

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in Information Technology

By
LEELAVATHI V

38120045



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMILNADU



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" Grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.
Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344
www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of who carried out the project entitled "YOLO OBJECT DETECTION FROM THE IMAGE " under our supervision from Jan 2021 to April 2022

Internal Guide

DR. P. Jeyanthi

Head of the Department

Dr. R. Subhashini M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I hereby declare that the Project Report entitled "**YOLO OBJECT DETECTION FROM THE IMAGE**" done by me under the guidance of Dr. P. Jeyanthi at Sathyabama Institute of Science and Technology (Deemed to be University), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119 is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 22/01/2021

Place: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge our sincere thanks to Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey our thanks to **Dr. T. SASIKALA M.E., Ph.D., Dean, School of Computing** and **Dr. R. SUBHASHINI M.E., Ph.D., Head of the Department, Department of Information Technology** for providing us the necessary support and details at the right time during the progressive reviews.

I would like to express our sincere and deep sense of gratitude to our Project **Dr.P.Jeyanthi** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind Encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing ,**Dr.SUBHASHINI M.E., Ph.D. ,** Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. P.Jeyanthi ,for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



ABSTRACT

YOLO object detection from the image

Object detection is very useful in the future technology like AI, ML, DL etc. Object detection is one of the typical task of Computer vision The **purpose of computer vision** is to program a **computer** to "understand" a scene or features in an image. This is the real time project and it is very useful for robotics vision to detect the object.

CHAPTER NO	TITLE NAME	PAGE NO
	ABSTRACT	7
	LIST OF FIGURES	9
	LIST OF ABBREVIATION	10
1	INTRODUCTION	11
	1.1 GENERAL	11
	1.2 OUTLINE OF THE PROJECT	11
2	AIM AND SCOPE	12
	2.1 AIM	12
	2.2 PROBLEM STATEMENT	12
	2.3 SCOPE	12
3	SYSTEM ANALYSIS AND DESIGN	13
	3.1 GENERAL	13
	3.2 SOFTWARES USED	14
	3.3 ALGORITHM	17
	3.4 PROJECT DESCRIPTION	18
4	RESULT AND DISCUSSION	19
	4.1 SYSTEM REQUIREMENTS	19
	4.2 RESULT	19
5	CONCLUSION AND FUTURE WORK	20
	5.1 CONCLUSION	20
	5.2 FUTURE WORK	20
	5.3 REFERENCE	20
	APPENDIX	21
	A.SCREENSHOTS	21
	B.SOURCE CODE	23

FIGURE NO	FIGURE NAME	PAGE NO
3.1	DATA FLOW	18
4.1	OUTPUT	19

LIST OF ABBREVIATIONS

ABBREVIATIONS

EXPANSION

CV	COMPUTER VISION
YOLO	You only Look Once
XML	Extensible markup language

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Computer Vision is a field of artificial intelligence that trains computer to interpret and understand the visual world, using digital images from cameras and videos and deep learning models, Machine can accurately identify and classify object and then react to what they “see”. And the computer vision task include methods for acquiring processing, analyzing and understanding digital images and extraction of high-dimensional data from the real world in the order to produced numerical or symbolic information

AI and CV share other topics such as pattern recognition and learning techniques. In the CV there have a typical task like Recognition Motion analysis, scene reconstruction and the Image restoration

Recognition the classical problem in CV, image processing, and machine vision is that of determining whether or not the image data contains some specific object, feature or activity.

Detection the image data are scanned for a specific condition.

3.1 OUTLINE OF THE PROJECT

Computer Vision studies consist of different challenges when aiming to build a high-performance predictive model. It is crucial to select the right algorithms to solve the problem at hand, and in addition, the algorithms and the underlying platforms need to be capable of handling the volume of data. This project mainly focuses on creating the Detection of object by using the OpenCV recommendation system using YOLO algorithm to detect the object .and we load the algorithm by 3 xml files

CHAPTER 2

AIM AND SCOPE

2.1 AIM

The main objective of this project is to predicting the location of the object along with the class label by the bounding box.

2.2 PROBLEM STATEMENT

This project is designed for machine to trains computer to interpret and understand the visual world.

2.3 SCOPE

With further research on and refinement of the technology, the future of computer vision will see it perform a broader range of functions. Not only will computer vision technologies be easier to train but also be able to discern more from images than they do now. This can also be used in conjunction with other technologies or other subsets of AI to build more potent applications. For instance, image captioning applications can be combined with natural language generation (NLG) to interpret the objects in the surroundings for visually challenged people. Computer vision will also play a vital role in the development of artificial general intelligence (AGI) and artificial super intelligence (ASI) by giving them the ability to process information as well as or even better than the human visual system.

Considering the capabilities of present-day computer vision, it might be hard to believe that there are more benefits and applications of the technology that remain unexplored. The

future of computer vision will pave the way for artificial intelligence systems that are as human as us.

CHAPTER 3

SYSTEM ANALYSIS

3.1 GENERAL

3.1.1 Computer Vision

Computer Vision is a field of artificial intelligence that trains computer to interpret and understand the visual world, using digital images from cameras and videos and deep learning models, Machine can accurately identify and classify object and then react to what they “see”. And the computer vision task include methods for acquiring processing, analyzing and understanding digital images and extraction of high-dimensional data from the real world in the order to produced numerical or symbolic information

.

3.1.2 PYTHON

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

3.1.3 IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

IDE stands for Integrated Development Environment. It's a coding tool which allows you to write, test, and debug your code in an easier way, as they typically offer code completion or code insight by highlighting, resource management, debugging tools,... And even though the IDE is a strictly defined concept, it's starting to be redefined as other tools such as notebooks start gaining more and more features that traditionally belong to IDEs. The IDE used in this project is Anaconda Navigator.

3.1.4 OPENCV-Python

It is a library of python binding designed to solve Computer vision problems. All the Opencv array structures are converted to and from Numpy arrays

3.2 SOFTWARES USED

3.2.1 CODE EDITORS USED

Pycharm idl

- It is a dedicated python Integrated Development Environment (IDL) providing a wide
- Range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python , web, and data science development.

3.2.2 PROGRAMMING LANGUAGES USED

Python

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its

language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

3.2.3 LIBRARIES AND FRAMEWORKS USED

NUMPY IDL

Numpy is a Python used for working with arrays .It also has functions for working in domain of linear algebra, and matrices.

.

OPENCV-PYTHON

It is a library of python bindings designed to solve Computer vision problems All the OPENCV array structured are converted to and from NUMPY arrays.

3.3 ALGORITHM

3.3.1 YOLO

YOLO stands for “You Only Look Once”. This algorithm looks at the entire image in one go and detects objects.

In order to load the algorithm, we need these 3 files:

- **Weight file**: The trained model that detects the objects.
- **Cfg file**: The configuration file
- **Name files**: Consists of the names of the objects that this algorithm can detect

3.4 PROJECT DESCRIPTION

The aim of this project is to predicting the location of the object along with the class label by the bounding box. . This project mainly focuses on creating the Detection of object by using the OpenCV recommendation system using YOLO algorithm to detect the object . And the computer vision task include methods for acquiring processing, analyzing and understanding digital images and extraction of high-dimensional data from the real world in the order to produced numerical or symbolic information

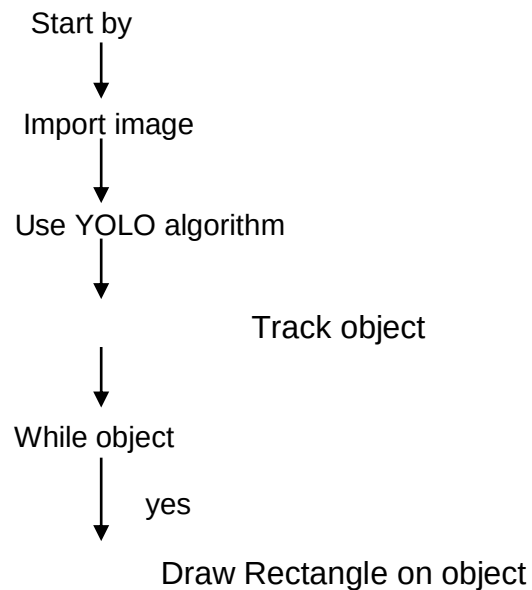


FIG 3.1 DATA FLOW

CHAPTER 4

RESULTS AND DISCUSSION

4.1 SYSTEM REQUIREMENTS

- Operating System: Windows/Mac
- RAM: 4GB
- Processor: 64x 1.0Ghz processor
- ROM: 8GB

4.2 RESULT

The Results shows that we can detected the image using by the YOLO algorithm

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

Object detection is a key ability for most computer and robot vision system , we are still far from achieving human-level performance in particular in terms of open-world learning

This project is undertaken using Computer vision and evaluates the performance by using YOLO(you only Look Once) algorithms. In our proposed model among our

algorithm is the to compare other algorithm

5.2 FUTURE WORK

This project describes the object detection is very useful in the technology like AI, ML, DL, etc. this typical task of computer vision is to program a computer to understand a scene or images by the ability of the algorithm. In future we can determine the efficient algorithm based on this topic that will helps to choose an efficient algorithm for object detection.

5.3 REFERENCES

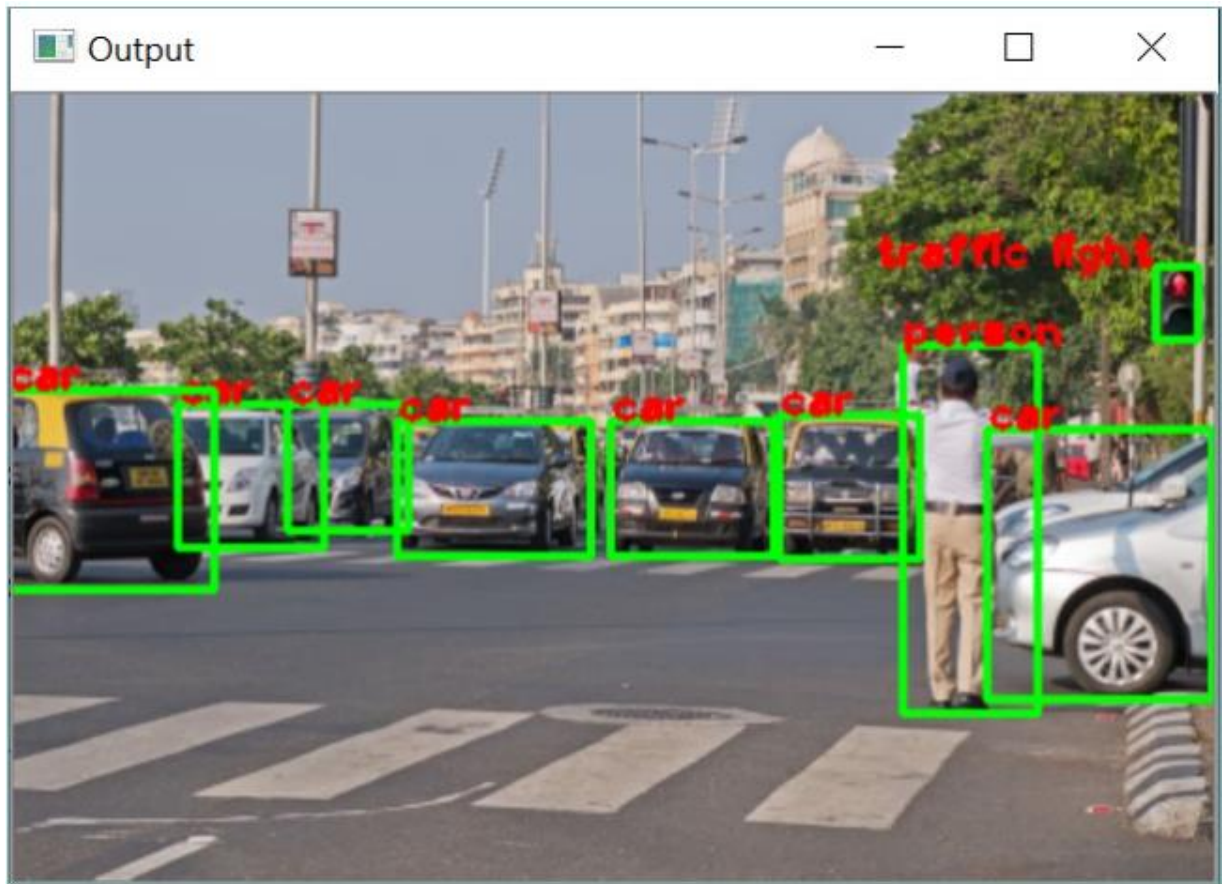
[Weight file](#)

[Cfg file](#)

[Name files](#)

APPENDIX

A.SCREENSHOTS



B.SOURCE CODE

```

import cv2
import numpy as np

#Load YOLO Algorithm
net=cv2.dnn.readNet("yolov3.weights","yolov3.cfg")

#To load all objects that have to be detected
classes=[]
with open("coco.names","r") as f:
    read=f.readlines()
for i in range(len(read)):
    classes.append(read[i].strip("\n"))

#Defining layer names
layer_names=net.getLayerNames()
output_layers=[]
for i in net.getUnconnectedOutLayers():
    output_layers.append(layer_names[i[0]-1])

#Loading the Image
img=cv2.imread("Road.jpg")
height,width,channels=img.shape

#Extracting features to detect objects
blob=cv2.dnn.blobFromImage(img,0.00392,(416,416),(0,0,0),True,crop=False)

#We need to pass the img_blob to the algorithm
net.setInput(blob)
outs=net.forward(output_layers)

```

```

#print(outs)
#Displaying informations on the screen
class_ids=[]
confidences=[]
boxes=[]
for output in outs:
    for detection in output:

        #Detecting confidence in 3 steps
        scores=detection[5:]          #1
        class_id=np.argmax(scores)    #2
        confidence =scores[class_id]  #3
        if confidence >0.5: #Means if the object is detected
            center_x=int(detection[0]*width)
            center_y=int(detection[1]*height)
            w=int(detection[2]*width)
            h=int(detection[3]*height)

            #Drawing a rectangle
            x=int(center_x-w/2) # top left value
            y=int(center_y-h/2) # top left value
            boxes.append([x,y,w,h])
            confidences.append(float(confidence))
            class_ids.append(class_id)

            #cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
#Removing Double Boxes
indexes=cv2.dnn.NMSBoxes(boxes,confidences,0.3,0.4)
for i in range(len(boxes)):
    if i in indexes:

```

```
x, y, w, h = boxes[i]
label = classes[class_ids[i]] # name of the objects
cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
cv2.putText(img, label, (x, y), cv2.FONT_HERSHEY_PLAIN,
1, (0, 0, 255), 2)

cv2.imshow("Output",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```