Object Detection for Self Driving Car

A Synopsis Report submitted in partial fulfillment of the requirement for the degree of

B.Tech.

In

Computer Science & Engineering

Under the

Supervision of

"Mr. Mahesha A.M"

By

Yogesh Chandra Mishra	**(1609110125)**
Yash Saraswat	**(1609110123)**
Vaibhav Tyagi	**(1609110119)**
Shaurya Vardhan Singh	**(1709110920)**

JSS MAHAVIDYAPEETHA

JSS ACADEMY OF TECHNICAL EDUCATION

TABLE OF CONTENTS

Page

Session: - 2019-2020

VISION OF THE DEPARTMENT:

To spark the imagination of the Computer Science Engineers with values, skills and creativity to solve the real world problems.

MISSION OF THE DEPARTMENT:

Mission1: To inculcate creative thinking and problem solving skills through effective teaching, learning and research.

Mission2: To empower professionals with core competency in the field of Computer Science and Engineering. Mission3: To foster independent and life-long learning with ethical and social responsibilities.

PROGRAMME EDUCATIONAL OBJECTIVES:

PEO1: To empower students with effective computational and problem solving skills.

PEO2: To enable students with core skills for employment and entrepreneurship.

PEO3: To imbibe students with ethical values and leadership qualities.

PEO4: To foster students with research oriented ability which helps them in analyzing and solving real life problems and motivate them for pursuing higher studies.

PROGRAMME OUTCOMES:

Engineering Graduates will be able to:

PO1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

PROGRAMME SPECIFIC OUTCOMES:

PSO1: An ability to apply foundation of Computer Science and Engineering, algorithmic principles andtheory in designing and modeling computation based systems.

PSO2: The ability to demonstrate software development skills.

COURSE OUTCOMES

C410.1 Identify, formulate, design and analyze a research-based/web-based problem to address societal and environmental issues.

C 410.2 Communicate effectively in verbal and written form.

C410.3 Apply appropriate computing, engineering principle and management skills for obtaining effective solution to the formulated problem within a stipulated time.

C410.4 Work effectively as a part of team in multi-disciplinary areas.

C410.5 Consolidate the final outcome in the form of a publication.

CO-PO-PSO MAPPING

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO12
C410.1	3	3	3	3	2	3	3	3	3	3	2	3	3	3
C410.2	2	2	2	2	2	2	2	2	2	3	2	3	2	3
C410.3	3	3	3	3	3	3	2	3	3	3	3	3	3	3
C410.4	3	3	3	3	2	3	2	3	3	3	3	3	3	3
C410.5	3	3	3	3	3	3	3	3	3	3	3	3	3	3
C410	**2.80**	**2.80**	**2.80**	**2.80**	**2.40**	**2.80**	**2.40**	**2.80**	**2.80**	**3.00**	**2.60**	**3.00**	**2.80**	**3.00**

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature: Yogesh Mishra

Name : Yogesh Chandra Mishra

Roll No.: 1609110125

Date: 08/05/2020

Signature: Yash Saraswat

Name: Yash Saraswat

Roll No.: 1609110123

Date: 08/05/2020

Signature: Vaibhav Tyagi

Name: Vaibhav Tyagi

Roll No.: 1609110119

Date : 08/05/2020

Signature: Shaurya Vardhan Singh

Name: Shaurya Vardhan Singh

Roll No.: 1709110920

Date : 08/05/2020

CERTIFICATE

This is to certify that Project Report entitled "Object Detection for Self Driving Cars" which is submitted by G-42 Group in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science and Engineering of Dr. APJ Abdul Kalam Technical University, Lucknow is a record of the candidate's own work carried out by him/her under my/our supervision.The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Supervisor: Mr. Mahesha A.M

Date: 08/05/2020

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year.

We owe special debt of gratitude to Professor/Assistant Professor Mr. Mahesha A.M, Department of Computer Science.

Engineering, JSS Academy of Technical education, Noida for his/her constant support and guidance throughout the course of our work. His/Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us.

It is only his/her cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Prof. (Dr.) Vikram Bali, Head, Department of Computer Science & Engineering, JSS Academy of Technical education, Noida for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature: Yogesh Mishra

Name : Yogesh Chandra Mishra

Roll No.: 1609110125

Date: 08/05/2020

Signature: Yash Saraswat

Name: Yash Saraswat

Roll No.: 1609110123

Date: 08/05/2020

Signature: Vaibhav Tyagi

Name: Vaibhav Tyagi

Roll No.: 1609110119

Date : 08/05/2020

Signature: Shaurya Vardhan Singh

Name: Shaurya Vardhan Singh

Roll No.: 1709110920

Abstract

The detection,classification and tracking of objects around an autonomous vehicle is essential to operate the vehicle safely. This report presents an algorithm to detect, classify, and track objects. All objects are classified by their type (e.g. vehicle,pedestrian, or other). The proposed approach uses the state of the art deep-learning based technique YOLOv2 (You Only Look Once) to detect and classify the objects and estimate the position of objects around the car. The resultant solution aids in the localization of the objects within its environment, so that it can safely navigate the roads autonomously. The algorithm has been developed and tested using the dataset collected by COCO dataset and drive.ai. The Drive.ai Sample Dataset (provided by drive.ai) is licensed under the Creative Commons Attribution 4.0 International License. We are grateful to Kaggle and deeplearning.ai for providing this data.

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Full Form
ANN	Artificial Neural Networks
CNN	Convolutional Neural Network
NMS	Non-Max Suppression
YOLO	You Only Look Once
DATMO	Detection And Tracking of Moving Objects
SLAM	Simultaneous Localization And Mapping
GPS	Global Positioning System
GUI	Graphical User Interface

CHAPTER 1

1.1 Introduction

Objects present in an image can be easily detected by humans. The human visual system is very fast, accurate and can perform complex tasks like recognizing and identifying multiple objects and detect obstacles with little thought. With the availability of large amounts of data, faster GPUs, TPUs, and better algorithms, we can now easily train computers to detect and classify multiple objects within an image with high accuracy. Autonomous driving-object detection can detect different classes of objects present on the road to drive car autonomously and efficiently. The process of the autonomous vehicle guidance depends on three things: mapping, localization, and tracking objects. Localization is process of the identifying the position of the autonomous vehicle in environment. Mapping includes being able to make the sense of the environment. Tracking of the moving objects involves being able to identify and classify the moving objects and track them during navigation. The processes of localization and mapping have been explored through use of simultaneous localization and mapping (SLAM), which was initially proposed by Leonard and Durant-Whyte. Simultaneous Localization and Mapping enables autonomous vehicles to simultaneously create the map of the unknown environment and localize itself in the environment. The development of SLAM has been significant in the development of autonomous robots and machines. The detection and tracking of the moving objects (DATMO) is one of the most challenging 11 issues and is important for safety and essential for avoiding collisions. SLAM combined with DATMO helps solve this problem. With advancements in area of incremental improvements and deep learning in computing power, object detection using images outperforms other methods for the detection and classification of objects.

1.1.1 PROBLEM STATEMENT

Many problems in computer vision were saturating on their accuracy and speed of computation before a decade.A slightly complicated problem is that of image localization, where the image contains a single object and system should predict the class of location of the object in the image (a bounding box around the object). The more complicate problem of object detection involves

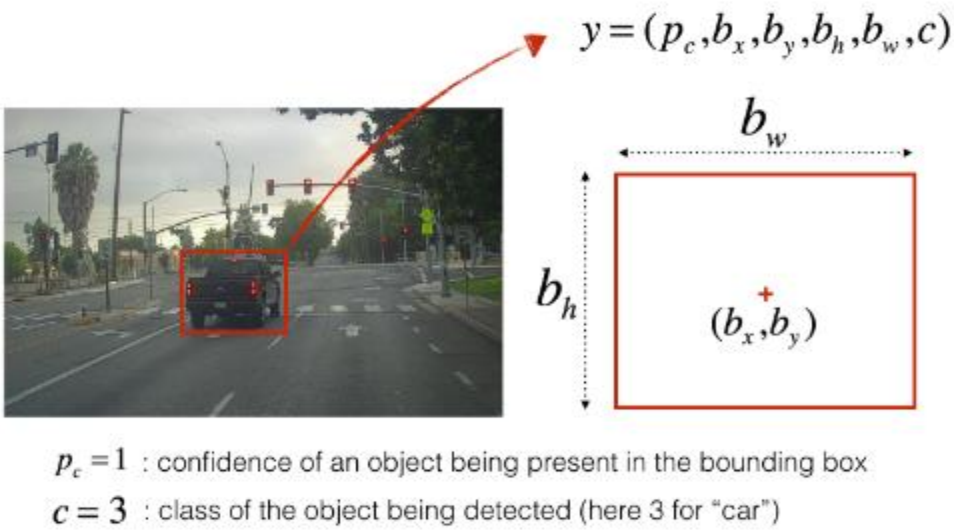


Fig 1.1.1 Definition of a box

both the classification and localization. In this case, the input to the system will be a image, and the output will be a bounding box corresponding to all the objects in the image, along with the class of object in each box.

1.1.2 Motivation

Efficient, accurate and fast object detection has been an important topic in the advancement of computer vision systems. With advent of deep learning techniques, the accuracy and the speed for object detection has been increased drastically. This project aims to work on state-of-the-art technique for object detection with the goal of achieving high accuracy with a real-time performance. A major challenge in most of the old object detection systems is the dependency on other computer vision techniques for helping the deep learning based approach, which leads to slow, less accurate and non-optimal performance. In this project, we use a completely deep learning based approach to solve the problem of object detection in an end-to-end fashion using YOLO and Darknet-19. The network is trained on the dataset taken from **drive.ai**, a company building the brains of self-driving vehicles. The resulting system is fast,reliable and accurate, thus aiding those applications which require object detection.

1.1.3 Project Objective

The Objective is to detection of objects using YOLO approach. This method has the several advantages as compared to the other object detection algorithms. In the other algorithms like Convolutional Neural Network(CNN), FastConvolutional Neural Network(FCNN) the algorithms will not look at the image completely but in YOLO the algorithm looks the image completely by predicting the bounding boxes using convolutional neural network and the class probabilities for these boxes and detects the image faster as compared to other algorithms . Deep neural networks have many computational stages, or levels in which neurons are simulated from the environment that activate the other neurons in the network The neural network performance depend on an extensive amount of data extracted from real life driving scenarios. The neural network is activated and “learns” to perform the best course of action.

1.1.4 Scope of Project

Object detection is breaking into a wide range of industries, with use cases in autonoumous car ranging from car security to distance estimation in the workplace. Object detection and recognition is applied in many areas of the computer vision, including the image retrieval, automated vehicle systems and machine inspection. Significant challenges stay on the field of object detection and classification. The possibilities are endless when it comes to the future use cases for the object detection.

1.2 Related Previous Work

- RCNN(Region Convolutional Network)
- Fast RCNN
- Faster RCNN
- SSD(Single Shot Multibox Detector)

CHAPTER 2

2.1 Literature Survey

There has been a lot of work in object detection and classification using traditional computer vision techniques (sliding windows, deformable part models). However, they lack the accuracy and computation speed of deep learning based techniques.Among the deep learning based techniques, two broad class of methods are prevalent: two stage detection (RCNN, Fast RCNN, Faster RCNN) and unified detection (Yolo, SSD).

Many of the ideas in this project are described in the two YOLO papers:

Redmon et al., 2016 (https://arxiv.org/abs/1506.02640) and

Redmon and Farhadi, 2016 (https://arxiv.org/abs/1612.08242).

Yan, L. et al. (2017) [10]proposed a deep learning model for iterative object recognition. Based on user needs, the program can classify a number of objects. It will become ever more intelligent during user interaction and identify moreclasses of objects. By studying few examples, including five pictures of an object type, it has high performance. Therecognition level is becoming higher and higher, and after studying the new samples, it can recognize ever more objectcategories. Through evaluating it on complex datasets,researchers have confirmed its efficient learning capability,high recognition levels, and wide range of applications. It also reported that our incremental learning method's object recognition level would be higher with the user's interaction. Xu, N. et al. (2019) [16] recommended an efficient objectdetection strategy when using a deep reinforcement learningsolution to help object recognition system dynamicallychange brightness. Experiments show the adaptive brightnesschange needs and the efficacy of the proposed effectiveobject recognition method. Future research is specificallyconcerned with in-depth verification and further progress inboth the imaging technique and the method of objectrecognition.

Tang, J., & Wen, G. (2016) [15] proposed a flexible objectrecognition approach through a classifier interface with multiple features. Through numerous characteristics involved with the classifier fusion, it shows power of dealing with pose variance, occlusion and lighting adjustments. The results of the experiment indicate that the proposed system of object recognition can correctly and robustly identify the object.

Hao, W. et al. (2018) [23] presented an advanced image recognition system model based on the CNN in conjunction with ideas for bottom-up regions. Target area choice and weight enhancement were our paper's key contributions. Based on the above enhancements, appropriate tests using a greater number of images are also used to illustrate the reliability and robustness of our system. Further changes in furture , however are still required. For example if the target sample is not selected accurately , due to ineffective selection, the error could be enhanced. In addition, the weight of the improvement is based on subjective evaluation, but there would be a potential for subjective evaluation.

Year	Author(s)	Method Proposed	Results (Outcomes)
2017	Wu, H. et al. [18]	CNN refinement method based recognition	method could contribute to retrieve pictures efficiently
2019	Cadoni, M. et al. [19]	Object model is constructed and developed because of significant point resolution.	Recognition rates hit 91.17 percent at rank one, which compares favorably with the 85.75 percent recognition rate achieved through a competitive Bag-of-Words-based process.
2015	Bai, Z. et al. [20]	ELM-LRF was proposed as a general framework for object identification	Reduced complexity and required amount of training data as compared to CNN
2016	Guo, Q. et al. [21]	Presented an integrated model by combining Convolutional Neural Network (CNN) and Hidden Markov Model (HMM)	Hybrid model will significantly increase the recognition quality
2018	Ciocca, G. et al. [22]	use of CNN-based features for food retrieval and recognition	features learned from the new database outperform

CHAPTER 3

System Design and Methodology

3.1 System Design

3.1.1 System Architecture

IMAGE (m, 608, 608, 3) -> DEEP CNN -> ENCODING (m, 19, 19, 5, 85)->Prediction of Bounding Boxes->IoU->Non-Max Suppression->Output

3.1.2 Model Details

First things to know:

The **input of the model** is a batch of images of shape (m, 608, 608, 3)

The **output of the model** is a list of bounding boxes along with the recognized classes. Each bounding box of the model is represented by 6 numbers (*pc*,*bx*,*by*,*bh*,*bw*,*c*) as explained above. If you expand *c* into an 80- dimensional vector, each bounding box of model is then represented by the 85 numbers.

We will use 5 anchor boxes. So you can think of the YOLO architecture of the model as the following:

Image (m, 608, 608, 3) →Deep CNN→Encoding (m, 19, 19, 5, 85) are required for intial part of Yolov2 Architecture.

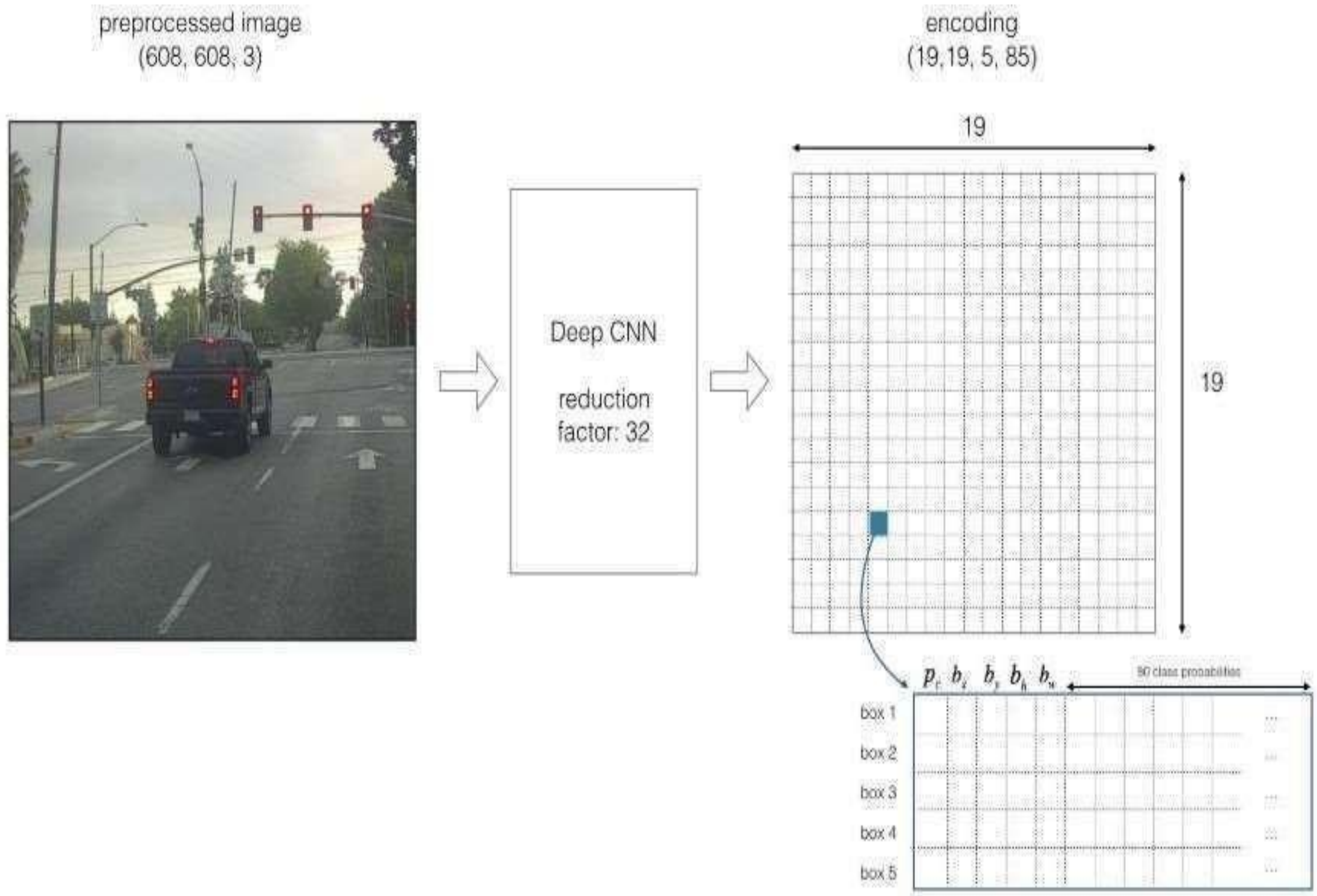
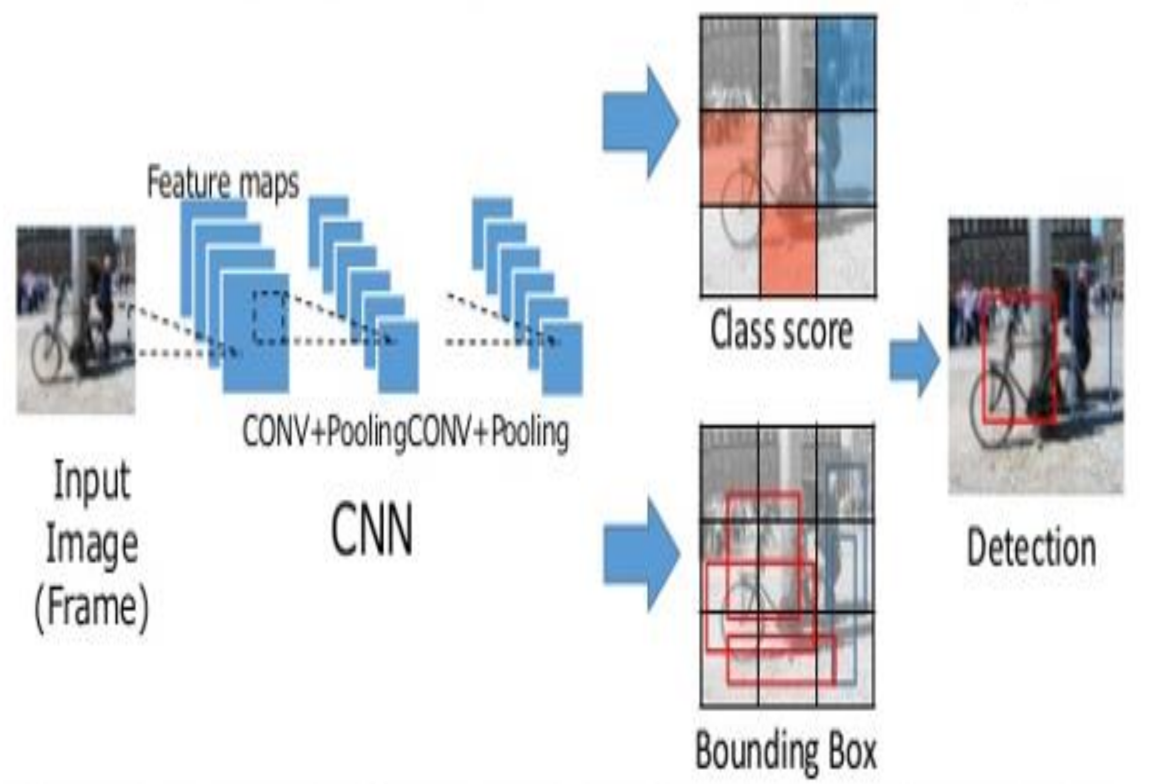


Fig 3.1.2 Encoding architecture for YOLO

3.2 Algorithms

3.2.1 Core

YOLO:YOLO ("you only look once") is a popular algorithm used for recognization of objects because it achieves high accuracy while also being able to run in real-time. This algorithm " You only looks once" sees at the image in the sense that it requires only one forward propagation pass through the network to make predictions.It then outputs recognized objects together with the bounding boxes after non-max suppression.



3.2.2 Predicting Bounding Boxes

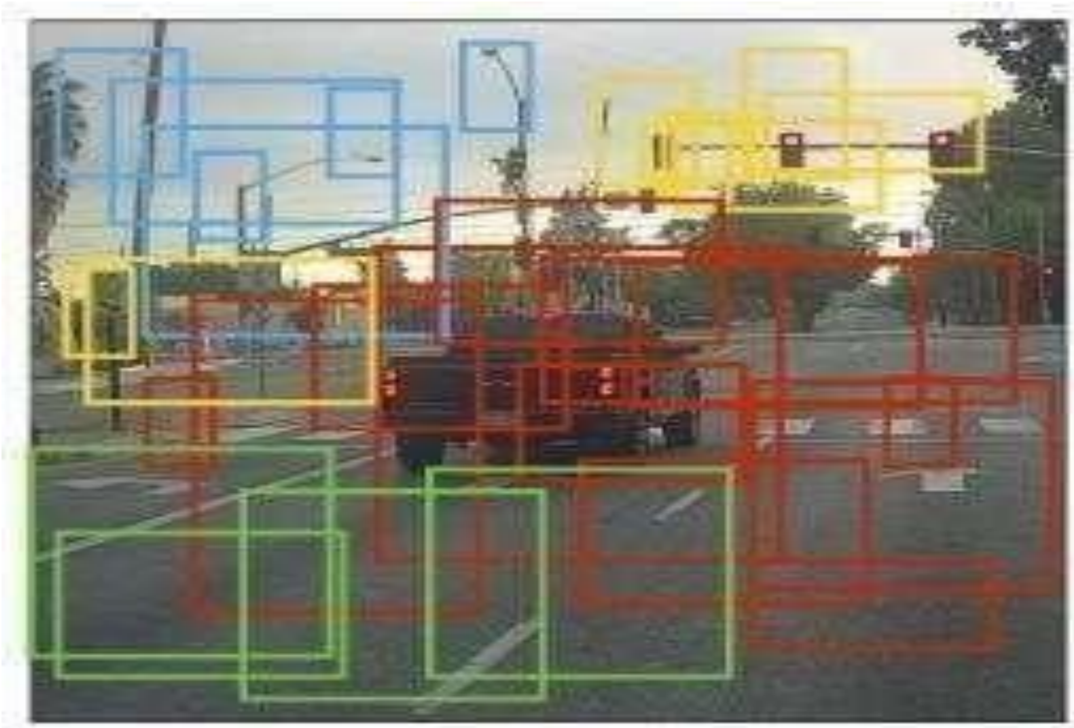
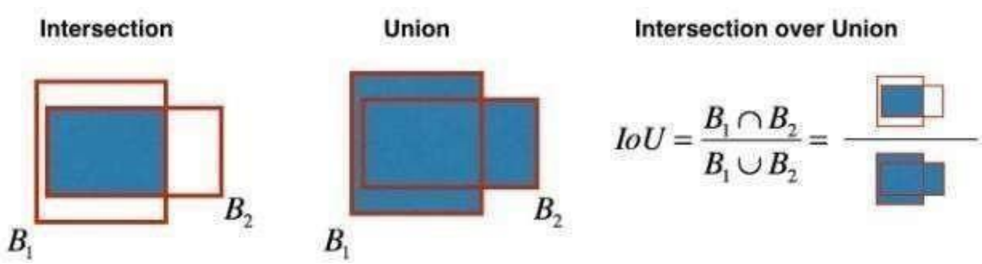


Fig 3.2.2 : A way to visualize YOLO's output is to plot the bounding boxes that it outputs.

Each cell gives you 5 boxes. In total, the model predicts: 19x19x5 = 1805 boxes by using yolo algorithms.



3.2.3 Intersection over Union(IoU)

Fig 3.2.3 : Definition of Intersection over Union

Intersection over union i.e IOU is a measure of the overlap between two bounding boxes.

3.2.4 Non-max Suppression

Non-max suppression is a way of eliminating points that do not lie in important edges.

The key steps of Non-max suppression are:

- Select the box that has the highest score of IOU.
- Compute overlap of above box with all other boxes, and remove boxes that overlap it more than iou_threshold.
- Repeat the step 1 and iterate until there's no more boxes with a lower score than the current selected box.

This process will be removed all the boxes that have a large overlap with the selected boxes. Only the "best" boxes remain.

Fig 3.2.4 N-MS

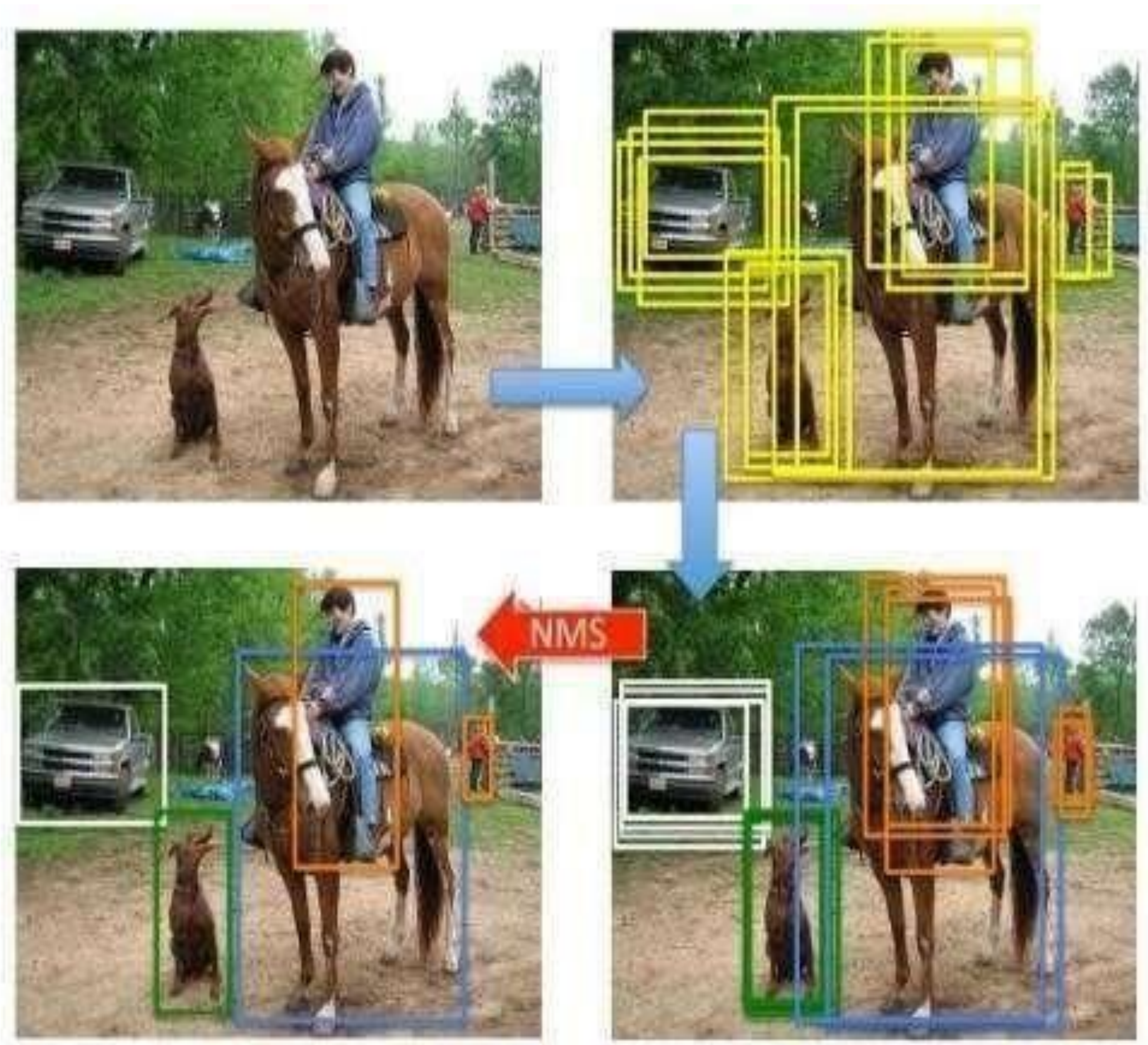
3.2.5 Combining Modules

We will combine all the modules here and run the YOLO model on the preprocessed images for cross validation .

If The cross validation is successful then we will run the model on our test set which consists of images captured by camera on top of car.

The output of the model will be the detection of 80 classes of objects in an image which can be used to guide the car to take suitable actions.

Finally we will use our model for real time simulation of object detection by car.



CHAPTER 4

Implementation and Results

4.1 Software Requirements

- Web Browser
- Google Colaboratory
- Tensorflow
- Keras
- YOLO model dataset

4.2 Implementation Details

4.2.1 Some Snapshots of code

+ Code + Text

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&res

Enter your authorization code:
··········
Mounted at /content/drive
```

```
[ ] cd /content/drive/My\ Drive/New folder
```

```
/content/drive/My Drive/New folder
```

Import libraries

Run the following cell to load the packages and dependencies that you will find useful as you build the object *detector*

```python
import argparse
import os
import matplotlib.pyplot as plt
from matplotlib.pyplot import imshow
import scipy.io
import scipy.misc
import numpy as np
import pandas as pd
import PIL
import tensorflow as tf
from keras import backend as K
from keras.layers import Input, Lambda, Conv2D
from keras.models import load_model, Model
from yolo_utils import read_classes, read_anchors, generate_colors, preprocess_image, draw_boxes, scale_boxes
from yad2k.models.keras_yolo import yolo_head, yolo_boxes_to_corners, preprocess_true_boxes, yolo_loss, yolo_body

%matplotlib inline
```

+ Code + Text

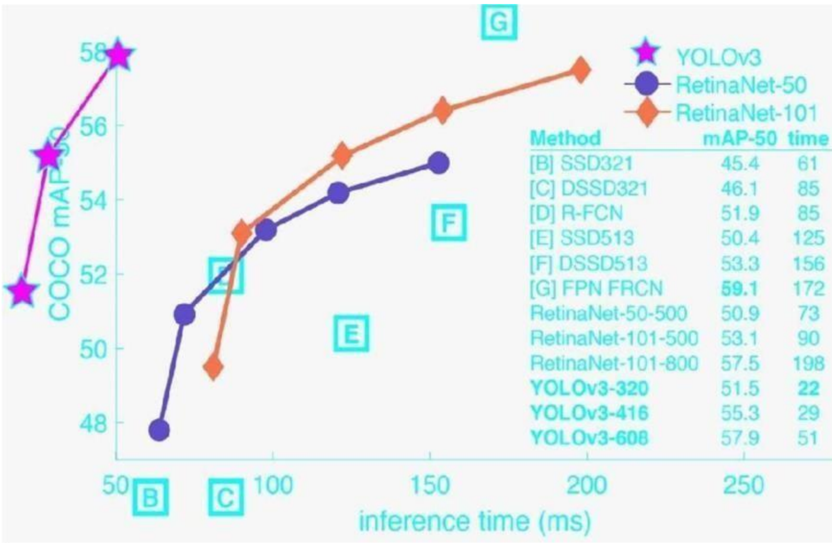
```python
def yolo_filter_boxes(box_confidence, boxes, box_class_probs, threshold = .6):
    """Filters YOLO boxes by thresholding on object and class confidence.

    Arguments:
    box_confidence -- tensor of shape (19, 19, 5, 1)
    boxes -- tensor of shape (19, 19, 5, 4)
    box_class_probs -- tensor of shape (19, 19, 5, 80)
    threshold -- real value, if [ highest class probability score < threshold], then get rid of the corresponding box

    Returns:
    scores -- tensor of shape (None,), containing the class probability score for selected boxes
    boxes -- tensor of shape (None, 4), containing (b_x, b_y, b_h, b_w) coordinates of selected boxes
    classes -- tensor of shape (None,), containing the index of the class detected by the selected boxes

    Note: "None" is here because you don't know the exact number of selected boxes, as it depends on the threshold.
    For example, the actual output size of scores would be (10,) if there are 10 boxes.
    """

    # Step 1: Compute box scores
    box_scores = box_confidence * box_class_probs # (19, 19, 5, 80)
    # Step 2: Find the box_classes using the max box_scores, keep track of the corresponding score

    box_classes =  K.argmax(box_scores, axis=-1)
    box_class_scores = K.max(box_scores, axis=-1)

    # Step 3: Create a filtering mask based on "box_class_scores" by using "threshold". The mask should have the
    # same dimension as box_class_scores, and be True for the boxes you want to keep (with probability >= threshold)

    filtering_mask = box_class_scores >= threshold  # (19, 19, 5)

    # Step 4: Apply the mask to box_class_scores, boxes and box_classes

    scores = tf.boolean_mask(box_class_scores, filtering_mask)
    print (boxes, filtering_mask)
    boxes = tf.boolean_mask(boxes, filtering_mask)
    classes = tf.boolean_mask(box_classes, filtering_mask)

    return scores, boxes, classes
```

```
[ ] yolo_model.summary()

Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 608, 608, 3)  0
__________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, 608, 608, 32) 864         input_1[0][0]
__________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 608, 608, 32) 128         conv2d_1[0][0]
__________________________________________________________________________________________________
leaky_re_lu_1 (LeakyReLU)       (None, 608, 608, 32) 0           batch_normalization_1[0][0]
__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 304, 304, 32) 0           leaky_re_lu_1[0][0]
__________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, 304, 304, 64) 18432       max_pooling2d_1[0][0]
__________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 304, 304, 64) 256         conv2d_2[0][0]
__________________________________________________________________________________________________
leaky_re_lu_2 (LeakyReLU)       (None, 304, 304, 64) 0           batch_normalization_2[0][0]
__________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None, 152, 152, 64) 0           leaky_re_lu_2[0][0]
__________________________________________________________________________________________________
conv2d_3 (Conv2D)               (None, 152, 152, 128 73728       max_pooling2d_2[0][0]
__________________________________________________________________________________________________
batch_normalization_3 (BatchNor (None, 152, 152, 128 512         conv2d_3[0][0]
__________________________________________________________________________________________________
leaky_re_lu_3 (LeakyReLU)       (None, 152, 152, 128 0           batch_normalization_3[0][0]
__________________________________________________________________________________________________
conv2d_4 (Conv2D)               (None, 152, 152, 64) 8192        leaky_re_lu_3[0][0]
__________________________________________________________________________________________________
batch_normalization_4 (BatchNor (None, 152, 152, 64) 256         conv2d_4[0][0]
__________________________________________________________________________________________________
leaky_re_lu_4 (LeakyReLU)       (None, 152, 152, 64) 0           batch_normalization_4[0][0]
__________________________________________________________________________________________________
conv2d_5 (Conv2D)               (None, 152, 152, 128 73728       leaky_re_lu_4[0][0]
__________________________________________________________________________________________________
batch_normalization_5 (BatchNor (None, 152, 152, 128 512         conv2d_5[0][0]
__________________________________________________________________________________________________
leaky_re_lu_5 (LeakyReLU)       (None, 152, 152, 128 0           batch_normalization_5[0][0]
__________________________________________________________________________________________________
max_pooling2d_3 (MaxPooling2D)  (None, 76, 76, 128)  0           leaky_re_lu_5[0][0]
__________________________________________________________________________________________________
conv2d_6 (Conv2D)               (None, 76, 76, 256)  294912      max_pooling2d_3[0][0]
__________________________________________________________________________________________________
```

```python
    # Use one of the functions you've implemented to perform Non-max suppression with
    # maximum number of boxes set to max_boxes and a threshold of iou_threshold (≈1 line)
    scores, boxes, classes = yolo_non_max_suppression(scores, boxes, classes, max_boxes, iou_threshold)

    return scores, boxes, classes
```

4.2.2 Implementation Results

Fig 4.2.2(i) Performance of Different Architecture



4.2.2(ii) Computation Time

Model	Train	Test	mAP	FLOPS	FPS
Old YOLO	VOC 2007+2012	2007	63.4	40.19 Bn	45
SSD300	VOC 2007+2012	2007	74.3	-	46
SSD500	VOC 2007+2012	2007	76.8	-	19
YOLOv2	VOC 2007+2012	2007	76.8	34.90 Bn	67
YOLOv2 544x544	VOC 2007+2012	2007	78.6	59.68 Bn	40
Tiny YOLO	VOC 2007+2012	2007	57.1	6.97 Bn	207
SSD300	COCO trainval	test-dev	41.2	-	46
SSD500	COCO trainval	test-dev	46.5	-	19
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40
Tiny YOLO	COCO trainval	-	-	7.07 Bn	200

Chapter 5

Conclusion

An accurate and efficient object detection system has been developed which gives us better understanding of how object detection works by using yolo algorithms and how self driving car uses it for detecting different objects on the roads such as cars,pedestrian,truck,signals etc..,and achieves comparable metrics with the existing state-of-the-art system. This project used recent techniques and concept in the field of computer vision and deep learning.We uses label Image which created Custom dataset and its evaluation was consistent.Which we can used in real-time applications and it require object detection for pre-processing in their pipeline. An important scope would be to train the system on a video sequence for usage in tracking and notion planning applications. Addition of few optimization technique in network training would enable smooth detection and more optimal than per-frame detection.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei- Fei. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR2009. IEEE Conference on, pages 248–255. IEEE, 2009.1

[2] J. Redmon.Darknet: Open source neural networks in c http://pjreddie.com/darknet/, 2013–2016.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXivpreprint arXiv:1506.02640, 2015.

[4] M. Lin, Q. Chen, and S. Yan. Network in network. arXivpreprint arXiv:1312.4400, 2013.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842, 2014.

[6] R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learn- ing for image recognition. arXiv preprint arXiv:1512.03385,2015.

[8] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift.arXiv preprint arXiv:1502.03167, 2015.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages1097–1105, 2012.

[10] M. Lin, Q. Chen, and S. Yan. Network in network. arXiv preprint arXiv:1312.4400, 2013.

[11] W.Liu,D.Anguelov,D.Erhan,C.Szegedy,andS.E.Reed. SSD: single shot multibox detector. CoRR, abs/1512.02325,2015. 5,6 [14] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller.Introductiontowordnet:Anon-linelexicaldatabase. International journal of lexicography, 3(4):235–244, 1990.