ABSTRACT

Deep learning, commonly known as a artificial neural network, is a form of deep learning. Self-driving cars, news aggregation and fraud detection news, entertainment and health care are some of the important applications of deep learning across industries. Deep learning will more deeply be seen as the potential of artificial intelligence thanks to the ongoing exponential growth in the volume of data resulting in increased computing capacity. Main advances in deep learning hierarchical features from various types of data, such as numerical picture, text and audio, entity recognition, regression, semi-supervised and unsupervised problems. Deep learning algorithms feature self-learning representations, relying on ANNs in that convolutionary neural network is a better option for pedestrian detection because it is a multi-layer, feed-forward neural network that uses perceptrons for supervised learning and data analysis. It is mainly used for visual data, such as image classification. A pedestrian detector locating pedestrians on a visual picture is a crucial factor involving a real-time answer. The vast range of individual images in diverse situations, clothes, illuminations, and environments has made the topic quite complex.

To research neural networks and image recognition techniques, build an image archive and remove HOG features from pictures, Pedestrian identification utilizing deep neural networks. The purpose of the detection is to locate the pedestrian in the picture frame. It is therefore important to optimize true detection while reducing false detections. The scheme is split into three sections. The first stage of the project is the probability of segmentation identification, which involves boundary identification and converts the initial picture to a gray image and converts it to a binary image. The second stage of the project is the extraction of the HOG feature and the last stage is the classification of the neuron networks and the choice of the different size window for the detection of pedestrians. The algorithm is being simulated in MATLAB ®. MATLAB operates very quickly and advanced data structures, integrated editing and debugging tools and support for object-oriented programming make MATLAB an excellent tool. MATLAB may also call functions and subroutines that are written using C.

For different modules alternate between the original image and the binary image based on the higher-level objective requirements. The pedestrians are identified using a window with an accuracy of 75% of the HOG attribute extracted using the convNet method in MATLAB.

TABLE OF CONTENTS

List of figures

List of Abbreviations

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
DNN	Deep Neural Network
GPU	Graphics Processing Unit
SDN	Switchable Deep Network
SNU	Seoul National University
SSD	Single Short Detector
AE	Auto Encoder
DBN	Deep Belief Network
RBM	Restricted Boltzmann Machine
BRNN	Bidirectional Recurrent Neural Network
GAN	Generative Adversarial Network

Chapter 1

INTRODUCTION

Deep learning is a class of machine learning. It is a multi-layered neural network with many parameters. Many deep learning approaches utilize the neural network model. This is also classified as Deep Neural Networks. Specific types of neural networks are used in deep learning, such as the Convolution Neural Network (CNN), the Recurrent Neural Network (RNN) and the Strong Short Term Memory (LSTM). But there are varying pros and cons to each of the architectures. The Convolution Neural Network is primarily used for the Image Recognition Architecture because it can effectively extract different features from the image. Deep learning to use is very effective in predicting known or unknown results. Deep learning works fine on a wide range of large datasets. Deep learning has been extensively used in visual images, speech recognition, facial recognition, recognition of fingerprints and recognition of iris etc... The technique of deep learning may be used to enhance facial recognition. We do use the Convolution Neural Network and is commonly called CNN. CNN solution is a single neural network system with the potential to recognize the properties of the existence of the image in the data. This technique allows the raw image to be transformed to a secure format. CNN is a combination of distortion and spatial transformation. The layer used to extract the image is called a convolutionary layer. CNN has four layer structures to measure the degree of change, scale, and distortion. The level of accuracy of CNN is relatively small. To build a pedestrian identification device that can be used in broader implementations such as a driverless car. This higher level of use of our method determines the specific aim of our project, which is essentially: assume that the input image is a street view image taken from the front camera of a car, filter the information to identify places where upright people are standing or walking. In other words, the aim of our project is to find upright pedestrians in an image.

1.1Literature Review

Sanjukta Ghosh, et.al, [1], presents that the system such as security, driver assistance and automated vehicles, the identification of pedestrians is a vital activity. a novel approach to the identification of Pedestrian to use a large, complicated neural network (CNN) specializing in pedestrian counting. This function removes the need to customize the positioning of the pedestrian by bounding boxes in the training information. Decoded outputs of the conditioned counting model filters are used to classify the pedestrians. Despite the flexibility of research using only pedestrian estimates, the total missed incident values for the datasets tested have been shown to be within the same spectrum of other strategies. It is known that this approach is suitable for the detection of pedestrians in intense occlusion areas as well as in less crowded regions.

Jae Kyu Park, et.al, [2], implemented an image recognition platform based on Deep Learning for pedestrian identification and monitoring enhancement of CCTV and carried out successful software tests by developing a test and assessment process. The research hypothesis is that if someone's face has not been recognized in the crime scene CCTV video, the same person will be identified and found in other CCTV image data using the Color Strength Recognition system as body features for clothing colors. Deep learning to improve the precision, speed of the tracking and at last the tracking of the pedestrians.

V. Campmany, et.al, [3], describes an Integrated Nvidia Tegra X1 GPUCPU hybrid device for real-time pedestrian detection. The detection line consists of the following state-of-the-art algorithms: histograms of local binary patterns (LBP) and histograms of oriented gradients (HOG) are features extracted from the input image; creation of candidates using the Pyramidal Sliding Window Technique; and definition using the Support Vector Machine (SVM). Experimental results show that the Tegra ARM architecture is twice as powerful as the GPU for a laptop and at least eight times faster than the CPU for a multicore laptop.

Pierre Sermanet, et.al, [4], implemented pedestrian detection is a subject of great practical importance. Adding to the list of successful applications of deep learning approaches to vision, the author reported state-of-the-art and competitive results on all major pedestrian datasets using a convolutionary network model. The model includes a number of different changes, such as multi-stage functions, layer-bypassing links to combine global shape information with local distinctive pattern information, and an unsupervised strategy for pre-training filters at each stage based on convolutionary sparse coding.

Xiaoheng Jiang, et.al, [5], describes that deep neural networks (DNNs) have shown state-of-the-art detection performance on pedestrian datasets. Due to its high computational complexity, the detection efficiency of graphics processing units (GPUs) is still a challenge. This paper proposes to allocate functions to improve identification efficiency across a class of DNNs suited to pedestrian models of varying sizes. By sharing features, the computational responsibility for removing features from the image pyramid may be reduced. At the same time, pedestrians will be detected on a single layer of an image pyramid of varying sizes. In comparison, the improvement in detection efficiency is achieved with a reduction in marginal detection accuracy.

Zhaowei Cai, et.al, [6], describes that the CompACT suggested improving the Complexity-Aware Detector Cascade learning algorithm. By minimizing classification vulnerability under a complexity constraint, CompACT generates cascades that force high-complexity features into the later stage of the cascade. This has been shown to allow the smooth integration of a number of functional families into a single interface. This versatility extends to functions that have traditionally been outside the scope of cascaded detectors, such as deep CNNs. The CompACT cascades have proposed to generalize the common combination of object proposals and CNN that they have been shown to outperform. Ultimately, it has been shown that the pedestrian detector built by applying CompACT to a broad feature pool achieves state-of-the-art detection speeds on Caltech and KITTI, with speeds much faster than competing approaches.

Ping Luo, et.al, [7], presents a switchable deep network for context noise modeling and dynamic pedestrian recognition of differences in appearance. This SDN strengthens the traditional neural convolution network by incorporating several switchable layers that are designed with a modern restricted Boltzman switchable system. This new deep model studies the hierarchical functions, the salience charts and the representations of the body parts together. It ensures state-of-the-art performance in the public comparison data collection.

Yonglong Tian, et.al, [8], presents that the details move from existing scene segmentation databases to a pedestrian dataset by incorporating a new deep model for learning high-level features from various tasks and specific data sources. Because different tasks have different convergence rates and different distributions of data from different datasets, a multitasking deep model is specifically designed to organize tasks and reduce variances between datasets. Extensive analyzes indicate that the latest approach outperforms the state-of-the-art of the rigorous Caltech and ETH datasets, decreasing the lost rates of previous deep models by 17 per cent and 5.5 per cent respectively.

Hailong Li, et.al, [9], presents that pedestrian detection method focused on a deep convolutionary multi-layer neural network. Allows the full use of the capabilities of the deep convolutionary neural network and delete features from the pedestrian recognition library. They used the edge box algorithm instead of the sliding window algorithm to remove windows to solve the question of too many redundant windows created by conventional methods. Finally, they had a smaller number of high-quality windows, which was very important for the subsequent task of classification. Multi-sets of contrast experiments were performed in this process. Experiments have shown that the deep-learning pedestrian detection method outperforms traditional methods based on features that are both built and trained.

Carlos Ismael Orozco, et.al, [10], proposed deep convolution network architecture to identify candidate regions previously created using a simple pyramid sliding window approach as pedestrian or non-pedestrian. A distinguishing characteristic of CNN in this method is the isolation of pedestrians from non-pedestrian photos without the assistance of a pre-classification point, and without the need for extra adjustment measures or initial requirements, rendering them straighter than most CNN-based approaches. The outcome for the validation collection was an average of 98% of completion.

Hyok Song, et.al, [11], presents a Pedestrian / Car Detection, Tracking and Action Recognition Program using SNU (Seoul National University) CCTV deep learning video streams allowed. This technique includes tuning of SSD (Single Shot Multibox Detector) and cell networks for quicker process and higher detection ratios.

Anelia Angelova, et.al, [12], describes a Deep Neural Network-based pedestrian detection algorithm that combines fast cascade ideas with a deep network. Implementation is easy because it is based on open source software. This system ranks among the best for pedestrian tracking and runs at 15 frames per second in real time. It's the only approach that's both real-time and high-precision.

Wanli Ouyang, et.al, [13], describes a unified deep model which study together four components: insulation, hand deformation, occlusion management and pedestrian identification. Joint learning produces the highest results on publicly accessible datasets by working with these interdependent elements, outperforming existing top-performing approaches by 9% on the largest Caltech dataset. Detailed theoretical analyses explicitly demonstrate that the proposed new model will optimize the power of each system as both components function together. The Author enriches the deep model with the introduction of the deformation layer, which has tremendous flexibility to incorporate a number of deformation solutions.

Hongquan Qu, et.al, [14], uses YOLOv3 to train the pedestrian detection model and increase detection accuracy, based on deep learning and enhancement of the Retinex image. The test results indicate that the model's overall identification rate is 94 per cent. The detection impact is good and the false detection rate is significantly lower than the model without improving the picture. The usage of the Retinex Picture Enhancement Method for pre-treatment of passenger flow photographs in the subway would then make it possible for YOLOv3 to learn a better design. In addition, the eligible model has an appropriate recognition score for scenarios where the number of single-image pedestrians is less than 25. This method is often realistic and can be applied to other places.

Keke Shi, et.al, [15], suggested substituting the typical rectangular neighborhood in crowded scenes with a circular neighborhood for pedestrian trajectory prediction, modeling the complex movement actions of pedestrians and demonstrated that utilizing circular neighborhood has a higher value than traditional rectangular communities in predicting trajectories. The author effectively incorporates pedestrian identification models, multi-target monitoring and pedestrian estimation to address the issue of manual marking of dataset workload and data set acquisition process. Irrespective of the increasing complexity of the algorithm, it can be shown that the algorithm has a greater error in predictive results than the predictive model trained in the actual data collection, since the yolov3 pedestrian detection stage creates a certain error relative to the conventional S-LSTM output.

Seong Kyung Kwon, et.al, [16], implemented an object tracking using a mirror, a LIDAR, and a RADAR was investigated. Camera-based methods, though, also provide a strong degree of image stabilization and are sensitive to light strength. LIDAR can calculate the distance from objects accurately, but the detection of objects is challenging. Furthermore, it is known that it is extremely difficult to detect them when the pedestrians are partially occluded due to insufficient data to identify them. We use LIDAR and RADAR sensors to improve detection precision to solve this issue. A low-complexity fusion system scheme for monitoring partly occluded pedestrians.

Enjia Chen, et.al, [17], presents in order to measure the similarity between two feature vectors more precisely, a linear combination of absolute difference and the product of two vector elements is used in this paper to optimize distance function, along with a modified pedestrian retrieval framework based on Faster R-CNN. The algorithm uses the Region Proposal Network (RPNN) to obtain a more accurate pedestrian retrieval result. The CUHK-SYSU dataset tests reveal that the new approach reaches 80.9 per cent in the top-1 CMC and 78.8 per cent in the mAP, rising by 2.0 per cent to ~18.0 per cent in the top-1 CMC and 3.0 per cent to ~23.0 per cent in the mAP relative to standard approaches.

Jiao Zhang, et.al, [18], presents that the neuro network based on R-CNN Quicker for detection of pedestrians with blurry visual pictures. The new Faster R-CNN improves the convolutionary neural network by introducing a multi-class classification layer. The new deep model studies specific features and features affected by blurred vision. It achieves state-of-the-art performance on skewed visual datasets. The division of people into many different groups to track down is different from past thoughts. And the class boundaries are not fixed.

MP Nkosi, et.al, [19], describes autonomous pedestrian recognition devices may provide a way to popular the deaths of pedestrians on the roads every day. Identification and tracking of pedestrians from a moving vehicle has been implemented. This research has shown that the software increases speed and performance by combining different algorithms. Three algorithms were used, namely the HOG, the Kalman filter and the particle filters. The HOG algorithm was used to classify pedestrians. It has been used as a pedestrian sensor in combination with a Kalman filter or a particle filter.

Peilei Dong, et.al, [20], presents that a region-specific R-CNN proposal algorithm. The goal of the ACF-based area proposal algorithm is to forecast regional proposals that are used in the R-CNN. Unlike a popular area proposal specific search algorithm that identifies generic object positions, the algorithm can only be used to produce region proposals for a pedestrian class. The author has explained the descriptions of our area recommendation algorithm and network architecture. The experimental results suggest that our algorithm improves the accuracy of the sector proposals. Compared to the state of the art of pedestrian detection, the hybrid method achieves competitive results and consumes less time for R-CNN training and testing.

Zhaowei Cai, et.al, [21], proposed the Compact Increase Complexity-Aware Detector Cascades Learning Algorithm. By reducing classification weakness under a difficulty restriction, CompACT creates cascades that transfer high-complexity features to the later stage of the cascade. This has been shown to allow the smooth integration of a number of functional families into a single interface. This functionality applies to features that have historically been beyond the reach of cascaded detectors, such as deep CNNs. The CompACT cascades have suggested generalizing the popular mix of CNN object proposals where they have been seen to outperform. The author reveals that a pedestrian detector trained from the Lightweight implementation of a variety of function pools achieves state-of-the-art detection speeds on Caltech and KITTI, with speeds much faster than competitive approaches.

S.P.Vimal, et.al, [22], describes that the reduction achieved in contexts is 58 per cent on average, 45 per cent on average and 41 per cent on average. In many cases, the reduction in detection windows was found to be over 90%. If the context is explored utilizing different background methods, such as geometric context elements, the capacity for success in the identification stage is higher.

Koti Naga Renu Chebrolu, et.al, [23], proposed the R-CNN is for walking for quick tracking. Models have been trained separately using color and thermal pictures. The MAPs of the corresponding Color and Thermal versions are '79.12 percent' and '80.41 percent' respectively. The Visibility Perception Model was developed to detect low light and daytime conditions for pedestrians. A brightness-conscious device can be used during the day as it utilizes both light and thermal models to identify pedestrians on the lane. The light vision trend has a '81.27 points' rating. Additional parameters can be used as part of future work to distinguish between day / night situations for a more comprehensive role of the model.

Weicheng Sun, et.al, [24], presents a small yet network used to delete a feature map as a simple network to illustrate the characteristics of a walking pedestrian. The regional proposal splits the network and uses the convolutionary feature maps to reduce the time spent in the process of generating regional proposals. A decision is taken to identify the tree in order to increase the accuracy of pedestrian detection.

Zahid Ahmed, et.al, [25], describes road is a vital part of daily traffic, but it remains equally dangerous for commuters, particularly pedestrians. Due to a major shift of meaning, as well as severe obstruction or occlusion, pedestrians remain a daunting task among the multiple classes that exist for object detection. Achieving high precision and speed in real time is also a major challenge, as pedestrian detection should be both reliable and easy enough to be carried out in real time. These requirements are using a depth-wise separable convolution and single-shot detector method utilizing numerous OpenCV activation maps to achieve precise, robust and competent deep-learning pedestrian detection for real-time activity.

Jorg Wagner, et.al, [26], presents first deployment of deep pedestrian recognition CNNs based on multi-spectral image data and a comparison of two deep architectures, one for early-and the other for late-fusion. Our research on the KAIST Multispectral Benchmark data set shows that the pre-trained late-fusion-based architecture will significantly outperform the state-of-the-art ACF+T+THOG approach, while the early-fusion architecture is largely unable to achieve state-of-the-art efficiency. This may be due to the failure of the early-fusion network to learn realistic abstract multimodal features in a given environment.

TianRui Liu, et.al, [27], presents pedestrian process identification class dependent on DPM speed limitations. The author describes a relatively fast method of head-to-back detection that generates accurate candidate windows in a short time to significantly reduce the overall computational costs of pedestrian detection. In order to enhance tracking precision, a two-pedestrian system is used, in particular in cases where pedestrians are close to each other. The INRIA dataset analysis reveals that the suggested form of pedestrian detection achieves a similar rate of detection to the DPM detector with increased level of implementation.

Ping Luo, et.al, [28], proposed that the SDN immediately explores structural features, salience maps and descriptions of various body parts as mixtures. Due to shifts in stance and orientation and other factors, the detection of pedestrians faces challenges of visible glare and large variations in the appearance of pedestrians. Some of the key improvements is the recommendation of the Switchable Restricted Boltzmann Machine (SRBM) to precisely model the complex mixture of multi-level visual variations. At the rate of operation, saliency maps for each test sample are dynamically determined to differentiate background clusters for pedestrian detection from segregated regions. This is capable of inferring the most effective design for the combining models of each component and the entire body at the stage of the component and the body. We have developed a new generative algorithm to effectively pre-train the SDN and then fine-tune it to back-propagate. Our system is evaluated on Caltech and ETH datasets and achieves cutting-edge detection performance.

Seong Pyo Jeon, et.al, [29], presents that the pedestrian identification is a critical tool that can be utilized in the advanced driver assistance system to avoid a collision between a vehicle and a pedestrian. Most of the pedestrian identification areas work on the basis of people standing or walking in the training process. The IRIA pedestrian dataset consists of people standing and facing the front, but other datasets include different types of pedestrians without a path grouping. In other words, the guidance of the pedestrian is not taken into account when making detectors. A pedestrian recognition program that utilizes pedestrian data divided into four by changing paths, such as front, back, left and right. It incorporates each of the detectors created by the classified data used for pedestrian detection. The histograms of the guided gradients are used for guidance on the distribution of the edges for training.

Carlos Ismael Orozco, et.al, [30], presents a strong results of vision experiments using deep convolution networks make them an attractive method of enhancing the functionality of pedestrian detection systems. A broad, convolutionary network architecture that classifies the candidate regions previously created using a simple pyramidal sliding window approach as pedestrian or non-pedestrial. A distinguishing feature of CNN in this approach is that it separates pedestrians from non-pedestrial images without the help of a pre-classification procedure and without the need for additional adjustment procedures or initial criteria, making it easier than other CNN-based methods. The data used for the preparation and assessment is derived from the Pedestrian dataset between Caltech and the USA. On the proposed configuration, we evaluated the effects of the classification and obtained an average score of 98% for the validation kit.

1.2 Motivation

Autonomous driving requires sensors to observe and understand the vehicle environment, to provide self-location, vehicle control, and route planning. A pedestrian detector positioning people on a digital image is a crucial element requiring a real-time response. The broad variety in the appearance of humans in various poses, clothing, illuminations, and backgrounds has made the issue very complicated and the subject of intensive work over the last 20 years.

1.3 Problem Definition

Pedestrian detection for autonomous driving vehicle using neural network.

1.4 Objectives

The objectives of this project are as follows:

- To study the neural networks and image processing techniques.
- Creating data base of images and extracting the HOG features in the images.
- Pedestrian detection using deep neural networks.

1.5 Methodology

The purpose of the detection is to locate the pedestrian present in the picture frame. It is therefore necessary to maximize true detections while minimizing false detections. The scheme is split into three parts as shown in the figure1.1. The first stage of the project is the possibility of segmentation detection, which contains border detection and then converts the original image to a gray image and converts it to a binary image. The second stage of the project is HOG feature extraction and the last stage is classification of neuron networks and chooses the different size window for detection of pedestrians.

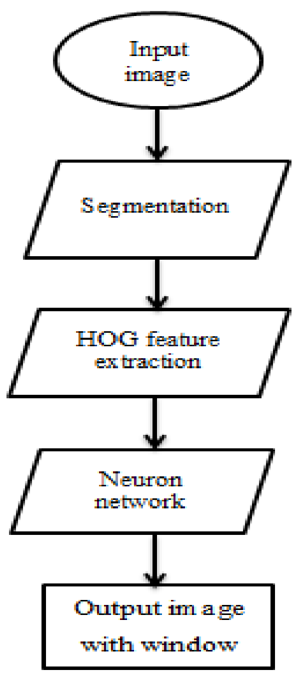


Figure 1.1 steps of pedestrian detection

1.6 Organization of the Report

The report shall be organized as follows,

Chapter 1 gives brief information about deep learning and convolutional network. It also includes literature survey, motivation, and problem definition, objectives of the work, usage of software, methodology and organization of report.

Chapter 2 explains about deep learning, the form of deep learning methods and the usage of deep learning, CNN and pre-trained network. Focus on explaining about the software used in the work and its advantages.

Chapter 3 The chapter describes how the convolutional network works and the steps involved in it.

Chapter 4 discusses about the performance analysis of the results.

Chapter 5 describes conclusion of project undertaken and future project scope.

Chapter 2

DEEP LEARNING

This chapter describes that the importance of the deep learning and its working. Mainly it describes the applications of deep learning and describes the CNN and MATLAB.

2.1Deep learning

Deep learning is a sub-part of machine learning. It has a great number of layers and parameters. Deep learning uses multiple layers of non-leaner processing units for the extraction and transformation of objects. The lower layers are similar to the data input learning basic features, and the higher layers learn more sophisticated features extracted from the lower layer features. Architecture is a powerful, hierarchical representation of features. This means that deep learning is appropriate to analyze and extract useful knowledge from enormous quantities of data and data from various sources. Typical deeper-learning techniques are autoencoder (AE), Deep Belief Network (DBN), Convolutionary Neural Network (CNN), Recurrent Neural Network (RNN). In the 1980s the neocognitron was introduced by Kunihiko Fukushima as a hierarchical artificial neural multi-layer network. It provides a method for handwriting character detection and other pattern recognition. They later developed neural networks of the Convolution. A Restricted Boltzmann (RBM) processor is a generative artificial neural stochastic network that can understand the probability distribution through its input sets. It was popular in the mid-2000s following the creation of fast-learning algorithms by Geoffrey Hinton and collaborators. The Recurrent Neural Networks (RNN), which is the computer of Boltzmann and the auto-encoder, helps to restore a stored pattern in a malformed version. In 1986, Michael Jordan developed Jordan networks, an early architecture for sequence analysis. LeCunet al. introduced first in its 1998 article the first CNN LeNet model, Gradient-based analysis used to classify a document. This was primarily used for OCR recording and character identification. In 1997, Mike Schuster and Kuldip Paliwal invented the Bidirectional Recurrent Neural Networks (BRNN). This works in both positive and negative time directions at the same time through a training network. In 1997, Hochreiter and Schmidhuber introduced the Long Short-Term Memory (LSTM). LSTM can learn how to cross the minimum time lags over 1000 discrete time steps by implementing the constant error flow of "constant error carousels" within special units. Multiplicative gate units learn how to open and close access to constant error flow. LSTM is local in space and time; its computational complexity is 0 (1) per step of time and weight. Neural capsule networks are recent deep learning inventions of Hinton. Highly popular deep learning today is caused by a drastic increase in the ability to process chips (e.g., GPU units), a significant reduction in computer cost, and recent advances in machine learning and signal / info processing. In new areas like medical imaging, profound sound learning, deep arts learning, computer hallucinations, robots, predictions, computer games, self-driving cars and big data deep learning applications should be investigated as shown in figure 2.1. Furthermore, there are few problems with deep learning, like automatic coloring, translation by automatic machine, automatic text generation, automatic manuscript, image recognition, automatic imaging captioning, publicity (data-driven predictive adverting, real-time RTB bidding, accurately targeted advertising for displays and cellphone advertising).

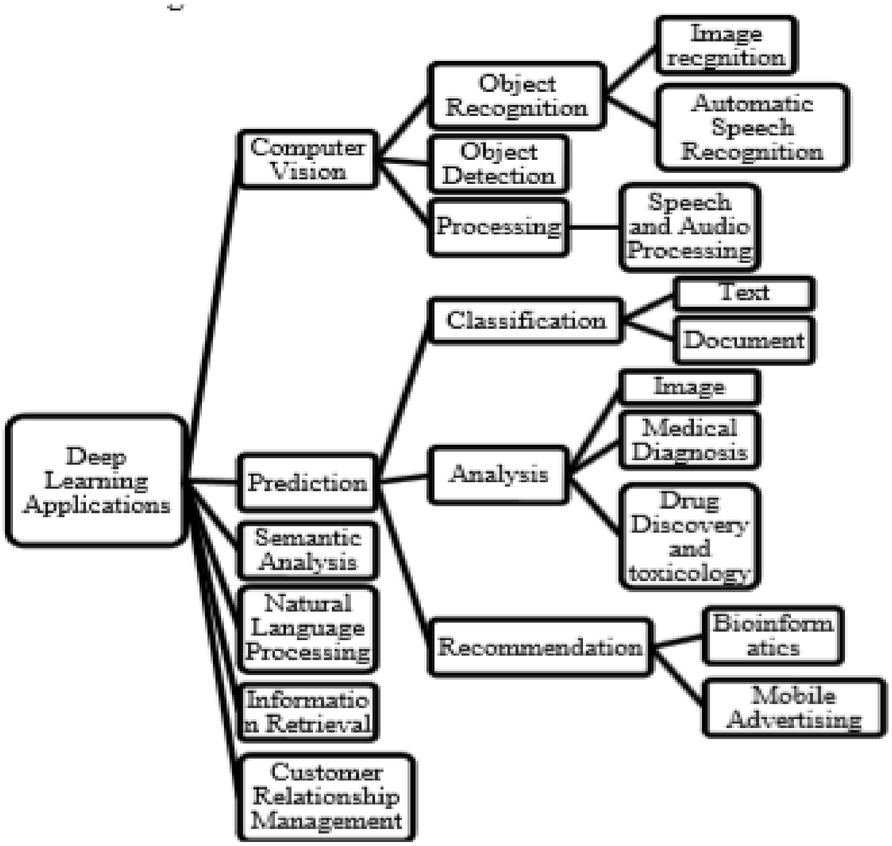


Figure 2.1: Deep learning applications

2.2 Convolutional Neuron Network

CNN is a neural network category which has been demonstrated to be very efficient in areas such as image recognition and graduation. The way neural biological networks are inspired collect knowledge about the human brain process; this helps to produce the ton of machine learning research such as speech recognition, computer vision and text processing. The following figure 2.2 illustrates the structure of the convolutionary neural networks, in that various there are layers including convolutionary layer, pooling layer, fully connected layer and soft max layer. The CNN achieves image recognition mainly. Here the image is transferred to the deep learning system. Various filters are implemented on each layer of CNN, and filtering can be performed for the various features according to different filters. Convolutionary layer functions are extracted from the image of the given input. That filter here has different features to correct prediction of class. To get an appropriate image resolution that reduces the pixels, we need to use a padding (zero padding). The next layer is the layer of pooling that was used to reduce parameters. The CNN performance is in the fully connected layer where the data from the other layers is compressed and passed to the soft-max level layer. The following sections define different models which fall within the system of the Convolution Neural Network.

2.3 Different models of CNN

2.3.1 AlexNet

The AlexNet dataset contains mainly 15 million annotated images of a total of 22000 categories. It employed the data enhancement techniques. The network is deep in 8 layers and allows images to be identified into 1000 styles such as the keyboard, mouse, pencil and several animals. This has led to the network learning rich feature representations for a wide range of images. The picture input element of the network is 227 by 227.

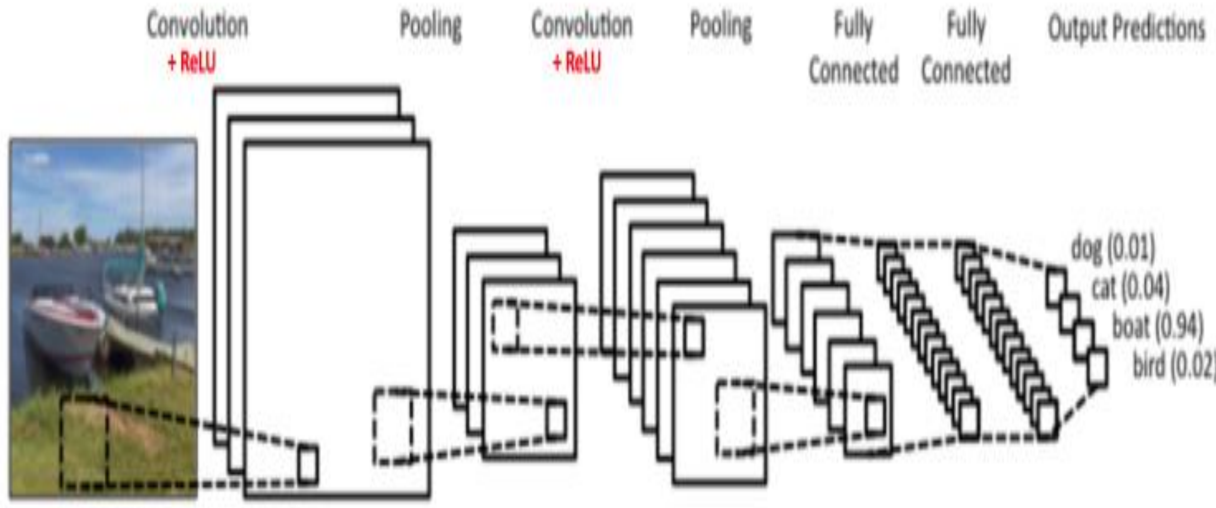


Figure 2.2: architecture of convolutional neuron network

2.3.2 VggNet

The VGG-16 is a neural network, which has trained over 1 million images from the ImageNet database. The network is thick in 16 layers and can recognize pictures in 1000 different styles, including buttons, cursors, crayons and various species. This has led to the network learning rich feature representations for a wide range of images. The image input size of the Network is 224 by 224.

2.3.3 Inception/GoogleNet

The start-up modules include "GoogleNet," essentially start-up architecture. It has more scope and a broad activation network to improvise the outcome slightly. It is therefore the most effective deep network framework for learning. The relation between the activations is insufficient, which is that all 512 output channels have no connection to all 512 output channels as input channels.

2.3.4 SegNet

This will identify the specific class such as car, tree and path, from the given input image. Two modules, encoder and decoder, are mostly used in SegNet. Encoder has 13 layers of Vgg16 conv that are not fully linked and are mainly useful for parameter reduction. In the sub-sampling method, maximal pooling of pixels is used to invariant translation over small spatial modifications in the image, combine it with sub-sampling and to monitor every pixel to a wider context. SegNet provides a better rating precision, however reduces the size of the chart that allows the picture to be shown negatively at blind borders. This makes the identification and storage of data possible with the decoder.

2.4 Compare of pre-trained networks

Pre-trained networks have several features to apply the problem when evaluating a network. Precision, speed and network size are the most important attributes of the network. The option of a network is a balance between efficiency, pace and network size. To make the predictions and to make the number of network parameters use the graph to compare the exactness of the ImageNet validation with that of operations. The plot links the networks on the exact speed limit and paints the networks at both the orange precision-speed and precision-size boundaries. On both frontiers are SqueezeNet, Google Net, ResNet-18, Inception-v3 and Inception-ResNet-v2, whereas VGG network is wider, slower and more accurate than other nets as shown in figure 2.3.

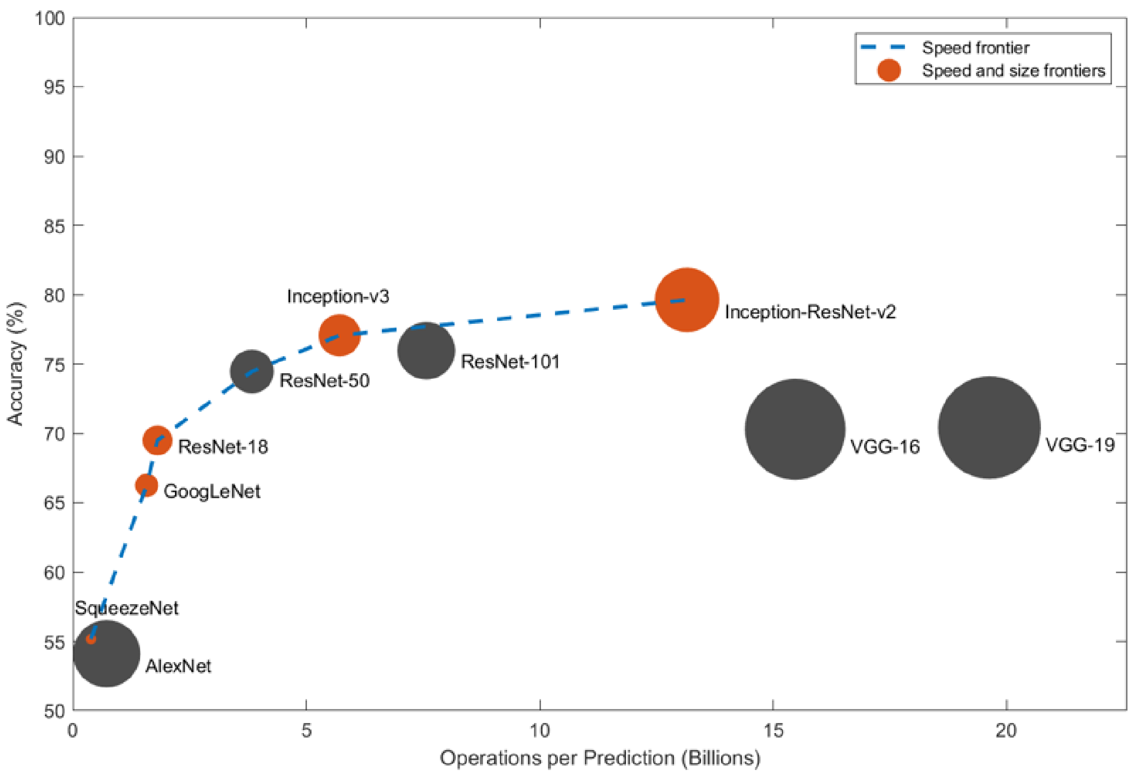


Figure 2.3: comparison of pre-trained networks

2.5 Usage of software

MATLAB® has been developed for scientists and engineers as a programming tool. The core of MATLAB is the MATLAB language, a matrix-based language which allows the most natural expression of computational mathematics. Data, algorithms of architecture, models and applications can be analyzed using MATLAB. The language, applications and integrated mathematical functions allow you to quickly explore a number of solutions to find a solution. A simple and easy language for the programming of a large library of mathematical features is the Matrix Laboratory (MATLAB). No declarations or dimensions such as C or JAVA languages are required for MATLAB. MATLAB contains live editor to create scripts which combines output, code and workspace. It contains thousands of built-in functions that can be used for wide variety applications. It also includes many toolboxes like Deep Learning Toolbox offers simple MATLAB® commands to build and interconnect profound neural network layers. MATLAB primarily originated as a programming of the matrix language. It enables the algorithm to be used to communicate with other programs written in other languages including C or C++.

It can be widely used in the areas such as

i. Audio, video, image and Signal Processing
ii. Wireless Communications.
iii. Test and Measurements.
iv. Machine Learning and artificial Intelligence.
v. In developing algorithms.
vi. Scientific and Engineering graphics.

As of 2018, it has nearly 2 millions of users over industries and academics. MATLAB performs operations very quickly and it uses language which is almost similar to C++.It is object oriented.

Advanced data structures, integrated editing and debugging tools and support for object-oriented programming make MATLAB an excellent tool. MATLAB can also call functions and subroutines that are written using C or FORTRAN, which highlights its nature as a multi-paradigm numerical computing environment. MATLAB mainly focuses on mathematical models and methods, hence the name. Unlike C and C++ MATLAB is a specialized language, not a general purpose language. MATLAB is an immersive platform, the heart of which is an array that needs no space. The graphical interface to workspace MATLAB is illustrated in Figure 2.4.

It has powerful commands which make the visualization of the result immediately after obtaining it.

These panels are on the desktop:

- Current Folder - Log in to your files.
- Command Window - Enter commands on command line, defined by the prompt (>>).
- Workspace - Explore data from files that you generate or import.

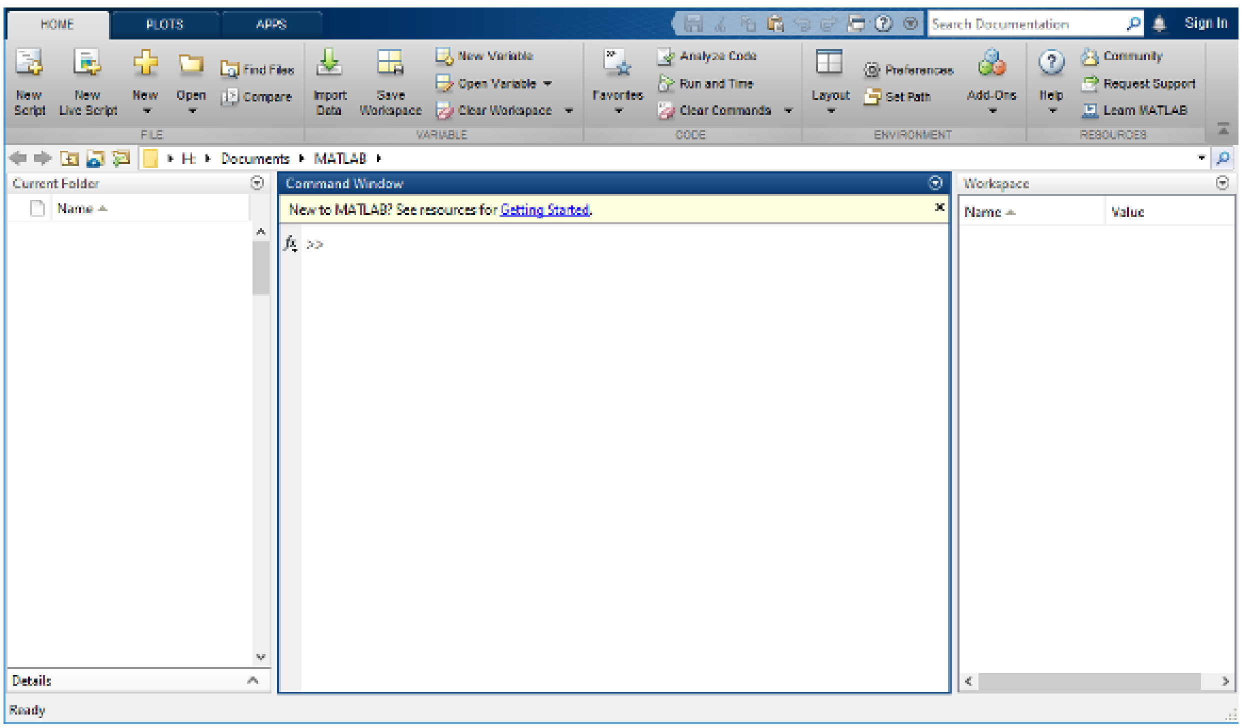


Figure 2.4: default layout of the MATLAB

Chapter 3

IMPLEMENTATION OF CNN

This chapter gives a detailed overview of the proposed framework. It deals with convolutionary network measures and the problem of pedestrian counting can be seen as a classification problem in which the model gives the likelihood of belonging to each class, where each class represents a different CNN number model resulting in a number of pedestrians using a 2D frame input.

3.1 Block diagram

Figure 3.1 shows the operations of the basic building blocks for each CNN.

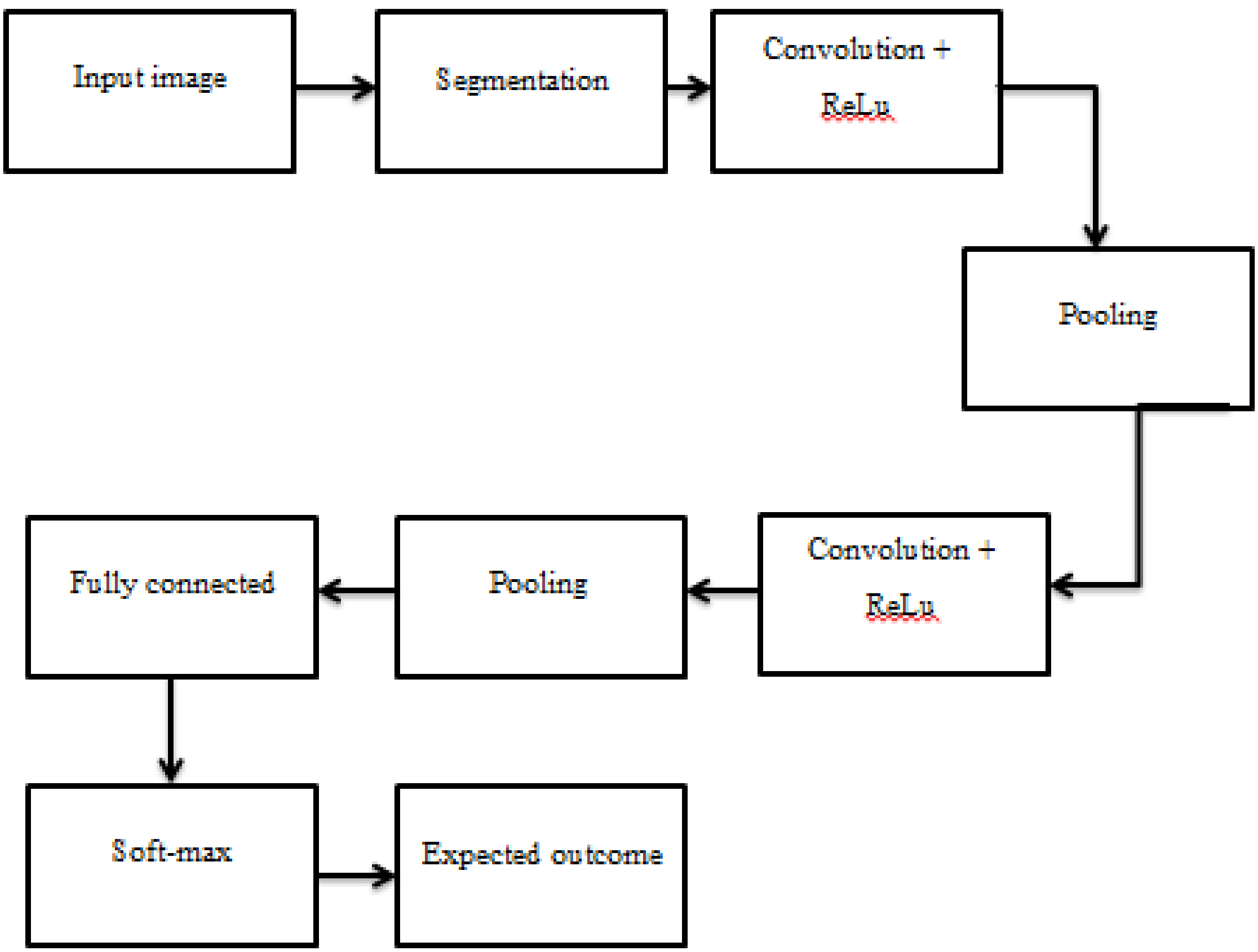


Figure 3.1: A simple CNN architecture

3.2 Expiation of Implementations

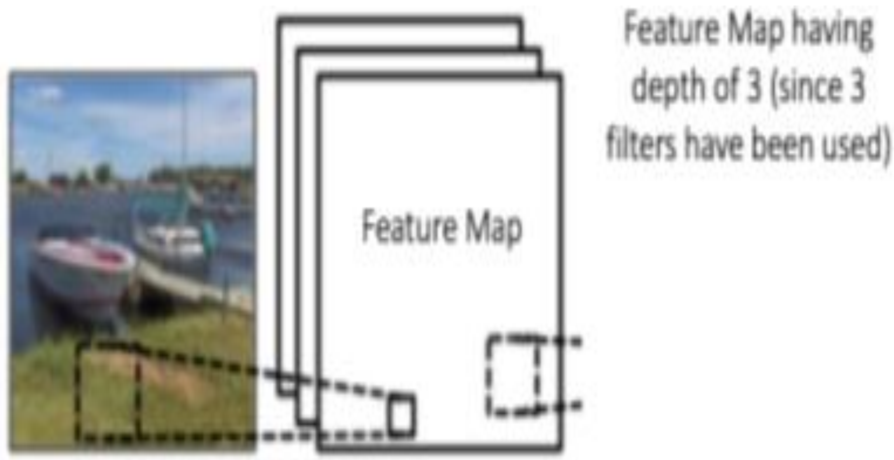
ConvNet comprises five main operations as shown in Figure 3.1

- Segmentation
- Convolution
- Non linearity (ReLu)
- Pooling or Sub Sampling
- Classification (Fully Connected Layer)

The above operations are describes as

Segmentation: This is usually one of the most important phases of the project. Segmentation includes some functions, first performs the detection of edges on a grayscale image. After identification of the edge, the image is transformed into binary data. The pixel values are divided into 0 and 1 wherever the edges are detected, which is 1 and the other part is 0. The binary format image is called imfill. After that the image will be carried out by the centroid, because in the binary image we will fill more of the contents of the image, the white pixel value will be increased so that, in order to perform the pedestrian detection, the centroid of the new enlarge component may not be accurately represented the pedestrian's location, So that we might lose some kind of detection. The code for this function is quite simple: for the whole binary image after the dilation, we set rows and columns to zero at certain intervals. As a result, the components in the image will be broken into a number of relatively small components. Windows around these component hubs, the windows will have a lot of overlapping. Thus, if the centroids are close enough, the windows corresponding to them are basically the same. After the chopping and non-maximum suppression step, the number of components increased and the influence of the large components ceased to exist, this can be demonstrated by the detection of the pedestrian on the right side of the image. Then we will mark the regions in the original image using rectangles and send these rectangular windows to the next module.

Convolution: CNN convolution’s primary goal is to remove the input image feature. Convolution maintains the spatial relationship between pixels by using small input data squares to learn image features as shown in figure 3.2.1.



3.2.1 Convolutional operation

Consider an image of 5 x 5 whose pixel values are only 0 and 1

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Consider another 3x3 matrix, too.

1	0	1
0	1	0
1	0	1

The configuration of the 5 x 5 image and the 3 x 3 matrix can then be computed as

4	3	4
2	4	3
2	3	4

The 3 x 3 matrix CNN term refers to the filter or kernel detector and is referred to as the transformed Element or the activation map for the matrix created by moving the filter over the image and calculating the point value. It is important to note that filters from the original input image act as a feature detector. The filter slides over the input image (Operation Convolution) to generate a function diagram. There are three parameters for the size of the feature map.

- Depth: Depth is the amount of filters we use in the process of convolution. Three 'depths' are given in the feature table.
- Stride: The number of pixels that we move over the input matrix is Stride. When the step is 1 then we move the filters one pixel at a time. If step 2 then the filters are moving around 2 pixels at a time. Smaller functional maps are produced in a larger step.
- Zero padding: Null padding is that it helps one to monitor the feature maps size. The use of zero-padding is often called broad convolution, because it would be a small convolution not to use zero-padding.

Non Linearity (ReLu): An additional procedure called ReLU was used following any Convolution cycle. ReLU is the linear rectified unit and is non-linear. Its output shall be OUTPUT = Max (zero, input). ReLu is applied to every pixel and replaces zero in the characteristic map for all negative pixel values.

Pooling or Subsampling: Pooling service as shown in figure 3.2.2. Spatial analysis reduces, while preserving the most significant details, the dimensory of each characteristic map. In this area, the largest element on the rectified feature map may be spatial pooling of various types:

Max, Average, Total etc. It was better that Max Pooling worked. Pooling makes it smaller and more manageable to display inputs.

Fully connected layer: In figure 3.2.3 is described the completely connected function. The completely connected layer is a standard multi-layer sensor, which uses the softmax activation function of the output layer. It means that each neuron of the previous layer is connected with each neuron in one of its nearest layers. The data from the convergence and pooling layers reflect high-level images. The goal of the Completely Linked Layer is to use this function to classify the image in the training dataset in different classes. The cumulative related output likelihood summation is 1. The Softmax function is expected to be used in the fully connected output layer of the network. The Softmax function takes a vector from a random real-world ranking and pulls it into a zero-to-one value matrix.

Feature extraction:

HoG's designer is a good descriptor for pedestrian sensing, and secondly, HoG can reduce its size and therefore reduce the computer complexity of our system, thus making our system even faster, instead of putting a complete window in the classifier. In each part of the picture, the HoG represents the gradient majority. Thus, the functional vector can represent the object shape in an image very effectively. The larger window size generally gives a larger HoG vector and reduces cross-entropy loss, but because of the small number of training data.

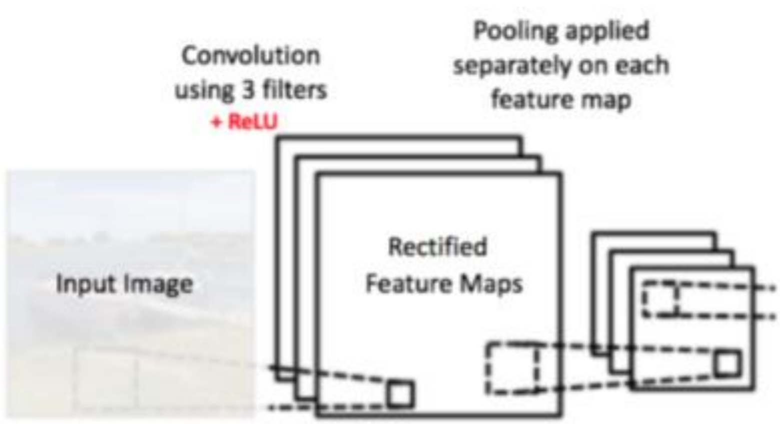


Figure 3.2.2 Pooling applied to Rectified Feature Maps

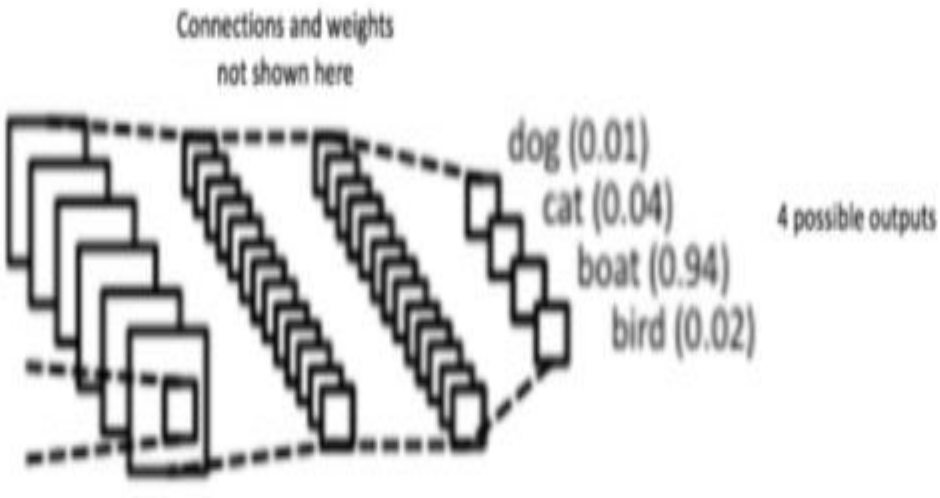


Figure 3.2.3 Fully Connected Layer -each node is connected to every other node in the adjacent layer

Chapter 4

SIMULATION RESULTS

This chapter gives a clear idea of the scenario considered for the simulation and performance analysis of the proposed work. It includes the result of every operation of the convolutionary neural network.

4.1Segmentation

The segmentation controls the grayscale image and the identification of the point. Figure 4.1.1 (a), (b) and (c) shows the grayscale images of the different input image and Figure 4.1.2 (a), (b) and (c) shows the detection of the edge of the image. Helps set the values to the foreground pixels and the background pixels.

Figure 4.1.1 (a) grayscale image of the low density pedestrian image

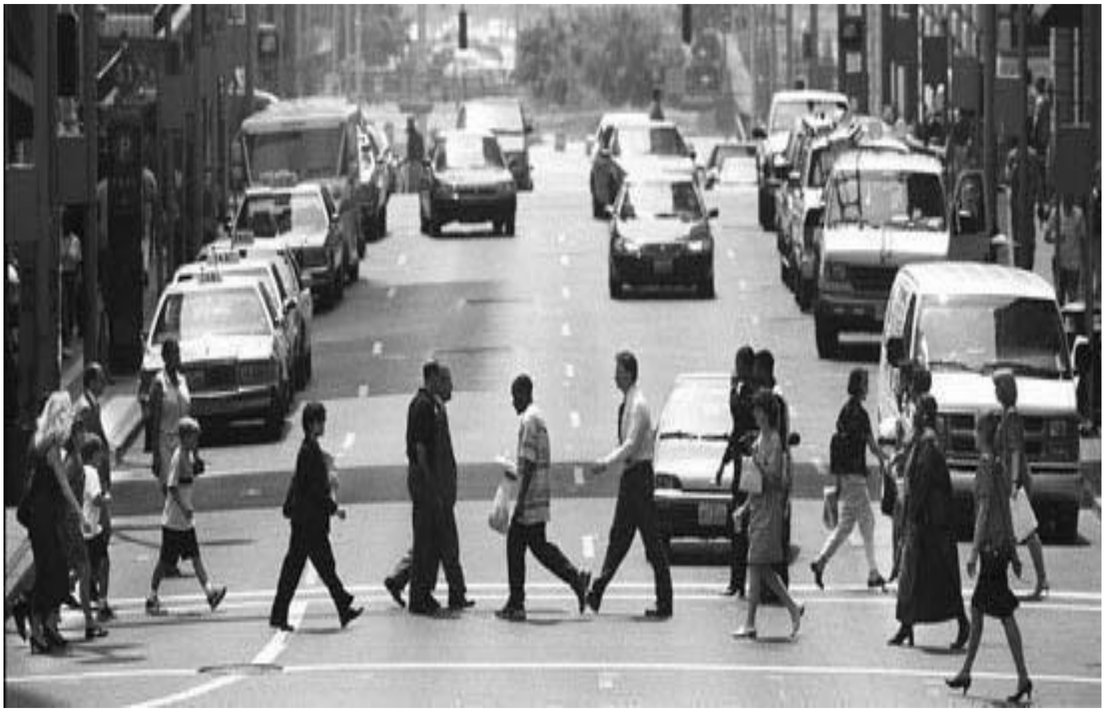


Figure 4.1.1 (b) grayscale image of the high density pedestrian image

Figure 4.1.1 (c) grayscale image of the accidental zone pedestrian image

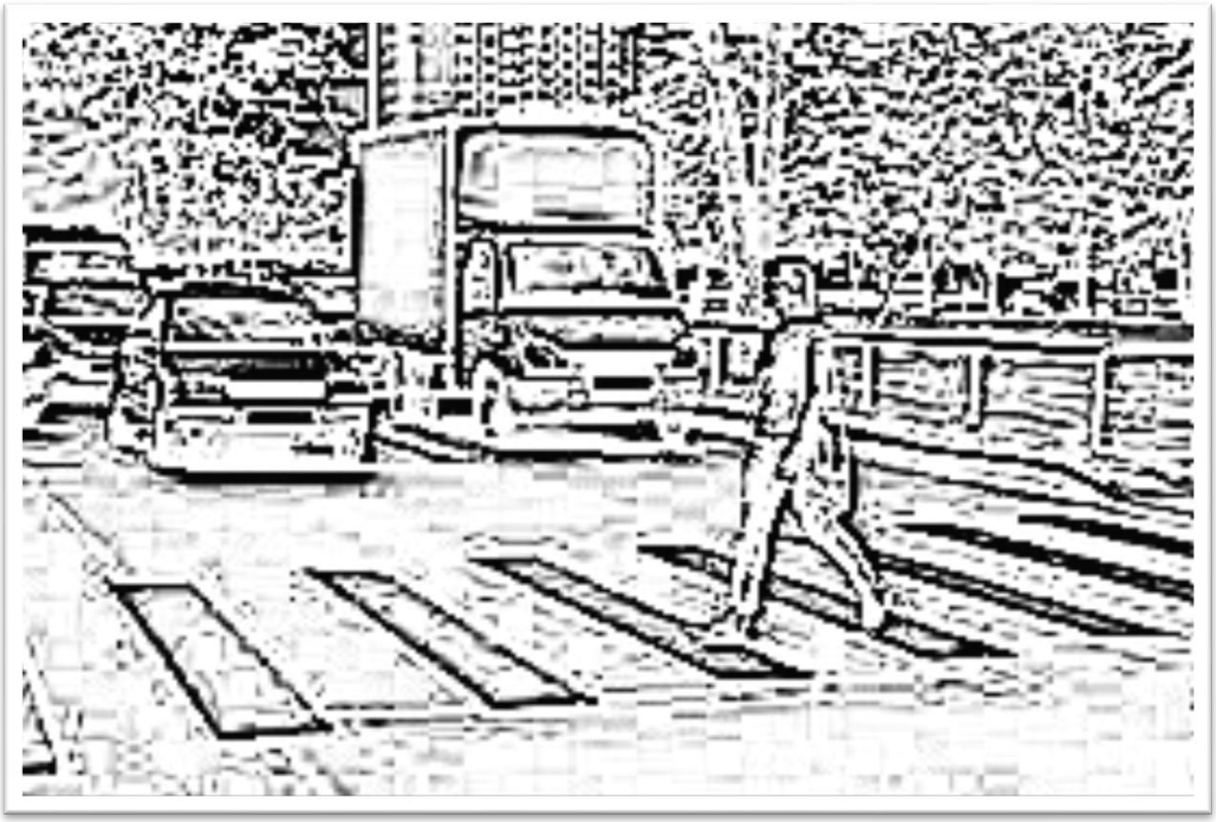


Figure 4.1.2 (a) the edge detection of grayscale image of low density pedestrian image

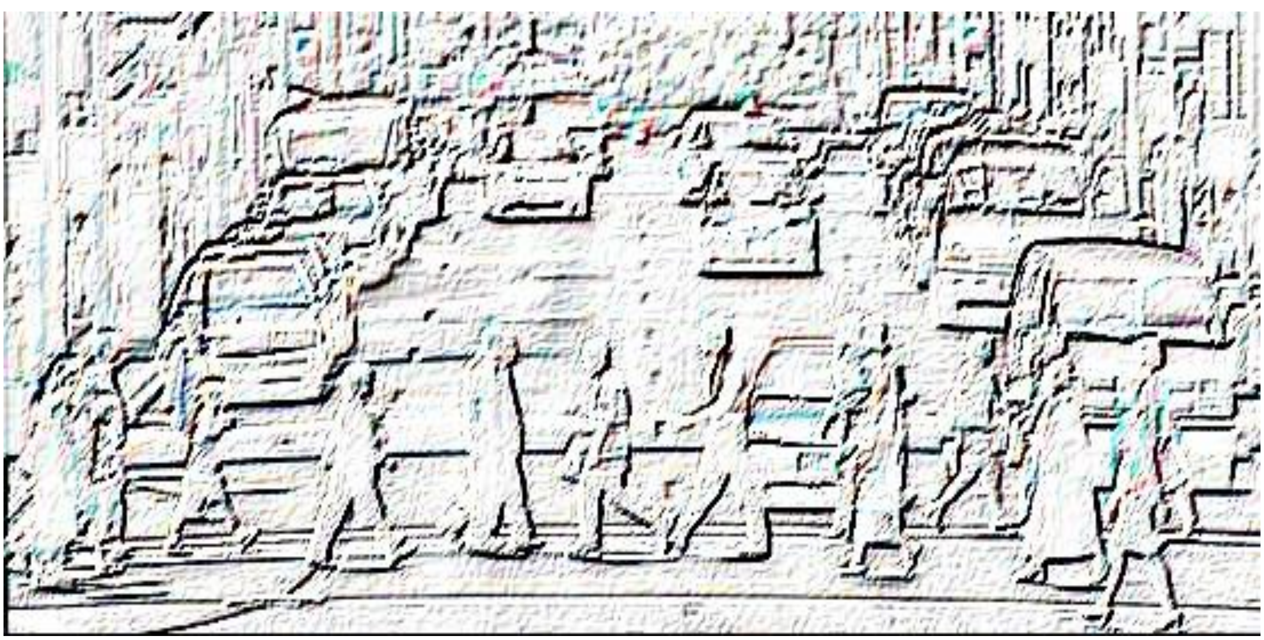


Figure 4.1.2 (b) the edge detection of grayscale image of high density pedestrian image

Figure 4.1.2 (c) the edge detection of grayscale image of accidental zone pedestrian image

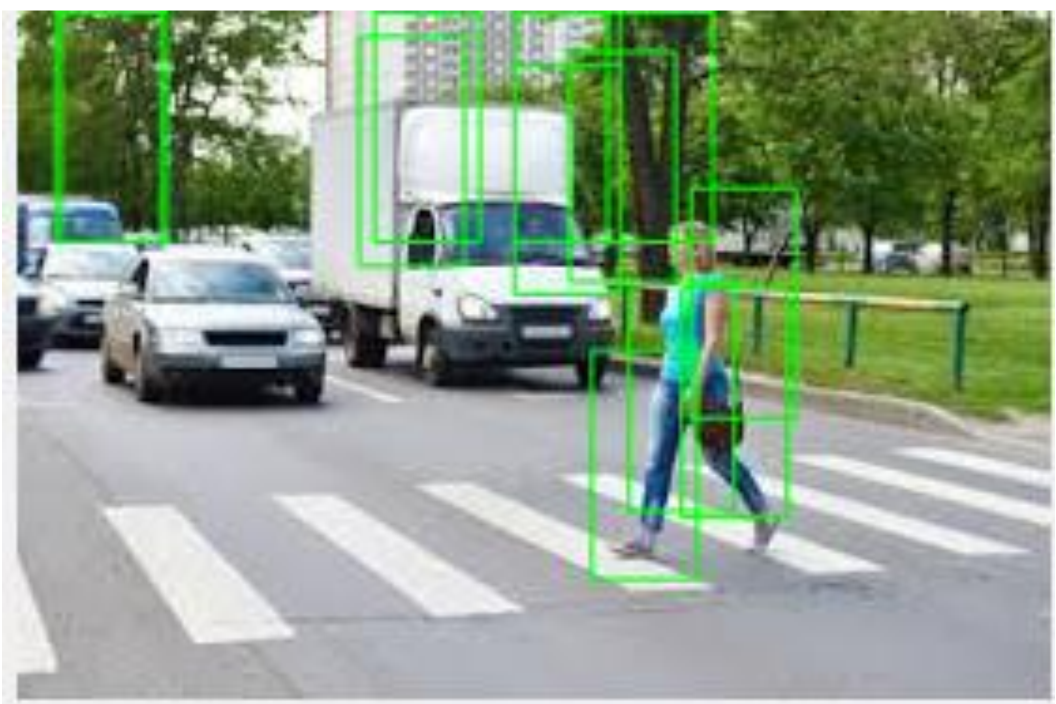


Figure 4.2.1 (a) result of pedestrian detection at first layer of ConvNet.

4.2 Convolutional neural networks

In this case the several layers of networks are executes like convolutional, pooling, fully connected and softmax. The procedure is followed until the pedestrian detected with number of different window size at each layer. Figure 4.2.1 (a), (b), (c), (d) with an accuracy of 75% the architecture of convNet deals with high speed simulation time.

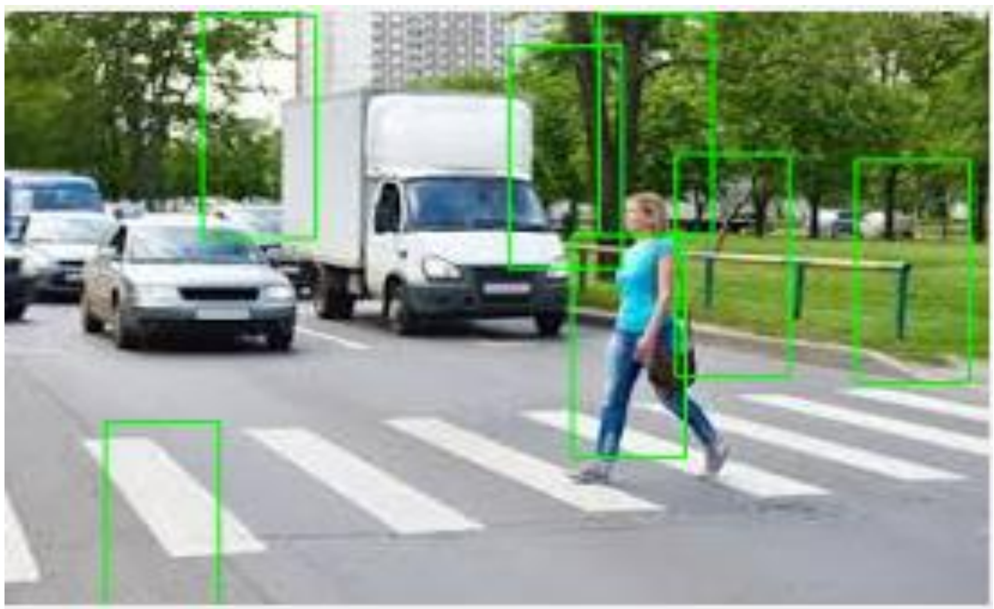


Figure 4.2.1 (b) result of low density pedestrian detection at the layer of ConvNet.

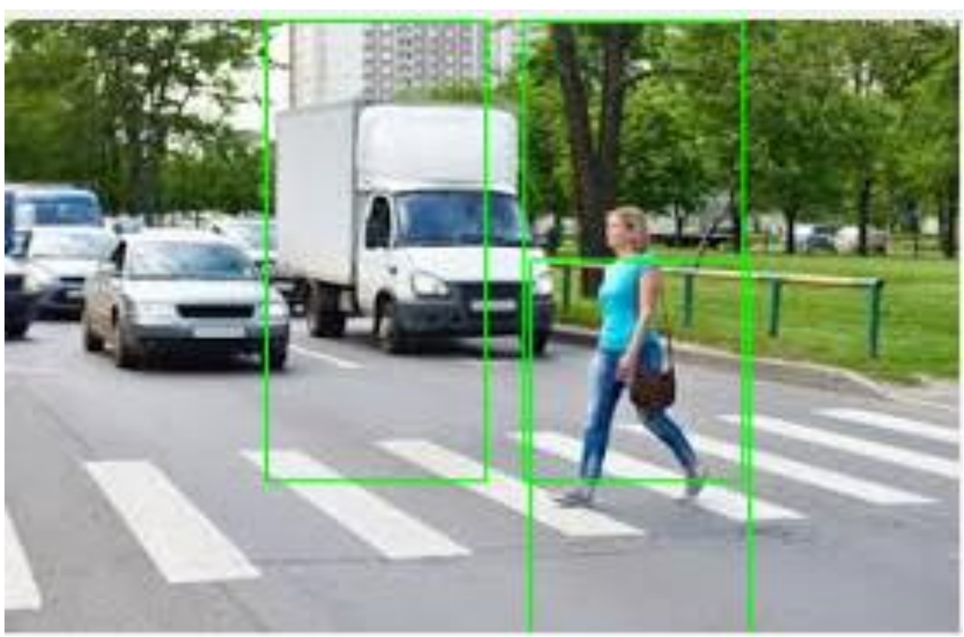


Figure 4.2.1 (c) result of low density pedestrian detection at another layer of ConvNet.

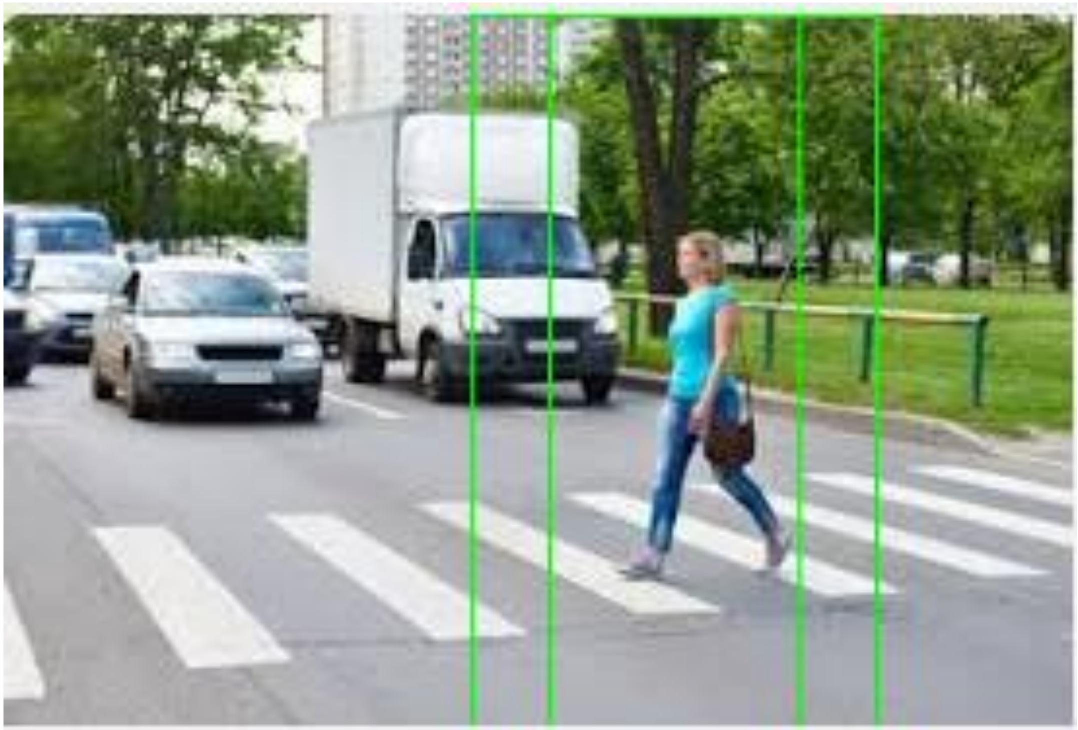


Figure 4.2.1 (d) result of low density pedestrian detection at last layer of ConvNet.

For the high density traffic areas pedestrian detection can be resulted as shown in Figure 4.2.2 (a) and (b) and in the accidental zone areas pedestrian detection can be shown in Figure 4.2.3 (a) and (b).

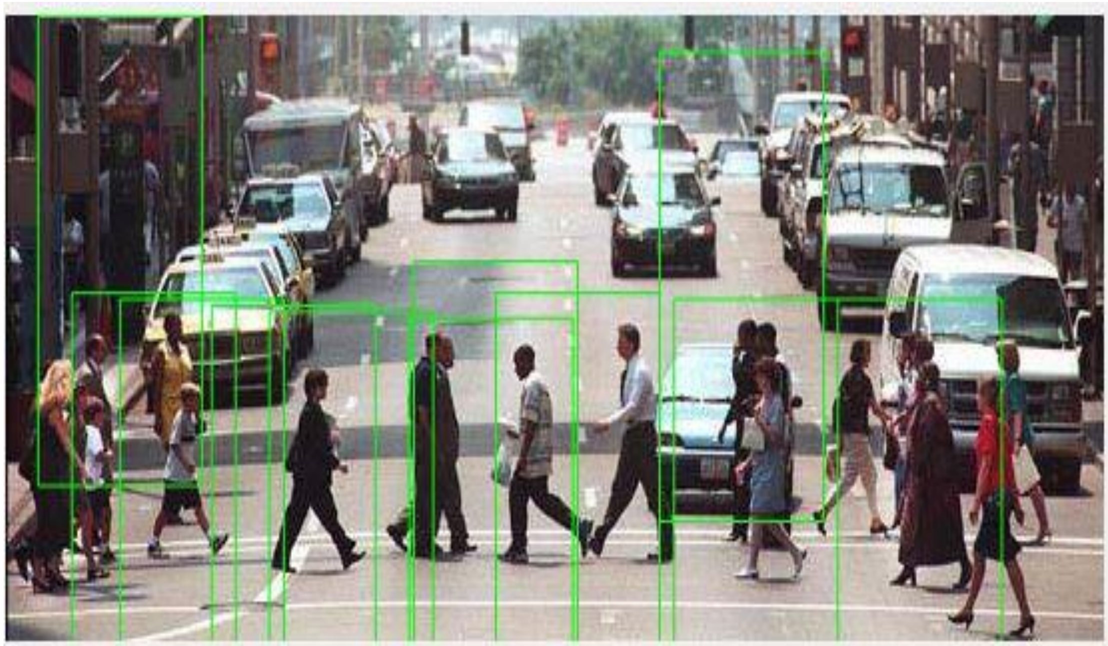


Figure 6.2.2 (a) result of high density pedestrian detection at the layer of ConvNet.

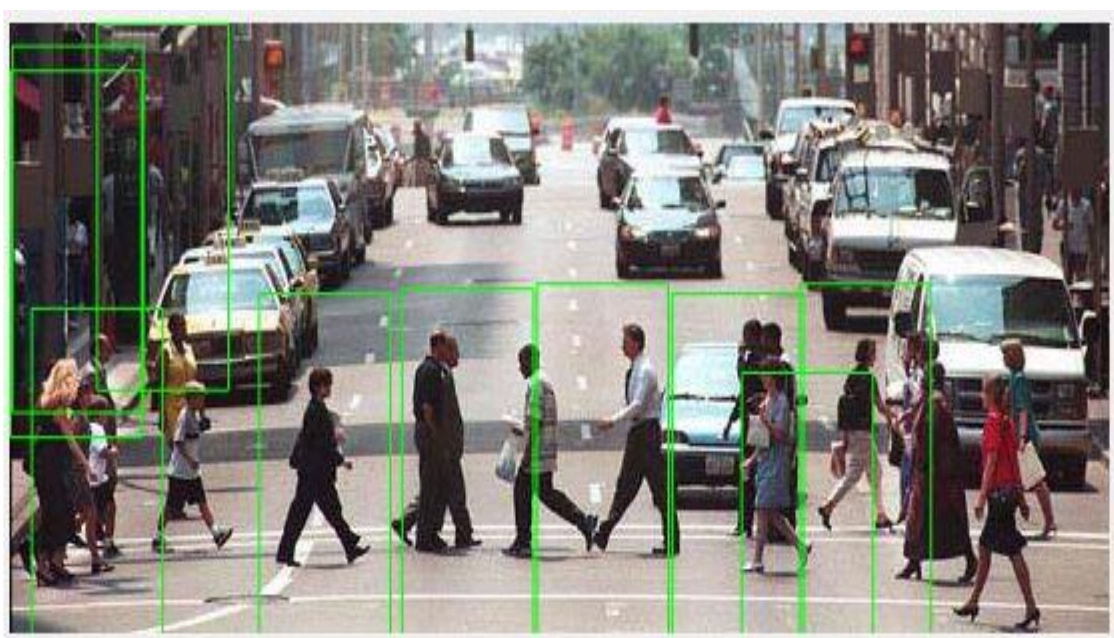


Figure 6.2.2 (b) result of low density pedestrian detection at last layer of ConvNet

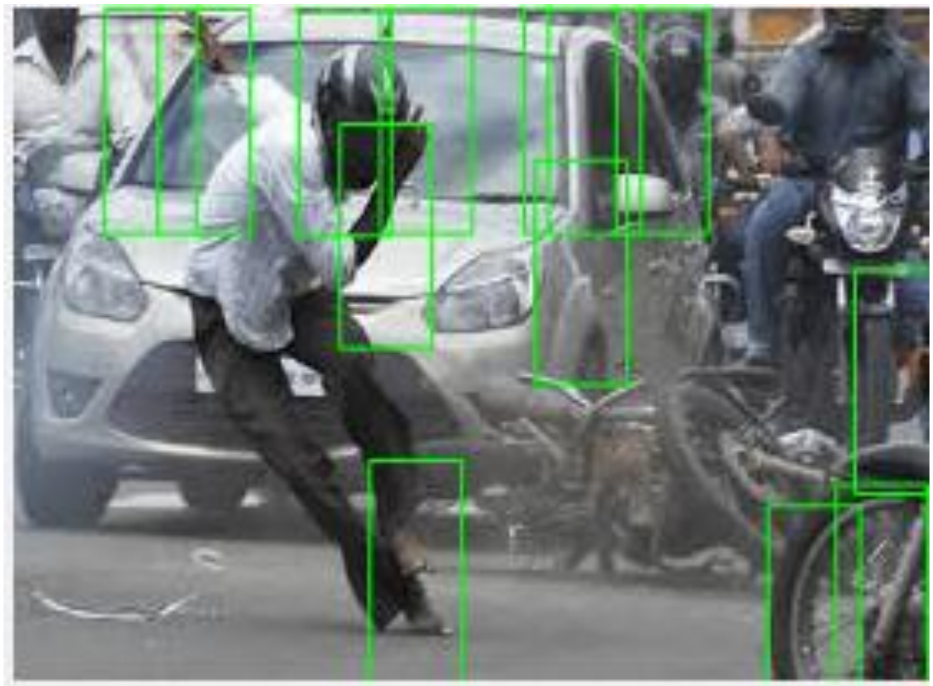


Figure 4.2.3 (a) result of accidental zone of pedestrian detection at layers of ConvNet

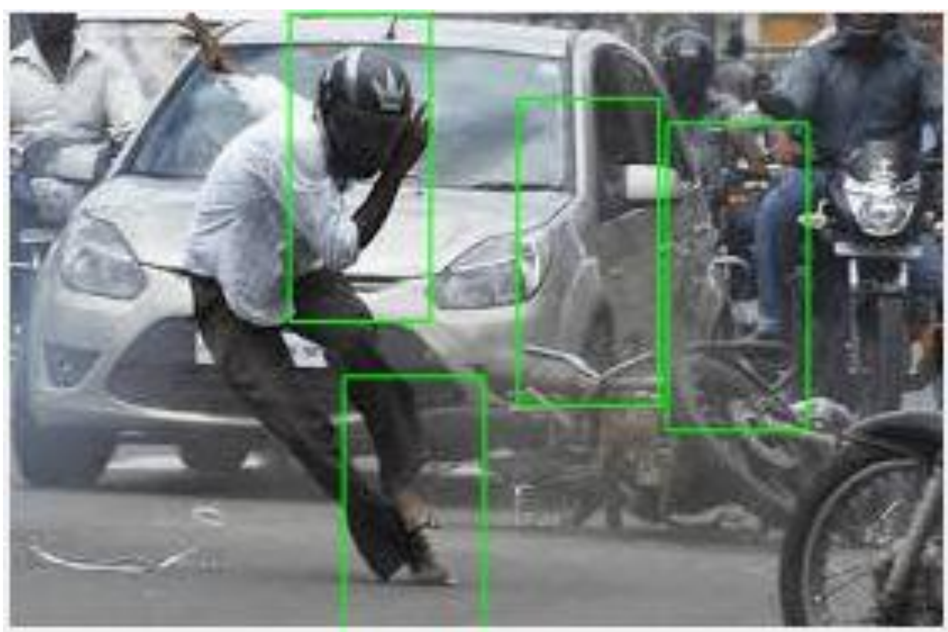


Figure 4.2.3 (b) result of accidental zone of pedestrian detection at last layer of ConvNet

Chapter 5

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

One of the most important lessons we have learned from this process is how to use the image information correctly for a vision system. For example, we can use gradient information in an image (i.e. borders) to extract possible pedestrians from an image with an accuracy of 75%, rather than naively sliding the window over an original color image, which can reduce complexity; to make use of the gradient information we opt to operate on binary images that is much simpler and quicker; instead, to make use of the gradient information. The methods we use for different modules alternate between the original image and the binary image based on the higher-level objective requirements. In other words, if you want simpler information, such as edges, it is enough to use only a grayscale or binary image, but if you want to determine whether a sub-image contains a human, you need to use the original colored image for more information.

7.2 Future scope

As can be seen from the experiment, the system can generally detect pedestrians precisely. One obvious problem with our system is that it has some unreasonably false positives. In this project, we designed a complete pedestrian detection system. During the design phase, many unseen problems were considered and resolved. There is a need for more research to reduce false negatives. Occlusion and, at the same time as passing through the road, the shape of the pedestrian was not clear. Efficiency must also be considered every time it is necessary to balance the strength.

References

[1] Sanjukta Ghosh, Peter Amon, Andreas Hutter, Andr´e Kaup,” Reliable Pedestrian Detection Using a Deep Neural Network Trained On Pedestrian Counts”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[2] Jae Kyu Park, Kyeong Seok Han, Seok Yong Yun,” Intensity Classification Background Model Based on the Tracing Scheme for Deep Learning Based CCTV Pedestrian Detection”, *IEEE 9th international conference*, 2018.

[3] V. Campmany, S. Silva, A. Espinosa, J.C. Moure, D. V´azquez, and A.M. L´opez,” GPU-based pedestrian detection for autonomous driving”, *The International Conference on Computational Science*, 2016.

[4] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, Yann LeCun,” Pedestrian Detection with Unsupervised Multi-Stage Feature Learning”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[5] Xiaoheng Jiang, Yanwei Pang, Xuelong Li, Jing Pan,” Speed up deep neural network based pedestrian detection by sharing features across multi-scale models”, Neurocomputing 185 (2016) 163–170.

[6] Zhaowei Cai, Mohammad Saberian, Nuno Vasconcelos,” Learning Complexity-Aware Cascades for Deep Pedestrian Detection”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[7] Ping Luo, Yonglong Tian, Xiaogang Wang, Xiaoou Tang,” Switchable Deep Network for Pedestrian Detection”, *IEEE International Conference on Image Processing*, 2014.

[8] Yonglong Tian, Ping Luo, Xiaogang Wang, Xiaoou Tang,” Pedestrian Detection aided by Deep Learning Semantic Tasks”, *IEEE International Conference on Image Processing*, 2015.

[9] Hailong Li, Zhendong Wu, Jianwu Zhang,” Pedestrian Detection Based on Deep Learning Model”, *9th International Congress on Image and Signal Processing*, 2016.

[10] Carlos Ismael Orozco, Mar´ıa Elena Buemi, Julio Jacobo Berlles, ”New Deep Convolutional Neural Network Architecture for Pedestrian Detection”, IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[11] Hyok Song, In Kyu Choi, Min Soo Ko, Jinwoo Bae, Sooyoung Kwak, Jisang Yoo,

"Vulnerable pedestrian detection and tracking using deep learning," 2015.

[12] Anelia Angelova, Alex Krizhevsky, Vincent Vanhoucke, Abhijit Ogale, Dave Ferguson, "Pedestrian detection in crowded scenes via scale and occlusion analysis" *electronic forms USA* 2015,

[13] Wanli Ouyang, Xiaogang Wang, "JointDeepLearningforPedestrianDetection" *IEEE International Conference on Computer Vision* (ICCV), Dec 2013.

[14] Hongquan Qu, Tongyang Yuan, Zhiyong Sheng, Yuan Zhang, "A Pedestrian detection method Based on YOLOv3 model and Image enhanced by Retinex," *11th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics*, 2018.

[15] Keke Shi, Yaping Zhu, Hong Pan, "A novel model based on deep learning for Pedestrian detection and Trajectory prediction" *IEEE 8th Joint International Information Technology and Artificial Intelligence Conference*, 2019.

[16] Seong Kyung Kwon, Eugin Hyuñ, Jin-Hee Lee, Jonghun Leẽ, Sang Hyuk Son, "WiP Abstract: A Low-Complexity Scheme for Partially Occluded Pedestrian Detection using LIDAR-RADAR Sensor Fusion" *IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications.* 2016

[17] Enjia Chen, Xianghong Tang, Bowen Fu, "A Modified Pedestrian Retrieval Method Based on Faster R-CNN with Integration of Pedestrian Detection and Re-identification" *IEEE,* 2018

[18] Jiao Zhang, Jiangjian Xiao, Chuanhong Zhou, Chengbin Peng," A Multi-Class Pedestrian Detection Network for Distorted Pedestrians" *IEEE,* 2018.

[19] MP Nkosi, GP Hancke, RMA dos Santos," Autonomous Pedestrian Detection" *IEEE,* 2015.

[20] Peilei Dong, Wenmin Wang, "Better Region Proposals for Pedestrian Detection with R-CNN", *IEEE VCIP*, 2016.

[21] Zhaowei Cai, Mohammad Saberian, NunoVasconcelos, "Learning Complexity-Aware Cascades for Deep Pedestrian Detection", *Computer version foundation.*

[22] S.P.Vimal, Bandla Ajay, P.K.Thiruvikiraman, "Context Pruned Histogram of Oriented Gradients for Pedestrian Detection", *IEEE*,2013.

[23] Koti Naga Renu Chebrolu and P.N. Kumar," Deep Learning based Pedestrian Detection

at all Light Conditions", *International Conference on Communication and Signal Processing*, April 4-6, 2019.

[24] Weicheng Sun, Songhao Zhu, Xuewen Ju, Dongsheng Wang, "Deep Learning Based on Pedestrian Detection", *IEEE*,2018.

[25] Zahid Ahmed, R. Iniyavan and Madhan Mohan P, "Enhanced Vulnerable Pedestrian Detection using Deep Learning", *International Conference on Communication and Signal Processing*, April 4-6, 2019.

[26] Jorg Wagner, Volker Fischer, Michael Herman, Sven Behnke, "Multispectral Pedestrian Detection using Deep Fusion Convolutional Neural Networks", *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium)*, 27-29 April 2016

[27] TianRui Liu, Tania Stathaki, "Fast Head-Shoulder Proposal for Deformable Part Model based Pedestrian Detection", *IEEE,* 2016.

[28] Ping Luo, Yonglong Tian, Xiaogang Wang, Xiaoou Tang, "Switchable Deep Network for Pedestrian Detection", *Computer vision foundation,* 2014.

[29] Seong Pyo Jeon, Yoon Suk Lee, Kwang Nam Choi, "Movement Direction-based Approaches for Pedestrian Detection in Road Scenes", *National Research Foundation of Korea,* 2010.

[30] Carlos Ismael Orozco, Mar´ıa Elena Buemi, Julio Jacobo Berlles, "New Deep Convolutional Neural Network Architecture for Pedestrian Detection",