Jimma University

Jimma Institute of Technology

School of Computing

Multiple object detection and recognition from live video cameras using deep convolutional networks for autonomous vehicles.

By: Kelemu Diress

Jimma, Ethiopia

June-2019

Jimma University

Jimma Institute of Technology

School of Computing

Multiple object detection and recognition from live video cameras using deep convolutional networks for autonomous vehicles.

By: Kelemu Diress

Advisor: Getachew Mamo (PhD)

Jimma, Ethiopia

June-2019

Jimma University

Jimma Institute of Technology

School of Computing

Multiple object detection and recognition from live video cameras using deep convolutional networks for autonomous vehicles.

By: Kelemu Diress

A Thesis Submitted to the Department of Information Technology in Partial Fulfillment for the Degree of Master of Science in Information Technology

Jimma, Ethiopia

June-2019

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged.

Declared by:

Name: **Kelemu Diress**

Signature: ________________________

Date of submission: June, 2019

Jimma University, Ethiopia

This thesis has been submitted for examination with my approval as a university advisor.

Confirmed by advisor:

Name: **Getachew Mamo (PhD)**

Signature: _________________________

Jimma University

Jimma Institute of Technology

School of Computing

Multiple object detection and recognition from live video cameras using deep convolutional networks for autonomous vehicles.

By: Kelemu Diress

This is to certify that the thesis prepared by Kelemu Diress, titled *Multiple object detection and recognition from live video cameras using deep convolutional networks for autonomous vehicles* and Submitted in partial fulfillment of the requirements for the Degree of Master of Science in *Information Technology* compiles with the regulations of the University and meets the accepted standards with respect to originality and quality.

Approved by board of Examining Committee:

	Name	Signature
Dean, Faculty of computing:	Getachew Mamo(PhD)	
Advisor:	Getachew Mamo (PhD)	
Internal Examiner:		
External Examiner:		

Jimma, Ethiopia

June -2019

Acknowledgment

I first and foremost, praise and thanks to God, the almighty, for His blessings throughout my research work.

I would like to express sincere gratitude to my adviser, Dr. Getachew Mamo for his constructive comments and advices. He is always kind to correct my work, consistence until the end of this thesis.

My acknowledgment shall also pass to Haramaya University which provide me the opportunity to join in this program and support me until I finish my study.

I want also acknowledge all staff of Jimma University school of Computing for all of their cooperation. Specially, Zerihun Olana for help me document edit.

I acknowledge the Karlsruhe Institute of Technology and Toyota Technological Institute for their critical role in the creation of the free publically available KITTI datasets used in this thesis work.

I’m absolutely delighted to say thanks to my wife Tigest Bayu and my daughter Yohana Kelemu to help and give me happiness in my life.

Last but not the least, thanks my parents and to my brothers and sisters for supporting me spiritually throughout writing this thesis and all my friends.

Abstract

In self-driving technologies the system perceived the environment without human intervention in which system can detect different obstacles and make decisions for smart transportation. In this studies we adapt and design different technique for detecting and recognize an objects which used different components like data processing, noise removal, input resize, input vector preparation, feature extraction, classification and regression problems. However, in the current research, the performance of the detector is not reach matured. And they used fully connected layers in detection networks for the detector models. Due to this problem the performance in the detection networks is not satisfied and doesn't extract many features. We developed a new model in detection networks using convolutional neural networks and extracted different level of features which helps the model to extract more usable information to the classification and regression problems in the detector. In the proposed model we used 3 layers of fully convolutional neural networks and two fully connected layers to develop the model. In the experiment, we have evaluate both the localization and classification mAP of the networks. And, we obtained 84% mAP model performance. And also, we evaluate the quantitative and qualitative results for the networks for each categories in the input data. Thus, our model, we detected more objects that doesn't detect in the previous works.

Key word: multiple object, object detection, detection network, CNN

Table of Contents

List of Figures

Lists of Tables

List of Algorithms

Lists of Acronyms

2D	Two Dimensional
3D	Three Dimensional
AI	Artificial Intelligence
ANN	Artificial Neural Network
AV	Autonomous Vehicles
BPNN	Back Propagation Neural Network
CNN	Convolutional Neural Networks
COCO	Common Objects in Context
DN	Detection Network
FCL	Fully Connected Layer
FCN	Fully Convolutional Network
FN	False Negative
FP	False Positive
GPU	Graphical Processing Unit
HOG	Histogram of Oriented Gradients
ILSVRC	ImageNet Large Scale Visual Recognition challenge
IoU	Intersection over Union
KITTI	Karlsruhe Institute and Toyota Technological Institute
LRF	Local Receptive Fields
mAP	Mean Average Precision
NMS	Non-Maximum Suppression
pr	Precision recall
RCNN	Regional based Convolutional Neural Networks
ReLu	Rectified Linear Units
RoIP	Region of Interest Pooling
RPN	Region Proposal Network
SDAE	Stack Denoising Auto-Encoder
SIFT	Scale Invariant Feature Transform
SRF	Speed up Robust Features
SSD	Single Shoot Detector
SVM	Support Vector Machines
TN	True Negative
TP	True Positive
VGG	Visual Geometric Group
XML	Extensible Markup Language
YoLo	You only Look once

CHAPTER ONE

INTRODUCTION

1.1. Background

In recent year, there has been a significant increase in research interest supporting the development of Autonomous vehicles which are gaining attention in the world. Potentially these technology is capable of perceive the environment without human intervention. If this system realized and adapted by the society, benefits such as reduction of traffic accident facilitation of smooth transportation are expected. It also, benefit go from reducing contamination of the environment by improving driving and fuel efficiency and help controlling the traffic flow and parking problem. So, autonomous driving is clear and predicted that will dramatically change transportation [5]. Since, self-driving cars have been pushed to the fact that government, research institution and industry are investing vast amount of human resource, time and money.

However, achieving fully autonomous driving is very challenging problem and, today its one of the main drivers of the development of a broad range of new technology. The key challenge for developing automated vehicle is to manage and combine the significant amount of data coming from the different sensor and to create a consistent model from this data that make decision. In autonomous vehicle object detection and recognition belongs to the core ability as they are requires to perceive the surrounding environment. Research and development of object detection and recognition techniques have significantly progressed by advancement of deep learning techniques. In particular, Deep Convolutional Neural Networks (CNNs) was one of the earliest algorithm to employ in object detection and recognition problem.

In order to overcome the limitations of the earlier neural networks, Geoffrey Hinton introduces a deep learning application which was Deep CNN in 2017, mainly to simulate the learning process of the human brain [7, 12]. It is specifically, simulating the human brain's multi-layer abstraction mechanism to achieve an abstract expression of an object. Since, it is a new area of machine learning research, which has been introduced with the objective of moving machine learning closer to one of its original goals of AI[8]. And also, it is machine learning algorithm based on learning multiple label of representation and allows to train an AI to predict outputs, given a set of inputs.

Recently, Convolution Neural Networks (CNNs) do end-to-end machine learning trainable multi-stage feed forward networks[7,12]. The input and the output of each stage in CNN are called feature maps. For the first stage, input features maps are images or videos. The output at each stage are the feature extracted from all the locations of the input feature maps of this stage. Each stage consists of three layers, i.e. convolutional layer, non-linearity layer and pooling layer. The last stage is a fully connected layer which outputs prediction on classes.

Object detection and Recognition are one of the current problem for computer vision states of art. It is detecting and localizing an object from the whole images. Since, it considered to be the most basic application of computer vision. Rest of the other development in computer vision are achieved by making small enhancements on top of this. In multi-class object detection is detecting multiple objects (such as pedestrian, vehicles, cars and bicycle) from input data. So, object detection in autonomous vehicle is one of the most challenging and important states of art in current research area.

For detecting and recognizing an object by using deep CNNs, automatically learns features representation from a large amount of data. Hence, learn multiple level of feature transformation of data that make it easier to extract useful information when building detectors or predictors. When we train the model, the Deep neural networks gets weights that find the relevant pattern to make better prediction and deep networks internally build representation of pattern in the data[12].

Our proposed system will solve multi-class detection and recognizing problem of autonomous vehicle from video camera by using deep CNNs which compare images piece by piece(the pieces that is looks for are called features). This will be done by classifying an objects in its class like pedestrian, vehicle and cyclist. Then the system can detect and recognize an objects when they are and localize the exact position from the input data.

1.2. Motivation of the Study

In most investigation show that about 80%-90% of all traffic accidents occur due to human error. The large number of death and injury that occur due to take alcohol, lack of attention and lack of experience. Even if, when the vehicle have a partial automated which detect and recognize the maximal closer to other objects might be reduce the number of accidents by giving alarm to the driver before the accidents occur.

Autonomous vehicles are getting more and more important, this technology still far from being mature. In our country, the road are very unpredictable dynamic environment where multiple actors such as pedestrian, car, or other vehicles coexist together. In this way, it is necessary to provide autonomous vehicle with robust perception systems in order to correctly understand the environment. Therefore, being able to interpret what is happening in the environment to act in consequence.

1.3. Statement of the problem

The detection and classification of multiple object is one of key importance in numerous object detection applications. In computer vision systems, object detection generally carry important information about the object itself and thus are targets for recognizing. Since, features form the main content of input data. Object detection is a very important step to predict the exact locations of each objects in a given input data and recognizing their categories[10].

Object detection on computer vision are very common for autonomous vehicle in current object detectors shows not detect all instances in the inputs. Since, in autonomous vehicles the system detected and recognized many instances in the road for different objects. And also, its features are difficult to detect and recognize them. For these problems many works have done to improve object detectors. The recent approach done as an option to overcome these problems is using fully connected in detection networks which learn features from different object class and learn about the environment. Even though, it improves some problems still it is difficult to detect and recognize objects according to their different views and features of objects.

In many computer vision, object detection and recognition is the first task being performed as it allows to obtain further information regarding to the detect object. When an object instance has been detected we can obtain further information like to recognize the specific objects and tracks an object. And also, object detection has been used in many applications, such as person counting, human-computer interaction, recognition and tracking for security, image retrieval and transportation. But, each of these application has different requirements to detect and recognize an instance of objects and many application consider a single object class detection. In multiple class detection used different requirements and models to perform in autonomous system from input data.

In addition to this, the other problem if a large number of object class is being detect and recognize the kind of classes that the system can handle without accuracy loss. By developing multiple class classifiers designed specifically to detect multiple instance of object. In current works, the features are detected using fully connected layers at detection networks. However, the fully connected networks have many parameters and sparse connections. Since, when many parameters used in the networks it needs more computational resources.

In our proposed system, the Deep CNNs approach uses the input data with have different categories. Different object class have their own many features for identifying and localizing the position from input data. Then the system learn from large datasets for detect and localize different objects from an image and make decision. The Deep CNNs need to be able to detect and recognize pedestrian, car, truck, tram, van and cyclist to safely navigate their environment.

1.4. Objective of The study

1.4.1. General Objective

The main objective of this thesis is automatically detect and recognize instance like pedestrian, Vehicles, truck and Cyclist objects from live video camera by using Deep neural network in particular, by using Deep Convolutional Neural Networks in the domain of Autonomous Vehicles.

1.4.2. Specific objectives

Under the above general objective the thesis has the following specific objectives:

- ✓ To remove noise and artifacts from input data
- ✓ Review previous research papers for developing multi class object detection and recognition system
- ✓ Selecting the appropriate mechanism and algorithm for multi class object detection and recognition of an object from video camera
- ✓ Developing multi class object detection and recognition systems for autonomous vehicle from video camera
- ✓ To evaluate the performance of the result obtained from proposed model using test images.

1.5. Methodology

It is how to work each specific objectives of our proposed system. So, collecting dataset from different sources, reviewing different related works, implementing the proposed work and evaluating the performance of the result obtained.

A number of methods (techniques) are employed for the successful completion of this study. Some of them are discussed below

1.5.1. Review of literature

A number of related works and resources have been reviewed. This consists of conference and journal articles, white papers and Object Detection systems developed by others. The large portions of reviewed materials are conference and journal articles.

1.5.2. Design and implementation tools

The design procedure of the research comprised a series of preprocessing, deep feature extraction, classification of different categories and prediction of bounding box for each objects in an image.

The programming aspect of the detecting and recognizing multiple objects is done using python programming language. The reason to use Python is on one hand, the exposure of the researcher to the language are platform Independence and on the other hand, due to the fact that more AI-based tools that can be used for this research are Python based. It combines become the power general purpose programming languages with easy to use domain specific scripting languages like MATLab or R. python has libraries of data loading, visualization and image processing.

In order to develop the proposed system, we have used different open source python application interfaces like Anaconda. Which is used a python distribution made for large scale data processing, predictive analytic and scientific computing. And also, it comes with different libraries like numpy, scipy, matplotlib and pandas for used in image Processing.

1.5.3. Data collection

The dataset we utilize in our thesis is the KITTI object detection benchmark 2012[5]. This dataset consist of 7481 training and 7518 testing real data. Each image is accomplished by a list of detection, where each detection gives for both classification and regression problems of the object detection. In classification it gives the category of different objects and in regression problem it gives the image bounding box coordinates, width and height of detection in meters and 3D position and orientation of the detection in x, y and z coordinates.

1.6. Significance of the study

Multiple object detection system is an important tool in almost all computer vision application areas. Proper identification and classification of multiple object are very crucial and pose a very big challenge to the object detection researchers [9, 20, 25]. Object detection application serves as an important preprocessing tool for tasks such as information extraction, information retrieval, person and vehicle counting and other information processing applications. Based on these facts, object detection system plays a crucial role in information extraction based researches and applications associated to the objects. It also simplifies development of the following.

- ✓ Since there are many vehicles in Ethiopia, and the driving system are not yet self-governance and this work to motivate researcher to develop self-controlled vehicles.

- ✓ The system support the driver by learning the environment and govern by itself and bring moral satisfaction using the technologies.

- ✓ As academic exercise, this research work increase the experience of other researcher about autonomous vehicles.

- ✓ The system greatly improve traffic flow safety and creating clean smart and safe cities.

1.7. Scope

This work attempts to detect and recognize different objects from a given inputs via deep ConvNets. It helps to find the exact position of an objects and recognize each classes from KITTI datasets. In general, this thesis might be wide from data processing to reduce false positives using image processing techniques, training CNNs for high level feature representations of classification and regression, BPNN for region proposal networks and Detection Networks. And also, we categorize the classes of objects and predict the bounding box. The model have proper classification of the different object class and draw bounding box around them.

1.8. Thesis Organization

The remaining of this thesis report is organized as follows. Chapter Two, presents literature review on object detection and recognition of different objects. Chapter Tree, introduced related works, which is discussed research that have conducted on object detection and recognition for self-driving vehicle. The design of object detection and recognition of self-deriving car is presented on chapter Four. The evaluation, test result and discussion are described in chapter Five. Lastly, conclusion and future works are pointed in chapter Six.

CHAPTER TWO

LITERATURE REVIEW

This chapter presents the states of the art in object detection and recognition in the domain of autonomous technologies with overview their components and the algorithms developed. This part of study review of literature on the concepts that are basis for the research. The chapter begin with a brief introduction to object detection and recognition for self-driving car.

2.1. Introduction

An autonomous vehicles means guide itself without human conduction. The vehicles have an able to detect and recognize them all obstacles like pedestrian, car, truck, tram, van and cyclist on the road to learn the environment. And also, the system on this technology recognize an object on the road with the proper position where they are. Self-driving vehicles has become a concrete reality and may pave the way for future systems where computers take over the art of driving.

In autonomous technologies not only one object is detect may have different objects needed to detecting and may have different categories. An object detection task can be described as composition of sub-tasks data per-processing, object identification/segmentation and object classification and regression. The object detection sub-task is concerned with identifying an object among other object is called classification and identifying the position of different objects where they are called regression from video frames.

Once all object from input data detected, they are passed for classification in a predefined set of categories such as picture, person, location, car or others. This task is known as object classification. The classification is done by classifiers based on some patterns and a set of features. The regression problem determines the boundaries (bounding box) of each object (the place from where it starts and the place it finishes) in an image. The final multiple object detection system performance is measured considering the object localization and classification tasks together.

2.2. Object detection in the domain of autonomous driving

An object detection is a computer technology related to computer vision and image processing that deals with detecting instance of semantic objects of a certain class(such as pedestrians, car, truck, cyclists) in a digital images or video data. In the domain of object detection, a given input images or videos might have different categories of objects and might have a number of object in the same categories. Hence, object detection technologies classified each categories and predict the exact position of each instance of objects in a given input data.

Autonomous systems are an application to get information from the environment through sensor or from database and then analyze and interpret of the pattern to do tasks without human intervention. In the domain of autonomous vehicles, we have different obstacles should be detect and recognize in the environments. Since, autonomous vehicles able to detected different obstacles/objects in input images/videos and localize where they are in order to perceive the environments.

In order to build up an object detection for self-driving car, it combines classifications and localizations multiple tasks: an object classification, which involves predicting the class of an object in input data. And object localization, which identifying the location of one or more objects in an input data and draw a bounding box around their extent. The object detection algorithm not only have label this as say a car but also responsible for putting a bounding box, or drawing a rectangle around the position of an object in the images. Hence, in the autonomous vehicles different instance of objects are predict to detect the positions and recognize the exact class classification with localization each objects where they are present in image/video data.

2.2.1. Video Camera

The primary way of self-driving technologies is to get information from the environment via different sensors. In recent year, different sensors are came to the market such as video cameras, automotive radars and laser scanners to perceive the environment. Smart cameras with real-time video object generation by a real time video object generation and MPEG-4 compression for maintaining quality and bandwidth reduction, by using intelligent vision sensors called smart cameras, which execute autonomous vision tasks and report events and data to a remote base-station. Their system implementation was modular, that involving multiple threads that are synchronized for the tasks of grabbing, detection, tracking, camera control, compression, and visualization. The performance of the system demonstrated in experiments which was on both static and moving object that shows good performance.

2.3. Image processing

Digital image processing (DIP) is a method to convert an images into digital form and perform computer algorithm (operations) on it, in order to get an enhanced images or extract some patterns or useful information from an images. And also, the purpose of DIP step is to prepare our datasets in the way that suitable for machine learning or deep learning algorithm.

In machine learning specifically, deep learning algorithm uses neural networks with a lot of hidden layer and large amount of training data in our work images data from video records. So, different states of art[17,41,42,53,54] use image processing methods to enhance the performance of the model.

2.3.1. Preprocessing

In the preprocessing step preparation are made to extract features that are detecting and recognizing an objects in each inputs[41,42]. It is the transformation on the video frames/images before feed into machine learning or deep learning algorithm. It is a method of enhancing the image for feature extraction. The choice of preprocessing method to be adapted on video frames that recorded from street on the days, dims light, snows, rainy time depends on the types of application for which the image is used. It also, mainly to reduce the noise and unwanted artifacts in the video frames/images and misrepresentation of the images, and also applied on images at lowest level abstraction. So, our proposed approach uses data normalization, median filter and histogram equalization to enhance the image quality.

2.3.1.1. Normalizing input image

Data normalization is an important step which ensures that each parameter (pixels, in this thesis) has similar distribution[53]. This makes convergence faster while training the network. Data normalization is done by subtracting the mean from each pixel, and then dividing by the result by the standard deviation. For image inputs we need the pixel numbers to be positive, we might choose to scale the normalized data in the range [0, 1] or [0,255]. For our dataset we used the range of [0, 1]. Data normalization can be calculated using equation (2.1),

$$\boldsymbol{Normalized = \frac{x-xmin}{Xmax-Xmin}} \text{ ……………………………………………………...(2.1)}$$

Where, x_{max} and x_{min} are the maximal and the minimal values for the variable x data respectively.

2.3.1.2. Noise removal

In video sequence denoising an important step specifically in real time image processing, where the original images are poor due to noise and artifacts. Media filter is a non-linear operation used to reduce artifact and noise [54]. As sharpening and shadows perform using high frequency signal in image, the noise of the image will get higher. Noise removal using media filter is more effective in terms of eliminating noise and preserving edge and fine details of digital image. As applied many works [17], to reduce noise they used medial filter. There are a number of different filter such as high pass, low pass, and mean media available [17]. We used media filter technique for our works to remove some of the noises because under certain condition it preserve Sharpe edge of the image and fine details of digital images in camera recorded video sequences data while removing noise. The media filter noise calculated using equation (2.2)

$$\boldsymbol{y(m,n) = median[x[i,j], (i,j)\varepsilon\omega]} \text{ …………………………………...(2.2)}$$

Where ω represents a neighborhood defined by the user center around location (m,n)

2.4. Machine learning

Machine learning is a method that capable of acquiring and integrating the knowledge automatically [11,17]. It is a sub-part of AI which is designed an approach to achieve Artificial Intelligence based on the idea that learn from data, identify pattern and make decision without being explicit programmed. The algorithm learn from analytical observation, training, experience, and other means, results in a system that can exhibit self-improvement, effectiveness and efficiency. Knowledge and a corresponding knowledge organization are usually used by a machine learning system to test the knowledge acquired, interpret and analyze. One of the machine learning algorithms is taxonomy based depending on outcome of the algorithm or types of input available[7].

Machine learning task are usually described in terms of how the machine learning system should be process an example. An example is a collection of features that have been quantitatively measured from some object or event that we want the machine learning system to process. We typically represent an example as vector $x\epsilon R$ where each entry x_i of the vector is another features. The feature of an image are usually the values of the pixel in the image.

The machine learning models can learn by analyzing the training data, and then make a prediction for new data. The models are used to solve classification and regression for object detection and recognition problem. In object detection for different objects they have its own features are extracted from dataset and pattern learn from extracted features then predict the objects class/categories and bounding box. But for predict the types of objects by writing only hard coded without using Machine learning it might be write a billion of line for extracted some features of an object. So, machine learning has a big advantage to learn features from data and predict the output in small code rules by using large data sets as input. In our work also, features are extracted from video/images data automatically with Machine learning; rather than relying on handcrafted feature extractor.

In machine learning different types of techniques may apply to extracting features and predict the classification and regression for object detection and recognition problem. There are three types of machine learning model algorithms. The first, supervised learning is a machine learning model which learn pattern from a given labeled data and the desired output. The algorithm to identify the rules of pattern by using the labeled data and outputs as an input. The second, unsupervised learning is learning from unlabeled data/only input data. This model gives input data to the algorithm and it learns and predict from experience, which is mostly through clustering and dimensional reduction. The third, Semi supervised learning model used to learn both labeled and unlabeled data. The model is first per-trained using unsupervised data and then improved with supervised data.

2.4.1. Classic object detection and recognition

Standard methods for object detecting and recognizing are commonly based in feature matching. These types of detector or recognizer are aims to find sets of per-learned shape features/descriptors of the object in images. And, the algorithm have been designed to extract local features like edges, contours. These features are the key descriptors of images, which are used as data to be as an input to algorithms for further creation of hypothesis. Some descriptor algorithm such as Scale Invariant Feature Transform(SIFT)[1,2] or Speed up Robust Features(SRF)[3], have been typically used extracting low level features from images for locating points of special interest between images. The SIFT algorithm is to find extreme key points in scale and rotation change. In SIFT, descriptor are supposed to be invariant against various transformation which might make image look different although they represent the same object. The SRF algorithm extracting key points(features) from different regions of images then find the right orientation of the key point and finally computation of a descriptor of the key point is performed by matrix thus is very useful in finding similarity between images.   Higher level feature can be extracted with other advanced descriptor like Haar features[16,49](which have edge, line and four rectangle features) or Histogram of Oriented Gradients(HOG)[13,14,48], which have shown good performance in human and pedestrian detection. Best descriptor should be scale, rotation and illumination invariant as well as pose and occlusion aware. This usually involves hand crafting and personalizing them to the application of image space being used, which reduce its generalization capability.

Neural in Networks [27] in pattern recognition have historically performed pretty well in situations where convolutional featured based approach struggle. They were taken with great optimism in the 1980s, specifically after back-propagation was expanded widely known. In a standard Neural network any neuron is connected to every other neuron is the adjacent layers. If more neuron are added, the connection will grow exponentially and the network become very huge in dimension.

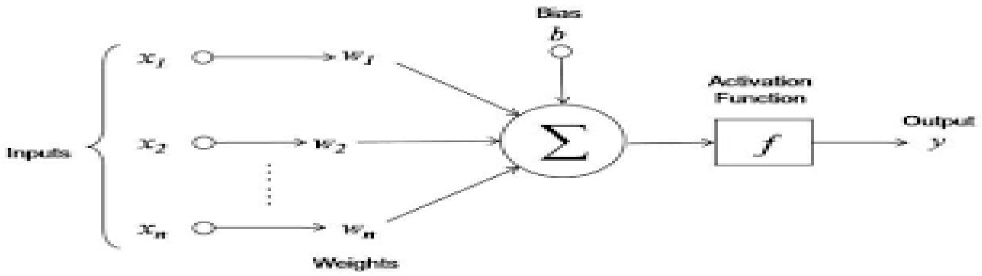


Figure2-1: Standard Neural Networks

The activation function y of each hidden neuron is computed as the summation of each of the weight input plus the bias bj. The equation to perform neural network is as follows:

$$Y = \sum (weight * input) + bias \quad \text{...} (2.3)$$

In each neuron different weight and bias value are computed and when the network gives deep the total number is grow. In a neural network would have 10,000 input units for 100x100 input images, and each of the hidden network 20,000 values would need to be computed (10,00 weights and 10,00 bias). So, due to these dimensional problem and computational cost standard neural network is not preferable for computer vision problems like object detection, object recognition and classification.

2.4.2. Deep learning

In the last year, neural network have again raised attention of the researcher. Specifically, deep neural networks are setting all sort of records and defeating standard approach such as SFIT, SRF HOG and Haar-Like (disused on section 2.4.1) on many computer vision problems such as object detection, medical imaging, voice identification and many others.

Deep learning is a sub-field of machine learning which is sub-filed of AI(show figure(2.2)) with several level of data representation which is extract the set of features that characterized the combination of color, texture and shape of the input images. The algorithm of deep learning inspired by the structure and function of human brain called artificial neural networks, learn from large amount of data. It is an improvement of artificial neural networks run data through several deep layers or consists of more hidden layers, each of which passes a simplified representation the data to the next layer. The algorithm have an ability to extract large number of features and learn complex function mapping the input to the output directly from a thousand of row data. Today, several deep learning-based computer vision applications such as CNN, DBN, RNN and SDAE are performing complex task even better than human. It becomes extensively applied to computer vision, speech recognition, natural language processing, online advertising, logistics, object detection and recognition and many applications. Since, Deep learning computer vision helping self-driving cars figure out where the other cars, pedestrian, traffic light around and recognize them. It have the capable to learn from video/image data for detecting the exact position of an object or recognizing the categories of each object in video sequences or images.

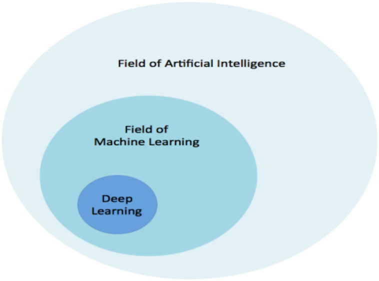


Figure 2-2: Relation of deep learning with machine learning and AI

A deep neural network can essentially memorize all possible mapping after successful training with a sufficiently large knowledge database and making intelligent predictions such as doubtful objects and not-doubtful objects from videos/images of unseen data. The performance of deep learning algorithm is high when we have a large amount of trainable datasets. In our time of information age, inspired by the application of computers, vast number of data are generated in each second. Since, deep learning algorithm require thousands of data to learn from. The success of the algorithm comes from their ability to create powerful object representations from large amounts of data such as video, image, voice and text data without the need to hand designed features.

In Deep learning, particularly ConvNets extract and learn features from input data end to end. Which means the algorithms extracts different features from the input data and then learns from the trained. In convolutional neural networks built by more than one hidden layers to extract different level of features.

2.5. Convolutional Neural networks

A convolutional Neural Network(CNN) also called ConvNets is a Deep learning algorithm which simplifies the process of feature extraction through convolution[23,25,41]. Convolution is a mathematical operation, which map out an energy function, which is a measure of similarity between two images. It use image as an input with assign learnable weights and bias into different objects and be able to differentiate one from the other. It is able to capture the spatial and temporal dependencies in image through the application of relevant filter/kernel. Since, Convolutional Neural Networks(CNN) is a special family of deep learning algorithm and serve as rich feature extractor which are used in image classification and object detection, object segmentation, image and video recognition, recommendation system and many other advanced tasks.

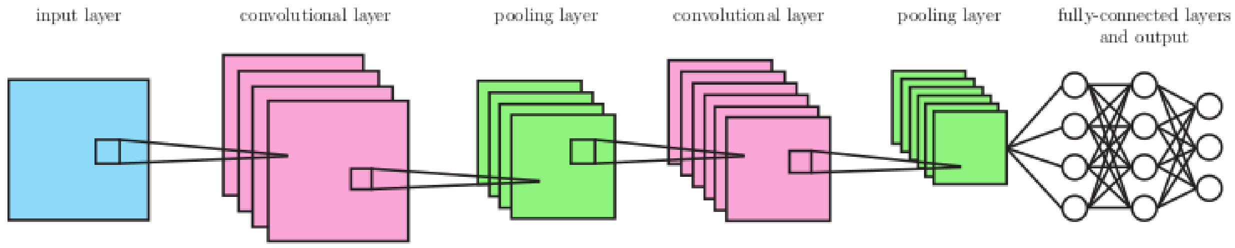


Figure 2-3: Simple ConvNets Architecture[7]

The main advantage of CNNs over standard fully connected Neural Networks are able to take into account the local spatial structure of the images. Therefore, the far away and those which are closed by input image pixels are not treated in the same way in the models. Moreover, CNNs performs a better fitting to the datasets due to the reduction in the number of parameters involved and reusability of weights. And also, in CNNs model allowing to faster training and can create deeper networks with many layers. It use a special architecture which is particularly well adapted to classification problems, allowing a faster training and therefore enabling the creation of deeper networks with many layers. Due to these, deep convolutional networks in most algorithms for image recognition and classification [46].

CNNs as Standard Neural networks have multiple sequential blocks of layers in a path that the output from one layer is serve as input for the following one. So, in ConvNets many of the standard Neural Networks concepts such as regularization, back-propagation, gradient descent, etc. are applied. However, CNNs allow to train many-layer networks and prevent dimensionality problem by introducing local receptive fields, shared weights, and pooling characteristics.

i. Local Receptive Fields

One of the basic concepts of Deep CNNs is Local Receptive Fields(LRF)[18], of a unit in certain layer of a network connect each neuron to only a local region of the input volume. The ConvNets use LRF where convolutional layer input pixel connected to the next hidden layer neurons. In the input layer it is represented as a height with width arrangement unites as shows figure(2.4),which is different from representation of standard neural networks(it have flatten vector).

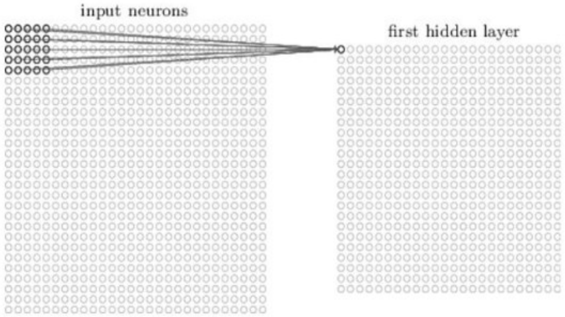


Figure 2-4: Local Receptive fields[41]

In addition, Convolutional layers input pixels are connected to the next layer of hidden unit through the convolution process, which applies only over a small chunk of the inputs map. This means that, instead of making fully connections between all the neurons, convolutional layers only make connections in a small, localized region of input image. Such localized architecture ensures that the kernel produce the strongest response to a spatially local input pattern. Although these filters apply locally, using many hidden layers leads to non-linear filters that become increasing global, as they respond to larger regions of previous layers. Therefore, Convolutional Neural Networks are able to create at lower layers simple representations of small parts of the input, whereas at upper layers generate more abstract descriptors of larger areas.

ii. **Pooling**

Another characteristic that distinguish main difference of CNNs with respect to standard neural networks is the existence of pooling layers [21]. The ConvNets in the pooling to reducing the dimension to simplify the information at the output of the convolutional layers without hurt the feature maps (the detail description on section 2.5.1).

iii. Shared Filters and Feature Maps

In CNNs, a unique set of weights and bias (also called kernel of filter) is shared across the entire input field creating a feature map in each hidden layer neurons [8]. This means that all the neurons in a given convolutional layer will respond in the same way to the same feature over the previous layer. This is done because it is highly likely that a learned feature would be useful at other places of the image. Sharing weight enables the network to learn a single kernel for an object no matter where the object is placed everywhere in the given image.

The main consequence of sharing these kernels, is that the feature can be detected regardless its position in the input field, obtaining the translation invariance property that convolutional neural networks have and that hand-crafted descriptors struggled to get. Still, the described structure would not be practical for doing robust image recognition, as it can only detect just a single kind of feature. More than one feature map is needed therefore, enabling to learn more kernels and obtaining a much more robust network.

Another important consequence, is that CNNs are enable to reduce dramatically the number parameters learned at each layer. For example, a 100*100 pixels small input image and setting a local receptive field of 3x3, the output of a convolutional hidden layer would have 98x98 units. These units result from moving 97 times the reception field over the whole input image. Nevertheless, in this case the CNN only need to learn one filter (vector of weights and a bias) for generating a full feature map, so that 3*3+1=10 parameters. Given the case that we choose to have for example 20 of those feature maps in the first CNN hidden level, we would need to learn 10*20=200 parameters. In order to compare, a classical neural network first layer with 20 neurons would require for the same example 100*100*20=200,000 weights plus another 20 biases, thus would have 200,020 parameters. This substantial reduction of weights of the CNN, leaves space to create bigger and deeper networks with several layers.

2.5.1. Basic Convolutional Neural Network structure

The ConvNets basic architecture must have convolution layer, pooling layer and fully connected layer. And also, its core structure of alternating convolutional layers with max-pooling.

i. The Convolution Layer

The convolution layer makes use a sets of learnable kernels, which used to detect the presence of specific features or patterns present in the input image. In this layer, the features extracted from an image by convolved the kernel across the width and height of input images, and dot product is performed to give the feature/activation map. The learned feature are a consequence of mathematical operation between each element from the input image and the filter matrix. The output feature map of a convolution layer is calculated as:

$$a_{x,y} = \sigma\left(b + \sum_{f_1}\sum_{f_2} w_{f_1,f_2} \cdot a_{j+f_1,k+f_2}\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.4)$$

Where

- $a_{x,y}$ is the output unit; b is the shared value given to the filter bias;
- f_1 and f_2 goes from 0 respectively to the height and width of the filter used,
- $w_{f1,f2}$ are the learned values for the Kernel weights;
- k,j is the top-left corner of the local receptive field in the input map;
- $a_{j+f1,k+f2}$ are the input values under the filter patch (input activation $a_{x,y}$); and
- Finally σ is the neural activation function providing the non-linearity.

Additionally, an activation function usually used in this layer. The activation function is a non-linear differential mathematical formula to compute the output of the specified neurons using the weighted input signals. The purpose of the activation function is to introduce non-linearity into the network[19,30], so that nonlinear models can be learned. It means, basically decide whether a neuron should be activated or not. And, whether the information that the neuron is receiving is relevant for the given information or should be ignore it. It final values are bounded between two values. Commonly used activation functions are hyperbolic tangent (tanh) function, the sigmoid function and ReLu function.

In our works, we use ReLU activation function because ReLUs units require only comparison, addition and multiplication operations, so that it allows for faster and effective training of deep neural architecture on large and complex datasets. Another important advantage is that ReLU units are able to replace all negative pixel values inside the feature map per zero. And, ReLU activation function improves the performance of CNNs for object detection and recognition.

ii. The pooling Layer

Pooling layer is a non-trainable layer which takes region of input and applies a simple function like max, mean, sum to each of these regions independently. The responsibility of this layer is to progressively reducing the special size of convoluted feature maps. Hence, reducing the amount of parameters and computational power required to process the data in the network. Moreover, the pooling operation provides to extracting dominant features in a form of translation invariance. Intuition says that once a feature has been found, its exact location is not as important as its rough location relative to other features, so in this way we are allowed to keep only the most activated unit.

The most common function used are max function and mean function for max-pooling and average pooling respectively. In max-pooling, each pooling unit take group of input and output the maximum value of each one of these groups(a region of the image covered by the kernel). Whereas, average pooling returns the average of all the values of region of image covered by the kernel.

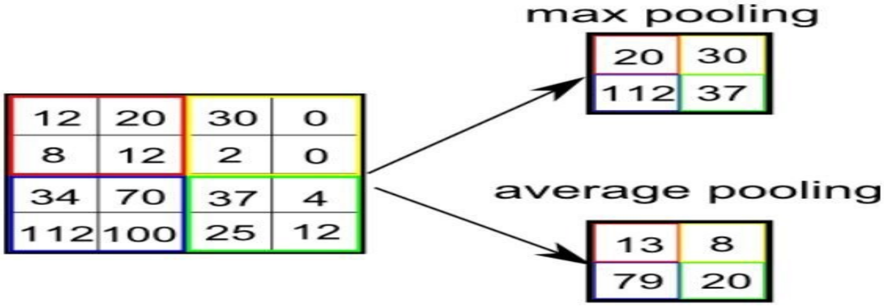


Figure 2-5: the right top max pooling and the right bottom mean average pooling.

Another pooling technique is Fractional pooling [21], which is having quite good results by smoothing the rapid reduction special size of max-pooling layers allowing to a special lxl region, where l can be non-integer value.

iii. The upper layers

ConvNets perform the high-level reasoning in the upper layers via fully connected layers and coming right before the output layer. The units in a fully connected layer are connected to all the feature maps in convolution and pooling layers between a fully connected layer neurons. Hence, in this layer perform learning non-linear combination of the high level features as represented by the output of the convolution and pooling layer. Therefore, their activation can be easily computed with a matrix multiplication followed by a bias offset.

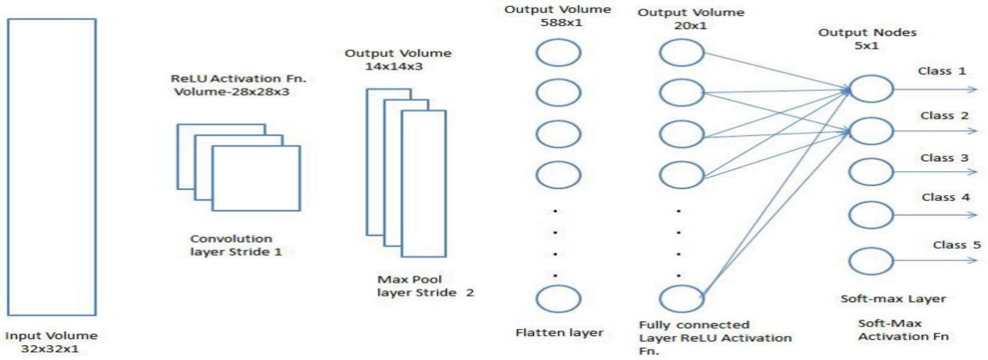


Figure 2-6: the left input layer, and follows 2 convolutional layers, the last three fully connected layer[17]

The fully connected layers also include the loss layer, to specify how the network training penalizes the deviation between the predicted values and the true ones. The For depending on the applications softmax, which predicting n independent classes by probability values in the range of [0,1] or cross-entropy loss, or Euclidean loss which is used for performing logistic regression to real-value labels in the $[-\infty, \infty]$ interval loss function uses. The flattened outputs feed-forward neural networks and back-propagation applied to every iteration of training.

Over a series of epochs, the model is able to distinguish between dominant and certain low-level features in image and classify them. Since, this layer to use the output features from the convolution and pooling layer and classify the instances/categories in the image based on the training datasets.

2.5.2. Deep Convolutional Neural Network models

ConvNets ability for building multiple layers of abstracted features seems to be fundamental to sense and understand our world, which is making CNNs a very hot research topic. And, many architectures are recently been proposed for different computer vision problems such as feature extraction, object detection and recognition. However, some of them are only slightly modifications or tweaked versions of other well-known networks.

Nevertheless, we present here a brief summary of the most famous CNNs architectures that have served as base for many other authors, along with its main characteristics and achievements:

i. LeNet-5

The first earliest and famous Convolutional Neural Networks architecture receives the name of LeNet-5[19], after one of its authors, Yann LeCun. This model is the reference network in the field. Basically, LeNet-5 consist on an input layer followed by 5 hidden layers and two fully connected layers at the performing the final classification. The hidden layers are a sequence of a convolution layer (C1) followed by a max-pooling sub-sampling layer (S2). This structure is repeated one more time (C3 and S4). The last hidden layer (C5) corresponds to another convolution layer that reduce the 3 dimensions of the whole set of feature maps of S4 to a two-dimensional vector enabling to link to the fully connected layers (F6 and output soft-max layers).

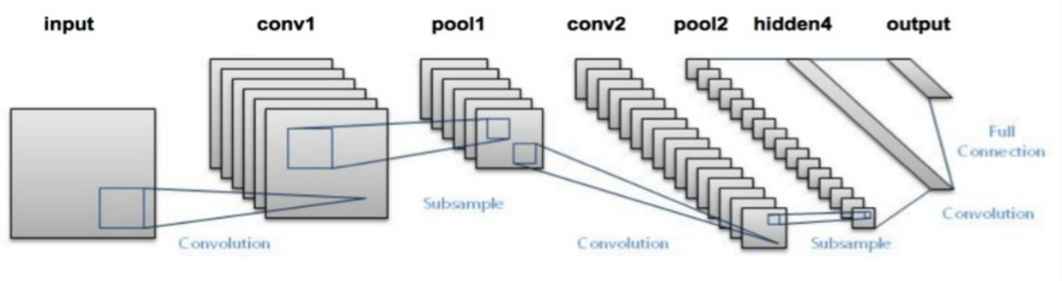


Figure 2-7: LeNet-5 architecture[19]

It was the first successful application of convolutional networks to real computer vision problems, mainly to recognize handwriting digits.

ii. AlexNet

The AlexNet, named Alex Krizhevsky[23] represented the real break-through of convolutional networks in computer vision. This neural network actually has a basic architecture similar to the LeNet[19], but is bigger and deeper. This architecture was one of the first deep networks and contains about 650K neurons, 60 million parameters and 630 million connections.

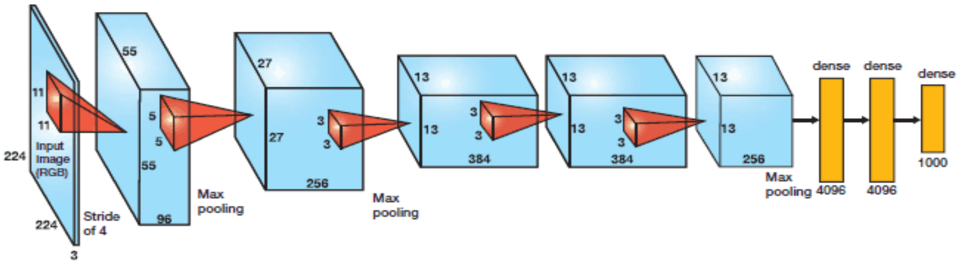


Figure 2-8: AlexNet Architecture[23]

The architecture that uses convolutional layers stacked on top of each other's, a single convolution layer immediately followed by a pooling layer. It contains 7 hidden layers, the 5 being convolutional (some with max-pooling) and the next 2 layers being fully connected. In fully connected layer, it has a 1000-unit soft-max output layer as output classification for the 1000 image classes.

iii. VGGNet

The VGGNet [25] is deep ConvNets for object recognition developed and trained by Oxford's renowned Visual Geometry Group (VGG). It makes the improvement over AlexNet, it contains a total of sixteen convolution layers and fully connected layers with eleven filter size in the first convolution layer and five for the second convolution layer. Simplicity is provided by applies from the beginning to the end blocks of two convolutions followed by one max-pooling layer.

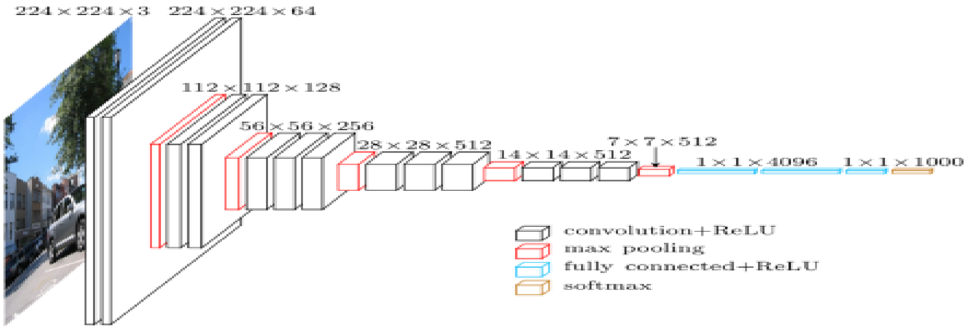


Figure 2-9: VGG Network[25]

The VGGNets to be one of the most preferred chose in the community when extracting CNN features from images. However, a drawback of VGGNet is that it's more expensive to evaluate and use a lot more memory and parameters, as it has around 140 Million parameters.

iv. Inception Network

Inception network[24] is an efficient deep network architecture for computer vision. It is basically a ConvNets which have 27 layers, and combines convolutional plus pooling layers with their inception modules. The inception module is a combination of 1x1, 3x3 and 5x5 convolution layer with their output filter banks concatenated into a single output vector forming the input of the next stage. Its main contribution is the development of the inception modules, inspired by the network-in-network[27].

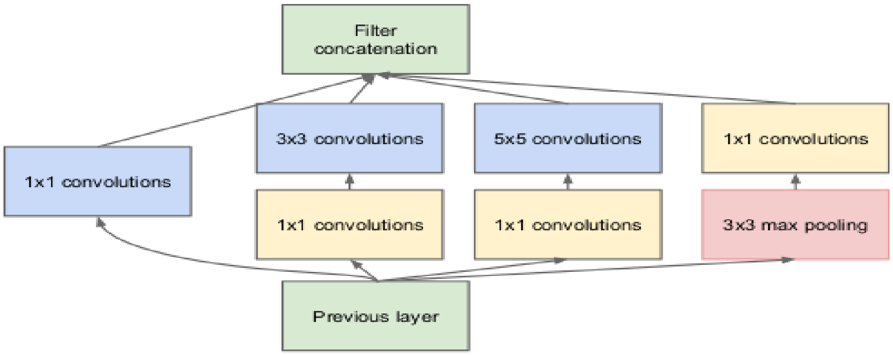


Figure 2-10: Block diagram of Inception layer[24]

Inception Network to replace the fully connected layer at the end with a simple average pooling, which average the channel value of feature map after the last convolutional layer. This approach reduced the number of parameters in the network to 4 million parameters and allowing to have deeper networks.

v. Residual Network

In general, in a deep convolutional neural network, several layers are stacked and are trained to the task at hand. The network learns several low/mid/high level features at the end of its layers. In residual learning[28], instead of trying to learn some features, we try to learn some residual. Residual can be simply understood as subtraction of feature learned from input of that layer. ResNet does this using shortcut connections (directly connecting input of nth layer to some (n+x)th layer. It has proved that training this form of networks is easier than training simple deep convolutional neural networks and also the problem of degrading accuracy is resolved[28].

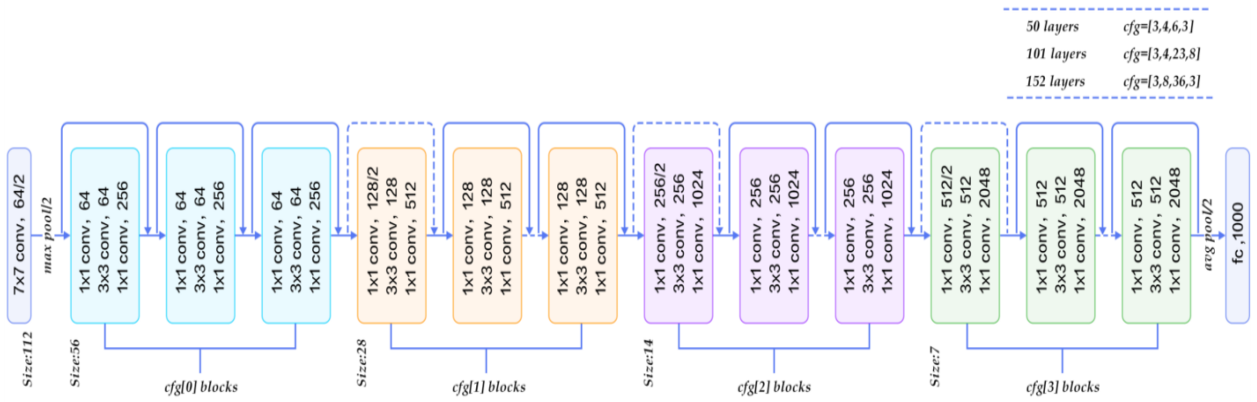


Figure 2-11: Residual network[28]

2.5.3. Convolution Neural Network Object Detection and Recognition Algorithms

A typical ConvNets is to classify categories of object from input data. But object detection is more challenging problem in states of the art[37,29,40,41,42], and require detecting several objects/instances in the same image that have different aspect ratio and scales. Some of the datasets, challenges and competitions have contributed to the development of algorithms and other CNNs architectures able to solve the object detection problems.

In object detection literatures[36,37,39,40], there are different algorithm to effectively detect an objects from input data. Different algorithms are perform in different domains and datasets. Since, one algorithm in a specific domain might accurate and in another domain might be poor performed. Thus, selecting an algorithm is important to create models to enhance the performance of the detectors. We divide mainly into three main categories i.e. sliding windows based, regional based and grid level based object detection algorithm approaches.

2.5.3.1. Sliding window approaches

Initial approaches use well known multi-scale and sliding window methods in combination with CNNs extracted features and a final classifier. However, Sermaner et al showed in their Overfeat work[36] that end-to-end train CNN architectures designed to produce an integrated approach to object detection, recognition and localization. They explored CNN special characteristics of location invariance and weight sharing to develop an inherently efficient sliding window approach. They also introduce a novel method learn to predict object boundaries to create bounding boxes, all in the same CNN architecture called Overfeat.

2.5.3.2. Regional approaches

In object detection, there is a set of idea called Region proposals that's been very influential in computer vision states art.

A. R-CNN

The algorithm was proposed by in [37], as what is called region-based CNNs. It starts with a pre-processing region proposal step that outputs 2000 proposals as shown figure(2-12). Next, it uses a pre-trained AlexNet classification network to extract a 4096 feature vector for each of the regions. Finally, it classifies each region with a category-specific linear Support Vector Machine and with the results they fine tune the CNN end-to-end for detection.

As shown below in algorithm (1), the selective search algorithm proposed different number of regions from input data.

Algorithm 1: selective search[31]

```
Selective Search algorithm:
Generate initial sub-segmentation, we generate many candidate
regions
Use greedy algorithm to recursively combine similar regions into
larger ones
Use the generated regions to produce the final candidate region
proposals
```

Then for region with ConvNets used the following algorithms

Algorithm 2: Region with Convolutional Neural Networks algorithm[37]

```
First takes input data like images
Then we gets the ROI using selective search algorithms
All these regions are then reshaped as per the input of the
ConvNets and each region is passed to the ConvNets
Then feature are extracted by CNN from each regions and apply
different SVM for classified the regions into different classes
Finally, a bounding box regression is used to predict the
bounding box for each identified regions
```

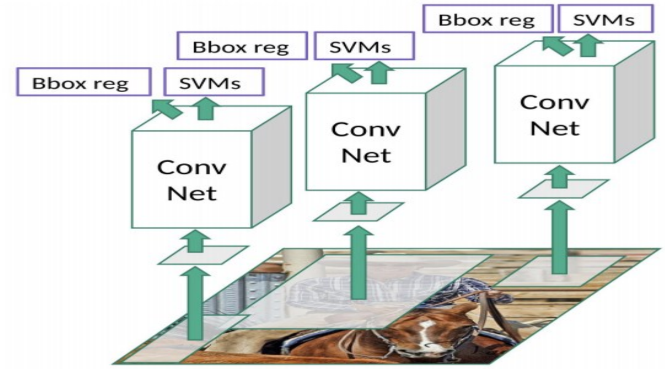


Figure 12: Region with Convolutional Networks[37]

In addition to predicting the presence of an object within the region proposals, the algorithm also predicts four values which are offset values to increase the precision of the bounding box.

B. Fast R-CNN

Further works on Object detection with CNNs has focus mainly on reducing the expensive computations of R-CNNs, which has been achieved successfully by sharing the convolutions across proposals [39]. The fast R-CNN work[39] speed up R-CNNs by moving the proposals warping and its interpretations after the last convolutional layer. The algorithm used selective search for candidate region proposals as describes algorithm (1). In this algorithm, feed the input image to the CNN to generate a convolutional feature map as shown figures(2-13). Then, from the convolutional feature map, identify the region of proposals and warp them into squares and by using a RoI pooling layer, which is used for reshape different sized feature maps into a fixed size. Then finally, the fixed size features are fed into a fully connected layer. The algorithm used softmax to classify each instance of objects and use regressor to draw bounding box around each objects in input data from the RoI feature vectors.

The Fast R-CNN algorithm performs as follows:

Algorithm 3: Fast R-CNN algorithm[39]

```
Takes an input data like images
The input data passed into a ConvNets, which is returns the RoI
Apply RoI pooling, which makes all RoIs have the same size
```

```
Finally, passed onto a fully connected layers, which classifies into each classes by using softmax and draw bounding box by linear regression simultaneously.
```

The image blows show the fast R-CNN algorithm from the input data to softmax classifier and bounding box regressor.

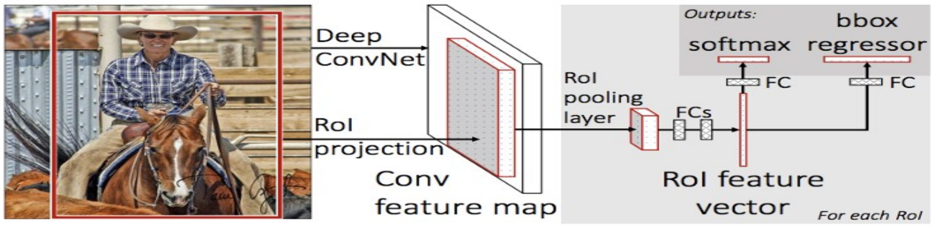


Figure 2-13: Fast R-CNN[39]

C. Faster R-CNN

Both RCNN[37] and Fast RCNN[39] uses selective search to find out the region proposals. Selective search is a slow and time-consuming process affecting the performance of the network. Since, those algorithm cannot use for real time object detection problems. Hence, for real time detector use faster RCNN algorithm[40]. The faster RCNN, uses the convolutional neural networks to propose regions from the input data as described in the algorithms.

The following algorithms describes how the Faster RCNN works:

Algorithm 4: Faster R-CNN Algorithms[40]

```
Takes image data as input and passed into ConvNets, which returns feature maps
Apply RPN on the feature maps, then returns the object proposal along with their abjectness score
Apply RoI pooling to make the proposals to the same size
Finally, the proposals are passed to a fully connected layers, which have a softmax layer and a linear regression layer for classifies into each categories and draw bounding box.
```

The input data fed into the ConvNets and produced feature maps as shown below figure(2-14).

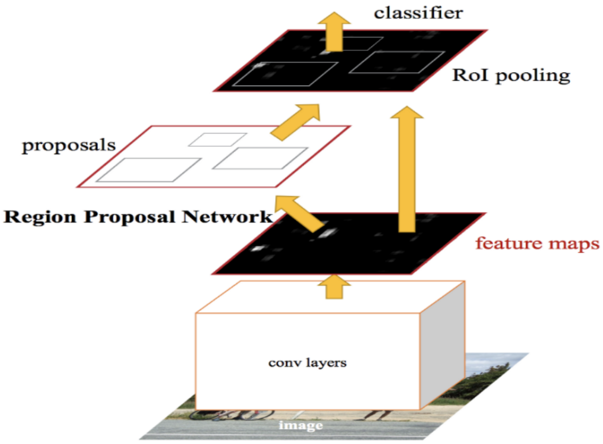


Figure 2-14: Faster R-CNN[40]

Next, use regional proposal networks used to find up to a predefined number of regions (bounding box, which may contain an objects. Then the predicted region proposals are reshaped using a RoI pooling layer. And finally, the reshaped tensors are used to classify the image within the proposed region and predict the offset values for the bounding boxes.

2.5.3.3. Gird label based

A. You only Look once

The algorithm[42] take an image and split it into an SxS grid as shown fig, within each of the grid take m bounding boxes. For each of the bounding box, the network outputs a class probability and offset values for the bounding box. The bounding boxes having the class probability above a threshold value is selected and used to locate the object within the image.

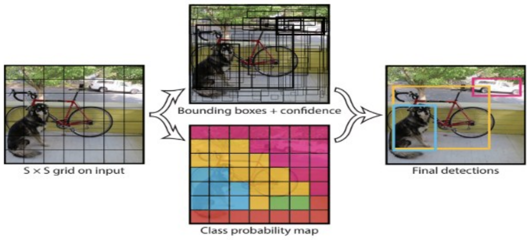


Figure 2-15: you only Look once(YoLo)[42]

Algorithm 5: You only Look once[42]

```
Takes the input images and then divides into mxm grids
Apply classification and regression on each grids
Apply anchor boxes to identified different objects in the same grids
```

2.5.4. ConvNets hyper-parameters

In ConvNets training hyper parameters are play a significant role on its performance. So some consideration will be taken training ConvNets algorithm. The main parameters are training epoch, batch size, number of hidden layers, number of computing units, learning rate and others. So to implement this thesis preliminary experiments are done to select the appropriate parameter values and some of them are taken as their default value on the ConvNets.

i. Batch size

It is the number of training samples that the training will be used to make one update to the model parameters. The training sample to calculate the gradient for every single update. The typical value of batch size is depending on the number of layer of the model and training data. Literature recommend values on 2^n where, n=any positive integer value but the batch size should be less than or equal to the training size. In our thesis experiment used 64 batch size.

ii. Training epoch

It is the total training steps of the learning algorithm using the entire training data. To train the CNN to evaluate do the forward not only one sample but with a lot of them. It is the total number of iteration in the one step of the whole training data and update its weight and bias values and finally it calculate the reconstruction error statistics. Its value extends in the range of positive integer and the specific value for this thesis will be determine using preliminary experiment in Chapter Five.

iii. **Number of hidden layers**

The number of hidden layers affects the fitting degree of data directly. In theory the more layers of network there are the more complicated the network structure is, making the network express data precisely and ultimately obtaining a higher accuracy [37, 48]. However only increasing the number of hidden layers may lead to difficulty in neural network training, greatly extend the learning time and decrease the accuracy. The number of hidden layers is studied in this thesis. Setting the number of hidden layers as 2, 3, and 4 respectively (excluding the input and output layers), the accuracy is calculated and we used 3 as the number of hidden layers in this thesis by making preliminary experiment in Chapter Five.

iv. Number of computing units of hidden layer

Hidden layer is feature extraction part of CNN. For such purpose it uses a number of computing units. Because the number of units of hidden layers L difficult to ascertain and the selection method is very subjective, there is no convincing study on it [46]. Its range is extended in positive integer. So, to decide the specific number of units in each hidden layer preliminary experiments should be taken.

v. Momentum

Momentum simply adds a fraction of the previous weight update to the current one [47]. The momentum parameter is used to prevent the system from converging to a local minima or saddle point. A high momentum parameter can also help to increase the speed of convergence of the system. However, setting this parameter too high create a risk of overshooting and the minimum which can cause the system to become unstable. A momentum coefficient that is too low cannot avoid local minima, and can also slowdown the training of the system. The value of momentum range in [0, 1]. In this thesis we used momentum value of 0.9 as recommended by many researchers.

vi. Learning rate

Learning rate directly affects the stability and convergence of the network. It is the controlling parameter for weight and bias update values. If the learning rate is too high, the reconstruction error may grow dramatically, and weight may change too much and skip optimal solution. This mean the weight change in each iteration is large then finally the weight will explode and the system may overfit earlier than it was expected. If the learning rate is too low, the reconstruction error may be significantly reduced. The network will stay near local extreme for long time, greatly extending the convergence rate. For this thesis it will be determined through simple experiments.

vii. **Weight decay**

Through training epoch weight decay shrinks the weights towards smaller values and this tends to control overfitting of the model. There are two version of weight decay, the first one is absolute value decay (L1) that push a lot of the weights to be exactly zero while allowing some to grow large and the other one is square value (L2) which tends to derive all the weights to smaller values. L2 is used for this thesis work. The recommended value of L2 by [48] ranges from 0.01 to 0.00001 and 0.00001 is used in this thesis.

viii. **Weight and bias initialization**

Weight initialization has been widely recognized as one of the most effective approaches in speeding up the training of machine learning. In fact, it influences not only the speed of convergence, but also the probability of convergence and generalization [47,48]. Using too small or too large values could speed up the learning, but at same time, it may end up performing worse. In addition, the number of iterations of the training algorithm and the convergence time would vary depending on the initialized values. In Convolutional Neural Network weight is used to tune the connection between computing units and between two consecutive layers. Then the algorithm learns the pattern from the training dataset by adjusting its weight parameter through its learning epoch. The bias is used as initialization signal for computing units of their respective layers. The bias and weight in ConvNets should be trainable variable, their variable allowed to change during training.

2.6. Back-propagation training for Feed forward ConvNets

Deep ConvNets are effective tools in the field of object classification using training and testing data to build a model. It derives the power for classification and recognition due to their massively parallel structure and also ability to learn from experience. It was proposed based on the working principle of natural neural cell and their connection between them to form nerves system [19,30,47]. Neural cell is composed of nucleus where electro-chemical reaction takes please, dendrites that used to connect to other neural cell axon for synapses input, and axon that expend to connect other neuron cell dendrites as output of synapses. The strength of the axon is a matter of the level of signal received by the next neurons for electro-chemical reaction. Then in this way number of them connects to form nerves system.

In ConvNets, the filters is flipped and slide across the input feature map in equal and finite strides to compute convolutional operation. Since, at each location the product between kernel and feature map element is computed and sum up to the result. Hence, this procedure is repeated using different kernels to form as many feature map outputs.

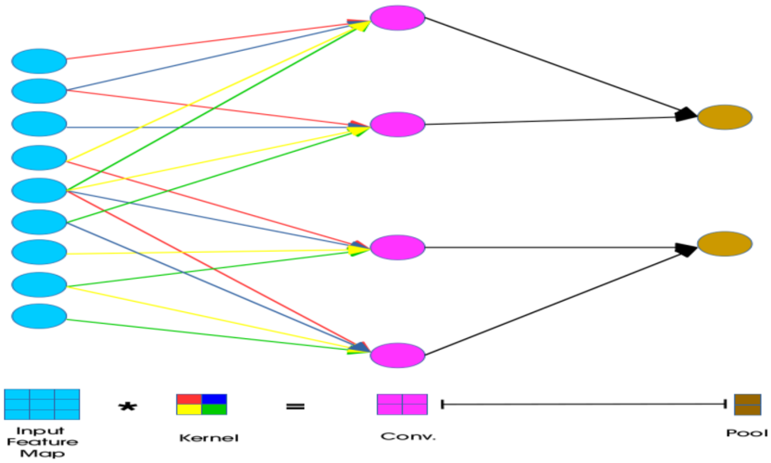


Figure 2-16: Feed Forward in CNN

Units in convolutional layer as shown fig have receptive fields of size 4 in the input feature map and are thus only connected to 4 adjacent neurons in the input layer. This is the idea of sparse connectivity in CNNs where there exists local connectivity pattern between neurons in adjacent layers.

Back propagation algorithm is the method of training multi-layer feed forward ConvNets using the gradient optimizations method [19,30]. The basic BPNN consists of three steps. The input pattern is given to the input layer of the network. Then these inputs are propagated to through the network until they reach the output units. This forward pass produces the actual or predicted output pattern. BPNN is a supervised learning algorithm where the desired outputs are given as part of the training vector.

Allows the information to go back from the cost backward through the network in order to compute the gradient. Therefore, loop over the nodes starting at the final node in reverse topological order to compute the derivative of the final node output with respect to each edge's node tail as shown figure(2-17)

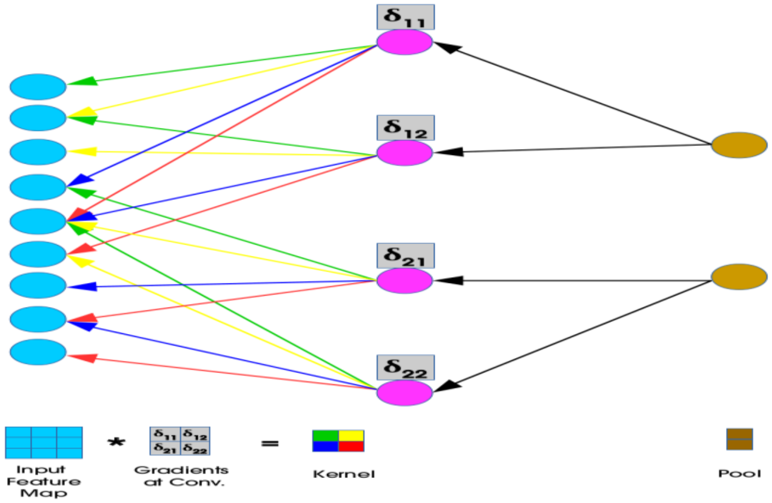


Figure 2-17: Architecture of back-propagation networks

Feed forward back propagation has two phases to train the given neural network. The first phase is feed forwarding of the input signal through each layer until output layer. The second phase is back propagating the error between the desired and the resulting output back to each layer until before reaching the input layer. The output of any neuron in respective layer computed using equation (2.5) in combination with the activation function.

$$\boldsymbol{x_j = \delta\left(\sum_{i:\perp}^{I} x_i w_{j_i} + b_i\right)} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots .. (2.5)$$

Where:

- δ is the activation function.
- xi is the input at layer i
- wij the weight of connection from neuron i to j

The result of one layer will feed to the next layer until the output layer and when it reaches the output layer it will be end of feed forward computation and it became the beginning of the next back-propagation process. The basic element of this algorithm is the energy function that defined as a quadratic sum of the difference between the actual output signals and the desired values as defined by equation (2.6).

$$E = \frac{1}{2}\sum_{i=1}^{p}\sum_{k=1}^{m}\left(y_k^{(i)} - d_k^{(i)}\right)^2 \quad \text{……………………………………………………..(2.6)}$$

Where:

- E is the energy function which is the square of the error between the output and the desired signal.
- P is number of training vector.
- M is number of classes or output layer neurons.

So, to reduce the above energy function value through training, BPNN use the following steps.

1. Apply the actual input signal vector X.

(i) Calculate the output signal under each hidden and output layer using equation (2.5)

(ii) Calculate the gradient of activation function in each neuron of each layer using the derivative of the activation function using equation (2.6).

2. Create the back propagation network by reversing the direction of signal transmission.

(i) Replace the activation function by its derivative

(ii) The input vector at former output layer and the current input layer is the error between the actual and the desired value.

(iii) The weight modifications proceed on the bases of the result in one feed forward and backward propagation using equation (2.7).

3. Repeat one and two for all training samples as much time until the stopping criteria of the algorithm is reached.

The weight modification in each training steps can be computed using the equation (2.7).

$w\ ij\ (t + 1) = w\ ij\ (t) - \varepsilon\nabla\ E(w)$---(2.7)

Where:

- w ij is the weight of connection from neuron i to j.
- E(w) is the gradient of the energy function.
- ε is the training coefficient.
- t + 1 is the next training time and t is the current training time.

The above formula states that the current weight update is the difference between the previous weight and training coefficient multiplied by the gradient of the energy function. The previous weight is obviously known but the gradient of the energy function obtained by differentiating the energy function with respect to respective weight of neurons as in equation (2.8).

$$\boldsymbol{\nabla E(w) = (o - t) * x_{ji}\frac{df(y)}{dy}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..\ (2.8)$$

Where,

- o − t is the error of the current training step.
- x ji is the signal of the current connection which is under consideration to modify its weight.
- df(y)/dy is the derivative of the activation function.

The final formula for weight update of back propagation algorithm is between two neurons of consecutive layer is computed using equation (2.9).

$$\boldsymbol{w_{ji}(t+1) = w_{ji}(t) - \varepsilon(o - y)^{*} x_{ji}\frac{df(y)}{dy} + \partial\Delta w_{ji}(t)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(2.9)$$

$$\boldsymbol{\Delta w_{j_i}(t) = w_{ji}(t) - w_{ji}(t-1)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(2.10)$$

Where,

- t is current execution time

- t + 1 is next execution time and t − 1 previous execution time
- o is desired output, y is current system output, *df(y)/dy* is is the gradient of the activation function and ∂ is momentum co-efficient.

2.7. Performance matrices

After designing and implementing the system there should be some performance evaluation mechanisms. They are the means of measuring the performance of the system based on the output of the system. There are a number of classification task evaluation mechanisms as stated by literature [50, 52, 53], Based on the number of class in the system output classification system can be grouped in to two groups. Binary classifier with two classes and multi-class classifier with more than two classes. This proposed object detection system is multi-class classifier because the dataset contains more than two classes (pedestrian, car, cyclist, truck, etc.).

For object detector used two main performance matrices of the model. The first, by calculating the mean average Precision of each class of object that the object detector is looking for over the entire datasets. And the second, Intersection over Union is calculating the intersection between the actual bounding box predictions with the ground truth bounding box over the entire datasets.

CHAPTER THREE

RELATED WORKS

3.1. Introduction

In this chapter, we discussed various approaches towards object detection for autonomous system. Several system have been proposed for object detection to improve the performance. Researcher select different tools and techniques depending on the object detection problem. This problem various from dataset to datasets, algorithm to algorithm and so on. Thus, we have to analyze and discuss different studies from different aspects of their work, such as methods datasets, limitations, algorithms, their experiment setup and result. We have mainly discussed the classical and deep learning object detection approaches.

3.2. Classical object detection approaches

The author in [33], deals about improving the performance of vehicle detection with single camera and proposed a Histogram Oriented Gradient (HOG) descriptor for feature extractor and Support Vector Machine (SVM) for classifier to forward vehicle detection. The general approach of their study was on detecting underneath vehicle shadow and background of the input image and they proved to perform well in different lighting conditions. They resized the input image to 24 by 24 gray-scale images. They were different stages as follow. Firstly, the underneath vehicle shadow was segmented by using histogram analysis method. Then the initial candidates were generated by horizontal and vertical edge features, and these initial candidates were further verified by support vector machine methods. The dataset used here was obtained from road images collected from Beijing with the resolution 720 by 480. They used 3600 training sample which contains 1207 positive sample and 2393 negative samples. Finally, they achieved 96.87% true positive rate with 2.77% false positive rate under normal lighting condition.

The researcher proposed in[35], perform a combination of Scale Invariant Transform Features(SIFT) and bag-of-words to detect and represent valuable features. Their proposed detection system consist of the following step, first they used to extract the background around the object in images then the image is segmented then the SIFT descriptor used to extract features. They used on the precision of hand engineered features that are extracted using SITF feature descriptor. They used Support Vector machine for classify the vehicle object in training and testing case. The dataset used here was obtained from publicly available NTOV-MMR frontal vehicle image dataset. The used 2,748 images for training and 3,274 images for tasting. In total they implemented 6055 image datasets. The author perform a feature matching to evaluate the result. Finally, they got a result of 89% accuracy. But in a deep learning algorithm, all the features are extracted by the model rather than hand crafted feature extraction. They extracted features by using hand engineering. So, for real time system features are extracted automatically by the model.

3.3. Deep learning object detection approaches

The research in [36], introduced the novel deep learning approach to localize the bounding box for each objects in the image. Their proposed system explored object localization, classification and detection computer vision task. And also, they released OverFeat feature extractor model to automatic feature extractor to provide powerful feature for computer vision research problems. They used AlexNet Network architecture as a base network for classifications computer vision task with some modifications like none-overlapping pooling, small stride in the first two layer feature maps and omitting contrast normalization. In their model, they replace the fully connected layer of classification network by regression network and train it to predict the bounding box around the objects for localization tasks and it trained four bounding coordinates at each spatial location and scale. Then, they combine the regression predication together along with the classification result at each location for object detection task. In their model, the classification and localization networks are trained simultaneously. Then, their result are merged to get final prediction. The datasets used here from ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2013 and obtained 50.23% mAP and it is competitive result for object detection and localization tasks.

The Researchers proposed in[37], proposed an object detection and semantic segmentation using region proposal Convolutional Neural Networks(R-CNN). They used selective search algorithm to proposed possible region of interest and standard CNN to classify the class of each objects in the input image/video. In their work shows CNN is achieved better result than low level feature descriptor methods like HOG[33] and SIFT[35]. Their methodologies for object detection and semantic segmentation tasks as follows. Firstly, the R-CNN proposed 20K regions for each input data by using selective search algorithm. Then, the 4096 dimensional features are extracted from each proposed regions using AlexNet Convolutional Neural Network architectures. Finally, each features (i.e 4096) are classified using SVM classifier where each SVM has been trained for a specific classes. They have done the task based on ILSVRC 2013 detection datasets and they obtained 70.23 % mAP results. So, their work(R-CNN) scored better result than OverFeat[36]. However, they perform ConvNets for each of the regions and use different SVM for each of object categories to classify instance of objects in input data. Thus, the algorithm takes huge amount of times to train the model to classify 2000 region proposals per input data.

As the author proposed in[38], disused about object detection for autonomous vehicles from street images. He have done predicting bounding box for an image without the need of expensive preprocessing or expensive deep evaluation. So, he mainly focuses to predict the bounding box for each object from the input data of object detection problem. They used SimpleNet five and three layers Convolutional Neural network algorithm to classify the objects and predict the bounding box of an objects. He have done the task based on publicly available KITTI object detection benchmark datasets. The author follow as, first take the input data from KITTI datasets, then he perform pixel by pixel convolution to extract a particular pattern from the input images and applied ReLu activation function to a feature map, next to that applied max pooling for dimension reduction purpose in the feature map and finally a dense/fully connected layer used for classify an object with back-propagation algorithm. In the result he made a prediction at a rate of 11 frames per second(0.09 seconds for each images) and achieves a mAP 12.83% on validation set. But this result was minimum according to deep CNN algorithm for real time object detection. And that is why, he did SimpleNet CNN algorithm with five layers networks is not extracted all low-level, meddle level and high-level features of the input data to learn the model.

The researchers in[39], proposed a regional-based convolutional Neural Networks for real-time object detection and recognition. They used a selective search region proposal algorithm to extract features of objects in the input images. Their methodologies for real-time object detection and recognition as follows. Firstly, the network takes as input the entire images and a set of object proposal. The network process the whole image with several convolution layers and pooling layers to produce convolution feature map. Then selective search regional proposal algorithm, each object proposal a region of Interest(RoI) pooling layer extract a fixed length feature vector from the feature maps. Finally, each feature vector are fed into a sequence of fully connected layer for softmax probability estimation over K number of object classes and four real valued bounding box for each K object classes. They have done the task based on MS COCO 2012 detection datasets. They trained on 80,000 image training set for 20K iterations and they obtained 65.7 mAP results. But, the selective search algorithm for region proposal is a fixed algorithm and doesn't learn from experience. So, the selective search might proposed bad regions to the model.

The authors in[40], proposed a real-time object detection networks using region proposal algorithm to hypothesized object localization. They introduced a novel Region Proposal Network (RPN), which shares full image convolutional features with the detection networks. The RPN is fully convolutional networks that predict abjectness score and the bounding box value for each object classes. In their work the input image are fed into VGG16 Network architecture and produced feature maps with their abjectness score. Then the RPN networks are applied to feature maps to predict the bounding box for each object classes and predict the abjectness score. They used MS COCO object detection datasets and they used 80K images for training set, 40K images for validation set, and 20K for test-dev set. They obtained 5 frames per second and 73.45% mAP accuracy to their result. However, they used fully connected networks in the detection, so the number of parameters and a number of connection in this networks is high. Since, the networks takes more computational resources and cannot extract more features to enhance the performance of the classification and regression in the input images.

The researcher in[41], they designed a regional-based object detection networks to recognize object. Their proposed networks have two stages. In the first stage, they used CNN architecture region network with convolutional layers that activates semantically meaningful regions for detection proposal to localize objects. These activated regions are used as input to the Object CNN to extract deep features. They exploit the features from the final convolutional layers of the ConvNets to active the receptive fields to obtain the region when an object might be appear. Then, these proposed regions are used as input to Object CNN networks to extract features. Finally, they train a set of class- specific binary classifiers to predict the object labels from extracted features. In their work, they used PASCSL, SUN, MITGA and MSRC detection datasets.

The researchers in[42], proposed the most recent real-time object detection and recognition approaches. They used the whole images instead of using the separate region proposal in their network architecture to predict the bounding box confidence and class probability. The network architecture is based on GoogleLeNet where Inception modules are replaced by reduction layers. In their proposed network model use twenty pre-trained convolutional layers and added another four convolutional with two fully connected layers to their network model. In their network, each input image is divided into a grid cell. And also, a single network predict bounding box and class probabilities directly from each grid in full images in evaluation. Since the whole detection and recognition pipeline is a single network, it can be optimize end-to-end directly on detection performance. In here used ImageNet detection datasets with 1000 object categories and they obtained 63.4% mAP and 45 frames per second in the result.

The researcher in[43], develop single shot object detector method for detect an object in images that does not re-sample pixels features for bounding box. Their proposed system used ConvNets that produced a fixed size bounding box when the object appear in the box, followed by a non-maximum suppression step to produce the final detection. In their work used data augmentation to make the model more robust to various input object size and shapes. They have done on ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and they obtained 75.1% mAP as result.

The author in[44], proposed a new model for real-time multi feature object detection by using ConvNets free from region proposals. In their work to improve the detection of small objects in an image. The networks takes the whole image as input and perform the ConvNets instead of taking the region proposal methods for predict the bounding box around each instances/objects. They designed the model as follows: firstly the image are resized into 352x352 square dimension before feed into the networks. After resized, the input images are divided into SxS grid cell, which have a responsibility to fall each object to predict bounding box with confidence scores. Then, they used five convolution layers to extract features from each grid cells. And in each grid cell, if confidence score is 1 it contains an objects and 0 for no objects/background. And, when more than one object appear in one grid cell they used different sized vertical and horizontal anchor box in the grid cell. The confidence value represents Intersection over Union (IoU) between ground truth bounding box. They implement their model as Fully Convolution neural network(FCN) by removing the fully connected layers of Darknet architecture. In their work used PASCAL VOC 2007 and 2012 dataset and they obtain 73.2 mAP. And they compare with R-CNN,YOLO and SSD current states of art approach.

CHAPTER FOUR

DESIGN FOR MULTIPLE OBJECT DETECTION AND RECOGNITION SYSTEM

4.1. Introduction

As discussed in Chapter 3, object detection and recognition system had been considered by different researchers with different algorithm/approach and datasets. In this chapter, we outline the problem we are trying to solve a real-time multiple object detection and recognition for autonomous vehicles system with the design goal that enables to attain accurate measurement in detecting different objects and recognize them in proposed. And, we are going to accomplish each specific objectives to achieve the main objective stated in chapter one. In the following sections, the details about the approaches/techniques and the model developed in the proposed solution are discussed. And also, in this chapter we have discussed different components of the proposed model for the domain of autonomous vehicles.

4.2. The proposed multiple object detection and recognition system architecture

The proposed system architecture for multiple object detection and recognition which is illustrated in Figure(4-1), presents the model and algorithm used to achieve the task to be done in the work. The architecture of the proposed methods contains different image processing techniques with the integration of current states of art.

The proposed system architecture consists of different components working together, Video to frame conversation, image processing, Feature extraction, Object classification and bounding box predictions, Training and Testing. In the first component Video to frame conversation, converts the acquired video file to frame images and then the frame images are prepared in the second components. In this image preparation stage, the image are normalize, remove noise, resizing and balance poor illumination preprocessing task are done before fed into the feature extraction components. The region proposal components, the main objects are isolated from the background. The next component is input vector preparation, which is a suitable size for all the input data and performs usable to the next stage in ConvNets model training to provide standardized, reduced, and compact format as input vector used by CNN algorithm. After input preparation vectors the next component is feature extraction. In this component have different feature extraction layers, in lower layer extract simple features like color, texture, etc. in middle layers extract some complex features like edges and in top layers learn more complex feature like parts of an objects in different ConvNets layer. Finally, the predict object classes and bounding box are obtained from classification and regression layers respectively in the model.

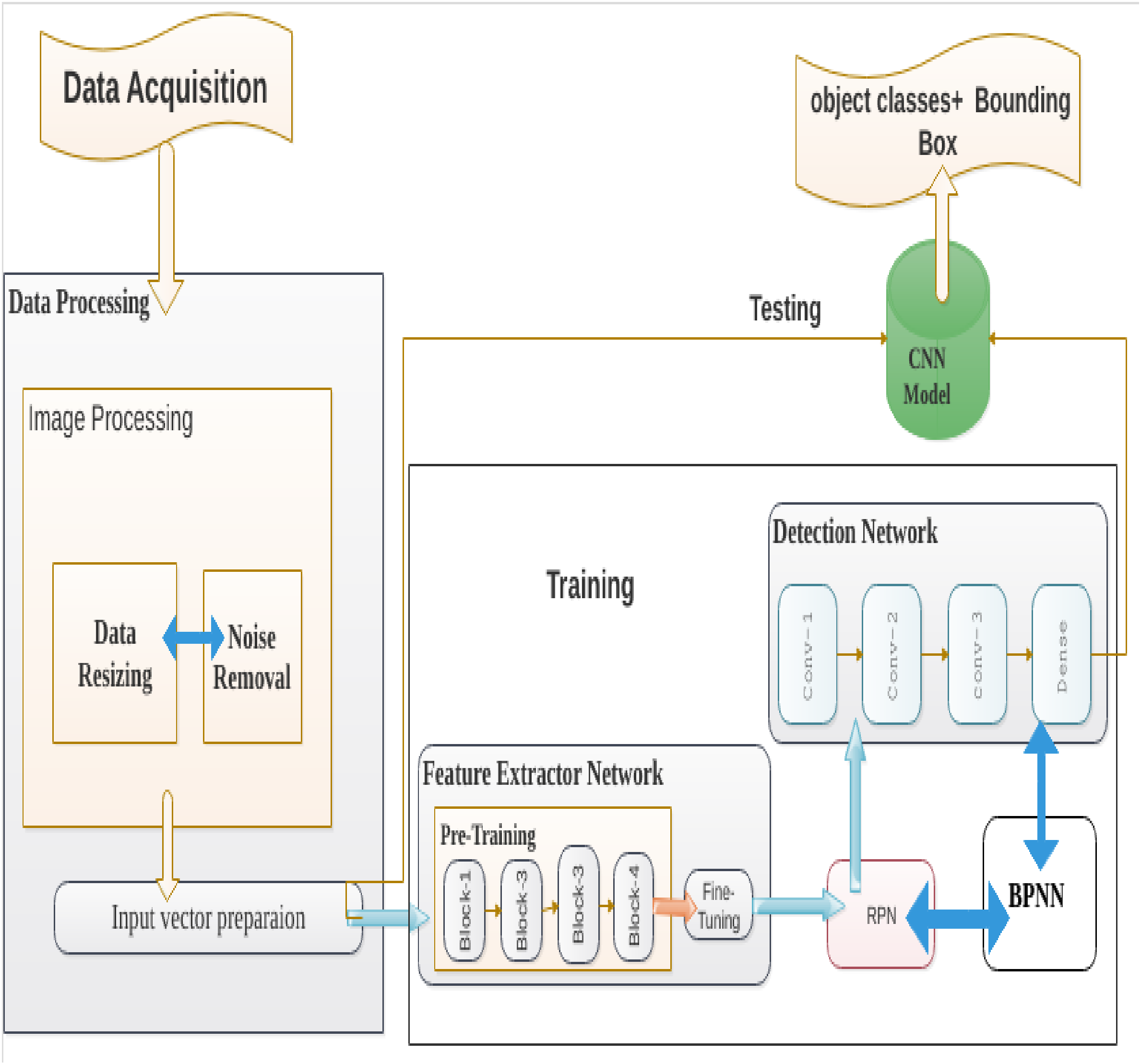


Figure 4-1: *The Architecture of the proposed Object detection and recognition system*

4.3. Components of the proposed system

4.3.1. Data Acquisition

The first step in our proposed methods is KITTI file acquisition from the resources of Karlsruhe Institute of Technology databases. The KITTI files recorded by left, right, rare and front side video cameras with high resolution, which make this dataset is unique[5,52]. Then the recorded video extracted into frames/images with high 1240 x 375 height to width ratio. We got this data sets from publicly available databases.

Figure 4-2: *Sample images from the KITTI datasets*

The datasets are annotated and labeled by the institute of technology researchers in Karlstuhe. It is aimed for training development and evaluation of autonomous vehicle object detection system. The labels contains both 2D and 3D bounding box for different class of objects like pedestrian, car, truck, cyclist etc. In KITTI data sets the labels provide 3D position of the center of bottom side of the bounding box, its rotation around Y axis, width and height as shown Fig 4.3.

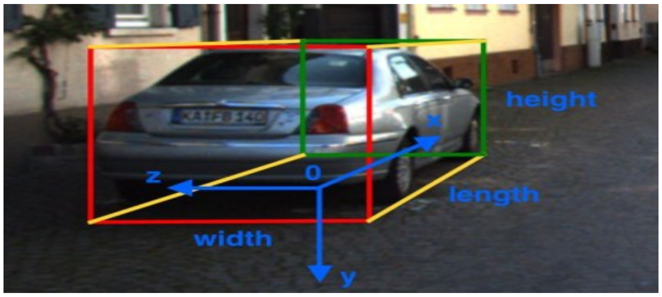


Figure 4-3: *Annotated in 3D Bounding box*

4.3.2. Data processing

In this section describes the data preprocessing step of the final solution. The original frames are exposed to noise and the size are different during data capturing. Hence to reduce these problem, we apply the following preprocessing task: resizing, normalizing the data, remove noise, histogram equalization, media filter and generate more training data.

The ConvNets model usually takes the same size input images data, thus all the image data are all resized into the same size. And also the bounding box have to accurate coordinate the resized images data. Data normalization, it is important step which ensures that the input parameter that is pixel has a similar data distribution. The normalization has done by subtracting the mean from each pixel values in input image then dividing by their standard deviation. Hence, we need the pixel number to positive, and use a scale to normalize data in the range [0,1]. The next preprocessing is data augmentation, this technique involves augmenting the existing datasets with perturbed version of existing images data. Data augmentation is done to expose the model to a wide variant of variation. The other data processing technique is media filter, is a non-linear filter used to remove high frequency signal noise and salt pepper noises from input images. The last technique we used in this thesis is Histogram equalization, it is method in image processing for contrast adjustment using the image's histogram. It used for intensity transformations, that change the given image distribution to a uniform distribution.

4.3.3. Image annotations

In deep learning world, one of the most wanted feature is to be able to recognize different objects on one image to be able to do need to train a model with annotated images. Image annotation is time consuming task to annotate different objects on thousands of images while to train the model with this data. This process is to do bounding on different objects on an image that need to be manually handled before the training.

Manual annotating an image, for this dissertation was not only label the object presented in the image but link that label to specific coordinate by drawing a bounding box that surrounds to each objects that would be used in training model. Figure(4-4) shows the annotation of an objects in images.

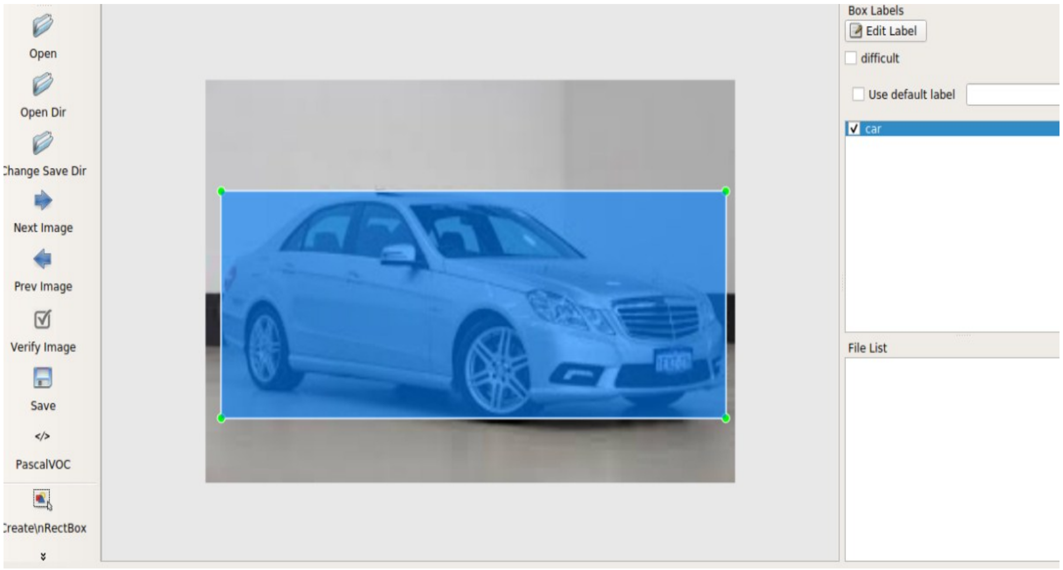


Figure 4-4: *Image annotation*

The process of image annotation using LabelImage Tools, requires a few step, namely:

1. Select data directory, then load the image dataset folder
2. When the datasets properly loaded, then configure the color of bounding box(red, blue, yellow) for different categories, bounding box type(rectangle, square) around each objects and the respective labels/classes of the object(car, pedestrian, truck, bus)
3. Finally, manually tagged for each images available in the dataset by drawing a box around objects and select respective class/categories.

Hence, after the data are annotated then use in Training, test-dev distribution for the model. In this thesis used 80% data as training and the rest used for test-dev.

4.3.4. Input vector preparation

For the purpose of getting usable data for the Deep convolutional Neural Network input, we perform dataset preparation. Input feature vectors are prepared by taking the original annotated data with their coordination to map together from each objects on an image. The data in KITTI detection benchmark put the image data in one folder as .png format and their corresponding labels are saved in another folder as text format, which contains the categories, x coordinate, y coordinate etc. of each objects in an images as shown figure(4-5).

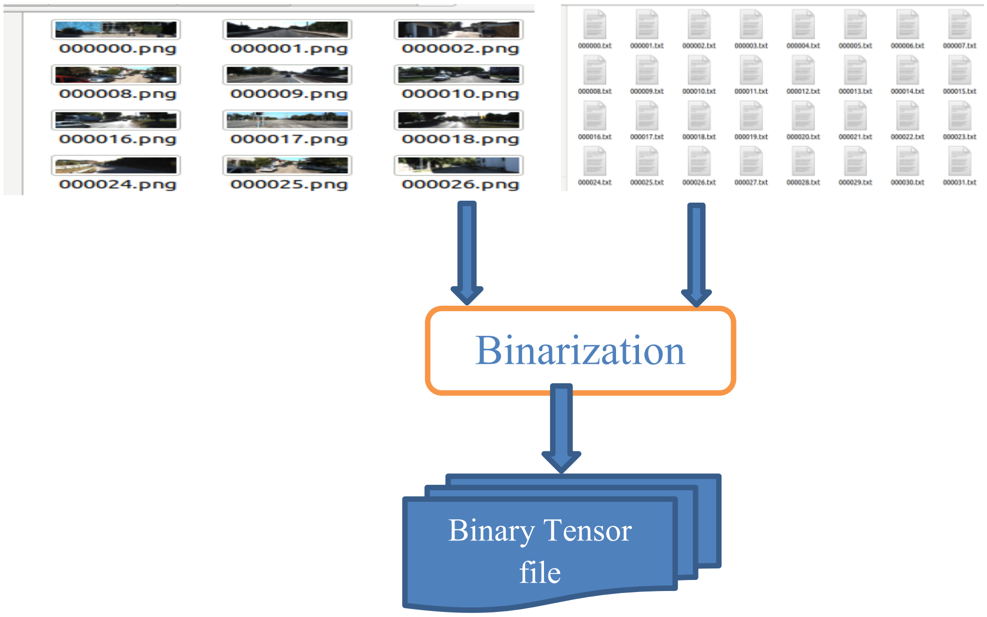


Figure 4-5: *Mapping annotated data*

In addition, deep learning algorithm are working on large datasets. Since tensor file format efficiently store large amount of data in compact binary format. It have significant impact on the performance on the training time of the model. In this thesis the image data and their respective bounding box map are converted into binary files. Hence, takes less time to copy and can be read much more efficiency from disk during training and testing of the model. The binary file in our thesis contains the mapped of each object with the bounding box coordinates and their class in the images.

4.3.5. Feature Networks

This part of the model is more important for detecting and classifies an object in input images. In this thesis used a Deep Neural networks, Specifically ConvNets mainly automatically detect/extract the important features. Feature is a piece of information that relevant for solving computation task related to a certain application without human intervention. Feature may be specific in the image such as points, edges and objects. Extracting as much information as possible from the available datasets is the main thing to creating an effective solution of object detection and recognition problem. Feature are detected and extracted in convolutional layers in the ConvNets model. Since, it learns distinctive features for each class by itself from many input datasets/images.

4.3.6. Region Proposal Network (RPN)

In the last feature networks outputs a set of feature maps, which stores information about pixel of an images. The information in the feature map is needed to classify and detect an object in the model. Hence, those feature map and anchor box region is feed into the regional proposal network two predict the effective bounding box around an objects and categorize them in input image. The network identified by the pixel co-ordinate of two diagonal corners and the value(1,0,or -1). When the value is one for in the region there is an object, the value is zero for background and its -1 the box can be ignored.

In this network, we takes all the reference box/anchors and scans each region then predict whether or not an object is present on that region/anchors. Since, it scans on feature map not the actual input images. Only those box with a corresponding high probability of object being present are further processed. And also, the two different outputs for each of the anchors feed into two layers. The first outputs, the probability that an anchor is an object. Here, the network identified only there is an object or not but don't care what class of object it is. We used objectless score to filter out the bad prediction for feed into classification layer. And the second output is the bounding box regression for adjusting the anchors to better fit the object it is predict and feed into regression layer.

During region proposal many region are might be identified on a particular objects. As a region can be highly overlapped with each other in a single object, we used non-maximum Suppression (NMS) to reduce applying filters. Hence, in our object detection model we use NMS ensure that a single object identified only once.

4.3.7. The Detection Network

In our model, we have built a ConvNets in detection networks, which consecutively modeling small pieces of information and combining them deeper in network. In the proposed model, we used from the base networks and the region proposal networks as input.

The first layer to detect edges and from templates for edge detection. Then subsequent layers to combine them into simpler shapes and eventually into different object positions. The final layers match an input image with all the templates and the final prediction is like a weighted sum of all of them. So, our deep CNNs are able to model complex variation and behavior giving accurate perdition. Thus, we designed a ConvNets with a very powerful and efficient model which performs automatic feature extraction to achieve superhuman accuracy.

In object detection network we recognized and detected different objects in input images. In our proposed system, the object can be recognized and identified the exact position in the images/video data. In our model the detection network takes from both feature network and region proposal networks as input to classify each objects and draw bounding box around them. We used 3 fully convolutional layers for the object networks. Here, there are two stacked common layer shared by a classification layer and regression layer to classify only the inside of the bounding boxes. Hence, the feature are cropped according to the bounding box.

In our proposed model built network contains three convolutional layers and one fully connected layer in the detection networks. The only fully connected layer is the final layer after the last convolutional layer. We use patch of region of intersection in the input layer of size in the first layer. And also, we use stride with two in order to reduce the size of feature maps. Then after each of ConvNets, we use ReLu activation functions to remove negative features. We used it for compute the positive feature maps. Lastly, we use softmax activation for classify each class of objects in the predicted bounding box.

In the proposed networks, the detection networks have 64 filters of size 5x5 in the first hidden layers, 128 filters of size 3x3 in the second hidden layers and 256 kernels with size 5x5 in the third layers. We used stride of two in all convolution layers to reduce the size of feature maps.

The output layers in the classification has 6 units to classify the probability of each object in a given input. And, in regression has 4 units to predict the bounding box localization around each objects.

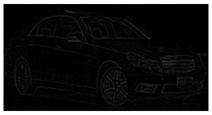
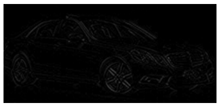
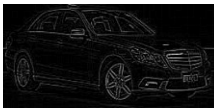
A. The Convolutional Layer

It is the main building block of automatic feature extraction for our ConvNets model. Convolution is a mathematical operation two make two set of information. We applies the input image data using convolution filter to produce a feature map in the model. And we extract the features from an image preserving the spatial connection from the pixel and the learned feature inside the image with the use of small quality sized tiles. Our learned feature are a consequence of a mathematical operation between each element from the input images and kernel matrix. We slides the filter matrix through all element of image pixel value and convolved by the kernel matrix and then produced a single matrix called feature map. And in addition, we used 3D (2D matrix with RGB image Channel) kernel size, which used to detect or extract RGB color input images. Because, computer sees an input image as array of pixel based on HxLxC(height, width and number of RGB color channels). Since, the same we used the same number of channel in filters and input data.

To detect and extract different features from the input data we used different filters as shown in table 4-1. They involves going through the input data/images and applying filter to some patterns. When one filter detect vertical edge detection while used another filter for horizontal edge detection and another used for some bluer effect features and so on. The filters/kernels are stack of weights represent as a vector, which are multiplied by some section of the input image and produced the feature maps. Since, we train the model, changes the weights or kernel values. The combination of high weights from different filter the network predict the content of an image.

After we perform a convolution on each of image section then normalization to the feature map using non-linear activation function. We applied an activation function on after every convolution layer. We used Rectified linear unit (ReLu) activation function to normalize the feature map as most researcher recommended. Thus, the algorithm perform the maximum between the ‘x’ value from feature map and zero (0). It takes an input ‘x’ and return ‘x’ when it is positive else return zero which means replaced all negative values by Zero.

Table 4-1: Filter/kernel metrics

Operation	Filter/Kernel	Convolved image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge Detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box Blur	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

Since, our network performs two main component classification and regression. Here, when two car and 3 pedestrian are appear on the input data then the detection networks classify all the objects on their classes/categories(car, pedestrian, cyclist, truck…) and draw a four coordinate(the most left top x,y coordinate and the most right bottom x,y coordinate) bounding box around the recognized objects.

4.4. Training and Model Construction

4.4.1. Training Deep CNN

For training, our model used to train both the region proposal and detection networks. In this research, used Residual Network to extract feature of an object with different categories (car, truck, pedestrian, cyclist) from input data and produced feature map for the RPN networks.

For training our model, first generate a number of bounding boxes by a mechanism called anchor boxes in the input images. Every 'pixel' of the feature image is considered an anchor. Then, generate twelve (with a scale value=0.25, 0.5, 1, 2 and aspect ratio = 0.5, 1, 2) different shape and size rectangle boxes in each anchor box by used as a center. Here, there are 10s of thousands of anchor boxes per image. Since, the number of anchor box are very high number and we should to reduce to improve the performance and computational problem. In the first reduction used NMS algorithm to removes boxes that overlaps with other boxes that has high score values. Here, about 2000 boxes are extracted for further reduction process.

Then the IoUs are used to label the 300 ROIs as foreground and background, and ignored labels. We takes the IoU of the bounding box against the ground truth boxes and put into foreground, background and ignore labels. Here, when the Intersection over Union (IoU) of the overlap anchors with the ground truth is greater than or equal to threshold value (in this research threshold value=0.5) are considered as foreground (1) label. And, the IoU value is less than or equal to 0.1 are considered as background (0) label while the IoU values are between 0.1 and 0.5 for ignore label. These labels are then used to calculate the cross-entropy loss, after first removing the ignored (-1) label class boxes. If there are not enough foreground and/or background ROIs to fill the fixed number, then some ROIs are duplicated at random.

In addition, bounding box regression to tighten the center and the size of the anchor boxes around the target are trained. The anchor box around the target need to be generated and losses needs to be calculated for back propagation. The distance vector from the center of the ground truth box to the anchor box is taken and normalized to the size of the anchor box. That is the target delta vector for the center. The size target is the log of the ratio of size of each dimension of the ground truth over anchor box. And the loss is calculated by using an expression called Smooth L1 Loss.

The features are cropped to 14x14 according to the size of the ROIs (for this, ROI width and heights are scaled to the feature size). Then the set of cropped features for each image are passed through the Detection Network as a batch. The final dense layers output for each cropped feature, the score and bounding box for each class. In this research the number of classes is 6 and one background. Hence, 300 x C, 300x4C in one-hot encoding form, where C is the number of classes (6 classes).

To generate label for Detection Network classification, IOUs of all the ROIs and the ground truth boxes are calculated. Depending on IOU thresholds (in our research, foreground above 0.5, and background below 0.1), labels are generated for a subset of ROIs.

4.4.2. Model Construction

The object detection model is constructed from feature network, region proposal network and detection network. The feature networks are constructed from different block of Resnet pre-trained and fine tuning model. The contracted model takes 7000 input data as an input neuron. And the input neurons are passed through 5 Resnet blocks and in each blocks have 4 convolution and polling layers that have the same behaviors. Since, this network outputted 1000 feature maps which stored pixel information.

And the RPN network constructed from 3 fully Convolutional Layers and outputted a 38x57 feature map with 12(anchor boxes) times 2(foreground or background) labels and 12 times 4(bounding box coordinates). Then the detection networks, contracted from four fully connected layers. The first fully connected layers takes 4098 neurons and decrease the number through the second, third and fourth FCN layers of the model. Then, the final FCN layers produced classifier and regressor. Finally, the classifier classifies into each 6 class probabilities and regressor predict the bounding box regression values for each object classes.

4.5. Classification and Regression

In object detection model constructed, the model performs both the classification and regression to detect and recognize each objects in input data. After constructed the classifier and regressor we used unseen test data to evaluate the model performance. To evaluate the performance for the model for the proposed model, we used the same classification and regression problems using the same data and target output. It's trained with back-propagation.

We have used 10% data to test the performance of this designed model. In the detection networks the performance is measured in the classification and the regression. Since, the performance is not measured in the single value. So, to measure the performance of the model by using mAP of the model.

CHAPTER FIVE

EXPERIMENTATION RESULT AND DISCUSSION

5.1. Introduction

This chapter presented the implementation details and experimental result of the proposed design for multiple object detection in the domain of self-driving vehicles. We studied and compared both the classification and regression performance with previous model done by other researchers. Since, we need to have analysis our data sets that we used for our multiple object detection and recognition model. A comprehensive set of experiments was performed to verify the performance of the proposed approach. Section 5.2.1 presents the datasets used in training and testing the system. Section 5.2.2 describes the implementations of the proposed system. Section 5.2.3 describes evaluation methods we used to evaluate our proposed approach. Section 5.3 presents the test results found. Finally, discussions are made in the last section of this chapter.

5.2. Experimentation

In the next section we presented the datasets used for our model experimentation, programming language for model implementation and evaluation methods used to measure the performance of our object detector and recognizer system.

5.2.1. Datasets

In order to test our multiple object detection and recognition model in the domain of autonomous vehicles, we used real-world Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) detection benchmark database. It is well known benchmark for object detection in autonomous domain. The KITTI datasets are publicly available data that suitable for Stereo, Optical flow, Visual Geometry, object detection and tracking computer vision states of art research problems [5]. Since, we used this datasets for development, training and evaluation of multiple object detection and recognition in the domain of self-driving car systems. The raw data sets are divided in Person, Residential, Road, Campus and City categories which collected from different distributions. The data in this datasets are a real data that collected from both Urban and Rural area street in different air condition and different lighting.

For each frame, we have images file and label files with the same name but different extension(.png and .txt). The image file are a regular .png files while the label files are text format files which contains the bounding box for objects. In the label files, each row stored one object information about its class and bounding box coordinates. The database have 7481 images data for training and 7512 images for testing for the domain of computer vision state-of-art. And also, up to 15 cars and 30 pedestrian are appear in one image data. Since, as shown fig 5-1 car and pedestrian are the two predominant object classes that often an object occurs in an images. We focused on the detection model those two object classes with other classes of objects (Tram, Van, Truck,Cyclist, Mise and siting person).

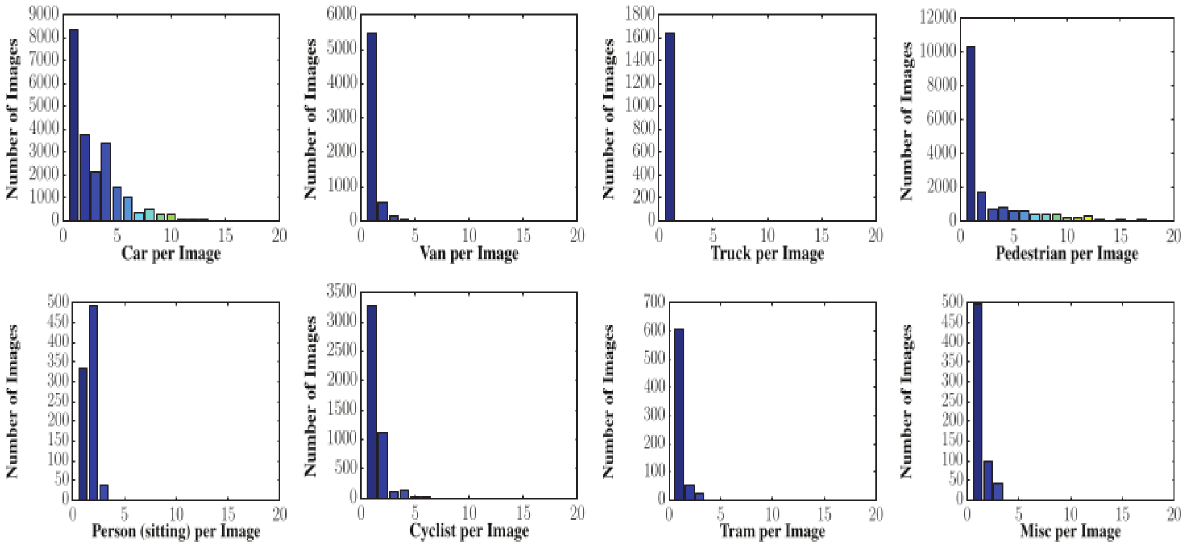


Figure 5-1: Number of object labels per class and image

To demonstrate in our research, we used 5000 images for training and 500 images data for testing due to machine constraints from the total data.

5.2.2. Implementation

To implement and design our object detector and recognizer system using python programming language. It’s the most programming language for image processing and computer vision problems. The design system is implemented on Toshiba core i5 8 GB RAM and trained on

Google Colabs which have Graphics Processing Units(GPU) with 12 GB RAM and 312 GB disk space in the cloud.

5.2.3. Evaluation Methods

To evaluate our object detector system, not used simple accuracy methods for the evaluation, because there are determine whether an object exist in the image(classification) and determining the location of the object(localization and regression) two distinct task measured. Furthermore, we have many class and their distribution is not uniform in our detection datasets. Thus, we have associate confidence score with each bounding box detected and to assess the model at various level of confidence. Since, we used standard Mean Average Precision(mAP) to specific Intersection Over Union threshold for evaluation methods.

In the classification task, we have used the confusion matrix for the detailed breakdown of correct and incorrect classification for each class in our results. As shown table 5-1, we built the confusion matrix by placing each prediction in the row of the target values what the model should have predicted the ground-truth(the original class) and the column of the predict class what the model actually predicted. The following table shows the description of all matrices with correspond to each one of the classes supported by our detector.

Table 5-1: Confusion Matrix

	Predicted class		
Actual Class		**Class=Yes**	**Class=No**
	Class=Yes	**TP**(True-Positive)	**FN**(False-Negative)
	Class=No	**FP**(False-Positive)	**TN**(True-Negative)

With the confusion matrix information, we defined the precision and recall for each one of the classes.

The precision measures the False Positive (FP) rate or the ratio for True object detection to the total number of objects that the classifier predict. We have measured the probability that a class is a true positive class given that our detector said it is positive.

$$\text{Precision}=\frac{TP}{TP+Fp} \quad \text{...……………………………………………………………………… (1)}$$

The Recall/sensitivity measures the False Negative(FN) rate or the ratio of true object detection to the total number of object in the datasets. We have measures the number of the negative classes have been identified as being negative.

$$\text{Recall}= \frac{TP}{TP+FN} \quad \text{…………………………….……………………….(2)}$$

We then calculate the mean precision for a specific class the Precision Recall(PR) curve is computed our detector outputs, by varying the model score threshold that determines what is counted as a model predicted positive detection of the class. To evaluate the detection performance more objective, we used PR curve to quantitative analyzed the classifier. It used to visualize the trade-offs between the true positive rate and the positive productive value for a predictive model using 11 threshold values(i.e 0,0.1,0.2,…,1.0). In our dataset we used imbalance number of classes. Since, PR curve is more appropriate for imbalance datasets. Precision and recall are always between 0 and 1. Therefore, AP falls within 0 and 1.

And in the Regression task, we used Intersection Over Union(IoU) evaluation methods to compute our predicted bounding box overlap with the ground truth(real object bounding box). When overlap value exceeds the 50% thresholds value, then the estimation is considered True Positive while the overlap under the threshold is false positive[50]. Equation 4 compute IoU is the area of Intersection divided by the area of union.

$$a_o = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \quad \text{……………………………….……………………………...(4)}$$

Finally to evaluate our object detector, we used mean Average Precision(mAP) used to evaluate the performance of detection algorithm. The mAP is defined as the average of the maximum precision at different recall values.

5.3. Experimentation Result

We analyzed the result obtained in our study to show what achieved and how much the problems stated are solved in the proposed model. In the proposed method different parameter were tasted, and the model with small error rate are selected as a detector for our problem. In our model we used 100 hidden layers for pre-trained base networks, 4 hidden layers for RPN and 4 hidden layers for detection networks. And also, in the model we used 0.0001 learning rate value, 32 for mini batch size, 8000 number of iteration. Here three parameters, namely number of hidden layers, number of units in hidden layers and learning rate are taken into consideration with experiments.

5.3.1. Number of hidden layers

In our experiments we tried to observe model losses in deciding the number of hidden layers to be used to achieve better performance. The number of hidden layers in the network affected both the speed and accuracy in detection and recognition systems. Hence, we used the better number of hidden layers in the network via experiment result. We decided the number of hidden layers that have less model losses in the networks. In our experiment the network lose decrease until three hidden layers and after three hidden layers slightly increase. Hence, for our model five number of hidden layers selected to object detection and recognition system.

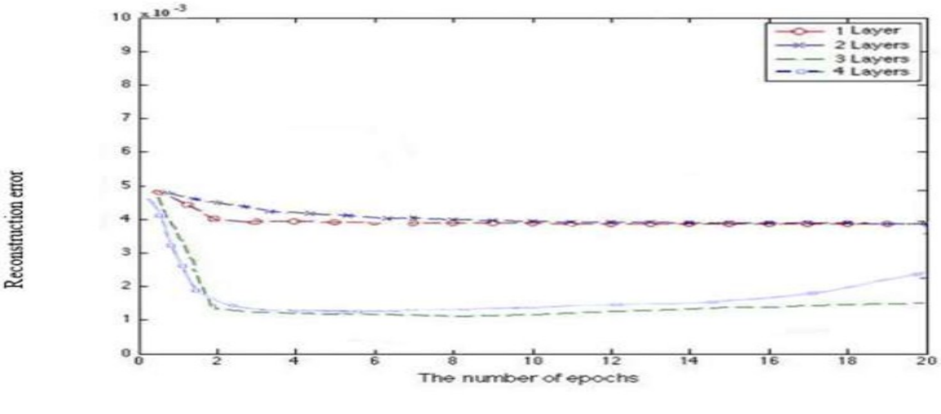


Figure 5-2: *Number of hidden layers*

5.3.2. Number of iteration for CNN training

The number of iteration in the network is the total forward and backward process in a given batch sizes. Since, to decide the number of training iteration of ConvNets which have less both localization and classification losses from experimental results. In our experiment set the number of iteration for this experiment is from 0 up to 6.124K. As shown in the figure(5-3) the losses dropped smoothly up to 3.336k and slightly increase up to 3.535K. When the number of training iteration increase and the losses also increase it show that the occurrence of overfiting. Hence, 3.336k is selected for model contraction in this work.

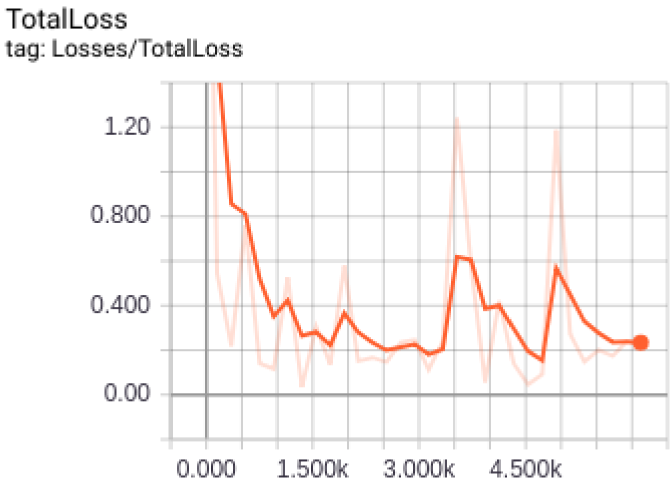


Figure 5-3: *number of iteration for CNN training*

5.3.3. Learning Rate

In our model start to set the learning from maximum numbers(i.e 1.000e-4) and decrease 10x through the training iteration until get very small number. The initial point of the learning rate is most CNN used this learning rate as starting points. As shown figure(5-4) the learning rate decrease up to 2k and after that goes slowly with very small numbers.

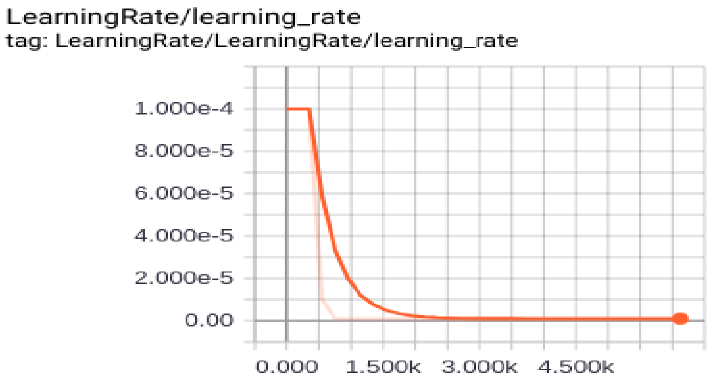


Figure 5-4: Cyclic learning rate for different iteration

5.3.4. Classification and localization losses

In object detection and classification problem two types of losses are experimented in our study. Firstly, the classification of the model performance are experimented and recorded in each iteration. And the second is localization losses which is the model performance regarding to bounding box prediction. As shown in figure(5-5) the 3.36k training iteration performs better for both classification and localization losses. So for our study we used 3.336k number of training iterations.

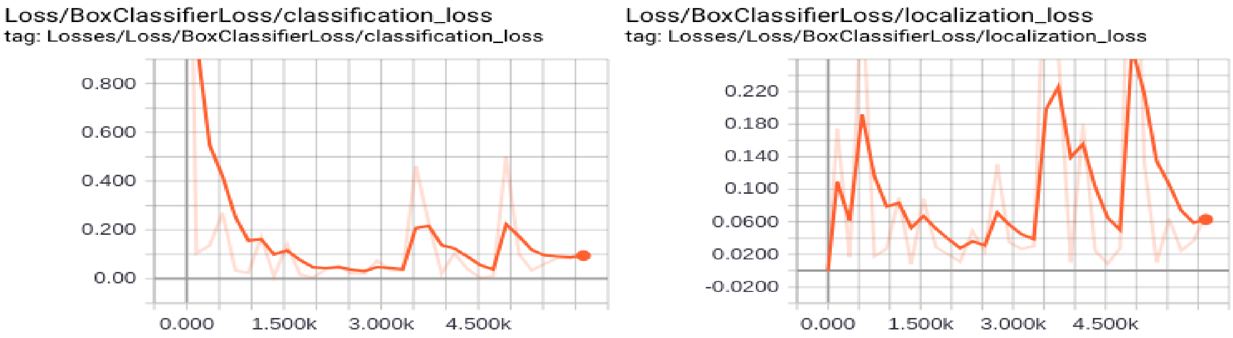


Figure 5-5: Classification and localization losses

5.4. Discussion

We have proposed multiple object detection and recognition methods using Convolutional neural network algorithm. Our proposed model evaluated by 80% for training sets and obtaining their accuracy by using the remaining 20% for testing and validation in experiments.

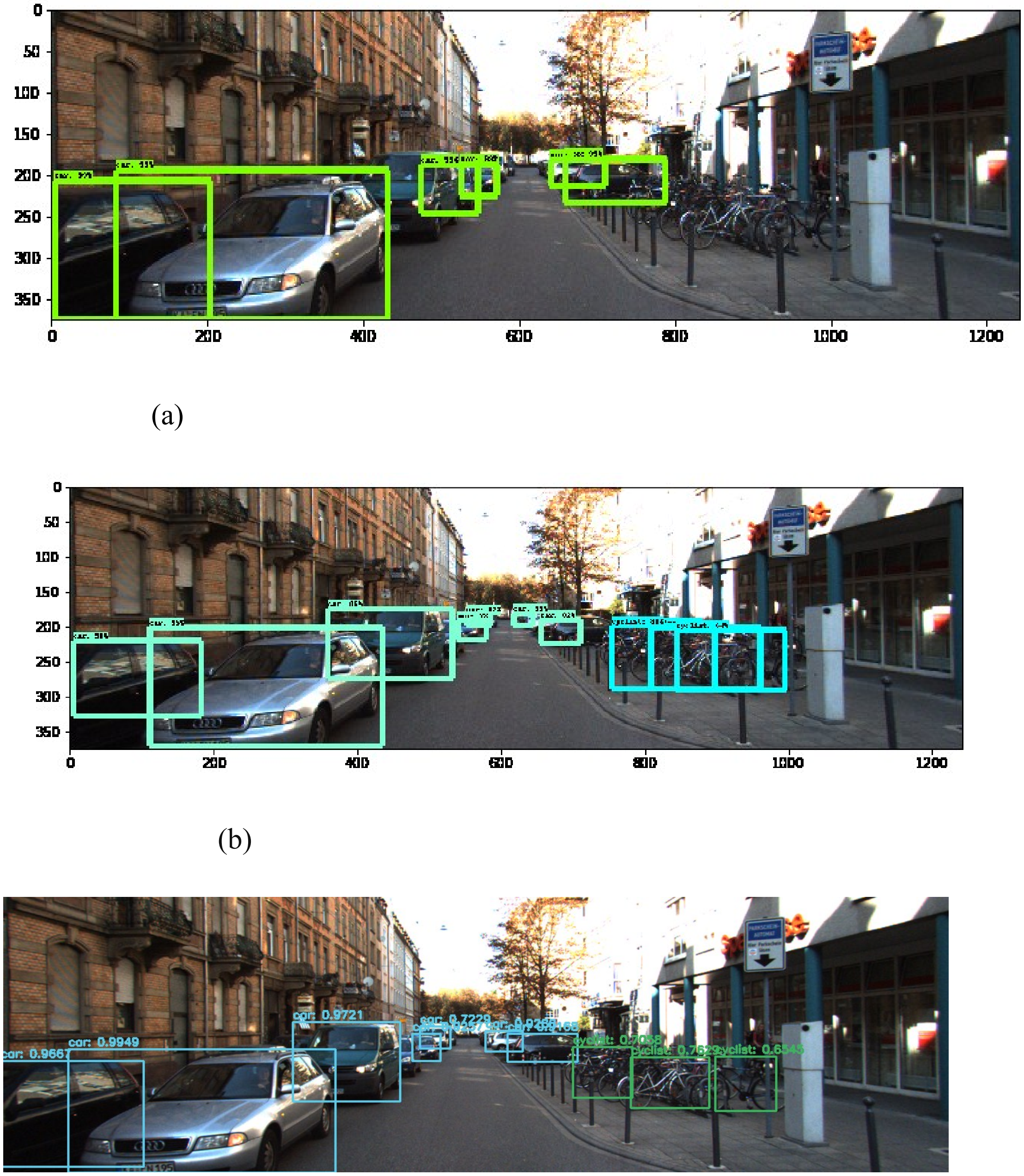
In our experimentation results, the performance of our model highly depends on the dataset size, model parameters and the architecture of the algorithm employed. Since, in the proposed model additional features are learned in ConvNets than the previous fully connected layers. Thus, when the more features are learned the false positive in the detection model decrease. So, it enhance the accuracy of the model. Multiple objects are detected and classified effectively from the given input and the ability to ConvNets to automatically extract both low level, middle level and high level features from the input data. Our model have the ability to classify categories of each objects and localize around them as per the test results.

The proposed model compared the quantitative result with other existing algorithm with the same datasets. As mentioned by the previous work[39,42], we evaluated in KITTI datasets and we obtained 0.838 average precisions and 0.662 average recall. In faster R-CNN[39] obtained 0.71 average precion and 0.682 avarage recall. And lastly by YoLo[42], we get average precision of 0.602 and average recall of 0.578 as shown table(5-2).

Table 5-2: the precision and recall @0.5 IoU for different class

	YOLO		**Faster R-CNN**		**Our Model**	
	Precisions	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
Car	0.71	0.70	0.88	0.84	0.96	0.91
pedestrian	0.63	0.60	0.75	0.72	0.93	0.60
cyclist	0.59	0.55	0.73	0.70	0.86	0.67
Truck	0.57	0.53	0.65	0.62	0.74	0.55
Van	0.52	0.51	0.54	0.53	0.70	0. 58
Avarage	**0.602**	**0.578**	**0.71**	**0.682**	**0.838**	**0.662**

In addition, we observe the qualitative results of the our studies with previous works[39,42] and in the result of test experiment our model obtained better qualitative results. Figure(5-6) shows some example testing results using these three models.



(a)

(b)

(c)

Figure 5-6: Qualitative Result of YoLo(a), fater R-CNN(b) and our model(c)

As shown in the above figure the testing for our CNN models performs much better than the two YOLO and faster R-CNN models. In YoLo models, some vehicles are doesn't detect because of the algorithms perform in grid level and cannot detect small objects. The faster RCNN used FCL at detection networks and this cannot learn additional features for classified and localize them. So, this algorithm detect more objects but cannot localize perfectly. Our ConvNets model can detect multiple cyclist on the right-hand side and small vehicles and localize them. The precision recall values for car are highest because the number of cars in each images is more than the other categories.

CHAPTER SIX

CONCLUSION AND FUTURE WORK

6.1. Conclusion

In self-driving technologies the system perceived the environment without human intervention. Since, the system can detect different obstacles and make decisions for smart transportation. In this studies we adapt and design different technique for detecting and recognize an objects in the input data. In our works, used different components like data processing, noise removal, input resize, input vector preparation, feature extraction, classification and regression problems.

In order to perceive the environment we used object detection computer vision states of art. Object detection and recognition problems which detected different objects from the input images and recognize the exact location for the domain of autonomous vehicles. However, in the current research, the performance of the detector is not reach matured. In computer vision, the number of objects and their locations are not constant which means it varies data to data.

In current research, they used fully connected layers in detection networks for the detector models. However, when we used fully connected layers the number of parameters are high and have many number of connection in the model. Due to this problem the performance in the detection networks is not satisfied and doesn't extract many features.

We developed a new model in detection networks using convolutional neural networks and extracted different level of features. It helps the model to extract more usable information to the classification and regression problems in the detector. We used 3 layers of fully convolutional neural networks to develop the model.

In the experiment, we have evaluate both the localization and classification mAP of the networks. And, we obtained 82% mAP network performance. And also, we evaluate the quantitative and qualitative results for the networks for each categories in the input data. In our model, we detected more objects that doesn't detect in the previous works.

6.2. Future Work

We have adopted and designed different techniques for multiple object detection and recognized them for self-driving with an encouraging result. However, we believe that incorporation of the following ideas as a future works, would achieve better result:

- Implementing the whole dataset of KITTI detection benchmarks improve the robustness and sensitivity of the model.
- Using unsupervised CNN for pre-training and fine-tune it with SVM, Random Forest or Softmax classifiers and try to compare each other will widen to choose the best result
- Using improved model architecture selection mechanisms and batch normalization will improve the model performance.
- Developing a system which able to reduced number of proposal regions.

References

[1]. D. Low “object recognition from local scale invariant features” in international conference on computer vision(ICCV),1999

[2]. J. Zhao “A more brief and efficient SIFT image matching algorithm in computer vision” in international conference on computer and Information Technology IEEE, 2015

[3]. L. Van,H. Bay, and T. Tuytelaars,”Speed up robust features” in lecture notes in computer science,2006

[4] L. Yali, L, Wanhao, S. Wang and X. Ding “Local Haar-Like feature in Edge Maps for Pedestrial Detection”, in international congress on image and signal processing,2015

[5] A. Geiger, P. Lenz and R. Urtasum, “Are you ready for autonomous driving? The kitti vision benchmark suit.” in conference on computer vision and pattern recognition(CVPR),2012

[6] M. SHAKTI and D. SHIV, "Comparative study of various image segmentation methods," International Journal of Multidisciplinary Acadamy, pp. 1-12, 2013.

[7]. A. Ardakani, M. Ahmadi and J. Gross “An architecture to accelerate convolutional in deep learning” IEEE transactions on circuit and systems,IEEE ,2017

[8]. C. Fernandez-Maloigne, "Advanced Color Image Processing and Analysis," Springer Science and Business Media New York, 2013.

[9]. T. Acharya and K. Ray “Image Processing Principles and Applications” in international Journal of research in computer Science, 2005.

[10]. E. Anjna and R. Kaur, “Review of image segmentation technique” in international Journal of research in computer Science, volume 8, No. 4,2017

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation.” In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[12]. Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," Nature, pp. 436-444, 2015.

[13]. K. Dramdahl “Morphological Operations Applied to Digital Art Restoration” in Scholarly Horizons,University of Mnisota, Volume 1, 20114

[14]. Z.-R. Wang, Y.-L. Jia, H. Huang, and S.-M. Tang, “Pedestrian Detection Using Boosted HOG Features,” in IEEE International Conference on Intelligent Transportation Systems, 2008.

[15]. N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Computer Vision and Pattern Recognition (CVPR), 2005.

[16]. P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Computer Vision and Pattern Recognition (CVPR), 2001.

[17]. A. Baareh ” Feature based 3D Object Recognition using Artificial Neural Networks,” International Journal of Computer Applications, Volume 44– No.5, April 2012.

[18]. L. Wenjie, L. Yujia, R. Urtasun and Z. Rechard “Understanding the Effective Receptive Field in Deep Convolutional Neural Networks” in conference of neural information processing, Barcelona, Spain, 2016

[19]. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient Based Learning Applied to Document Recognition,” Proceedings of the IEEE, vol. 86, 1998

[20] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?” in International Conference on Computer Vision (ICCV),2009.

[21] B. Graham, “Fractional max-pooling,” arXiv preprint arXiv:1412.6071, 2014.

[22] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in International Conference on Artificial Intelligence and Statistics (AISTATS), 2011.

[23]A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems (NIPS), 2012.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” arXiv preprint arXiv:1409.4842, 2014.

[25] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[26]. H. Kaiming, Z. Haingu, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition” in Microsoft research,IEEE,2015

[27]. M. Lin, Q. Chen, and S. Yan, “Network In Network,” arXiv preprint arXiv:1312.4400, 2013.

[28]. D. Kingma and B. Jimmy, “Adam: A method for stochastic optimization,” in International Conference for Learning Representations, 2017.

[29]. J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, “Speed/accuracy trade-offs for modern convolutional object detectors” In IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[30]. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard,W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural computation, vol. 1, no. 4, pp. 541–551,1989.

[31]. R. Uijlings, A. Sande, T. Gevers , and M. Smeulders,”Selective Search for Object Recognition” Technical Report 2012, submitted to IJCV, University of Trento, Italy,2012.

[32]. S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, “Object detection networks on convolutional feature maps” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 7, pp. 1476–1481, 2017.

[33]. X. Li and X. Guo, “A HOG feature and SVM based method for forward vehicle detection with single camera,” in Intelligent Human-Machine Systems and Cyber-netics (IHMSC), 2013 5th International Conference on, vol. 1, pp. 263–266, IEEE, 2013.

[34]. C. Narhe and M. Nagmode, “Vehicle classification using SIFT,” International Journal of Engineering Research and Technology ESRSA Publications, 2014.

[35]. M. A. Manzoor and Y. Morgan, “Vehicle Make and Model classification system using bag of SIFT features” in Computing and Communication Workshop and Conference (CCWC), IEEE 7th Annual, pp. 1–5, IEEE, 2017.

[36] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks,” in International Conference on Learning Representations (ICLR), 2014.

[37] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Computer Vision and Pattern Recognition (CVPR), 2014.

[38]. G. Lewis,”Object detection for autonomous vehicles” in Computer Vision and Pattern Recognition (CVPR), Stanford, 2015

[39] R. Girshick, “Fast R-CNN,” in International Conference on Computer Vision (ICCV), 2015.

[40] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Advances in Neural Information Processing Systems (NIPS), 2015.

[41]. H. Bappy and K. Roy-Chowdhury, “CNN based Region proposals for efficient object detection”, Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521,IEEE,2016.

[42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. “You only look once: Unified, real-time object detection.” In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[43] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, F. Cheng-Yang, and C. Berg. “SSD: Single shot multibox detector.” In ECCV, 2016.

[44] G. Yajing, G. Xiaoqiang, Z. Jiang, A. Men and Y. Zhou “Real-Time Object Detection by Multi-Feature ConvNets”, Beijing University of Posts and Telecommunications, Beijing, China, 2017.

[45]. Junliang Li, et al.," Multiple Object Detection by a Deformable Part-Based Model and an R-CNN," IEEE Signal Processing Letters, Vol. 25, No. 2, February 2018.

[46]. Yohei Koga, et al.," A CNN-Based Method of Vehicle Detection from Aerial Images Using Hard Example Mining," Remote Sens.10, 124, 2018.

[47]. M. Manana, T. Chunling and A. Owloawi "Preprocessed Faster RCNN for Vehicle Detection" IEEE Department of Computer Systems Engineering Tshwane University of Technology, 2018.

[48] N.Dalal and B.Triggs, Histograms of Oriented Gradients for Human Detection. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 2005. 1: p. 886 - 893

[49] L. Yali, L, Wanhao, S. Wang and X. Ding "Local Haar-Like feature in Edge Maps for Pedestrial Detection", in international congress on image and signal processing,2015

[50] M. Everingham, A. Eslami, L. Gool, K. Williams, J. Winn, and A. Zisser-man, "The pascal visual object classes challenge: A retrospective," International journal of computer vision, 2015.

[51]. S.Yuheng and Yan Hao "image segmentation algorithm overview" SiChuan University, SiChuan, ChengDu, 2016

[52]. A. Geiger, P. Lenz, C. Stiller and R. Urtasun," Vision meets Robotics: The KITTI Dataset" International journal of Robotics Research(IJRR),2013

[53]. N. Vijayalakshmi, M. Senthilvadivu, "Performance Evaluation of Object Detection Techniques for Object Detection" R.V Engineering College and Research scholar, 2017

[54]. Y. Mariano, D. Mihalcik, M. Junghye, H. Liz, J. Park, D. Doermann and R. Kasturi, "Performance Evaluation of Object Detection Algorithms" Pennsylvania State University, University Park, PA 16802 USA, 2017

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in European Conference on Computer Vision (ECCV),2014.