A Thesis/Project/Dissertation Report

on

TITLE OF THESIS/PROJECT/DISSERTATION

Submitted in partial fulfillment of the

requirement for the award of the degree of

Real – Time Object Detection

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. V. Gokul Rajan

Submitted By

S.No	**Enrollment Number**	**Admission Number**	**Student Name**	**Degree / Branch**	**Sem**
1	19021011414	19SCSE1010225	SALONI GUPTA	B-Tech /CSE	V
2	19021011625	19SCSE1010447	MAHAK AGGARWAL	B-Tech /CSE	V

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
2021-2022

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"CAPS…."** in partial fulfillment of the requirements for the award of the Galgotias University–submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Name… Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name
Designation

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination held on____________________and his/her work is recommended for the award of Name of Degree.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: November, 2013
Place: Greater Noida

Acknowledgement

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. Wewould like to extend our sincere thanks to all of them. We are highly indebted to Mr. V. Gokul Rajan for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude towards the member of Galgotias University for their kind co-operation and encouragement which help us in completion of this project.

We would like to express our special gratitude and thanks to industry persons for giving me such attention and time. Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

TABLE OF CONTENTS

Abstract

Due to object detection's close relationship with video analysis and image understanding, it has attracted much research attention in recent years. Object detection is a computer vision technique that works to identify and locate objects within an image or video. Specifically, object detection draws bounding boxes around these detected objects, which allow us to locate where said objects are in a given scene. Object detection is commonly confused with image recognition, so before we proceed, it's important that we clarify the distinctions between them. Object detection is inextricably linked to other similar computer vision techniques like image recognition and image segmentation, in that it helps us understand and analyze scenes in images or video. The main aim of this project is to build a system that detects objects (person, bicycle, car, motorbike, aeroplane, bus, train, truck, boat etc.) from the image or a stream of images given to the system in the form of previously recorded video or the real time input from the camera. Bounding boxes will be drawn around the objects that are being detected by the system. This paper presents a simplified approach to achieve this purpose using some basic Machine Learning packages like OpenCV, YOLO algorithm and python.

Keywords: Object, Python, OpenCV and Object Detection.

Literature Review

Since AlexNet has stormed the research world in 2012 ImageNet on a large scale visual recognition challenge, for detection in-depth learning, far exceeding the most traditional methods of artificial vision used in literature. In artificial vision, the neural convolution networks are distinguished in the classification of images.

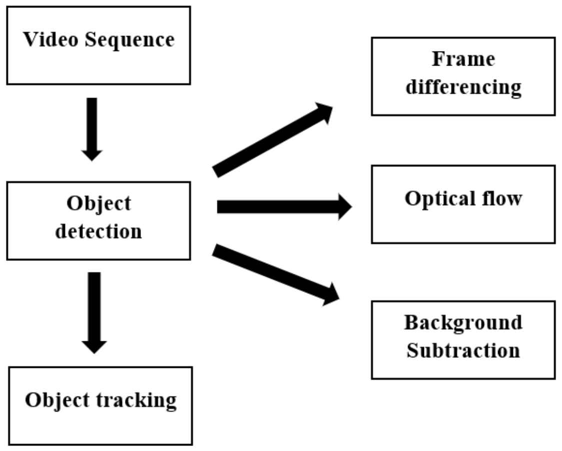


Fig. 1 Block diagram of Detection and Tracking

(Figure 1) shows the basic block diagram of detection and tracking. In this paper, an SSD and MobileNets based algorithms are implemented for detection and tracking in python environment. Object detection involves detecting region of interest of object from given class of image. Different methods are –Frame differencing, Optical flow, Background subtraction. This is a method of detecting and locating an object which is in motion with the help of a camera.

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 . YOLO based algorithm with GMM model by using the concepts of deep learning will give good accuracy for feature extraction and classification. Section II describes SSD and MobileNets algorithm, section III explains method of implementation, and section IV describes simulation results and analysis.

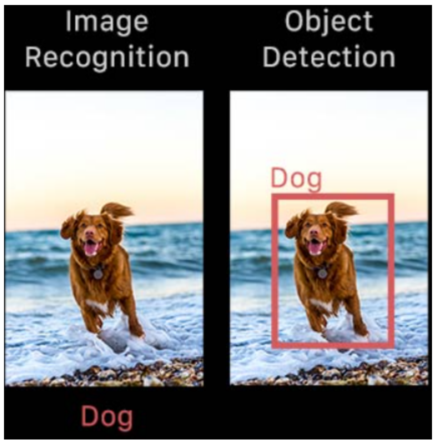


Fig. 2 Object Detection

What is object detection?

Object detection is a computer vision technique for locating instances of objects in images or videos. Object detection algorithms typically leverage machine learning or deep learning to produce meaningful results. The goal of object detection is to replicate this intelligence using a computer. (Figure 2)

Why is object detection important?

Object detection is inextricably linked to other similar computer vision techniques like image recognition and image segmentation, in that it helps us understand and analyze scenes in images or video. But there are important differences. Image recognition only outputs a class label for an identified object, and image segmentation creates a pixel-level understanding of a scene’s elements.

What separates object detection from these other tasks is its unique ability to locate objects within an image or video. This then allows us to count and then track those objects. Given these key distinctions and object detection’s unique capabilities, we can see how it can be applied in a number of ways:

- Crowd counting
- Self-driving cars
- Video surveillance
- Face detection

Problem Formulation

In this era, we face a lot of problem in late night driving we are not able to identify the thing that is coming in - front of us through our naked eyes or even our cars headlights are there and if we dropped something in the crowdy place. So using this object it would we easy for us to detect the object. Specifically, we'll examine how object detection can be used in the following areas:

• Video surveillance

• Crowd counting

• Self-driving cars

• Video surveillance

Because state-of-the-art object detection techniques can accurately identify and track multiple instances of a given object in a scene, these techniques naturally lend themselves to automating video surveillance systems.

For instance, object detection models are capable of tracking multiple people at once, in real-time, as they move through a given scene or across video frames. From retail stores to industrial factory floors, this kind of granular tracking could provide invaluable insights into security, worker performance and safety, retail foot traffic, and more.

• Crowd counting

Crowd counting is another valuable application of object detection. For densely populated areas like theme parks, malls, and city squares, object detection can help businesses and municipalities more effectively measure different kinds of traffic—whether on foot, in vehicles, or otherwise.

This ability to localize and track people as they maneuver through various spaces could help businesses optimize anything from logistics pipelines and inventory management, to store hours, to shift scheduling, and more. Similarly, object detection could help cities plan events, dedicate municipal resources, etc.

• **Self-driving cars**

Real-time car detection models are key to the success of autonomous vehicle systems. These systems need to be able to identify, locate, and track objects around them in order to move through the world safely and efficiently.

And while tasks like image segmentation can be applied to autonomous vehicles, object detection remains a foundational task that underpins current work on making self-driving cars a reality.

Required Tools

For the development and implementation of the project the tools we need is OpenCV, programming language is Python and there is a need of the algorithm that will we like a guider to our project that what to do and how to do. It will help in doing the project step by step and a good way.

What is Python?

Python is an object-oriented, high-quality language with active explanation. It is combined with powerful typing and active binding, makes it extremely attractive for faster Application Development, as well as for use as a writing. Python is simple and reduces system maintenance costs.

In this project we will be using python library namely OpenCV for the object detection. We can install OpenCV using pip the package installer in python. With the command pip3 install opencv-python. We are going to use the pip3 to install OpenCV as pip is the package installer in the python language.

What is OpenCV?

OpenCV is an open-source library dedicated to solving computer vision problems. Assuming you have python 3 pre-installed on your machines, the easiest way of installing OpenCV to python is via pip. You can do it by typing the below command line in your command prompt.

It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

What is YOLO Algorithm?

YOLO is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular because of its speed and accuracy. It has been used in various applications to detect traffic signals, people, parking meters, and animals. YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects. This means that prediction in the entire image is done in a single algorithm run.

The CNN is used to predict various class probabilities and bounding boxes simultaneously. The YOLO algorithm consists of various variants. Some of the common ones include tiny YOLO and YOLOv3. **(Figure 3)**

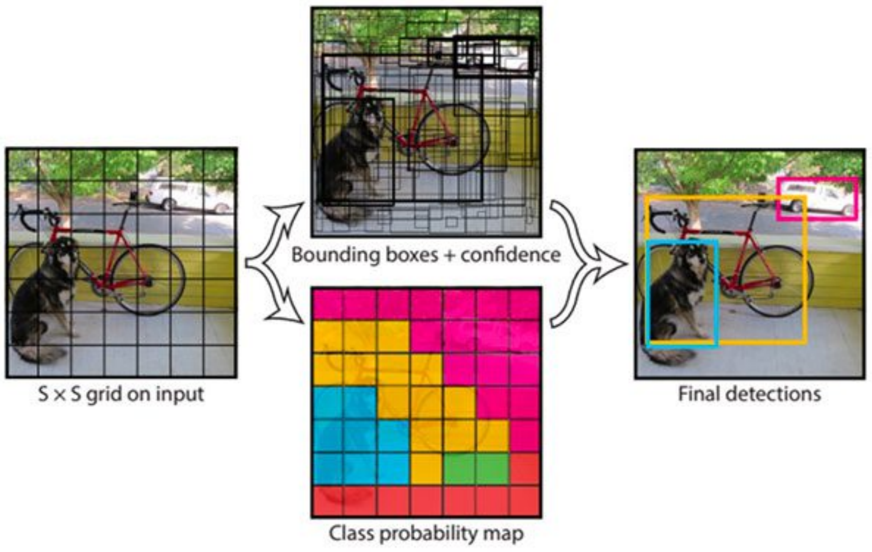
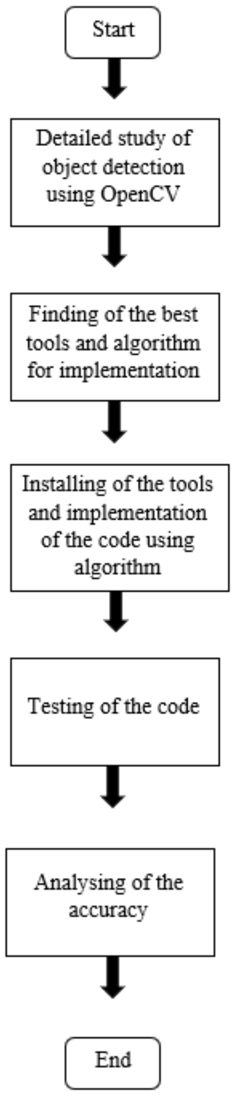


Fig. 3 YOLO Object Detection

What is Imutils?

Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and both Python 2.7 and Python 3.

Implementation Working Flowchart


Start
Detailed study of object detection using OpenCV
Finding of the best tools and algorithm for implementation
Installing of the tools and implementation of the code using algorithm
Testing of the code
Analysing of the accuracy
End

Merits Of Proposed Model

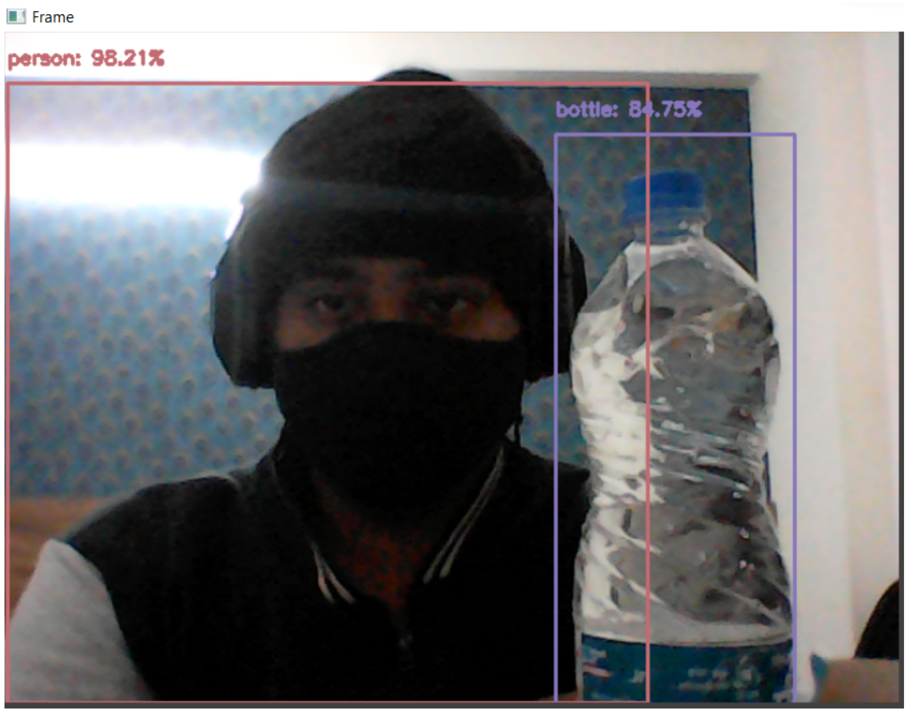
1. High detection precision.
2. Real - time detection, simple network structure.
3. Locate objects with bounding box.
4. The network can meet the real - time requirements with using the full image as context information.
5. The network is faster and the input image on be any size.

Implementation and Source Code

```
# python real_time_object_detection.py --prototxt
MobileNetSSD_deploy.prototxt.txt --model
MobileNetSSD_deploy.caffemodel

# import the necessary packages

from imutils.video import VideoStream
from imutils.video import FPS
import numpy as np
import argparse
import imutils
import time
import cv2
print("All libraries are imported...")

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()

ap.add_argument("-p", "--prototxt", required=True,
	help="path to Caffe 'deploy' prototxt file")

ap.add_argument("-m", "--model", required=True,
	help="path to Caffe pre-trained model")

ap.add_argument("-c", "--confidence", type=float, default=0.2,
	help="minimum probability to filter weak detections")

args = vars(ap.parse_args())

# initialize the list of class labels MobileNet SSD

# detect, then generate a set of bounding box colors for each
class

CLASSES = ["background", "aeroplane", "bicycle", "bird",
"boat",
	"bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
	"dog", "horse", "motorbike", "person", "pottedplant",
"sheep",
	"sofa", "train", "tvmonitor"]

COLORS = np.random.uniform(0, 255, size=(len(CLASSES),
3))
```

```
# load our serialized model from disk

print("Loading MobileNet model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"],
args["model"])

# initialize the video stream, allow the cammera sensor and
initialize the FPS counter

print("Opening Video wait...")
vs = VideoStream(src=0).start()
time.sleep(2.0)
fps = FPS().start()

print("Minimize the all tabs...")

# loop over the frames from the video stream

while True:
	# grab the frame from the threaded video stream and resize
it to have a maximum width of 400 pixels
	frame = vs.read()
	frame = imutils.resize(frame, width=700)
	frame = cv2.flip(frame, 1)
	f1 = vs.read()
	f1 = imutils.resize(f1, width=700)
	f1 = cv2.flip(f1, 1)

	# grab the frame dimensions and convert it to a blob
	(h, w) = frame.shape[:2]
	blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300,
300)),
		0.007843, (300, 300), 127.5)

	# pass the blob through the network and obtain the
detections and predictions
	net.setInput(blob)
	detections = net.forward()

	# loop over the detections
	for i in np.arange(0, detections.shape[2]):

		# extract the confidence (i.e., probability) associated
with the prediction
		confidence = detections[0, 0, i, 2]
		# filter out weak detections by ensuring the
`confidence` is greater than the minimum confidence
		if confidence > args["confidence"]:

			# extract the index of the class label from and
the bounding box for the object
			idx = int(detections[0, 0, i, 1])
			box = detections[0, 0, i, 3:7] * np.array([w, h,
w, h])
			(startX, startY, endX, endY) =
box.astype("int")

			# draw the prediction on the frame
			label = "{}: {:.2f}%".format(CLASSES[idx],
				confidence * 100)
			cv2.rectangle(frame, (startX, startY), (endX,
endY),
				COLORS[idx], 2)
			y = startY - 15 if startY - 15 > 15 else startY +
15
			cv2.putText(frame, label, (startX, y),
				cv2.FONT_HERSHEY_SIMPLEX, 0.5,
COLORS[idx], 2)

	# show the output frame
	cv2.imshow("Normal_Frame",f1)
	cv2.imshow("Frame", frame)
	key = cv2.waitKey(1) & 0xFF

	# if the `q` key was pressed, break from the loop
	if key == ord("q"):
		break

	# update the FPS counter
	fps.update()

# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

Output and Result



We have successfully implemented our project with the accuracy of around 80% to 85%.

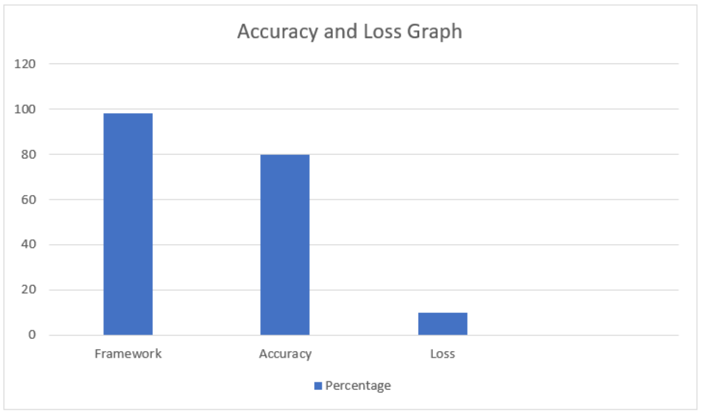


Fig 4

In the result we come to know that the real time object detection is easy, more accurate with the help of OpenCV and YOLO.

As we can in the above graph the framework and accuracy percentage are high. And loss is less. **(Figure 4)**

References

[1]. L. Carminati, J. Benois-Pineau and C. Jennewein,“Knowledge-Based Supervised Learning Methods in a Classical Problem of Video Object Tracking”, 2006
International Conference on Image Processing, Atlanta,GA, USA, ISSN (e): 2381-8549, year-2006.

[2]. Jinsu Lee, Junseong Bang and Seong-II Yang, “Object Detection with Sliding Window in Images including Multiple Similar Object”, 2017 IEEE International
Conference on Information and Communication Technology Convergence (ICTC), Jeju, South Korea, ISBN (e): 978-1-5090-4032-2, December-2017.

[3] Qichang Hu, Sakrapee Paisitkriangkrai, Chunhua Shen, Anton van den Hengel and Faith Porikli,“Fast Detection of Multiple Objects in Traffic Scenes with Common
Detection Framework”, IEEE Transactions on Intelligent Transportation Systems, ISSN (e): 1558-0016, Vol-17, Issue-04, Year-2016, pp. 1002-1014.

[4].https://jonathan-hui.medium.com/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359

[5] Justin Lai, Sydney Maples, “Ammunition Detection: Developing a Real-Time Gun Detection Classifier”, Stanford University, Feb 2017.

[6] Shreyamsh Kamate, “UAV: Application of Object Detection and Tracking Techniques for Unmanned Aerial Vehicles”, Texas A&M
University, 2015.

[7] Adrian Rosebrock, “Object detection with deep learning and OpenCV”, pyimagesearch.

[8] Mohana and H. V. R. Aradhya, "Elegant and efficient algorithms for real time object detection, counting and classification for video surveillance applications from single fixed camera," 2016 International Conference on Circuits, Controls, Communications and Computing (I4C), Bangalore, 2016, pp. 1-7.

[9] Akshay Mangawati, Mohana, Mohammed Leesan, H. V. Ravish Aradhya, “Object Tracking Algorithms for video surveillance applications” International conference on communication and signal processing (ICCSP), India, 2018, pp. 0676-0680.

[10] Apoorva Raghunandan, Mohana, Pakala Raghav and H. V. Ravish Aradhya, “Object Detection Algorithms for video surveillance applications” International conference on communication and signal processing (ICCSP), India, 2018, pp. 0570-0575.

[11] Manjunath Jogin, Mohana, “Feature extraction using Convolution Neural Networks (CNN) and Deep Learning” 2018 IEEE International Conference On Recent Trends In Electronics Information Communication Technology,(RTEICT) 2018, India.

[12] Arka Prava Jana, Abhiraj Biswas, Mohana, “YOLO based Detection and Classification of Objects in video records” 2018 IEEE International Conference On Recent Trends In Electronics Information Communication Technology,(RTEICT) 2018, India.