, E., & Yogamani, S.K. (2017). Deep Reinforcement Learning framework for Autonomous Driving. ArXiv, abs/1704.02532.

Schneiderman, H., & Kanade, T. (2000). A statistical approach to 3d object detection applied to faces and cars. Robotics Institute Carnegie Mellon University Pittsburgh, PA

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556.

Storey, G., Jiang, R., Keogh, S., Bouridane, A. & Li, C. (2019). 3DPalsyNet: A Facial Palsy Grading and Motion Recognition Framework using Fully 3D Convolutional Neural Networks. IEEE Access, 7. 121655-121664. ISSN 2169- 3536

Strachan, N. J. C. (1993). Recognition of fish species by colour and shape. *Image and Vision Computing*, Vol- 11, Issue 1, Jan,1993, 2-10, https://doi.org/10.1016/0262-8856(93)90027-E.

Suárez P.L., Sappa A.D., Vintimilla B.X. (2017). "Colorizing Infrared Images Through a Triplet Conditional DCGAN Architecture". In: Battiato S., Gallo G., Schettini R., Stanco F. (eds) *Image Analysis and Processing* - ICIAP 2017. ICIAP 2017. *Lecture Notes in Computer Science*, vol 10484. Springer, Cham. https://doi.org/10.1007/978-3-319-68560-1_26.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015). Going Deeper with Convolution. *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016*.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017) Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 31(1).*

Tan, T., Zhang, L., Lim, C., Fielding, B., Yu, Y. & Anderson, E. (2019) Evolving Ensemble Models for Image Segmentation Using Enhanced Particle Swarm Optimization. *IEEE Access, 7*. 34004-34019. ISSN 2169- 3536.

Turing, A. M. (1950. COMPUTING MACHINERY AND INTELLIGENCE, Mind, Volume LIX, Issue 236, October 1950, 433–460, https://doi.org/10.1093/mind/LIX.236.433.

Vincent, P. & Larochelle, H. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*. 11, 3371–3408.

Viola, P., Jones, M.J. & Snow, D. (2003). Detecting Pedestrians Using Patterns of Motion and Apperance. Proceedings Ninth IEEE International Conference on Computer Vision, 2003, pp. 734-741 vol.2.

Viola, P.A., & Jones, M.J. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 1, I-I.

Wakabayashi, D. (2018, March 19) Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam. *New York Times*. https://www.nytimes.com/2018/03/19/technology/Uber-driverless-fatality.html.

Wang, X., Yang, M., Zhu, S. & Lin, Y. (2015) Regionlets for Generic Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 10, 2071-2084.

Wang, K. & Liu, M. (2020). Object Recognition at Night Scene Based on DCGAN and Faster R-CNN. *IEEE Access.* 8. 193168-193182. 10.1109/ACCESS.2020.3032981.

Wang, C., Dong, S., Zhao, X., Papanastasiou, G., Zhang H. & Yang, G. (2020). SaliencyGAN: Deep Learning Semisupervised Salient Object Detection in the Fog of IoT. *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, 2667-2676, April 2020.

Wu, Q., Chen, Y., & Meng, J. (2020). DCGAN-Based Data Augmentation for Tomato Leaf Disease Identification. IEEE Access, 8, 98716-98728.

Xie, S., Girshick, R.B., Dollár, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5987-5995.

Xie, S., Shan, S., Chen, X., Meng, X., & Gao, W. (2009). Learned local Gabor patterns for face representation and recognition. Signal Process., 89, 2333-2344.

Yan, S., Xiong, Y., & Lin, D. (2018). Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Zhao, J.J., Mathieu, M., Goroshin, R., & LeCun, Y. (2015). Stacked What-Where Auto-encoders. ArXiv, abs/1506.02351.

Zhang, J., Chen, L., Zhuo, L., Liang, X., & Li, J. (2018). An Efficient Hyperspectral Image Retrieval Method: Deep Spectral-Spatial Feature Extraction with DCGAN and Dimensionality Reduction Using t-SNE-Based NM Hashing. *Remote Sensing*, 10(2), 271. MDPI AG.

Zhang, X., Gao H., Guo M., Li G., Liu Y. & and Li, D. (2016). A study on key technologies of unmanned driving. *CAAI Transactions on Intelligence Technology*, vol.1, pp 4-13, 2016, ISSN 2468-2322.

Zheng, Y., Wu, W., Chen, Y., Qu, H., & Ni, L.M. (2016). Visual Analytics in Urban Computing: An Overview. IEEE Transactions on Big Data, 2, 276-296.

Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). Object Detection in 20 Years: A Survey. ArXiv, abs/1905.05055.