Real time Object Detection, Tracking, Distance and Speed Estimation for self-driving cars based on Deep Learning using YOLO technique

Rashmi Kalinayakanahalli Anilkumar
10535943@mydbs.ie

Dissertation submitted in partial fulfillment of the requirements for the degree of
M.Sc. Data Analytics
at
Dublin Business School

Supervisor: Obinna Izima

August 2020

Declaration

I declare that this dissertation that I have submitted to Dublin Business School for the award of M.Sc. Data Analytics is the result of my own investigations, except where otherwise stated, where it is clearly acknowledged by references. Furthermore, this work has not been submitted for any other degree.

Signed: Rashmi Kalinayakanahalli Anilkumar
Student Number: 10535943
Date: 25 Aug 2020

Acknowledgment

Firstly, I would like to place on record my deep sense of gratitude to Dublin Business School for providing excellent Infrastructure and Academic Environment without which this work would not have been possible. I also wish to extend my thanks to Mr. Obinna Izima, my supervisor for his critical, insightful comments, guidance and impressive technical suggestions to improve the quality of this work.

Finally to all my friends and classmates who always stood by in difficult situations and helped me in some technical aspects. Last but not the least we wish to express deepest sense of gratitude to our parents who were a constant source of encouragement and stood by us as pillar of strength for completing this work and course successfully.

Abstract

In the modern age, the focus is more on self-driving cars for the betterment of technology. The precise detection of objects like vehicle, sign boards and obstacles can help the autonomous driving cars to drive safely. Detecting the real-time objects is always been a challenging task because of occlusion, scale, illumination etc. However many convolutional neural network models based on object detection were developed in recent years. But they can't be used for real-time object analysis because of slow speed of recognition. The model which is performing excellent currently is the unified object detection model which is You Only Look Once (YOLO). But in our experiment we have found that despite of having a very good detection precision, YOLO still has some limitations. YOLO processes every image separately even in a continuous video or frames. Because of this much important identification can be lost. After the object detection and tracking, speed and distance estimation is done. These are the two parameters which are used for self-driving cars. A number of companies are researching on this technology and it is expected to mature in the next decade or so.

Table of Contents

List of Figures

List of Equations

1 Introduction

1.1 Self-driving cars

Self-driving car or driverless car is a vehicle which operates with little or no human interactions or input which is capable of moving safely by sensing the environment. The Union of Concerned Scientists defines self-driving cars as cars that does not require human drivers for the safe operation of the vehicle. It is a combination of sensors and software which helps to drive the vehicle safely (LIU, 2017).

The aim of this work is the detection of objects for self-driving cars. Detection speed is the main factor to perform object detection in real-time (Joseph Redmon J. R., 2017). Our contribution aims on the development of perception system based object detection and tracking based deep learning with different approaches (Zhihao Chen, 2019). We have also found some parameters using YOLO model to help self-driving cars like speed estimation and distance calculation. Autonomous driving has been a promising industry in recent years. Both car manufactories and IT companies are competitively investing to self-driving field. Companies like Google, Uber, Ford, BMW have already been testing their self-driving vehicles on the road. Optical vision is an essential component of autonomous car (LIU, 2017).Detection and tracking algorithms are described by extracting the features of image and video for security applications. Features are extracted using CNN and deep learning. Classifiers are used for image classification and counting (Ayush Jain, 2018).

You Only Look Once (YOLO) is an Object Detection Algorithm. Yolo is a real time object detection algorithm. On a Pascal Titan X it processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev (pjreddie, 2016). When compared to classifier based methods, YOLO has several advantages. The predictions are made by the global context in an image because YOLO takes whole image for the testing. RCNN requires thousands of predictions for a single image. But in YOLO the predictions are made by the single neural network assessment. Because of this YOLO is tremendously fast. It is 1000 x times faster than RCNN and 100 x times faster than Fast RCNN (Farhadi, 2018).

1.2 Comparison of YOLO with other detection algorithms

In comparison to recognition algorithms, a detection algorithm does not only predict class labels but detects locations of objects as well. So, It not only classifies the image into a category, but it can also detect multiple Objects within an Image (Shahkaran, 2017).

It is extremely fast and accurate. In mAP measured at 0.5 IOU. It is on par with focal loss but about 4x faster. Moreover, you can easily tradeoff between speed and accuracy simply by changing the size of the model, no retraining required (pjreddie, 2016). And this Algorithm doesn't depend on multiple Neural networks. It applies a single Neural network to the Full Image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities (Ayush Jain, 2018).

1.3 Research Questions

1. To verify whether YOLO is good model for general object detection.
2. Is YOLO a potential object detection model for self-driving system or not?
3. Can YOLO be improvised to better fit in autonomous driving situation?
4. Whether YOLO can be applied to other tasks rather than object detection?

1.4 Hypothesis

When compared to other convolution neural network object detection models, YOLO works better with precision, accuracy, orientation estimation and detection speed.

2 Literature Review

2.1 Evolution of image recognition

Particularly in recent years, image processing has come a long way. Evolution can be majorly seen in technological fields like computer vision and software. First computer vision and study on images started in 1960s. Before this, image analysis was done manually. The major improvement in deep learning techniques and in image recognition technology took place in 2010. Now it is so advanced that we can write a program for supercomputers to train themselves (Alex Krizhevsky, 2015).

In early days, Feature extraction and classification paradigm was followed for object detection. Manually people need to define a specific feature which needs to be identified for extraction. After extraction of features, the objects or those features were represented in vector forms. These vector forms were used for training a model and for detecting an object while testing a model. It was a difficult task for detection of multiple objects since we have to find a general feature which can be found in multiple objects and can fit in different objects for training the model. The disadvantage is choosing the general feature which was a complex task and the detection accuracy was not that great (LIU, 2017).

In 2012, compared to other models which were already there, CNN gave a satisfactory results and good accuracy (Amy Jin, Standford). Though there was CNN model developed in 1990s, the accuracy was low due to improper training examples and fragile hardware. CNN model became strong when GPUs were prevailing. The CNN model built in 2012 was trained by the dataset which consists of 1.2 million images and 1000 categories. The experiment conducted by Krizhevsky proved CNN's powerful ability in images classification (Alex Krizhevsky, 2015). CNN methods can build feature filters while training the process which cannot be done in traditional methods. When compared to other models, CNN models are more friendly and have self-learning ability. (Shehan P Rajendra, 2019). Because of all these advantages, CNN became a major tool for image classification. To enhance the performance of CNN model, other regression heads were attached to the current model. This regression head is used to predict 4 coordinated after training it separately. Hence CNN allows both classification and regression head. While testing the model both classification and regression works simultaneously.

Classification predicts the class score and Regression helps for positioning. During 2012 to 2015, the experiments conducted were successful in attaching both classification and regression to CNN models Overfeat-Net, VGG-Net and ResNet. The error rate was reduced from 34% to 9% in these experiments (Amy Jin, Standford).

Since multiple object detection was failed in the experiments conducted in 2012, Researches in 2014 started conducting experiments to achieve the task of multiple object detection. In a single image more than five objects were to be detected. This can be done only when the system figures out object's class and location of the object. Usually deep convolution neural network works with the fixed size of image (e.g. 520 x 520). Because of this recognition accuracy might go low for the images and sub-images of arbitrary size or scale (Kaiming He X. Z., 2014). To overcome this issue Spatial Pyramid Pooling was introduced in 2014. Fixed length representation regardless image size or scale is achieved by developing a network structure called SPP-net. By removing the size or restriction, accuracy of the convolution neural network can be achieved. In SPP-net feature maps are computed only once to generate fixed length representations to train the detectors by pool features in the sub-images (Kaiming He X. Z., 2014). Repetition of computation of convolution features can be avoided by this method. This method was better than R-CNN and it gave satisfactory accuracy (Jifeng Dai, 2016).

Most of the ideas regarding CNN approach and classification came out in 2014. The main idea was to perform classification on every region that possibly contains objects. The region proposals and classification approaches achieved high accuracy and precision (Amy Jin, Standford) But these region proposals take a very long time to process which makes the speed of the entire system to go low. Because of this time consuming limitation, the region proposal approaches cannot be deployed in applications which are time critical like auto-driving, surveillance systems etc (Chadalawada, 2020).

Recently, YOLO (You Look Only Once) a unified object detection model was proposed by Joseph (Joseph Redmon A. F., 2016). Frame Detection in YOLO is considered as regression problem. It is a pre trained model which does not require a dataset to train the model. It consists of weights and object detection is done as boxes. The image which is inputted is regressed to tensor from the model directly which signifies the digit of every object's position and class score of the

object. The images which are inputted need not go through the YOLO network more than once. Because of this, processing of images is faster in this model.

When compared to other object detection models, Yolo has accomplished more than 50 times better accuracy. So currently YOLO is one of the best choices for real time object detections (Joseph Redmon J. R., 2017)

2.2 CNN based object detection

2.2.1 R-CNN

R-CNN stands for Region-based Convolution Neural Network. It combines region proposals with Convolution Neural Networks (CNN). R-CNN aids in focusing objects with deep neural network. It trains a model of high capacity with fewer amounts of annotated detection data. To categorize the object proposals deep convolution network is used and due to this R-CNN attains outstanding accuracy for object detection. Ability of R-CNN is high because numerous object classes can be scaled without resorting to estimated methods together with hashing (Chadalawada, 2020).

The researchers projected a multi stages purpose followed by classification. And classification was done using regions paradigm. The three main components of the developed system is feature vector extraction by CNN, classifier used which is Support Vector Machine and the last one is region proposal component (Jong-Min Jeongl, 2014) Feature vectors extracted from CNN are used to train the SVM classifier. Training is done on two datasets where CNN supervised is trained on one large dataset (ILSVRC) and one small dataset (PASCAL). During the testing time, the region proposal component used in this experiment is Selective Search. 2000 fixed size category independent regions which contain objects is produced by Selective Search (Jifeng Dai, 2016).SVM is used fot domain specific classification after a completely trained extractor of CNN converts every potential vectors into feature vectors. The two main problems that may arise are intersection-overunion (IOU) and duplicate detections. IOU will overlay the higher scoring region. These problems are eliminated by greedy non maximum suppression and refining the bounding box by using a linear regression model at the end (Kaiming He X. Z., 2015).

Satisfying accuracy for detection was accomplished by RCN when compared to any other detecting methods found in 2014. But RCNN also has many drawbacks because of complex multi-stage pipeline. The main role of CNN is to act as a classifier.

The region prediction is totally depended on exterior region proposal methods. This slows down the whole system while both training and detecting objects. Since RCNN has a separate training manner for every component which results in CNN, It is very difficult for optimization. Besides, CNN cannot be updated during the training of SVM classifier (Jifeng Dai, 2016).

2.2.2 SSD

Single deep neural network is used for detecting the objects in images by Single Shot Detector (SSD). The output spaces of bounding boxes are varied in SSD method. These boxes are set of default boxes over different aspect ratios. The approach is scaled to every feature map location after it varies. The predictions from multiple feature maps are combined in Single shot detector. Multiple feature maps are combined to handle objects of different sizes naturally (Amy Jin, Standford).

Some of the benefits of SSD are SSD totally removes the proposal generation. The following pixels or feature resampling stages are also eliminated which encapsulates every computation in a single network. Training in SSD is easy when compared to other models and it is forthright to assimilate into systems which needs a detection component. SSD accuracy can be increased by adding an additional method for object proposals. Since it is combined with other models, the training and inference is much faster. (Cr´eput, 2019)

2.2.3 R-FCNN

R-FCNN stands for region-based, fully convolutional networks. It is a simple framework used for efficient and accurate object detection. The other region based network detectors like F-CNN and Faster RCNN (Shaoqing Ren K. H., 2011), are based on per region sub network. But R-FCNN is entirely convolutional with every computation shared on the whole image. There is a predicament between image classification and object detection. Image classification has translation invariance issues and object detection has translation variance issues. To overcome this issue positive sensitive score maps are proposed. Thus, region based fully conventional

network can accept fully conventional image classifier like latest residual networks for detection of object (Jifeng Dai, 2016). PASCAL VOC datasets are used to show the modest results. ResNet with 101-layer is used. The results achieved by RFCNN are 20x better and faster than faster RCNN while both inference and training (Girshick, 2012).

2.2.4 Faster R-CNN

Faster region based convolution neural network is similar to RCNN which is an object detection algorithm. The features are extracted from the input image through convolution layers. Region proposal network (RPN) is used in Faster RCNN which shares the convolution features for each spatial location like objectness classification and bounding box regressor (Kaiming He X. Z., 2015). The F-RCNN network is cost effective than RCNN. It basically predicts the object boundaries and objectness scores for every position of the object. High quality region proposals are created and end-to-end training is done then this technique is used by Fast RCNN method for object detection (LIU, 2017).

When compared to other object detection methods, faster region based convolution neural network have less running time for detection of object. When feature maps are sent into RPN, feature maps projected region proposals are extracted. RoI pooling is done on feature maps. The end result of Faster RCNN classification will be multiclass classification and bounding box regressor for each RoI (Shaoqing Ren, 2011).

2.2.5 Fast R-CNN

Fast RCNN stands for Fast Region Based Convolution network. It is a training algorithm for detection of objects. Fast RCNN is better than RCNN and SPP net as it resolves almost all disadvantages and increases the speed and accuracy of RCNN and SPP net (Girshick, 2012). When compared to RCNN and SPP net, Fast RCNN has higher detection quality that is mAP. Training in Fast RCNN is done in single stage by means of multi-task loss. All the network layers can be updated during the training process. Disk storage is not utilized for feature caching by fast RCNN. The Convolution feature map is of Deep Convolution Network and RoI projection. The RoI pooling layer is extracted from the convolution feature map in RoI feature vetor. RoI feature vector is extracted for each RoI.The output will be softmax and bbox regressor (Juraj Ciberlin, 2019).

This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection (Shaoqing Ren K. H., 2011). By using the work of algorithms which are built previously, fast RCNN uses deep convolution network to classify object proposals efficiently. This helps Fast RCNN to achieve better detection accuracy and increase training and test speed.

The training done by fast RCNN on deep VGG16 network is 9x faster than RCNN and 213 x faster when compared to the test time. A good mAP on PASCAL VOC 2012 is achieved.

When Fast RCNN is equated with SPPnet, test accuracy is 10x faster and accurate and training of VGG16 is 3x times faster than SPPnet. Because of the detailed work carried out in this experiment, new insights are provided. The improved detector quality is achieved at the end. The main issue with other object detection algorithms are they are too expensive in time analyze in the past (Kaiming He X. Z., 2014).

2.2.6 HOG

Histogram of oriented gradients (HOG) is used for the detection of objects in computer vision or in image processing techniques using feature descriptor. Histogram of oriented gradients techniques includes restriction parts of an image in the orientation of gradient like detection of a window, Region of Interest (ROI) etc. It is very simple, user friendly and it is easy to understand the working of histogram of oriented gradients. From the input image gradient vector and cell histogram is formed and from these two HOG image is formed with Histogram of oriented gradients features (LIU, 2017).

2.2.7 SPP-net

CNN models works only with the fixed size of input image like 520x520. Because of this the recognition accuracy will go low. To overcome the above mentioned issue Spatial Pyramid Pooling was equipped. The fixed length of symbolization irrespective of size or scale can be generated by a Spatial Pyramid Pooling (SPP-net) network structure. (Kaiming He X. Z., 2014) Object deformation can be achieved by Spatial Pyramid Pooling. When compared all CNN based methods, Spatial Pyramid Pooling is an improved structure. Feature maps for the whole image can be computed at once in Spatial Pyramid Pooling method.

Pool features in sub images of fixed length is also computed to train the detectors. In the other methods, convolution featured are repeatedly computed which can be overcome in Spatial Pyramid Pooling. SPP-net is more weighted in object detection. When compared to RCNN method, SPP-net is 30-170× faster and when both the models were tested on Pascal VOC 2007, SPP-net gave better accuracy than RCNN (LIU, 2017).

2.3 Real time objects detection and tracking

Moving object detection and tracking is presented in (Cr´eput, 2019). Intuitive graphic interphase is achieved by means of new algorithm during the extraction of Silhouette. For the fast detection following algorithms were combined, frame difference method, background subtraction method, Laplace filter and Canny edge detector. The multivition dataset is used for testing the sequence images. The better performance object tracking algorithm is proposed. The detection algorithms and basic operation techniques are integrated and graphic user interface is used to make the process simple and straight forward.

World is adapting to artificial intelligence from past few years with influence of deep learning. (Ayush Jain, 2018) has compared many object detection algorithms like Region-based Convolutional Neural Networks (RCNN), Faster RCNN, Single Shot Detector (SSD) and You Only Look Once (YOLO). And the result id faster RCNN and SSD gives better accuracy with Yolo. Efficient implementation and tracking is done by combining deep learning with SSD and mobile nets. SSD helps in detecting the object and tracking them in a video sequence. They achieved in enabling good security utility for enterprise and order. The model created can be deployed in drones, detect attacks and CCTV cameras in government offices, colleges, hospitals etc.

Distance and estimation of real time video is achieved in (Zhihao Chen, 2019). Combinations of two deep learning models are developed to achieve object detection and tracking. The algorithms are tested on both railway and environment. Monodepth algorithm is applied for the estimation of object distance. Stereo image dataset and monocular images are used to train the model. Testing of both the models is done on another two datasets. They are Cityscape and KITTI datasets. Pedestrian and vehicle behavior tracking is done by developing a new method based SSD. The new SSD algorithm is developed by the coordinates of the output bounding boxes of SSD algorithm.

The whole development is tested on the real time data and the main objective is to monitor the tracks of pedestrians and vehicles to make sure it does not lead to any dangerous situations. Real time video of Routen tramway is taken by embedded cameras (LIU, 2017).

2.3.1 YOLO model

YOLO (You Look Only Once) a unified object detection model was proposed (Joseph Redmon J. R., 2017). Frame Detection in YOLO is considered as regression problem. It is a pre trained model which does not require a dataset to train the model. It consists of weights and object detection is done as boxes. The image which is inputted is regressed to tensor from the model directly which signifies the digit of every object's position and class score of the object. The images which are inputted need not go through the YOLO network more than once. Because of this, processing of images is faster in this model. When compared to other object detection models, Yolo has accomplished more than 50 times better accuracy. So currently YOLO is one of the best choices for real time object detections (Farhadi, 2018).

The base YOLO model can process real time images up to 45 frames per second where as Fast YOLO processes can process nearly 155 frames per second. The base version is the smaller version of the network. The natural images can be generalized very well using this model. According to recent studies, YOLO is one of the fastest detecting model when compared t other CNN object detection models (Joseph Redmon A. F., 2016).

2.3.2 Kalman tracking

In recent decades real time object tracking has been applied in multiple areas like human computer interaction, security, surveillance, video communication etc. Object tracking is the process of locating one or multiple moving object in the scene during continuous time. Some of the challenges faced are, Initial moving object segmenting - The goal of segmentation is to simplify or change the representation of the image into something that is more minimal and easier to analyze (Jovanny Bedoya Guapacha, 2017). Rapid appearance changes are caused by image noise, illumination changes, non-rigid motion and different poses. Tracking the moving target is complex in background. When tracking an object in real world background can be quite complicated for various depths in the background which can interfere their tracking.

So Kalman filter was introduced which is also called as linear quadratic estimation. It is an algorithm which uses the series of observing measurements over time (Alex Krizhevsky, 2015).

There are two main parts that contributes in Kalman tracking. They are Prediction and correction. Prediction will predict the project current state and estimate the next state. If there is any mistake in prediction, it goes to correction. In correction, Kalman gain is computed.
The system state is updated after Kalman gain is found and error covariance is also updated. Correction is in turn connected to prediction. The detecting range can be predicted by Kalman filter in order to accurately track object in occlusion which means a complicated background (Jovanny Bedoya Guapacha, 2017).

3 Background

This chapter gives the general review of basic neural networks. After this convolutional neural network is explained and one of the effective object detection model YOLO model and its working is explained in detail.

3.1 Neural network

Neural networks are the group of algorithms that are intended to identify patterns. Neural networks are modeled based on a human brain. The sensory data is perceived through the perception of machine. Neural networks also help in labeling and clustering of raw data or raw input. A simple neural network consists of 3 layers. They are: Input layer, Hidden layer and output layer. For every input layer there can be many hidden layers but for every hidden layer there is only one output layer (LIU, 2017).

Artificial neural networks or simulated neural network is the neural network which consists of artificial neurons. ANN is a unified set of natural or artificial neurons in which information is processed mathematically or computationally. Artificial network consists of neurons. Neuron is a mathematical function that works same as biological neuron. An average weight of the input and the total sum of the input is sent to a function which is nonlinear. And this function is referred as activation function same as sigmoid.

3.1.1 Structure of neural network

Basically neural network consists of three layers as shown in the figure below. All three layers consist of many neurons. The first layer is input layer, the middle layer is hidden layer and the last layer is output layer. All the neurons in each layer are connected and passes the information between them. The connection between the neural networks is called as weights. There is a connection between every neuron in the network hence neural network is a fully connected network. There connections can be measured by multiplying the neurons connected to adjacent layer. For example, if neuron in x layer is connected to neuron in y layer, then total connections will be $x \times y$.

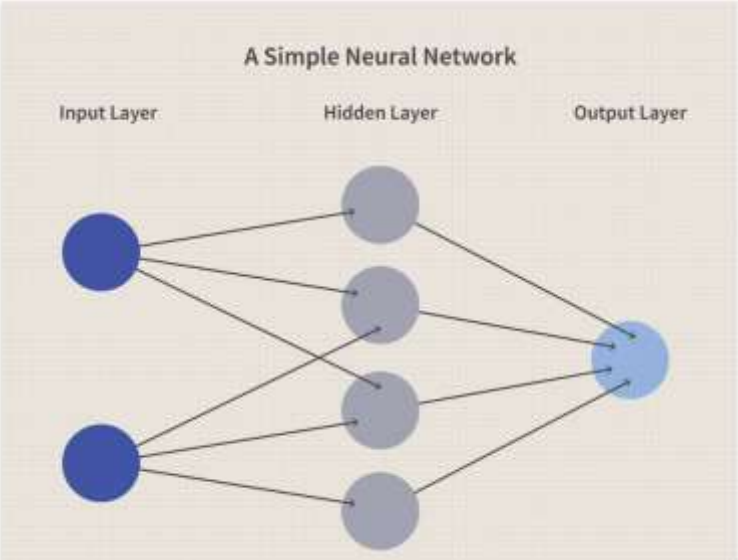

Figure 1: Structure of neural network

In neural networks, two layers can be connected to each other by making the outputs from one layer as the inputs to other layer. When the network starts working, input layer makes the simple decisions and send them to the hidden layer. According to the simple decision made by the input layer, hidden layer will make a better decision. Like this the information is passed to all the layers and hence the decisions will get better and better. It is a proven fact that, deep neural network can solve complex problems like in various fields like natural language processing, in biomedical fields, in speech recognition etc.

There are mainly 6 types of neural network which are being used currently in Machine Learning. They are:

- Feedforward Neural Network
- Radial basis function Neural Network
- Kohonen Self Organizing Neural Network
- Recurrent Neural Network(RNN) – Long Short Term Memory
- Convolutional Neural Network
- Modular Neural Network

3.1.2 Basics of neural network

In financial world, neural network helps in development of many processes like construction of proprietary indicators, forecasting the time series, security issues, trading of algorithms etc. Many layers of interconnected nodes are present in neural network. Every node acts as a multiple linear regression which is similar to perceptron. The signal produced by multi linear regression is converted into an activation function which is nonlinear.

If the perceptron's arranged in the interconnected layers then it is a multilayered perceptron (MLP). The input patterns are collected by input layers. The input signals are mapped to the output signal or the classification which output layer has. Hidden layer helps to fine-tune the weights of the input in of neural network. The marginal error is less in neural networks. Hidden layer has ability to predict the important features of the data which is inputted that have an extrapolate influence of the outputs. This is an example of feature extraction which helps to achieve a utility like principal component analysis which is a statistical technique.

3.2 Convolution Neural Network

Convolution neural network (CNN) is a deep learning algorithm. It has an ability to take an image as input, learn the weights and bias of the object or the image which is inputted and helps to identify or differentiate the inputted images from one another (Alex Krizhevsky, 2015). When compared to other classification algorithms, CNN does not require much of preprocessing. Convolution neural network has the ability to learn all characteristics if given an enough training. CNN is inspired by the animal visual cortex. The design of the CNN is similar to the arrangement of neurons in the human brain.

3.3 Partial Connected Manner

Partial Connected Manner stimulated by animal visual cortex. Every neuron responds to the stimuli. The response is given by neuron only in the restricted area of the field which is visible and this area is called as receptive field. Many receptive fields intersect to cover the whole visual field. The neurons in every layer in convolution neural network are arranged in two-dimensional array arrangement. Every neuron is connected to the previous layer in m x n fashion known as feature filter. Most of the time feature filter will be square in size.

When the state is active, feature map is generated. This action is performed when feature filter glides on to the input layer and convolution operation is performed. The layer in which convolution operation is performed is called as convolution layer. The networks related to this layer are known as convolution neural networks. In the input layer feature filter hunts for the exact patterns. The hunting process helps to understand the patterns more precisely during the training state. In testing phase, hunting process is done to check whether the pattern exists. Every neuron in the feature filter has the ability to share one feature filter. Therefore every feature filter represents one type of pattern. Due to this sharing feature weights action, the number of parameters used to name the connection between two layers are greatly reduced. Usually many feature filters are used to learn different patterns.

3.4 Pooling

Pooling layer and convolution layer are almost alike. The pooling layer helps to reduce the special size of the convoluted features. When spatial size is decreased, computational power is also decreased to process the data in dimensionality reduction. It also helps in extraction of dominant features. Since the dominant features are rotational invariant and positional invariant, it helps the process of training the model effectively. Pooling operation takes place after every convolution operation. It helps to abridge the information processed. Pooling process simply helps in compressing the earlier feature maps to summarized feature maps.

The commonly used pooling methods are Average pooling and Max pooling. Max pooling outputs the highest value and Average pooling returns the average of all the values. These values are based on the portion of the image enclosed by the kernel. When compared to both pooling methods, Max pooling is common method used in most of the CNN models.

Max pooling has the ability to suppress the noise. It eliminates all the noise together and de-noising is performed. It also helps to reduce the dimensionality reduction whereas the average pooling just helps in dimensionality reduction. Since max pooling performs lot better, it is used in most of the CNN models.

3.5 CNN for Image Classification

As mentioned earlier, CNN is used for image classification because of its high accuracy. The three layers in the neural network help to recognize the object. The first layer makes the simple designs. The middle layers helps to improve the designs built by the previous layer. The last layer makes the cultured design. Similar to this, CNN has an accumulative process.

But instead of designs creation, the layers help to recognize the patterns. The beginning layers recognize only few colors and edges. The next layers are little improved and recognize basic patterns like corners, or basic colors. The last layers have an ability to recognize the parts of the image or object like eyes. CNN gave a satisfactory results and good accuracy. Though there was a model developed in earlier, the accuracy was low due to improper training examples and fragile hardware. CNN model became strong when GPUs were prevailing. The CNN model built recently was trained by the dataset which consists of 1.2 million images and 1000 categories (Amy Jin, Standford). The experiment conducted by researchers proved CNN's powerful ability in images classification. CNN methods can build feature filters while training the process which cannot be done in traditional methods. When compared to other models, CNN models are friendlier and have self-learning ability. Because of all these advantages, CNN became a major tool for image classification. To enhance the performance of CNN model, other regression heads were attached to the current model. This regression head is used to predict 4 coordinated after training it separately. Hence CNN allows both classification and regression head. While testing the model both classification and regression works simultaneously.

3.6 General Object Detection Model

The region proposal component shadowed by CNN classifier is similar to the CNN model based on object detection. In region proposal methods many candidate regions are produced where every candidate region has at least one kind of object. And to do classification, all regions will pass through convolution neural network. The main reason to design this model is to translate multiple object detection problems to single object classification problem. When compared to classification part, region proposals are slow and this is the blockage of the whole system. Feature extraction and classification paradigm was followed for object detection. Manually people need to define a specific feature which needs to be identified for extraction. After extraction of features, the objects or those features were represented in vector forms.

These vector forms were used for training a model and for detecting an object while testing a model. It was a difficult task for detection of multiple objects since we have to find a general feature which can be found in multiple objects and can fit in different objects for training the model. The disadvantage is choosing the general feature which was a complex task and the detection accuracy was not that great. The main disadvantage of this system is the less accuracy (LIU, 2017).

3.7 Unified Detection model – YOLO

3.7.1 How it works:

Earlier detection models repurpose the classifiers to achieve detection. The model is applied to many location and scales in the image. If there is a maximum scoring regions on the image, then it is detected. But YOLO has an entirely dissimilar technique. A single neural network is applied to a whole image. In Yolo model, network spits the image into regions. After the splitting, bounding boxes and probabilities for each region is predicted. The predicted probabilities help to weigh the bounding boxes (Shahkaran, 2017) .

When compared to classifier based methods, YOLO has several advantages. The predictions are made by the global context in an image because Yolo takes whole image for the testing. RCNN requires thousands of predictions for a single image. But in YOLO the predictions are made by the single neural network assessment. Because of this yolo is tremendously fast. It is 1000 x times faster than RCNN and 100 x times faster than Fast RCNN (Farhadi, 2018).

3.7.2 Some of the dependencies required to build a YOLO in Tensorflow:

- Tensorflow (GPU version preferred for Deep Learning)
- NumPy (for Numeric Computation)
- Pillow/PIL (for Image Processing)
- IPython (for displaying images)
- Glob (for finding pathname of all the files)

Anaconda is suggested as it contains many libraries of machine learning and deep learning and interaction with spyder is easier.

3.8 Model Tuning and Hyperparameters

3.8.1 Batch Normalization

As said before features extraction is done and batch normalization helps in preprocessing these extracted features. Features should be normalized before sending to the next layers in the network. The input layer is normalizes by altering and scaling the initiations.

Normalization is done to increase the rate of learning. Like, if we have two features ranging from 0 to 1and 1 to 500, normalization is required for speed learning. If we do not normalize the features, the neural network accepts any features ranging from 1 to 500 as its highest priority in the dependencies of features. Because of this, every layer learns more about other layers not depending on anything. Every layer in YOLO undergoes the normalization technique. The main advantage of normalization is, it decreases the variance and total variance between units and trains the model faster (Joseph Redmon A. F., 2016).

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

Figure 2: Batch Normalization

Source: (Shahkaran, 2017)

3.8.2 Leaky ReLU

ReLU stands for Rectified Linear Unit. It is an initiation function developed in neural network. The advance version ReLU is leaky ReLU. If we assume that the output of the ReLU is 0, then gradient of it will also be 0. Only if there is a large negative bias in ReLU, the output of it can be 0. The error signals are passed from one layer to another. These signals are multiplied my 0 and hence error signal will never pass to the next layers and ReLU has expired. This is when Leaky ReLU comes into picture.

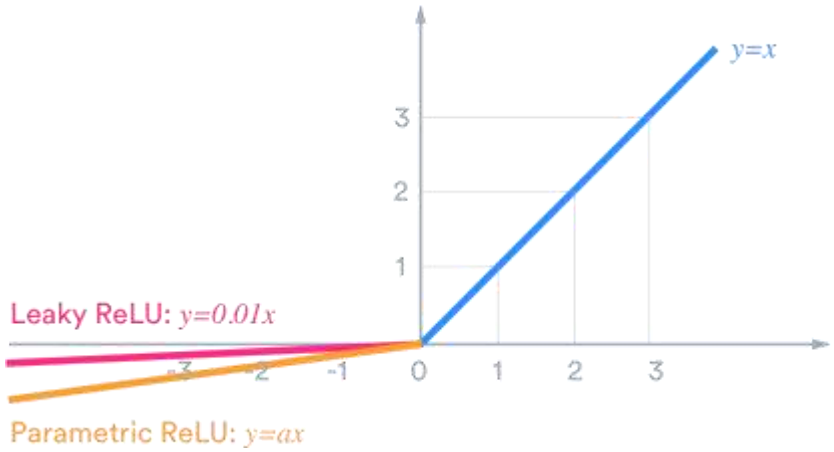

Figure 3: Leaky Relu

Source: (Shahkaran, 2017)

3.8.3 Anchors

One of the datasets used by YOLO is COCO dataset. Anchors are kind of bounding boxes that are calculated from the COCO dataset. Calculation is done by using k-means clustering. The width and height of the boxes are found through the centroid clustering. The sigmoid function is used to forecast the center coordinates of the box with respect to location of the filter application.

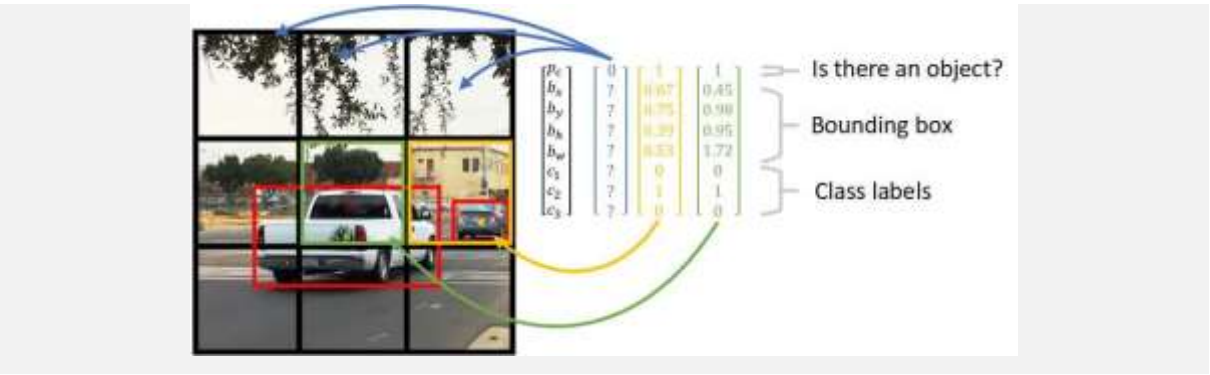

Figure 4: Anchor / Bounding Box illustration

Source: (pjreddie, 2016)

3.8.4 Implementation of Darknet-53 layers

In YOLO v3 paper, Researchers have explained about Darknet-53. Darknet-53 is a recent, deeper architecture of feature extractor. It consists of 53 convolution layers and hence the name Darknet-53. Every layer undergoes Leaky rectified linear unit activation function and batch normalization.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 5: Darknet-53

Source: (Joseph Redmon A. F., 2016)

3.8.5 Conversion of pre-trained COCO weights

YOLO is a detector's architecture. Training of YOLO model can be done in two ways. One can either use their own dataset to train the model or can use the pre-trained weights. These pre-trained weights are available for public use. The structure of the binary file used in YOLO can be explained as follows. The first 3 values that are int32 values contain the header information. The first 3 values are called as major version number, minor version number and subversion number. The int32 values are followed by int64 values. Int64 consists of the image count of the network while training the model. In float32, weights of all convolution layers and batch normalization layers are present. This binary file used is exactly opposite to tensorflow because binary file follows row major and tensorflow follows column major.

3.8.6 Reading the weights

Reading the weights starts from the first convolution layer. Every convolution layers are instantly tailed by batch normalization layer. The order in which weights are read are: read 4* number_filters weights of batch normalization layer: gamma, beta, moving mean and moving variance, then kernel_size[0] * kernel_size[1] * number_filters * input_channels weights of convolution layer. In some cases the convolution layer is not followed by the batch normalization. When this happens we should read number_filters weights instead of batch normalization parameters.

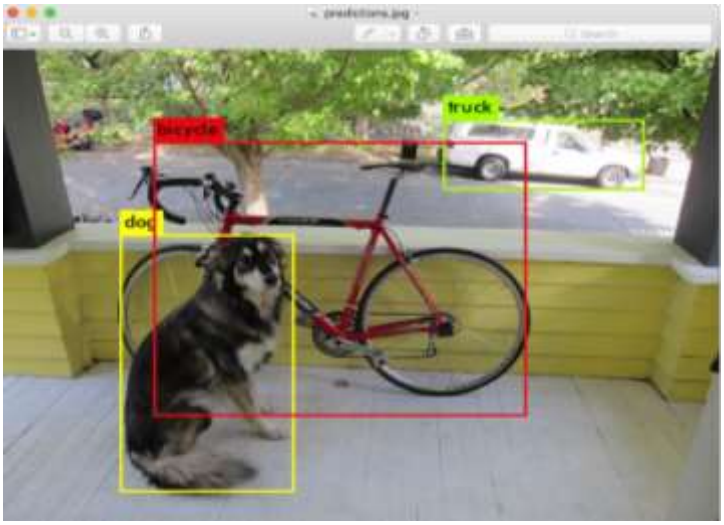

Figure 6: Detection of objects in an image

Source: (Joseph Redmon J. R., 2017).

4 Methodology

4.1 Object detection

There are mainly two ways of object detection. First one is to take object images and train our own Machine Learning model. When we train the machine learning model, main input is features. Based on the features, the model will learn and create weights for that object. But there are some disadvantages in this method. For example, when we consider the object as car, there are different types of cars based on their shapes. Sometimes even a truck might look like a car in the video. To overcome this issue, the feature extraction must be very much robust. Like the model should be trained by all the aspects like size, dimension and shape. This requires a large number of data. Because of this training will depend on our system. If the system's GPU is low then we cannot train our model at all or it might take a very long time to process. If we go for SVM, neural networks or Random Forest models or any basic type of modeling which takes less amount of data, it does not work with the real time data.

So we use a YOLO model which can be defined by a concept of convolution neural network. The one difference between YOLO and other CNN models are, YOLO has a moving or floating window. That means a window is created in YOLO which keeps moving from left to right. While moving if any object which is needed occurs on the screen, YOLO will highlight that object. With the weights which are already present in the model, it will try to detect the object and recognize it. For each objects there exists a different weights in YOLO. There are different types of YOLO. Some models may have 150 different objects and some might have 80. There is an option to limit the number of objects to whatever is required or one can use all the objects present in the model. In this research we have used a model with 80 weights. Every time the code runs we have to load the weights. Since we do not want our model to detect all 80 weights or objects we limit the weights for first 10. This is how detection of object takes place in YOLO.

4.2 Object tracking

Tensorflow used can help to detect the object but it will not track the object. To track the object, bounding box is given to all the objects present in the video or on the screen. Tensorflow gives the kernel dimension of the weights.

Out of eight coordinates in kernel dimension we extract four coordinates. We'll multiply the width and height of the coordinates of the kernel dimension because to fix the dimension of the object with is detected.

4.3 Distance calculation

Distance is calculated by the bounding boxes. The movement when other object is completely detected, the size becomes bigger. If the portion of the object detected is less, then we can assume that either the object is far from us or it is in the sideways like left or right. The bigger the size, the closer the vehicle or object is. And the smaller the vehicle, the distance is more. So from the bounding box distance is calculated. We are taking some constant and we are predicting the distance. Because in autonomous vehicle detection system, calculation of exact distance is not required. Our aim is to just stop the vehicle whenever it is closer to the other vehicle.

4.4 Speed calculation

Since camera is not outside the car or it is not still in one place, we cannot directly calculate the speed. The camera is moving along with the car that is it is fixed inside the moving car. So we will have to calculate the relative speed. Relative speed is a speed of the moving body with respect to another moving or still body.

$$\boldsymbol{Speed} = \frac{\boldsymbol{distance}}{\boldsymbol{time}}$$

Equation 1: Speed calculation

The relative speed is calculated by the difference between the two bodies which are moving in the same direction. And when bodies are moving in the opposite direction, the relative speed is calculated by the addition of the speeds of the two bodies. The x and y in the formula are the speed of two bodies which are moving in a same direction.

$$\boldsymbol{Relative\ speed} = \boldsymbol{x} - \boldsymbol{y}$$

Equation 2: Relative speed formula

We know the basic speed formula that is distance by time. As said earlier distance is calculated by the bounding boxes and time is considered as frames per second. For example, if the frame rate is 100 we get 100 frames per second. If the video has 1000 frames, its time code will be 30 frames per second or 50 frames per second. The frame time is calculated as $^{1}/_{frame\ rate}$. This explains us how long the frame is taken to condense.

For example, $^{1}/_{60}$ frames per second=16.6ms. The time frame is not always fluctuating; hence it will change the frame time from 16.6 to 17ms.

4.5 Evaluation Criteria

To authenticate the results obtained by object detection, three measures are used. They are Precision, Recall and Orientation accuracy. Orientation estimated task can be tested by orientation accuracy. The object detection results are tested by Precision and Recall. The below figure gives the explanation of precision and recall. The precision accuracy can be compared with rest of the state-of-art methods.

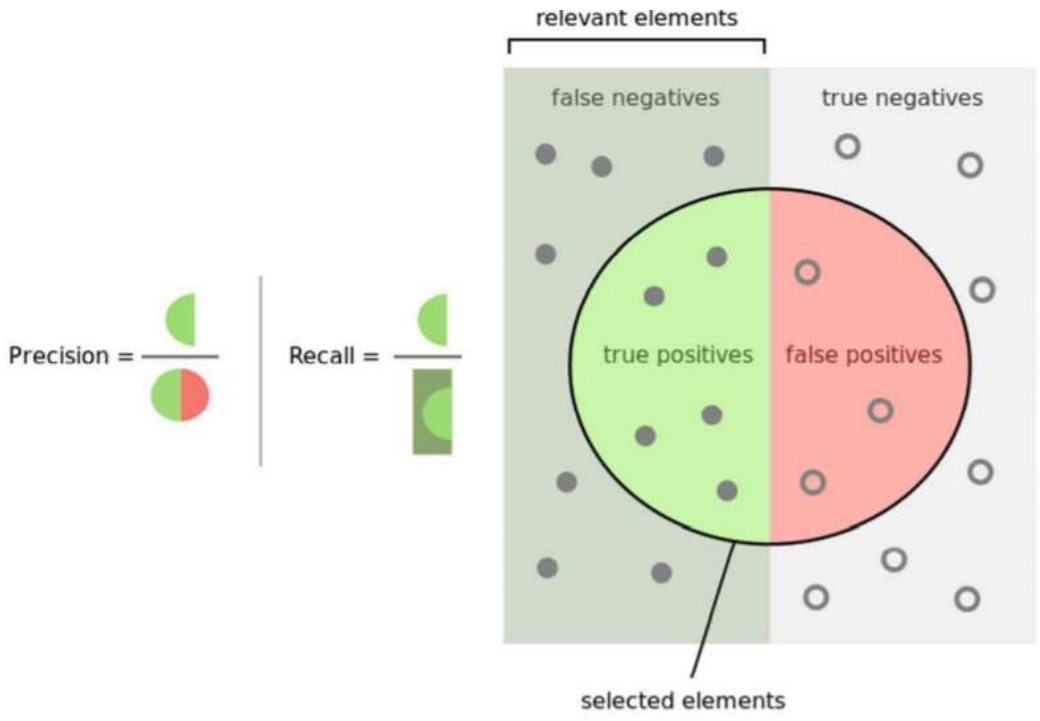

Figure 7: Precision and Recall

Source: (LIU, 2017)

The formula used to calculate the precision and recall is given below. Precision can be simply defined as number of predicting items which are relevant and Recall can be defined as number of relevant elements which are predicted.

$$\boldsymbol{Precision} = \frac{\boldsymbol{tp}}{\boldsymbol{tp} + \boldsymbol{fp}}$$

Equation 3: Precision calculation

Recall can be defined as number of relevant elements which are predicted.

$$\boldsymbol{Recall} = \frac{\boldsymbol{tp}}{\boldsymbol{tp} + \boldsymbol{fn}}$$

Equation 4: Recall calculation

The equation used to calculate the orientation accuracy is given below. The $\alpha_{predict}$ is the value which is predicted and the ground value is calculated by α_{truth}. Since we are testing the results with respect to the truth values or right detections, IOU should be more than 50% and there should be a correct class.

$$\boldsymbol{Orientation\ Accuracy} = \mathbf{1} - \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\mathbf{1}}^{\boldsymbol{n}} \frac{||\boldsymbol{\alpha_{predict}} - \boldsymbol{\alpha_{truth}}||}{\boldsymbol{\pi}}$$

Equation 5: Orientation accuracy formula

In the above equation n represents the total number of objects in the images used for testing. The predicting orientation value is given by $\alpha_{predict}$ and α_{truth} gives the ground truth value.

4.6 Evaluation

The experiment results were validated by orientation precision. The equation for the orientation precision is shown in the evaluation criteria. We have calculated the orientation precision separately by classes. Validation results were outstanding that the performance was topped by the detections of objects correctly. This happens when the class is right and IOU is over 50%.

4.7 Evaluation by Classes

In evaluation by class section, the orientation precision by class is shown. The classes used to predict the accuracy are car, pedestrian, cyclist and misc. All the objects which are not important for our research falls under misc category. According to the graph given below, it is found that prediction rate of car accuracy is high. When compared to pedestrians and cyclists, detection accuracy of cyclists is more than pedestrians. Pedestrians are the hardest class to detect. There can be two reasons that affect the accuracy of orientation. The first factor that affects the orientation is the object's size. If the object size is large, the feature extraction is easy hence detection of object is accurate. It is a clearly known factor that car is larger than the pedestrians or humans hence the precision orientation of car is higher than pedestrians.

The cyclist's size is in between car and human so its accuracy is in between the other two accuracies. The second reason that affects the orientation can be the structure of objects. The horizontal lines and patterns on the object is the main factor for the orientation. The objects which are flat will have long horizontal lines and patterns on its surface are easy for the prediction when compared to thin objects. Anyhow the hypothetical reasons should be given to prove the factors.

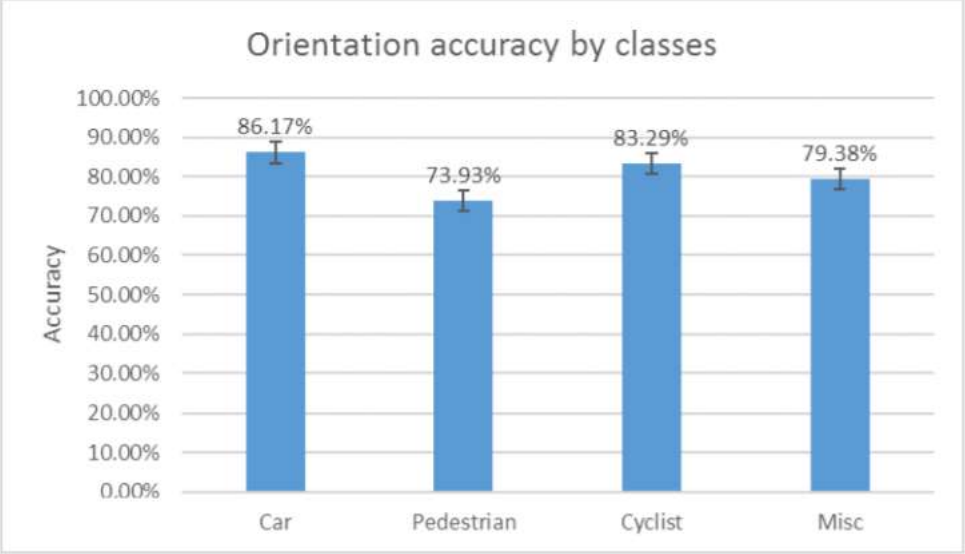

Figure 8: Orientation Accuracy by Object Classes

5 Results and Discussions

The below shown figure is the example of ground truth and prediction orientations. The results are assumed to be accurate when the model detects the objects correctly. Here we have discussed about the objects orientation that are detected correctly. During the validation of objects detected by the model, we have got the object accuracy of 84.89% for 0.031 seconds per image of processing speed.

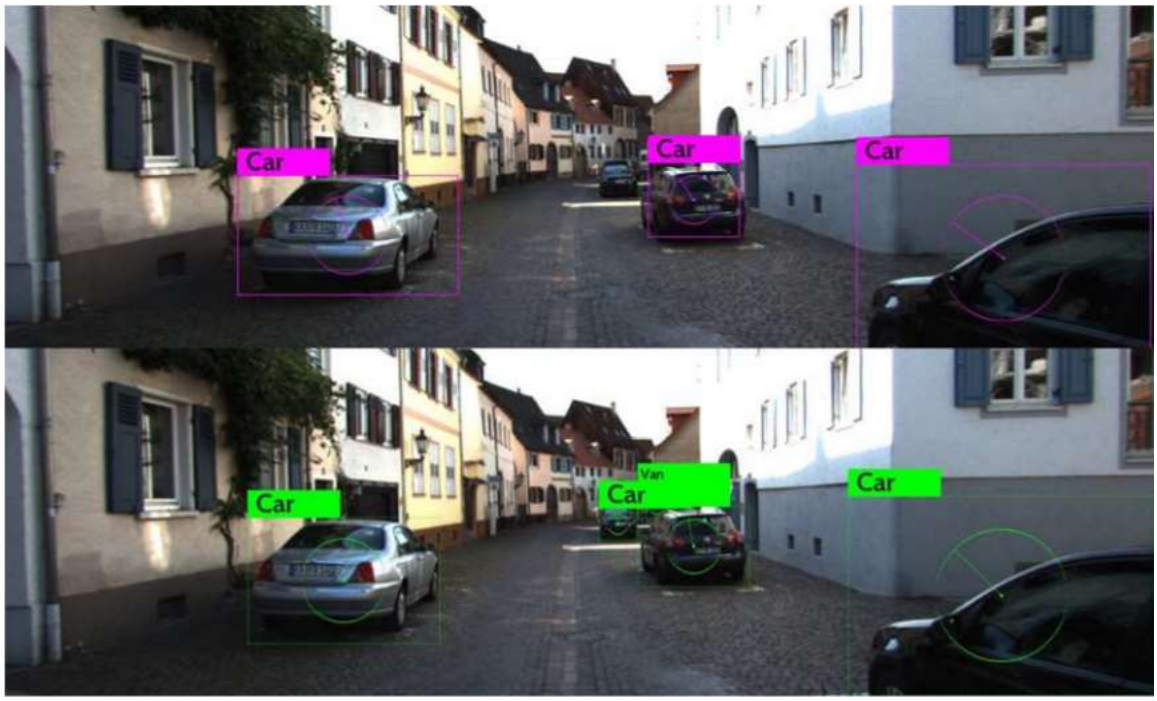

Figure 9: Orientation Results of Car

The detections in green color are the ground truth that is the bottom picture. The detections in the purple color are the predictions which are correct that is the top image.

The Figure 10, is the screenshot taken from the video analysis for self-driving cars. As seen in the picture, the speed and the distance between the vehicles are calculated. Distance is calculated by the bounding boxes. The movement when other object is completely detected, the size becomes bigger. If the portion of the object detected is less, then we can assume that either the object is far from us or it is in the sideways like left or right. The bigger the size, the closer the vehicle or object is. And the smaller the vehicle, the distance is more.

So from the bounding box distance is calculated. Since camera is not outside the car or it is not still in one place, we cannot directly calculate the speed.

The camera is moving along with the car that is it is fixed inside the moving car. So we will have to calculate the relative speed. Relative speed is a speed of the moving body with respect to another moving or still body. The relative speed is calculated by the difference between the two bodies which are moving in the same direction.

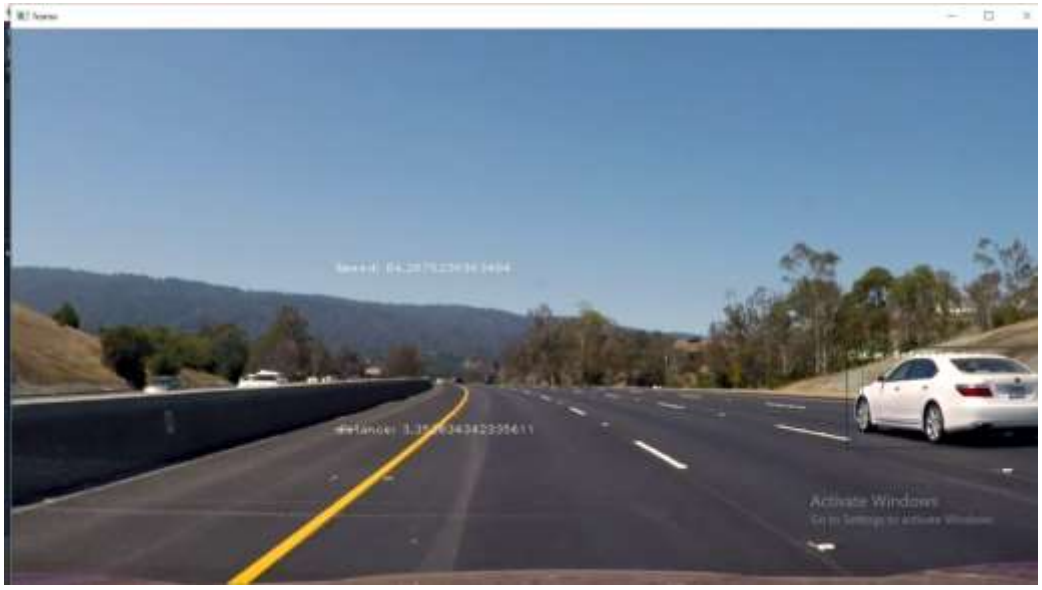

Figure 10: The output frame where the speed and distance is calculated

6 Conclusion

6.1 Summary

In the first chapter Introduction, the research plan is discussed and also the motivation for doing a research on object detection. In second chapter, a literature review on CNN models and YOLO model is given. The brief explanation of background of fully connected neural network is also provided. In third chapter explanation of the experiment flow and methodology is given and also overall system design is explained. In chapter four, the test validation results and the experimental results are explained.

6.2 Contribution

When we analyze the results we have got for precision and recall, it can be said that YOLO is one of the best model used for self-driving systems in detection of objects. YOLO model has achieved 85% of precision with 62% of recall with the time rate of 30 frames per second. We have also successfully found the speed of the other vehicles moving and the distance between the vehicles are calculated. This helps the autonomous driving system to compute braking system.

Even though there is a continuous data available in the video of real time driving condition, processing of images are done individually in YOLO. This can be a loop hole in YOLO model to fit in real-time autonomous driving system. To overcome the issue, the further more experiments should be conducted in the new technique called memory map technique. This helps YOLO to fit better in real-time video analysis. The memory map technique helps to gain the class confidence by using time based frames which in turn helps to decrease the precision and increase the recall.

The YOLO model is in the top place in the object detection speed when compared to other convolution neural networks. The detection speed that we have achieved is 0.03 seconds per image, which is 10 times faster than the already present object detection models. The YOLO model is the only model that has achieved this accuracy in real-time video streaming, which can be used for autonomous driving system.

From the computation of orientation estimation, we have found that YOLO has a good precision in prediction of object orientation. Since object's orientation has a main role in self-driving systems, with the accuracy we got for orientation estimation we can state that YOLO fits in the best for self-driving systems.

By all the experiments conducted, it is proved that performance of YOLO is high in both object detection and orientation precision. Since object's orientation has a main role in self-driving systems, with the accuracy we got for orientation estimation we can state that YOLO fits in the best for self-driving systems. We have also successfully found the speed of the other vehicles moving and the distance between the vehicles are calculated. This helps the autonomous driving system to compute braking system.

6.3 Research Questions and answers

1. To verify whether YOLO is good model for general object detection.

   By the research conducted and by reading many journal papers explained in chapter 2, we can say that for general object detection YOLO is the best model as its detection accuracy is high compared to other convolution neural network models.

2. Is YOLO a potential object detection model for self-driving system or not?

   In our experiment we have found that despite of having a very good detection precision, YOLO still has some limitations. YOLO processes every image separately even in a continuous video or frames. Because of this much important identification can be lost.

3. Can YOLO be improvised to better fit in autonomous driving situation?

   Since autonomous driving system depends mainly on optical vision, detection speed and precision in object detection should be high. YOLO should be improvised so that it does not miss any important information while processing every image separately.

4. Whether YOLO can be applied to other tasks rather than object detection?

   Yes other than object detection YOLO can be used for orientation estimation. Orientation estimation is to check the accuracy of the detection.

7 Future work

Even though there is a continuous data available in the video of real time driving condition, processing of images are done individually in YOLO. This can be a loop hole in YOLO model to fit in real-time autonomous driving system. To overcome the issue, the further more experiments should be conducted in the new technique called memory map technique. This helps YOLO to fit better in real-time video analysis. The memory map technique helps to gain the class confidence by using time based frames which in turn helps to decrease the precision and increase the recall. By the validation of orientation we can observe that increase in the recall accuracy can effect on the usage of memory map. Since the weights for each frame is already defined, training of memory map is not necessary. In future, training of memory map can be done to learn more and accurately about weights.

When it comes to autonomous driving system, we have found only two parameters that is speed analysis and distance calculation. To completely build self-driving cars, we have to consider many more parameters like road width, steering angle, acceleration etc. Our experiment is just a part of it. In future with better version of YOLO and with all the parameters considered for self-driving cars an efficient autonomous driving system can be developed.

References

Alex Krizhevsky, I. S. (2015). ImageNet Classification with Deep Convolutional neural networks. *IEEE*, 9.

Amy Jin, S. Y.-F. (Standford). Tool Detection and Operative Skill Assessment in Surgical Videos Using Region-Based Convolutional Neural Networks. *arXiv:1802.08774v2* , 9.

Ayush Jain, M. M. (2018). *Real Time Object Detection and Tracking Using Deep Learning and OpenCV.* Bangalore: 2018 International Conference on Inventive Research in Computing Applications (ICIRCA).

Chadalawada, S. K. (2020). *Real Time Object Detection and Recognition using deep learning methods.* Sweden: Faculty of Computing, Blekinge Institute of Technology.

Cr´eput, B. C.-C. (2019). *Matlab GUI Application for Moving.* France: Le2i FRE2005, CNRS, Arts et Mtiers, Univ. Bourgogne Franche-Comt´e,.

Farhadi, J. R. (2018). YOLOv3: An Incremental Improvement. *arXiv:1804.02767v1* , 6.

Girshick, R. (2012). Fast R-CNN. *IEEE*, 9.

Jifeng Dai, Y. L. (2016). R-FCN: Object Detection via Region-based Fully Convolutional Networks. *arXiv:1605.06409v2*, 11.

Jong-Min Jeongl, T.-S. Y.-B. (2014). *Kalman Filter Based Multiple Objects Detection-Tracking Algorithm Robust to Occlusion.* Sapporo: SICE Annual Conference.

Joseph Redmon, A. F. (2016). *YOLO9000:Better, Faster, Stronger.* Washington: arXiv:1612.08242v1.

Joseph Redmon, J. R. (2017). You Only Look Once: Unified, Real-Time Object Detection. *IEEE*, 10.

Jovanny Bedoya Guapacha, S. C. (2017). *Real time object detection and tracking using the Kalman Filter embedded in single board in a robot .* Brazil: IEEE.

Juraj Ciberlin, R. G. (2019). *Object detection and object tracking in front of the vehicle using front view camera .* Osijek: IEEE.

Kaiming He, X. Z. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 346–361, 2014.*, 16.

Kaiming He, X. Z. (2015). *Deep Residual Learning for Image Recognition.* tokyo: arXiv:1512.03385v1.

LIU, G. (2017). *REAL-TIME OBJECT DETECTION FOR AUTONOMOUS DRIVING BASED ON DEEP LEARNING.* china: Texas A&M University-Corpus Christi.

pjreddie. (2016). Yolo: Darknet. *IEEE*, 10.

Shahkaran. (2017). Yolo object detection algorithm in tensorflow. *medium-e080a58fa79b*, 10.

Shaoqing Ren, K. H. (2011). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Microsoft Research*, 9.

Shehan P Rajendra, L. S. (2019). *Real time traffic sign recognition using YOLOv3 based detector.* Kanpur: IEEE.

Zhihao Chen, R. K.-Y. (2019). *Real Time Object Detection, Tracking, and Distance.* France: IEEE.

Appendix

This section helps you to understand the necessary steps to be taken to implement a python code for Real time Object Detection, Tracking, Distance and Speed Estimation for self-driving cars based on Deep Learning using YOLO technique.

Contents of Artefact

Codes for YOLO model to detect and tract objects

- Yolov3.weights
- capture_frame.py
- detect.py
- detector.py
- dual.py
- helpers.py
- tracker.py

Code to run a project in real-time

- Realtime_output_yolo.py

Code to run a recoded video

- Record_yolo_video_with_input.py
- DNModel.py
- util.py