A FINAL REPORT ON

"REAL TIME OBJECT DETECTION & RECOGNITION USING MOBILENET-SSD WITH OPENCV"

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

HARSHAL GANESH HONMOTE	Exam No: B150734258
SHREYAS DATTATRAY GADEKAR	Exam No: B150734245
PRANAV RAVINDRA KATTA	Exam No: B150734274

UNDER THE GUIDANCE OF

Prof. Madhavi Kulkarni

DEPARTMENT OF COMPUTER ENGINEERING

JSPM'S, BHIVARABAI SAWANT INSTITUTE OF TECHNOLOGY & RESEARCH

GATE NO: 720/1&2, NAGAR ROAD, WAGHOLI, PUNE-412207

SAVITRIBAI PHULE PUNE UNIVERSITY

2021 -2022

CERTIFICATE

This is to certify that the project report entitles

“REAL TIME OBJECT DETECTION AND RECOGNITION USING MOBILENET-SSD WITH OPENCV”

Submitted by

HARSHAL GANESH HONMOTE	Exam No: B150734258
SHREYAS DATTATRAY GADEKAR	Exam No: B150734245
PRANAV RAVINDRA KATTA	Exam No: B150734274

Are bonafide student of this institute and the work has been carried out by them under the supervision of **Prof.Madhavi Kulkarni** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

(Prof. Madhavi Kulkarni)
Guide
Dpt. of Computer Engineering

(Dr. G. M. Bhandari)
Head,
Dpt. of Computer Engineering

(Dr. T. K. Nagaraj)
Principal,
JSPM’s, Bhivarabai Sawant Institute of Technology & Research, Wagholi, Pune-412207

Place: Pune

Date:

ACKNOWLEDGEMENT

This is a great pleasure & immense satisfaction to express our deepest sense of gratitude & thanks to everyone who has directly or indirectly helped us in completing our Project work successfully.

We express our gratitude towards guide **Prof**. Madhavi Kulkarni and **Dr**.**Prof**. G. M. Bhandari Head of Department of Computer Engineering, Bhivarabai Sawant Institute of Technology and Research, Wagholi, Pune who guided & encouraged us in completing the Project work in scheduled time. We would like to thanks our principal, for allowing us to pursue our Project in this institute.

HARSHAL GANESH HONMOTE	Exam No: B150734258
SHREYAS DATTATRAY GADEKAR	Exam No: B150734245
PRANAV RAVINDRA KATTA	Exam No: B150734274

Department Of Computer Engineering

ABSTRACT

Mobile networks and binary neural networks are the most commonly used techniques for modern deep learning models to perform a variety of tasks on embedded systems. In this paper, we develop a technique to identify an object considering the deep learning pre-trained model MobileNet for Single Shot Multi-Box Detector (SSD). This algorithm is used for real-time detection, and for webcam feed to detect the purpose webcam which detects the object in a video stream. Therefore, we use an object detection module that can detect what is in the video stream. In order to implement the module, we combine the MobileNet and the SSD framework for a fast and efficient deep learning-based method of object detection.

The main purpose of our research is to elaborate the accuracy of an object detection method SSD and the importance of pre-trained deep learning model MobileNet. The experimental results show that the Average Precision (AP) of the algorithm to detect different classes as car, person and chair is 99.76%, 97.76% and 71.07%, respectively.

This improves the accuracy of behavior detection at a processing speed which is required for the real-time Detection and the requirements of daily monitoring indoor and outdoor.

TABLE OF CONTENTS

LIST OF ABBREVATIONS	i
LIST OF FIGURES	ii
LIST OF TABLES	iii

Sr. No.	Title Of Chapter	Page No.
1	**Introduction**	10
1.1	**Motivation**	11
1.2	**Problem Definition**	11
2	**Literature Survey**	12
3	**Software Requirements Specification**	14
3.1	Introduction	14
	3.1.1 Project Scope	14
	3.1.2 Overview Of Responsibilities of Developer	14
3.2	Data Model and Description	15
	3.2.1 Data Description	15
	3.2.2 Data Objects and Relationships	15
3.3	**Functional Model and Description**	15
3.4	Non-Functional Requirements	16
3.5	Design Constraints	17
3.7	**Software Interface Description**	17
3.8	System Requirements	17
	3.8.2 Software Requirements	17
	3.8.3 Hardware Requirements	17
3.9	Requirement Analysis Model	18
3.10	System Implementation Plan	20
4	**System Design**	21
4.1	Introduction	21
4.2	Proposed System	21
4.3	SSD Architectural	21
4.4	Concept Used	22
4.5	Machine Learning	31
4.6	Steps Involved in Object Detection	36
4.7	Algorithm	37

5		**Project Plan**	38
	5.1	Project Estimates	38
		5.1.1 Reconciled Estimates	38
		5.1.1.1 Cost Estimate	38
		5.1.1.2 Time Estimates	38
		5.1.2 Project Resources	39
	5.2	Risk Management	40
	5.3	Project Schedule	41
		5.3.1 Project Task Set	41
		5.3.2 Task Network	42
	5.4	Team Organization	43
		5.4.1 Team Structure	43
		5.4.2 Management Reporting and Communication	43
6		**UML Diagrams**	44
7		**Testing**	50
	7.1	Black Box Testing (BBT):	50
	7.2	White Box Testing (WBT):	51
8		UI APP Screenshots	52
9		Other Specification	54
10		Conclusion And Future Work	55
11		References	56
12		Certificates	57

LIST OF ABBREVATIONS

ABBREVIATION	ILLUSTRATION
SSD	Single Shot Detector
BBT	Black Box Testing
WBT	White Box Testing

LIST OF FIGURES

FIGURE	ILLUSTRATION	PAGE NO.
3.1	Prototyping Model	18
3.2	Evolutionary Prototyping Model	19
3.3	System Implementation Plan	20
4.1	Proposed System	21
4.2	SSD Architecture	21
4.3	NumPy Features	23
4.4	Python Date & Time	26
4.5	OpenCV	28
4.6	Application of ML	34
5.1	Test Network	42
6.1	UML Class Diagram	45
6.2	UML State Diagram	46
6.3	UML Activity Diagram	47
6.4	UML Sequence Diagram	48
6.5	Data flow Diagram	49
8.1	Run Command	52
8.2	User Interface	52
8.3	Image with Detected Object	53
8.4	Output Obtained in real Time	53

LIST OF TABLES

TABLE	ILLUSTRATION	PAGE NO.
5.1	Time Estimates	38
5.2	Project Resources	39
5.3	Team Organization	43
7.3	Test Cases with Results	51

1. INTRODUCTION

The field of science is as big as the universe itself. Every passing day there are new developments if not big or groundbreaking, but constructive and leading towards a better tomorrow. Object detection is one of the most prominent fields of research in computer vision today. It is an extension of image classification, where the goal is to identify one or more classes of objects in an image and localize their presence with the help of bounding boxes.

Hence, object detection plays a vital role in many real-world applications such as image retrieval and video surveillance, to simply name a few. With this in mind, the main aim of our project is to investigate the inner workings of the "Single Shot MultiBox Detector" (SSD) framework for object detection. Our objective is to highlight some of the salient features that make this technique stand out as well as to address a few of its shortcomings as will be discussed in more detail in the rest of this post.

The integration of MobileNet into the SSD framework forms one of the core aspects of our work. However, it is worth pointing out that the combination of a highly efficient base network such as the MobileNet with the supremely effective SSD framework has been a hot research topic in recent times, largely due to dealing with the practical limitations of running powerful neural nets on low-end devices such as mobile phones/laptops to further extend the myriad of possibilities with regards to real-time applications.

1.1 MOTIVATION

Object detection is breaking into a wide scope of industries, with use cases going from personal security to efficiency in the working environment. Object detection and recognition is applied in numerous areas of computer vision, including image recovery, security surveillance, automated vehicle systems. For a vehicle to choose what to do what to do next: speed up, apply brakes or turn, it needs to know where all the objects are around the vehicle and what
those objects are That requires object detection
The information from the object detector can be used for obstacle avoidance and other interactions with the environment.

1.2 PROBLEM DEFINITIION

To detect Image in Real Time Faster, Accurately with Less memory consumption and reduce traffic consumption over Web Servers for classification of Image.

2. LITERATURE SURVEY

1. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications 2017:

   This paper proposes a new model architecture called MobileNets based on depth wise separable convolutions. The investigated some of the important design decisions leading to an efficient model. We then demonstrated how to build smaller and faster MobileNets using width multiplier and resolution multiplier by trading off a reasonable amount of accuracy to reduce size and latency. Then compared different MobileNets to popular models demonstrating superior size, speed and accuracy characteristics. Concluded by demonstrating MobileNet's effectiveness when applied to a wide variety of tasks. As a next step to help adoption and exploration of MobileNets, we plan on releasing models in Tensor Flow.

   In many real-world applications such as robotics, self-driving car and augmented reality, the recognition tasks need to be carried out in a timely fashion on a computationally limited platform. This paper describes an efficient network architecture and a set of two hyper-parameters in order to build very small, low latency models that can be easily matched to the design requirements for mobile and embedded vision applications.

2. Explicit Content Detection using Faster R-CNN and SSD MobileNet v2 Animesh Srivastava1, Anuj Dalvi2, Cyrus Britto3, Harshit Rai4, Kavita Shelke5 2020:

   This paper compares Faster R-CNN and SSD MobileNet v2, both object detection models to detect explicit content from an image in terms of speed, accuracy and model size. Instead of using selective search algorithm, used the in slower and time-consuming Fast RCNN, on the feature map to identify the region proposals, a separate network is used to predict the region proposals. The predicted region proposals are then reshaped using a RoI pooling layer which is then used to classify the image within the proposed region and predict the offset values for the bounding boxes, making Faster RCNNs ideal for real-time object detection. Hence, MobileNet v2 can be used in real time object detection.

3. Shoe Detection Using SSD-MobileNet Architecture 2020:

This Paper, proposes a deep learning model to detect in real time the position of the shoes in images. The system could detect the shoes with an average precision similar to other state of the art systems.

Paper presents a shoe detection model based on the SSD-MobileNetV2 architecture proposed in the proposed detection model was trained and tested using 10.000 images containing shoes from the Open Images V5 dataset. This research constitutes our initial work on the development of as system to combine camera and radar data to improve the accuracy of our feet position estimation system. Therefore, the proposed shoe detector has to be accurate but also it needs to be able to run in real time even in low computation devices.

4. Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems:

This paper aims to lightweight network architecture with improved feature extraction based on the Mobilenet-v2 backbone network. We combine Mobilenet-v2 and FPN models to enhance the feature map of the input image and effectively improve the detection accuracy of the back-end detection network

The paper would give an overall view of the Mobilenet-SSDv2 detector not only retains the advantage of fast processing of the original MobileNet-SSD detector, but also greatly improves the detection accuracy. These advantages indicate that the Mobilenet-SSDv2 detection model proposed in this paper is more suitable for embedded.

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 INTRODUCTION

3.1.1 PURPOSE AND SCOPE OF DOCUMENT

- **Purpose:** A software requirements specification (SRS) is a description of a software system to be developed, laying out function and non-function requirements, and may include a set of use cases that describe interaction the users will have with the software. Software requirement specification establishes the basis for an agreement between customer and contractor or suppliers (in market-driven projects, these roles may be played by the marketing and development division) on do. Software requirement specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

- **Scope:** The software requirement specification document enlists enough and necessary requirements that are re-quired for the project development. To derive the requirements, we need to have clear and thorough understanding of products to be developed or being develop. This is achieved refined with detailed and continuous commune- cation with the project team and customer till the completion of the software. The SRS may be one of a contract deliverable Data item Description or have other forms of organizationally mandated content.

3.1.2 OVERVIEW OF RESPONSIBILITIES OF DEVELOPER

Following activities are carried out:

- Design: Harshal Honmote, Shreyash Gadekar, Pranav Katta
- Implementation: Harshal Honmote, Shreyash Gadekar, Pranav Katta
- Testing: Harshal Honmote, Shreyash Gadekar
- Documentation: Shreyash Gadekar, Pranav Katta

3.2 DATA MODEL AND DESCRIPTION

3.2.1 DATA DESCRIPTION

Data objects that will be managed/manipulated by the software are described in this section. The database entities or files or data structures required to be described. For data objects details can be given as below

3.2.2 DATA OBJECTS AND RELATIONSHIPS

Data objects and their major attributes and relationships among data objects are de-scribed using an ERD- like form.

3.3 FUNCTIONAL MODEL AND DESCRIPTION

A description of each major software function, along with data low (structured analysis) or class hierarchy (Analysis Class diagram with class description for object-oriented system) is presented.

3.4 NON-FUNCTIONAL REQUIREMENTS:

Performance Requirements:

- System can produce results faster on 4GB of RAM.
- It may take more time for peak loads at main node.
- The system will be available 100% of the time. Once there is a fatal error, the system will provide understandable feedback to the user.

Safety Requirements:

The system is designed in modules where errors can be detected and fixed easily.

Security Requirements:

The system is designed in modules where errors can be detected and fixed easily.

Software Quality Attributes

- Usability**:**

This relates to how easily people can use your system. A measure of usability could be the time it takes for end users to become familiar with your system’s functions, without training or help.

- Reliability**:**

This is the percentage of time that your app works correctly to deliver the desired results, despite potential failures in its environment.

- Performance**:**

This is essentially how fast your system works. A performance requirement for the system could be start in less than 20 second.

3.5 DESIGN CONSTRAINTS

Any design constraints that will impact the subsystem are noted. There is a necessity to study design patterns in detail. Depending upon the various kinds of patterns available, different design constraints may be encountered such as supporting multiple operating systems which may not be possible. The schedule is tight and deadlines prove a reasonable constraint while adding features.

3.6 SOFTWARE INTERFACE DESCRIPTION

The interface must be easy to understand and use. It must be intuitive and explain the use at a glance.

3.7 SYSTEM REQUIREMENTS

3.7.1 SOTWARE REQUIREMENTS:

- Operating system: windows 8 or above
- Programming Language: python 3.8
- Visual Studio Code

3.7.2 HARDWARE REQUIREMENTS:

- System type: 64-bit operating system
- Processor: intel(r) core i3 or above.
- Installed memory (Ram): 4.00gb or higher
- Hard Disk: 40gb or above
- Camera/Webcam

3.8 REQUIREMENT ANLYSIS MODEL

Prototyping Model

It refers to the activity of creating prototypes of software applications, for example, incomplete versions of the software program being developed. It is an activity that can occur in software development and it used to visualize some component of the software to limit the gap of misunderstanding the customer requirements by the development team. This also will reduce the iterations may occur in the waterfall approach and hard to be implemented due to the inflexibility of the waterfall approach. So, when the final prototype is developed, the requirement is considered to be frozen.

It has some types, such as:

Throwaway prototyping: Prototypes that are eventually discarded rather than becoming a part of the finally delivered software.

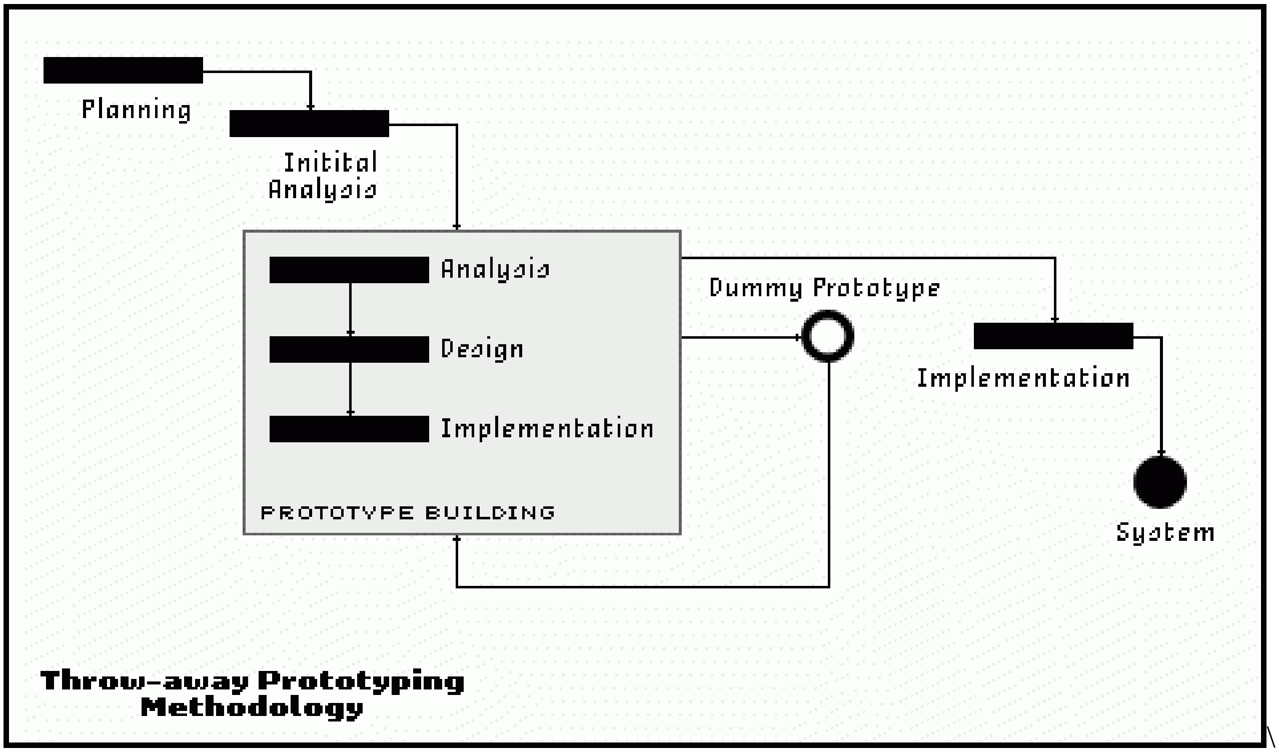


Fig 3.1 Prototyping Model

- Evolutionary prototyping: prototypes that evolve into the final system through an iterative incorporation of user feedback.

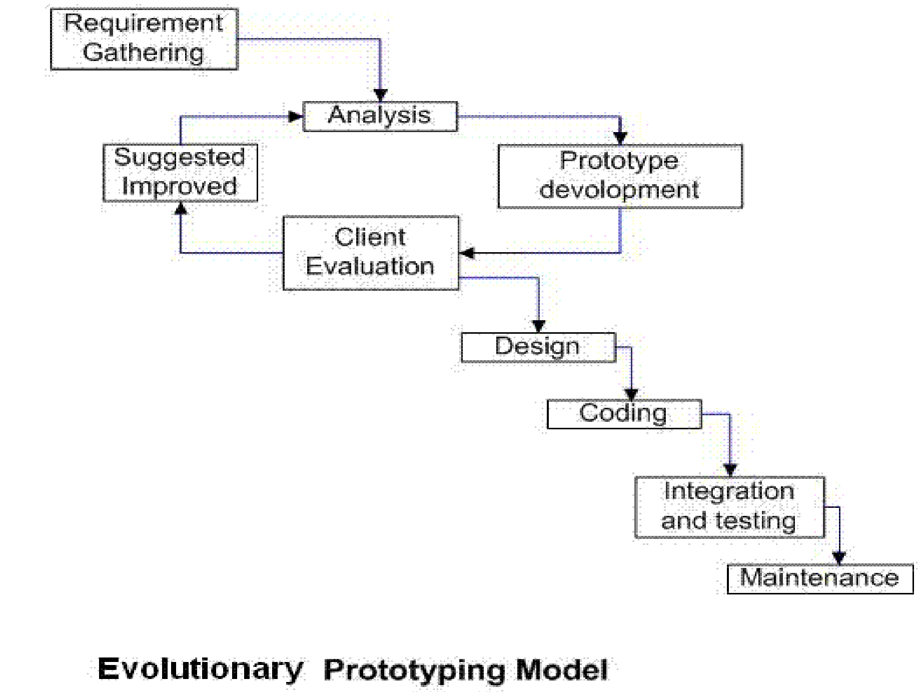


Fig 3.2 Evolutionary Prototyping Model

Advantages:

- Reduced time and costs, but this can be a disadvantage if the developer loses time in developing the prototypes.
- Improved and increased user involvement.

Disadvantages:

- Insufficient analysis. User confusion of prototype and finished system.
- Developer misunderstanding of user objectives.
- Excessive development time of the prototype.
- It is costly to implement the prototypes.

3.9 SYSTEM IMPLEMENTATION PLAN:

We are going to design our system using Python programming language on Windows platform.

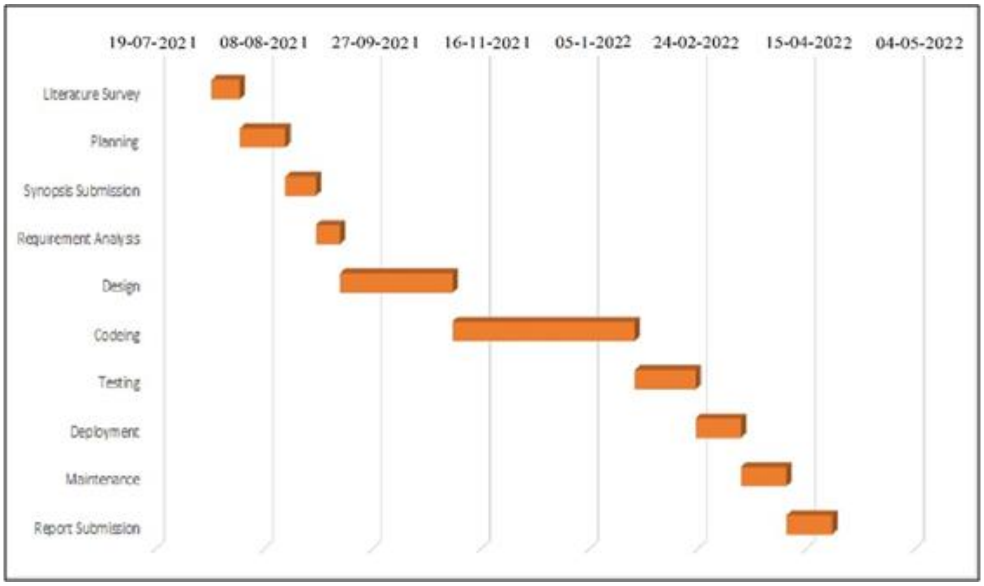


Fig 3.3 System Implementation Plan.

4. SYSTEM DESIGN

4.1 INTRODUCTION

This document specifies the design that is used to solve the problem of Product.

4.2 PROPOSED SYSTEM

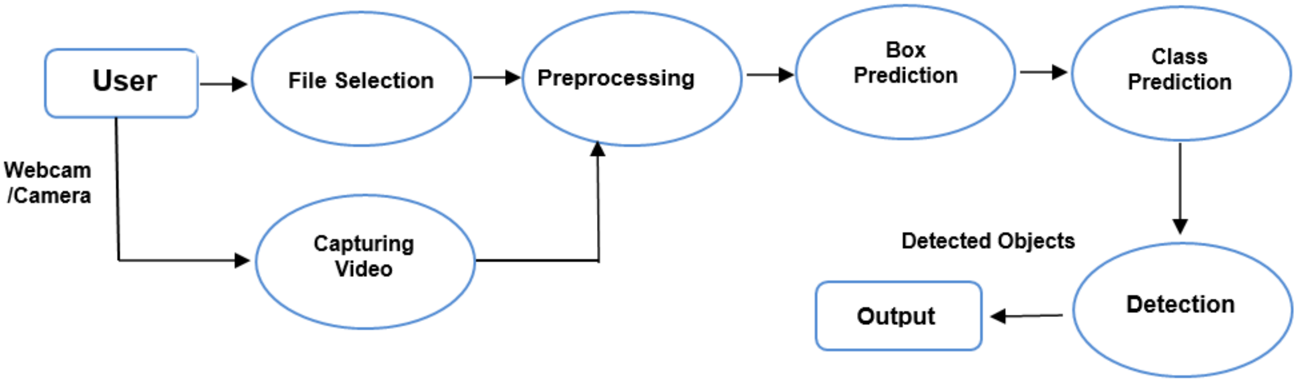
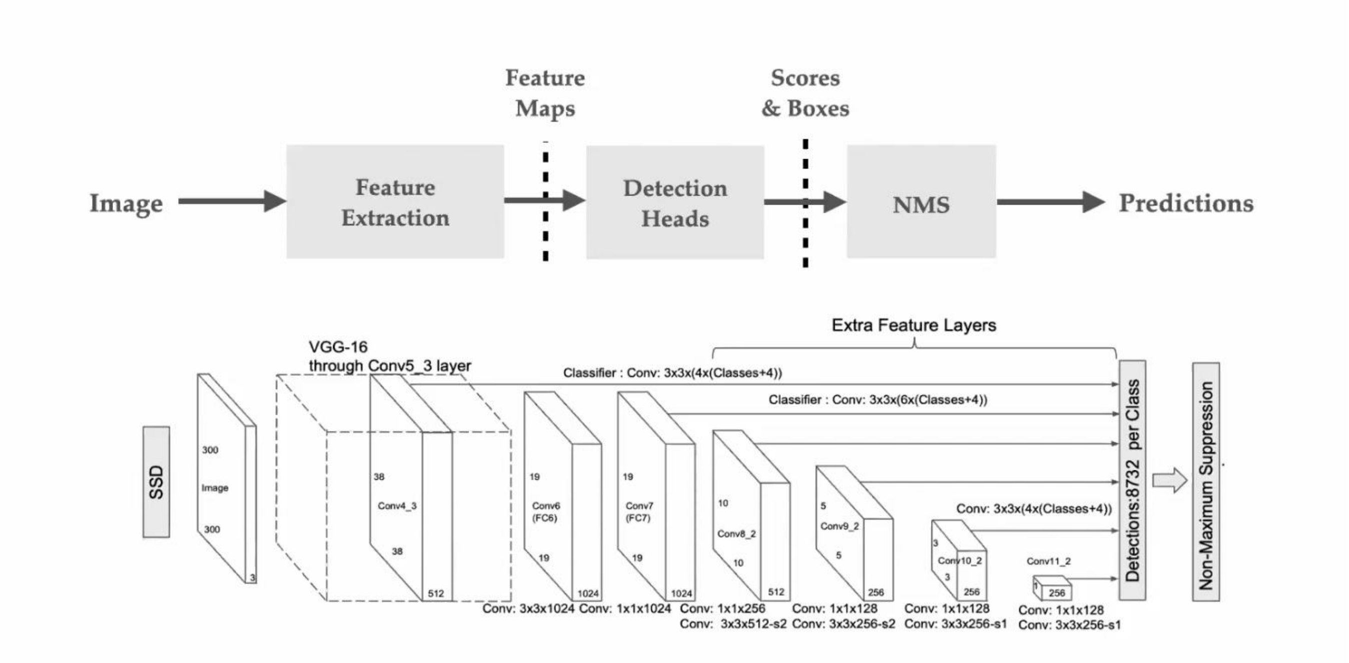


Fig 4.1 Proposed System

4.3 SSD ARCHITECTURE



4.4 CONCEPTS/PACKAGES USED

- NUMPY
- ARGPARSE
- IMUTILS
- TIME
- CV2
- PILLOW
- TKINTER

NumPy:

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Num array was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Num array into Numeric package. There are many contributors to this open-source project.

Operations using NumPy:

Using NumPy, a developer can perform the following operations −

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.
- Where source is the object that generates event.

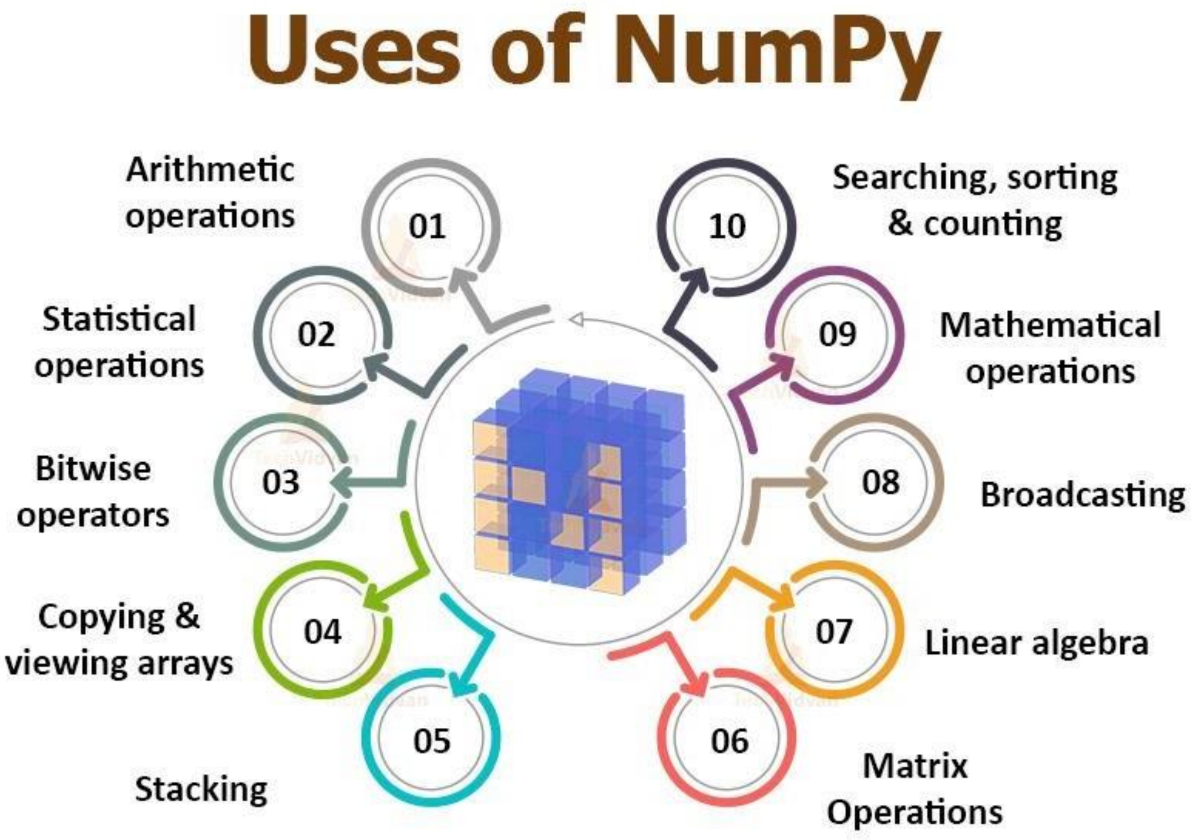


Fig.4.3 NumPy Features

<u>Argparse</u>: -

The argparse module in Python helps create a program in a command-line-environment in a way that appears not only easy to code but also improves interaction. The argparse module also automatically generates help and usage messages and issues errors when users give the program invalid arguments.

Why Use argparse?

argparse — parse the arguments. Using argparse means the doesn't need to go into the code and make changes to the script. Giving the user the ability to enter command line arguments provides flexibility.

<u>Imutils</u>:

Imutils are a series of convenience functions to make basic image processing OpenCV and both Python 2.7 and Python3.

Translation

Translation is the shifting of an image in either the *x* or *y* direction. To translate an image in OpenCV you would need to supply the *(x, y)*-shift, denoted as (t_x, t_y) to construct the translation matrix *M*:

Rotation

Rotating an image in OpenCV is accomplished by making a call to cv2.getRotationMatrix2D and cv2.warpAffine. Further care has to be taken to supply the *(x, y)*-coordinate of the point the image is to be rotated about. These calculation calls can quick add up and make your code bulky and less readable. The rotate function in imutils helps resolve this problem.

Resizing

Resizing an image in OpenCV is accomplished by calling the cv2.resize function. However, special care needs to be taken to ensure that the aspect ratio is maintained. This resizes function of imutils maintains the aspect ratio and provides the keyword arguments width and height so the image can be resized to the intended width/height while (1) maintaining aspect ratio and ensuring the dimensions of the image do not have to be explicitly computed by the developer.

Skeletonization

Skeletonization is the process of constructing the "topological skeleton" of an object in an image, where the object is presumed to be white on a black background. OpenCV does not provide a function to explicitly construct the skeleton, but does provide the morphological and binary functions to do so. For convenience, the skeletonize function of imutils can be used to construct the topological skeleton of the image. The first argument, size is the size of the structuring element kernel. An optional argument, structuring, can be used to control the structuring element -- it defaults to cv2.MORPH_RECT, but can be any valid structuring element.

Time:

As the name suggests Python time module allows to work with time in Python. It allows functionality like getting the current time, pausing the Program from executing, etc. So before starting with this module we need to import it.

The time module comes with Python's standard utility module, so there is no need to install it externally. We can simply import it using the import statement.

Sleep Method

Execution can be delayed using **time.sleep()** method. This method is used to halt the program execution for the time specified in the arguments.

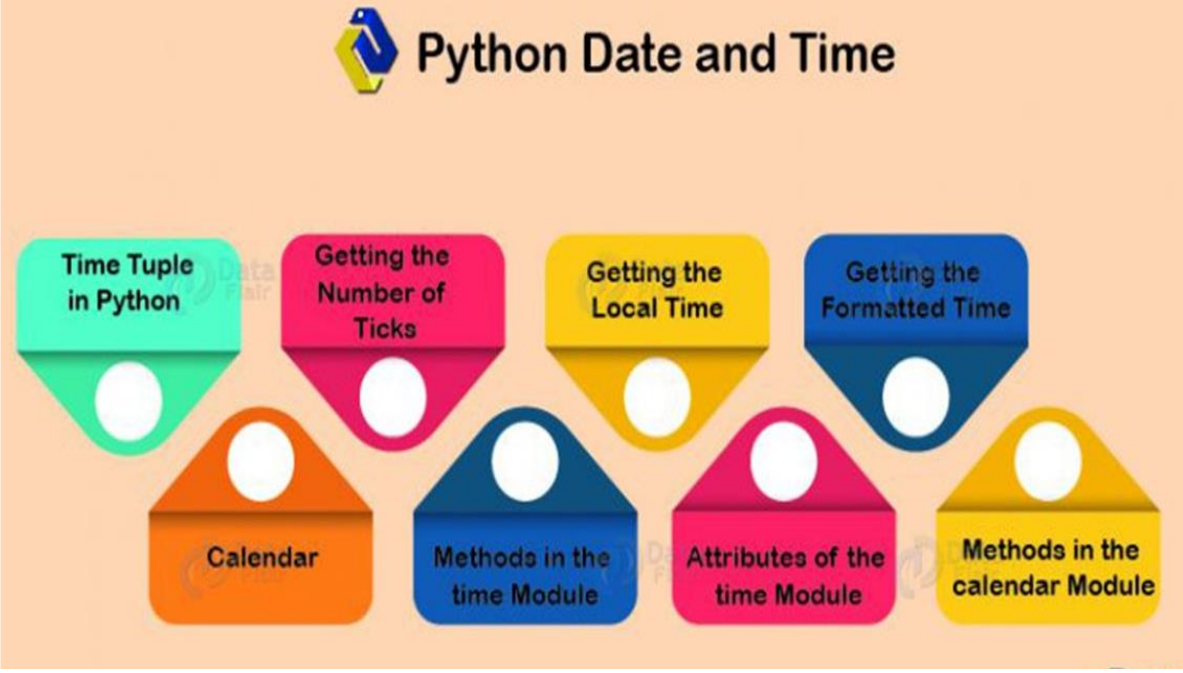

Python Date and Time
Time Tuple in Python
Getting the Number of Ticks
Getting the Local Time
Getting the Formatted Time
Calendar
Methods in the time Module
Attributes of the time Module
Methods in the calendar Module

Fig. 4.4 Python Date & Time

CV2:

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e., whatever operations one can do in Numpy can be combined with OpenCV.

This OpenCV tutorial will help you learn the Image-processing from Basics to Advance, like operations on Images, Videos using a huge set of Opencv-programs and projects.

Why cv2 is used in Python?

OpenCV stands for Open-Source Computer Vision Library, which is widely used for image recognition or identification. It was officially launched in 1999 by Intel. It was written in C/C++ in the early stage, but now it is commonly used in Python for the computer vision as well.

Imread Method:

cv2.imread() method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.

readNetFromCaffe Method:

cv2.dnn.readNetFromCaffe module to load our model. You will need these two types of files to work with any pre-trained model using dnn module:

1. . prototxt file: They basically contain a list of the network layers in the model that you’re using.
2. caffemodel file (in your case it might not be a Caffe model): This file contains the weights of the model.

You need both of these files to create the model, we’ll pass these two files as arguments to the **cv2.dnn.readNetFromCaffe** module to create our model.

Imshow Method:

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. cv2.imshow() method is used to display an image in a window. The window automatically fits to the image size.

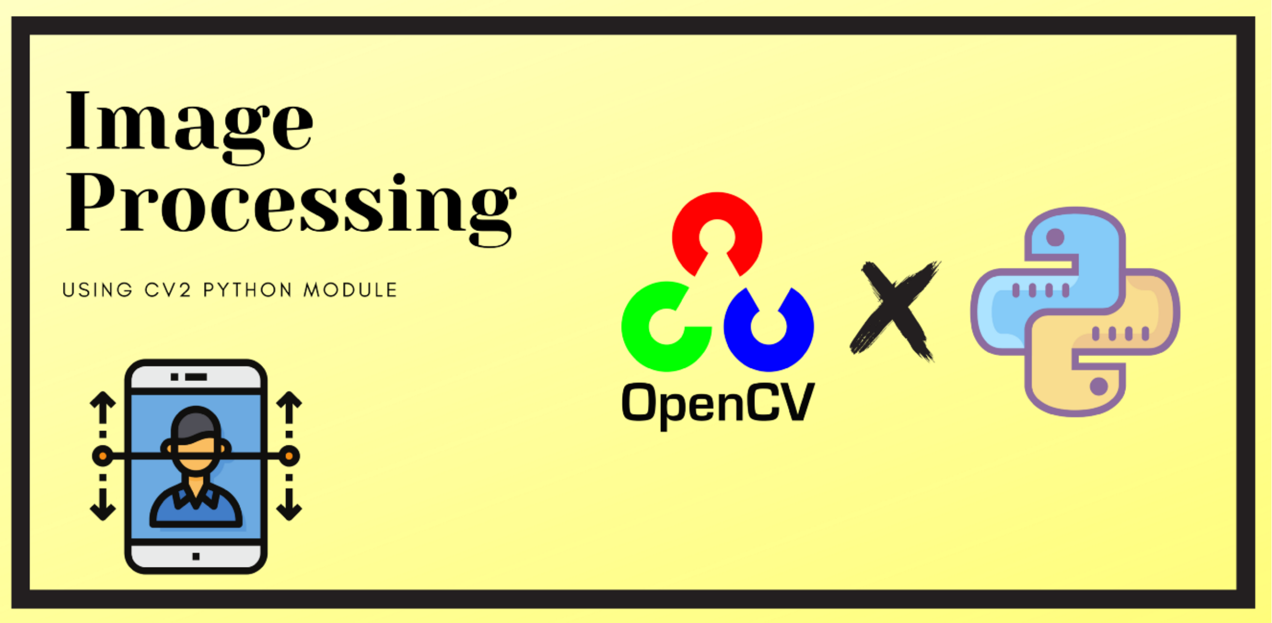


Fig. 4.5 OpenCV

Pillow:

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python3.x support to it. Pillow was announced as a replacement for PIL for future usage. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

Opening an image is a basic operation of the image processing. We import the image module from the PIL library to load the image. It provides the **Image.open()** method, which takes an image filename as an argument. This method returns the image object.

1. **Opening an image using open():** The PIL.Image.Image class represents the image object. This class provides the open() method that is used to open the image.
2. **Displaying the image using show():** This method is used to display the image. For displaying the image Pillow first converts the image to a .png format (on Windows OS) and stores it in a temporary buffer and then displays it.
3. **Resizing an image using resize():** Interpolation happens during the resize process, due to which the quality of image changes whether it is being upscaled (resized to a higher dimension than original) or downscaled (resized to a lower Image then original). Therefore resize() should be used cautiously and while providing suitable value for resampling argument.

Tkinter:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

1. **Tk(screenName=None, baseName=None, className=’Tk’, useTk=1):** To create a main window, tkinter offers a method ‘Tk(screenName=None, baseName=None, className=’Tk’, useTk=1)’. To change the name of the window, you can change the className to the desired one.
2. **mainloop():** There is a method known by the name mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.

tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

1. **grid() method:** It organizes the widgets in grid (table-like structure) before placing in the parent widget.

There are a number of widgets which you can put in your tkinter application. Some of the major widgets are explained below:

1. **Button**: To add a button in your application, this widget is used.

4.5 MACHINE LEARNING

4.5.1 WHAT IS MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through learning. In its application across business problems, machine learning is also referred to as predictive analytics.

4.5.2 FEATURES OF MACHINE LEARNING

- Business intelligence at its best.
- Accurate data analysis.
- The ability to change the mortgage market.
- The ability to take efficiency to the next level when merged with IOT.
- Automation at its best

4.5.3 ADVANTAGES & DISADVANTAGES OF MACHINE LEARNING

Advantages of Machine learning:

1. Easily identifies trends and patterns

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus software's; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data, you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine Learning:

With all those advantages to its powerfulness and popularity, Machine Learning isn’t perfect. The following factors serve to limit it:

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

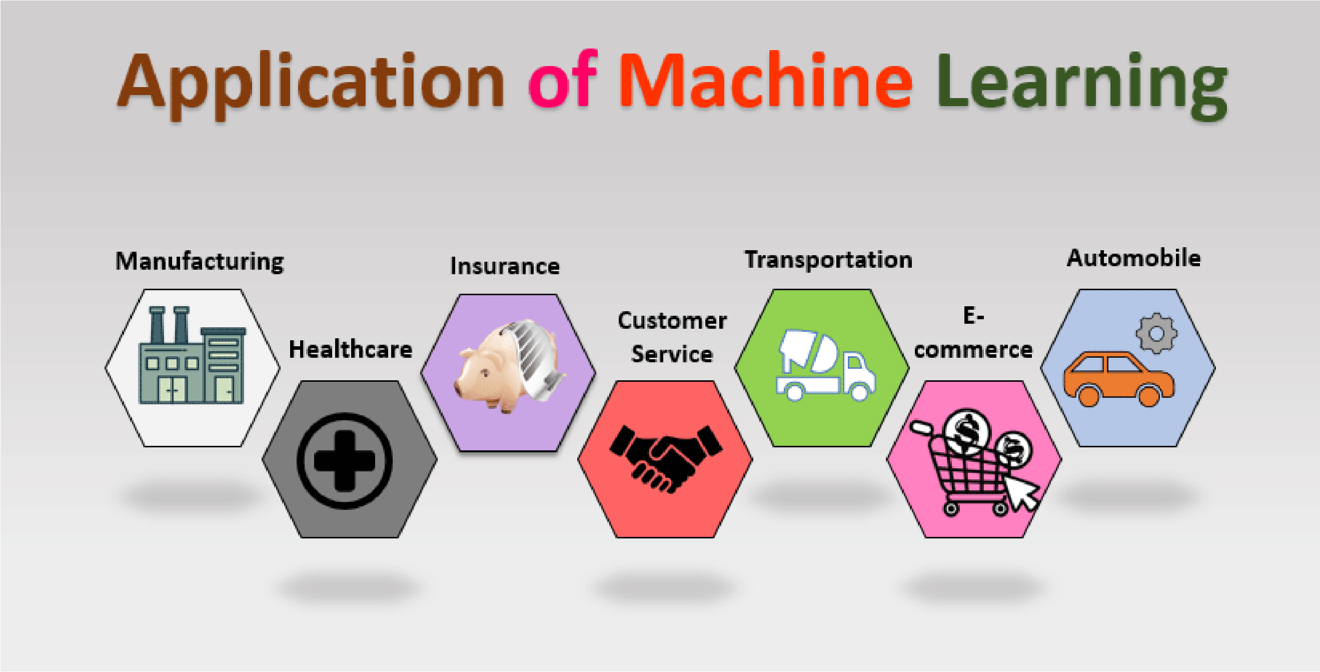


Fig. 4.6 Applications of Machine Learning

4.5.4 TYPES OF MACHINE LEARNING

Supervised Learning:

When the machine is supervised while it is “learning”, the training type is called supervised learning. But what does supervise a machine really mean? It means that we provide the machine with a ton of information about a case and also provide it with the case outcome. The outcome is called the labelled data while the rest of the information is used as input features. For example, we show the machine 1000 cases when customers defaulted on a loan and 1000 cases where customers did not default. Here default / not-default is the outcome and hence the labelled data while all the other characteristics like age, salary, loan amount, outstanding amount, other loan history etc. are the input features. In this case, the machine is supervised by the labelled data to learn what the relationships and dependencies are between the outcome of default and the borrower information.

There are two further categories into which we can divide supervised learning: regressions and classifications, and each has its own set of use cases and merits. Common supervised learning algorithms include: Linear regression; Naïve Bayes, Nearest Neighbors, Decision Trees, Support Vector Machines and Neural Networks.

Unsupervised Learning

As the name suggests, in case of unsupervised learning, there is no help from the user for the computer to learn. In the lack of labelled training sets, the machine identifies patterns in the data that is not so obvious to the human eye. So, unsupervised learning is extremely useful to recognize patterns in data and help us take decisions. For example, if we didn’t know which customers defaulted on loans, and but fed the borrower information to the machine, it would be able to pick out similar patterns among the different borrowers and grouped them in 3-4 buckets or clusters. Unsupervised learning is also often used for anomaly detection, like to uncover fraudulent transactions or payments. The most common use of unsupervised learning is in clustering problems, with the most talked about algorithms being k-means and hierarchical clustering, though other algorithms like Hidden Markov models, Self-Organizing Maps or Gaussian Mixture models are also often used.

Reinforcement Learning

Reinforcement learning is probably the closest to how we as humans learn. In this case, the algorithm or the agent learns continually from its environment by interacting with it. It gets a positive or a negative reward based on its action. Let’s consider the same example of customers with bank loans. A Reinforcement Learning algorithm looks at the information of a customer and classifies him / her as a high-risk customer. When the customer defaults, the algorithm gets a positive reward. If the customer doesn’t default, the agent receives a negative reward. The reward in both the cases helps the agent understand the problem and the environment better, and thus helps to make better decisions on our behalf. Common algorithms include Q-Learning, Temporal Difference and Deep Adversarial Networks.

Reinforcement Learning is probably the hardest to execute yet in a business environment but has been commonly used for self-driving cars or the famous Alpha Go chess match trials.

4.6 STEPS INVOLVED IN OBJECT DETECTION:

4.6.1 Data Acquisition & Image Tagging:

The journey of training an object detection model begins with data acquisition. Although there exist many ways of automating the process, in this specific training case, data acquisition was done manually to filter out noisy images and to ensure the quality of the training dataset. Once the data is acquired, the images in the dataset need to be tagged. Image tagging was completed using the LabelImg software, which is an open-source python-based implementation of a system that tags the bounding boxes and records them in an XML file.

4.4.2 Preparing dataset:

With the image acquisition and tagging done, you will be happy to know that most of the manual work is now completed. The output generated from the tagged image dataset is an XML file containing the bounding box coordinates of the hand-tagged images. However, neural net models cannot parse XML files and require a specific format called Tensor Records.

4.4.3 Configuring Training:

Configuration of the training involves the selection of the model as well as model parameters such as batch size, number of steps, training, and testing TF-Records directory. Thankfully, TensorFlow’s object detection pipeline comes with a pre-configured file that optimizes most of the configuration selection for efficient model training.

4.4.4 Training, Monitoring, and testing:

In technical terminology, we refer to this as out-of-distribution detection or anomaly detection. Using a reference dataset of images, you can perform ongoing monitoring of all new images to understand which ones are similar to the training data and which ones seem like anomalies. It is essential to know exactly when your model won’t generalize to new settings. For example, if your object detection model was trained primarily on images from outdoor locations in good weather and good lighting, it will likely underperform in rainy and dark conditions. you need a data drift detection tool that can automatically detect if these dissimilar images start coming in.

4.4.5 Matching strategy

During training we need to determine which default boxes correspond to a ground truth detection and train the network accordingly. For each ground truth box, we are selecting from default boxes that vary over location, aspect ratio, and scale. We begin by matching each ground truth box to the default box with the best jaccard overlap (as in MultiBox [7]). Unlike MultiBox, we then match default boxes to any ground truth with jaccard overlap higher than a threshold (0.5). This simplifies the learning problem, allowing the network to predict high scores for multiple overlapping default boxes rather than requiring it to pick only the one with maximum overlap.

4.7 ALGORITHM:

Step 1: Take Real Time (Video) Input from Webcam / CCTV Camera

Step 2: Divide the Video into Frames.

Step 3: The input Frame/image is divided into SxS grid.

Step 4: For each cell it predicts B bounding boxes Each bounding box contains five elements:
(x, y, w, h) and a box confidence score.

Step 5: SSD detects one object per grid cell only regardless of the number bounding boxes.

Step 6: If no objects exists then confidence score is zero Else confidence score should be greater or equal to threshold value.

Step 7. SSD then draws bounding box around the detected objects and predicts the class to which the object belongs

5. PROJECT PLAN

5.1 PROJECT ESTIMATES

5.1.1 Reconciled Estimates

5.1.1.1 Cost Estimate

1. Python & Visual Studio Code: Free
2. Hardware and Software:

5.1.1.2 Time Estimates

Procedure	Time
Literature Research	14 days
System Analysis	20 days
Design and Planning	31 days
Learning Required Technologies	31 days
Implementation	31 days
System Testing	28 days
Initial Report	14 days
Final Report	14 days

Table 5.1 Time Estimates

5.1.2 Project Resources

ID	Name of Person	Responsibility
1	Prof. Madhavi Kulkarni	Project Guide
2	Harshal Honmote	Developer
3	Shreyas Gadekar	Developer
4	Pranav Katta	Developer

Table 5.2 project resources

- Hardware:

1. RAM: 4 or 8GB for faster processing.
2. CPU: 2GHz+ for sufficient processing power

- Software

1. Operating system: Windows 10 or 11, Linux
2. Programming Language: Python

5.2 RISK MANAGEMENT

Risk is a possibility of loss or injury. Risk management is the identification assessment and prioritization of risks followed by coordinated and economical application of resources to minimize and control that would probability and impact of unfortunate events or to maximize the realization of opportunities. Risk can come from uncertainty in fi- noncoal markets, project failures (at any phase in design, development, production and sustainment life cycles), legal liabilities, credit risks, accidents, natural causes and disasters as well deliberate attack from an advisory of uncertain and unpredictable cause.

Using risk management techniques, we alleviate the harm or laws n software project or risk cannot be avoid but by perform in risk management we can attempt to ensure that right risks are taken at right time. Risk taking is essential to progress and failure is often key part of learning.

5.2.2 Risk Identification:

Our development identified some potential risks to the project. These risks were analyzed and were classified into various categories depending upon the threat they posed to the project. Some of these risks were ‘generic risks’ while others were ‘product specific risks’ A considerable amount of time was spent in analyzing the product specific risks.

1. Have top software and customer managers formally committed to support the project?

- The software manager and the customer mangers are fully committed to the project.

2. Are end-users enthusiastically committed to the project and the system/product to be built?

- The end-users have also committed to the project and to the product to be built.

3. Are requirements fully understood by the software engineering team and its customers?

- Requirements for citizens are fully understood by whole team from consistent feedback of citizens.

4. Have customers been involved fully in the definition of requirements?

- Yes, citizens have been fully consulted and are involved in the process.

5. Do end-users have realistic expectations?

- End users tend to have some unrealistic expectations, typically, on the various features of the product to be quickly delivered in a constrained manner.

6. Does the software engineering team have the right mix of skills?

- The team consists of the people with Managerial, Designing as well as Developing skill set.

7. Are project requirements stable?

- Project requirements are stable.

8. Is the number of people on the project team adequate to do the job?

9. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?

- Citizens have consistently agreed on the parameters of the project and its feasibility, and of the requirements for product to be built effectively.

5.3 PROJECT SCHEDULE

5.3.1 Project task set

Project scheduling is a mechanism to communicate what tasks need to get done and which organizational resources will be allocated to complete those tasks in what timeframe. A project schedule is a document collecting all the work needed to deliver the project on time.

A project is made up of many tasks, and each task is given a start and end (or due date), so it can be completed on time. Likewise, people have different schedules, and their availability and vacation or leave dates need to be documented in order to successfully plan those tasks.

Major Tasks in the Project stages are:

5.1.1.2 Task 1: Literature Research

5.1.1.3 Task 2: System Analysis

5.1.1.4 Task 3: Design & Planning

5.1.1.5 Task 4: Learning Required Technologies

5.1.1.6 Task 5: Implementation

5.1.1.7 Task 6: System Testing

5.1.1.8 Task 7: Initial Report

5.1.1.9 Task 8: Final Report.

5.3.2 Task network

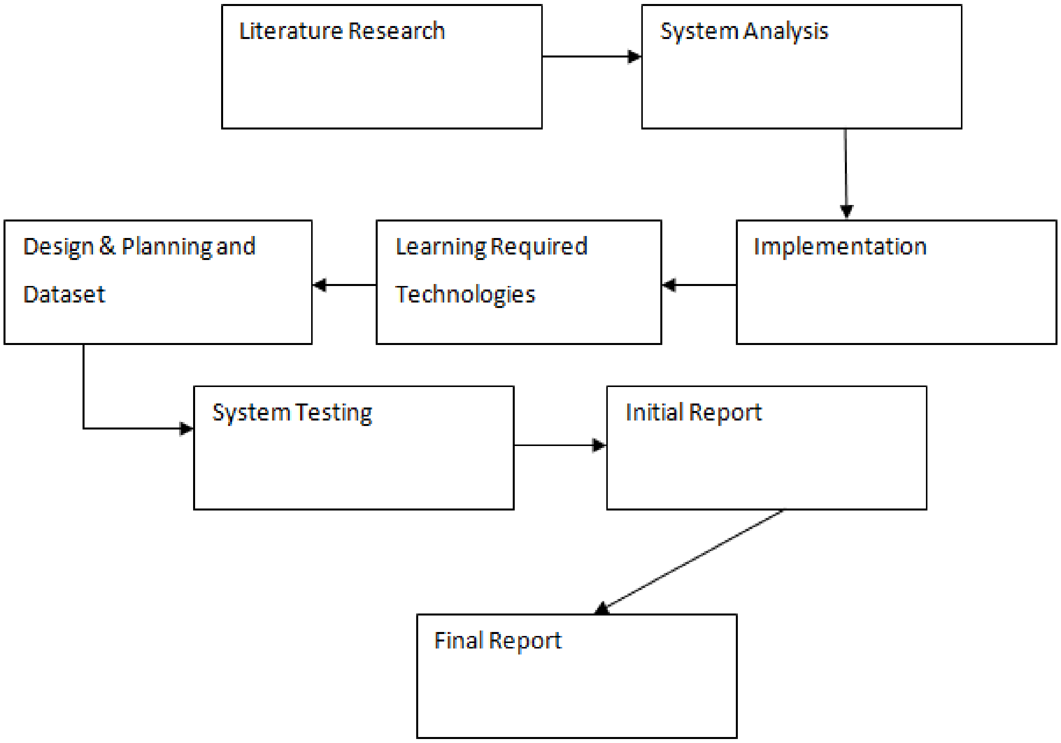


Fig 5.1 Test Network

5.4 TEAM ORGANIZATION

Sr. No.	Design/Develop	Name of Designers/Developers
1	Design	Harshal Honmote Shreyas Gadekar Pranav Katta
2	Implementation	Harshal Honmote Shreyas Gadekar Pranav Katta
3	Documentation	Shreyas Gadekar Pranav Katta
4	Testing	Harshal Honmote Shreyas Gadekar

Table 5.3 Team Organization

5.4.1 Team structure

The team structure for the project is identified. Roles are defined.

5.4.2 Management reporting and communication

Mechanisms for progress reporting and inter/intra team communication are identified as per assessment sheet and lab time table.

6. UML DIAGRAMS

UML, which stands for Unified Modeling Language, is a way to visually represent the architecture, design, and implementation of complex software systems.
When you’re writing code, there are thousands of lines in an application, and it’s difficult to keep track of the relationships and hierarchies within a software system.

UML diagrams divide that software system into components and sub components.

Why should you use UML diagrams?

UML is a standardized modeling language that can be used across different programming languages and development processes, so the majority of software developers will understand it and be able to apply it to their work.

Types Of UML Diagrams: -

1. Class Diagram
2. State Diagram
3. Activity Diagram
4. Sequence Diagram
5. Use Case Diagram
6. Data Flow Diagram

6.1 UML Class Diagram:

Class diagrams are the most common diagrams used in UML. Class diagram consists of classes, interfaces, associations, and collaboration. Class diagrams basically represent the object-oriented view of a system, which is static in nature.

Active class is used in a class diagram to represent the concurrency of the system.

Class diagram represents the object orientation of a system. Hence, it is generally used for development purpose. This is the most widely used diagram at the time of system construction.

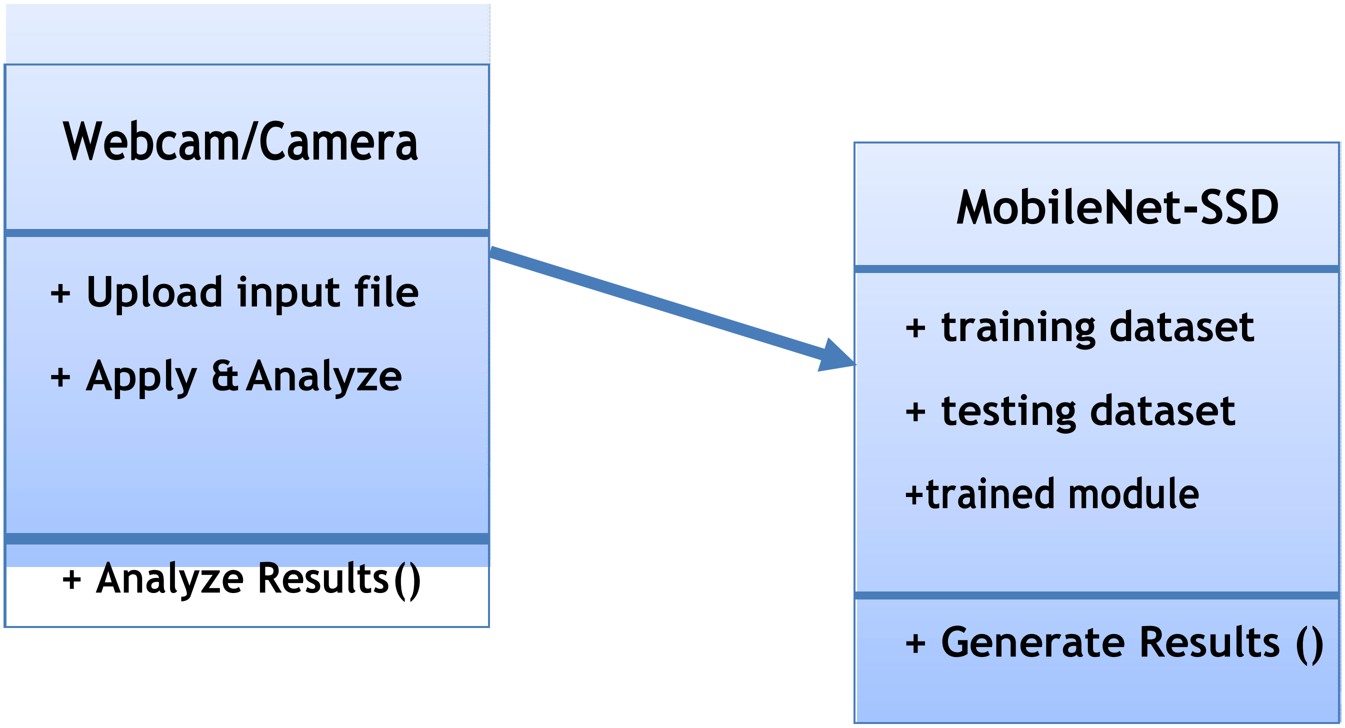


Fig 6.1 UML Class Diagram

6.2 UML State Diagram:

Any real-time system is expected to be reacted by some kind of internal/external events. These events are responsible for state change of the system.

State chart diagram is used to represent the event driven state change of a system. It basically describes the state change of a class, interface, etc.

State chart diagram is used to visualize the reaction of a system by internal/external factors.

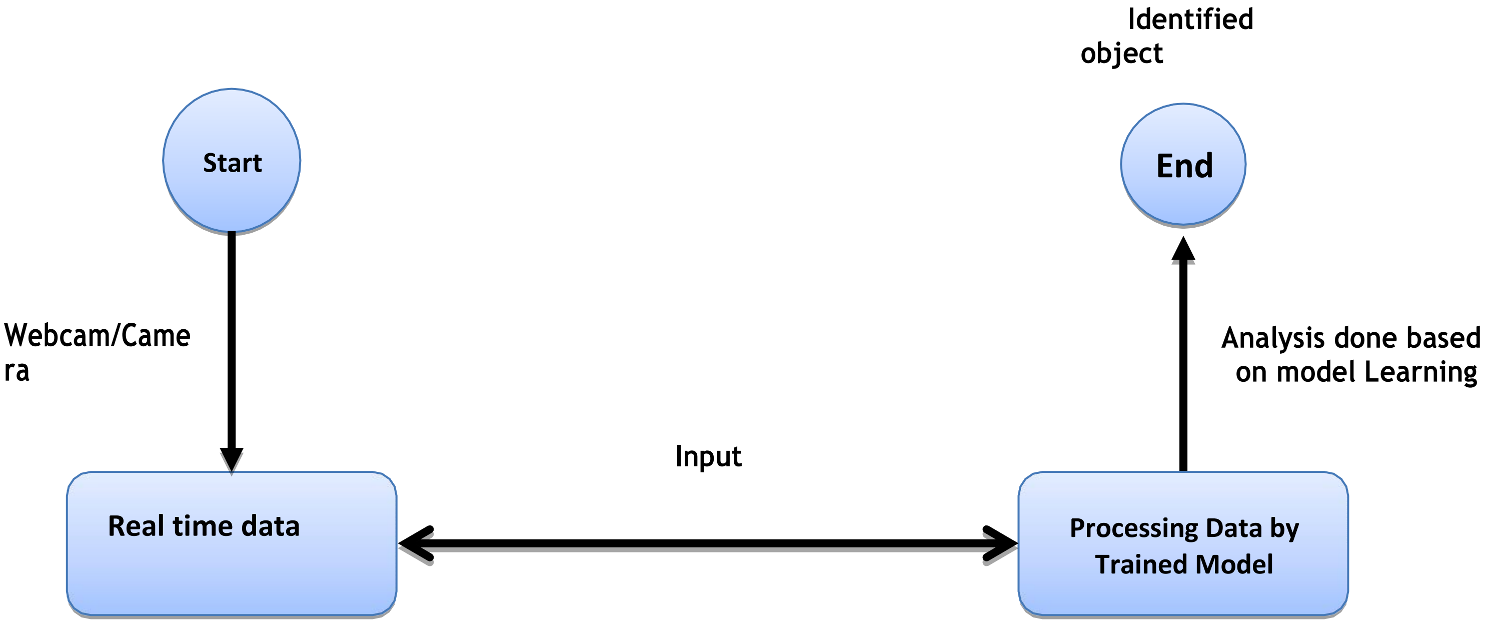

Identified object
Start
End
Webcam/Camera
Analysis done based on model Learning
Input
Real time data
Processing Data by Trained Model

Fig 6.2 UML State Diagram

6.3 UML Activity Diagram:

Activity diagram describes the flow of control in a system. It consists of activities and links. The flow can be sequential, concurrent, or branched.

Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.

Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

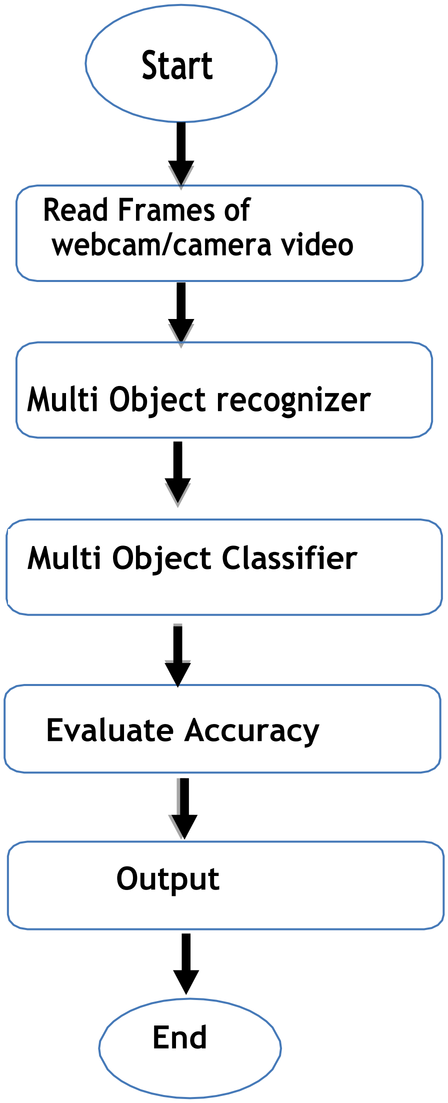


Fig 6.3 UML Activity Diagram

6.4 UML Sequence Diagram:

A sequence diagram is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another.

Interaction among the components of a system is very important from implementation and execution perspective. Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality.

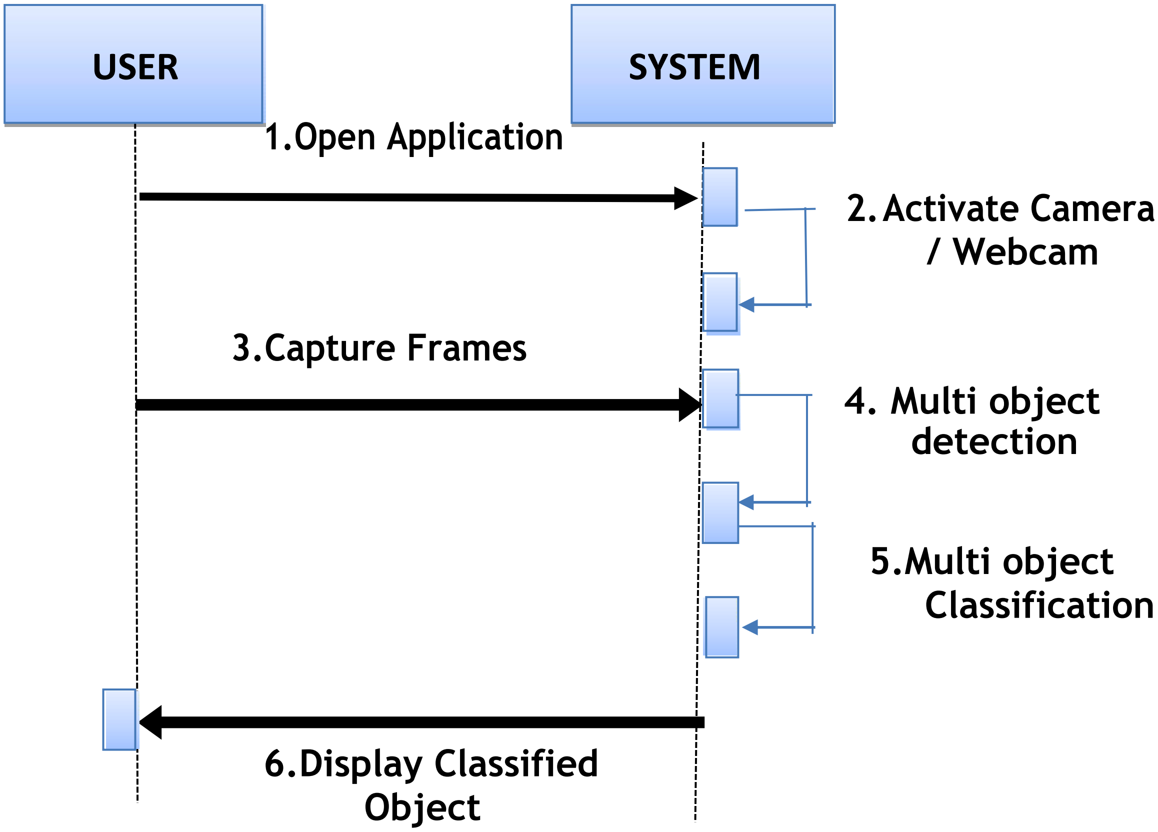

USER
SYSTEM
1.Open Application
2. Activate Camera / Webcam
3.Capture Frames
4. Multi object detection
5.Multi object Classification
6.Display Classified Object

Fig 6.4 UML Sequence Diagram

6.5 Data Flow Diagram:

Also known as DFD, Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

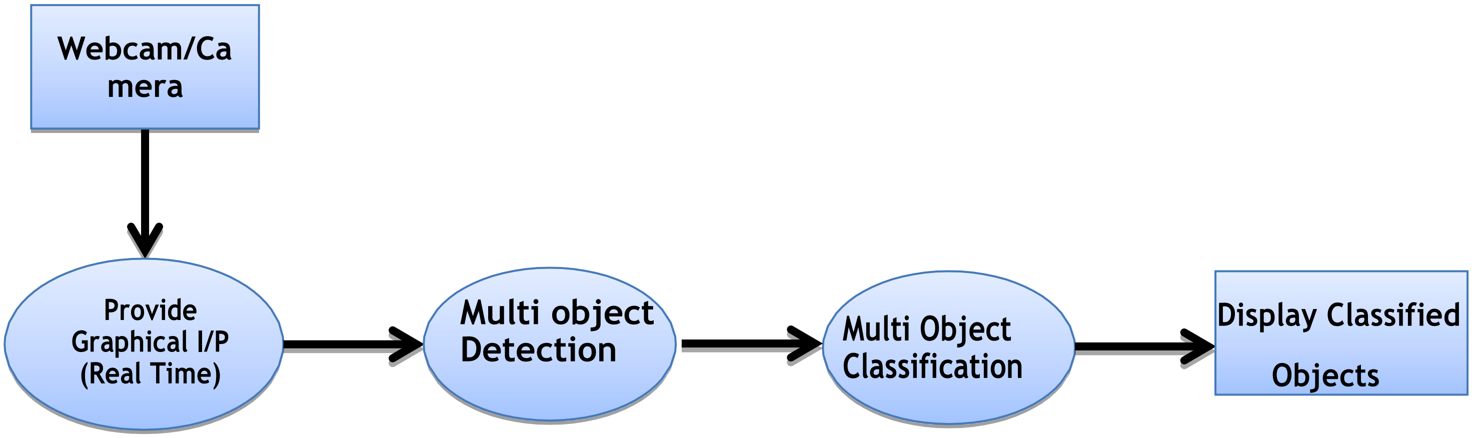

Webcam/Camera
Provide Graphical I/P (Real Time)
Multi object Detection
Multi Object Classification
Display Classified Objects

Fig 6.5 Data Flow Diagram

7 TESTING

Software testing is the process of executing a program with an objective of finding an error. Following are the different test strategies which were conducted on our system.

7.1 BLACK BOX TESTING (BBT):

BBT is related with the input and the output only and not with the internal structure of the program. It checked if some input is given then specific output is produced or not. Black Box Testing is also known as functional testing. A software testing technique whereby the internal workings of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs. The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specifications.

Black box testing implies that the selection of test data as well as the interpretation of test results is performed on the basis of the functional properties of a piece of software. Black box testing should not be performed by the author of the program who knows too much about the program internals. In new testing approaches, software systems are given a third external party for black box testing after having successfully finished the internal glass box testing exercises

7.2 WHITE BOX TESTING (WBT):

It is related with the internal logic of the program. Also known as glass box, structural, clear box and open box testing. A software testing technique whereby explicit knowledge of the internal workings of the item being tested are used to select the test data. Unlike black box testing, white box testing uses specific knowledge of programming code to examine outputs. The test is accurate only if the tester knows what the program is supposed to do. He or she can then see if the program diverges from its intended goal. White box testing does not account for errors caused by omission and all visible code must also be readable.

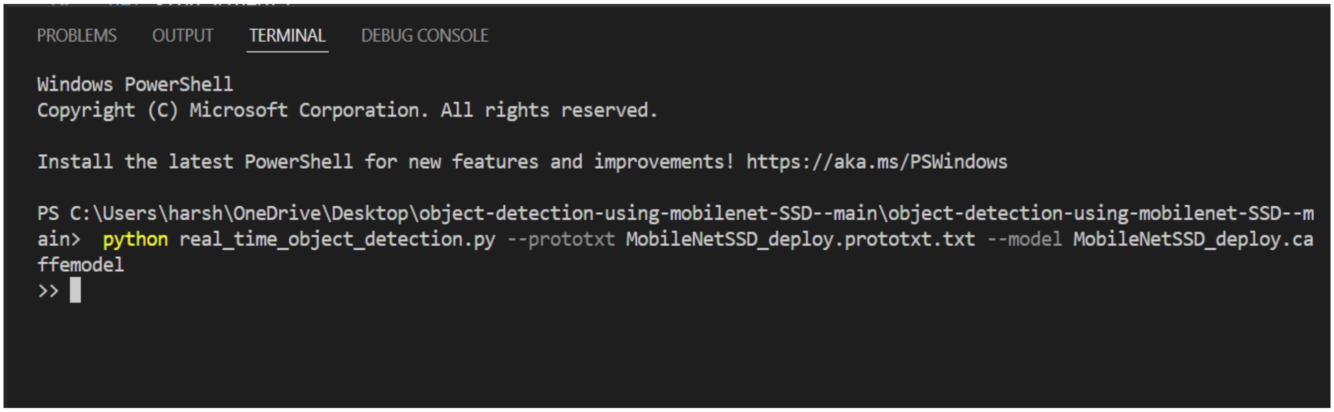
The intention in white box testing is to ensure that all possible feasible flow of control paths through a subprogram is traversed whilst the software is under test. This is not the same as saying that all statements in the subprogram will be executed as it is possible for all statements to be executed but for not all of the possible paths to be traversed. However, the converse is true; if all possible paths through a subprogram are traversed then all statements in the subprogram will necessarily be executed.

7.3 TEST CASES AND TEST RESULTS:

Test Case ID	**Test Conditions**	**Expected Result**	**Test Results**
TC1	When camera /Webcam is chosen as input	Objects detected in the real time with bounding box, confidence score and predicted class.	SUCCESSFUL
TC2	When image is c hosen as input	Image with bounding box around the objects and predicted class.	SUCCESSFUL
TC3	When black and white image is taken as input	Image with bounding box around the objects and predicted class.	SUCCESSFUL
TC4	Image with far objects is taken as input	Image with detected objects.	UNSUCCESSFUL

Table -7.3: Test Cases with Results

8. UI APPLICATION/OUTPUT SCREENSHOTS



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\harsh\OneDrive\Desktop\object-detection-using-mobilenet-SSD--main\object-detection-using-mobilenet-SSD--m
ain>  python real_time_object_detection.py --prototxt MobileNetSSD_deploy.prototxt.txt --model MobileNetSSD_deploy.ca
ffemodel
>> 
```

Fig 8.1 Run Command

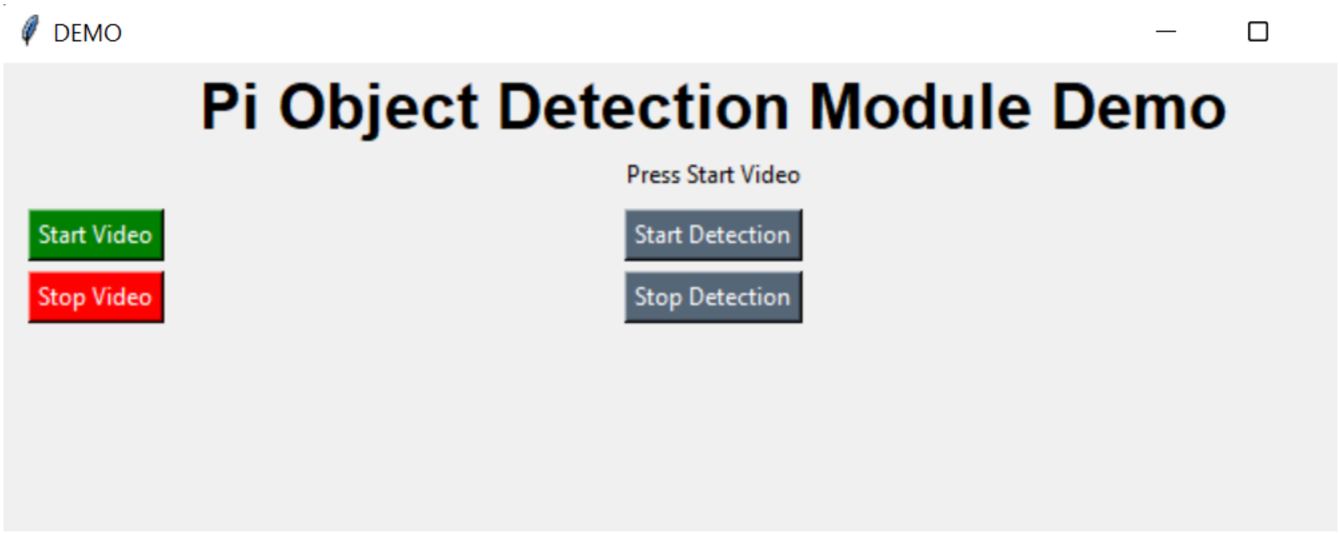


Fig 8.2 User Interface

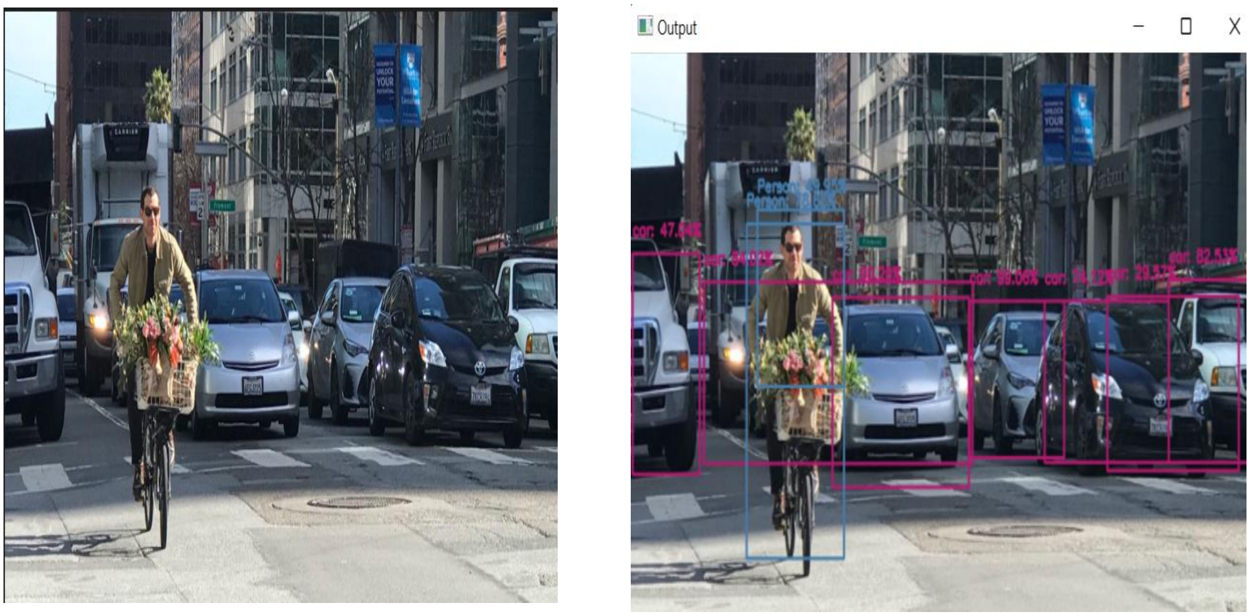


Fig 8.3: Image with Detected Object

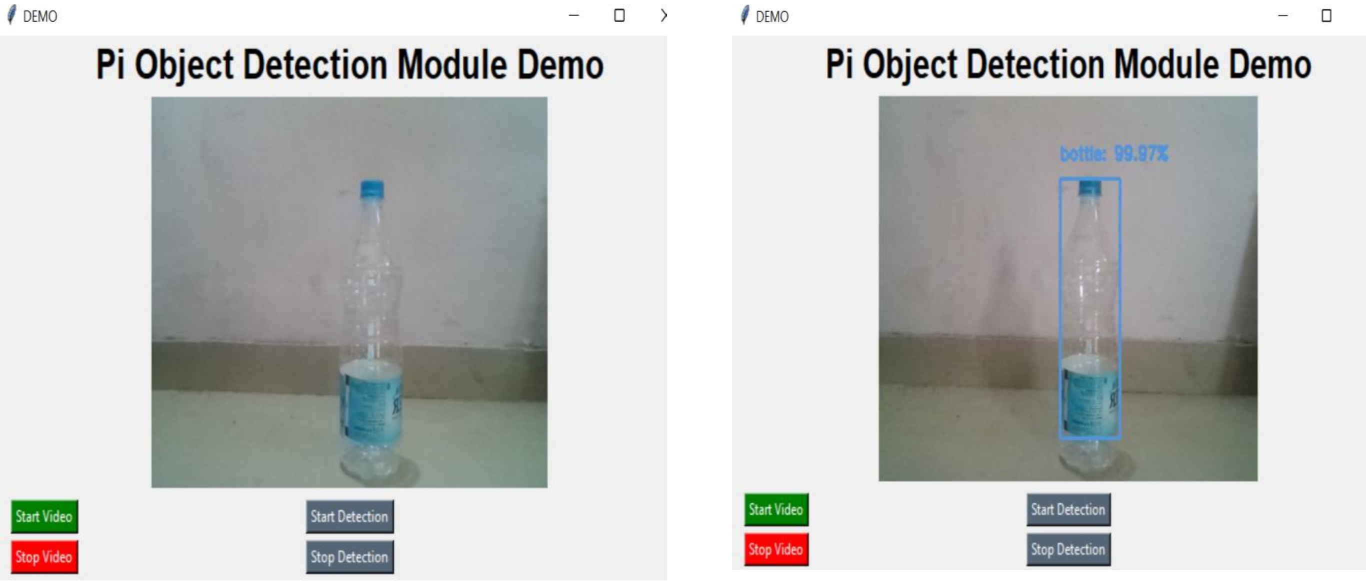


Fig 8.4: Output obtained in Real-Time

9. OTHER SPECIFICATION

ADVANTAGES:

1. Memory Efficient
2. Low Power Consumption
3. Optimized Software

APPLICATIONS:

1. Self-Driving Car
2. Real Time Object Detection
3. Object Tracking
4. Object Classification

10.CONCLUSION AND FUTURE WORK

The project is developed with objective of detecting real time objects in video and camera with faster, Accurately with Less memory consumption and reduce traffic consumption over Web Servers for classification of Image. Bounding Boxes are drawn around the detected objects along with the label indicating the class to which the object belongs. We have used CPU for the processing in the project.

FUTURE WORK

Future enhancements can be focused by implementing the project on the system having GPU for faster results and better accuracy. we will continue to optimize our detection network model, including reducing memory usage and increasing network computing speed.

11.REFERENCES

[1] MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications Andrew G. Howard Menglong Zhu IEEE

[2] Explicit Content Detection using Faster R-CNN and SSD MobileNet. Animesh Srivastava1, Anuj Dalvi2 IEEE.

[3] Shoe Detection Using SSD-MobileNet Architecture. Jeong-ah Kim, Ju-Yeong Sung IEEE

[4] Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems Yu-Chen Chiu, Chi-Yi Tsai October 18,2020 at from IEEE Xplore

[5] Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition. Jeong-ah Kim, Ju-Yeong Sung

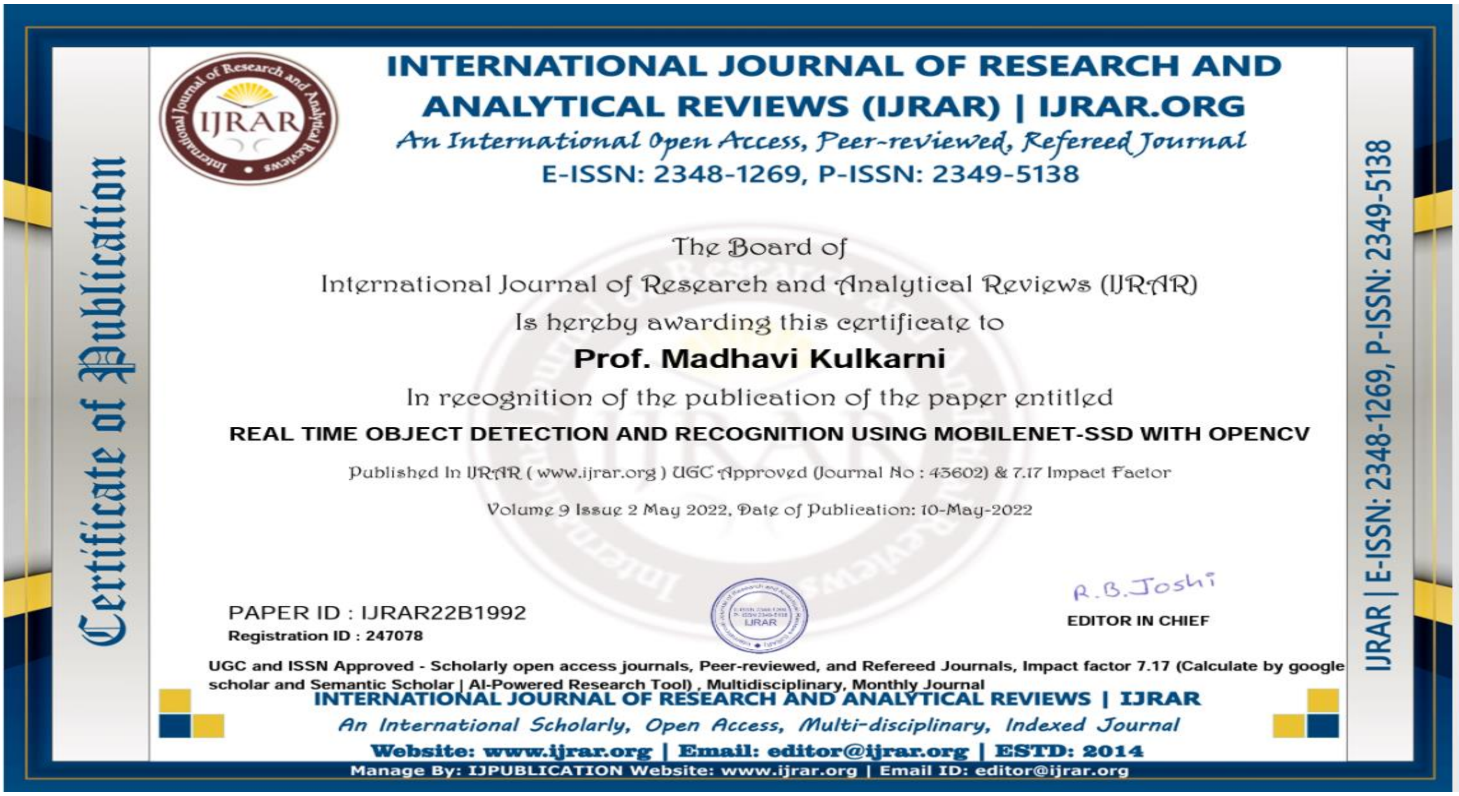
[6] Ojala. T., Matti. P., Topi, M. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 7, pp. 971–987, 2002

[7] Karen Simonyan and Andrew Zisserman.Very deep convolutional networks for large-scale image recognition.arXiv preprintarXiv:1409.1556, 2014.

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015

[9] Medical Waste Classification System Based on OpenCV and SSD-MobileNet for 5G Xiaoxiao Bian, Yuchao Chen, Shuyi Wang 978-1-7281-9507-0/20/$31.00 ©2021 IEEE.

12.CERTIFICATES



Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

Prof. Madhavi Kulkarni

In recognition of the publication of the paper entitled

REAL TIME OBJECT DETECTION AND RECOGNITION USING MOBILENET-SSD WITH OPENCV

Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor

Volume 9 Issue 2 May 2022, Date of Publication: 10-May-2022

PAPER ID : IJRAR22B1992

Registration ID : 247078

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

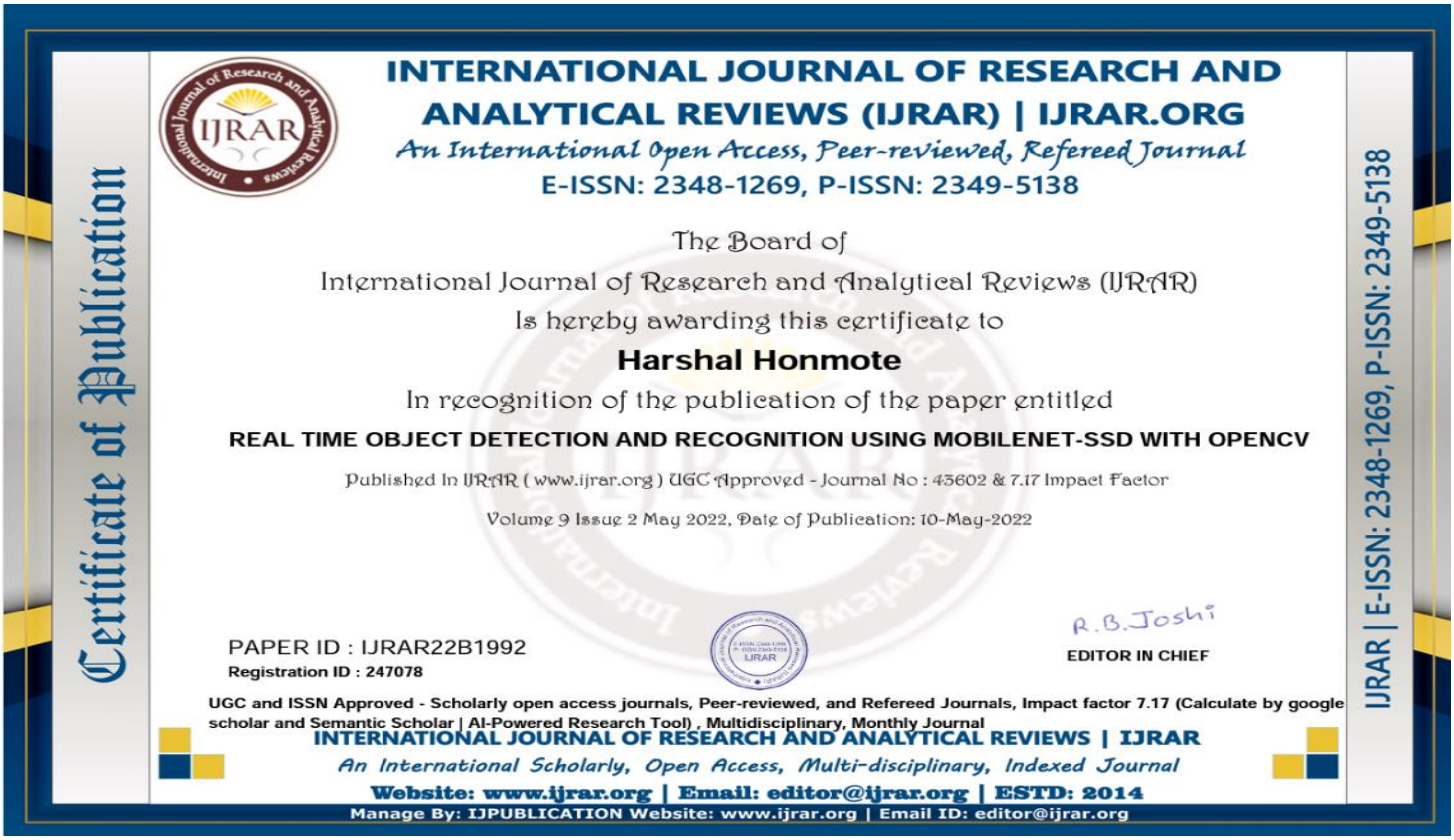
INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138



Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

Harshal Honmote

In recognition of the publication of the paper entitled

REAL TIME OBJECT DETECTION AND RECOGNITION USING MOBILENET-SSD WITH OPENCV

Published In IJRAR (www.ijrar.org) UGC Approved - Journal No : 43602 & 7.17 Impact Factor

Volume 9 Issue 2 May 2022, Date of Publication: 10-May-2022

PAPER ID : IJRAR22B1992

Registration ID : 247078

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

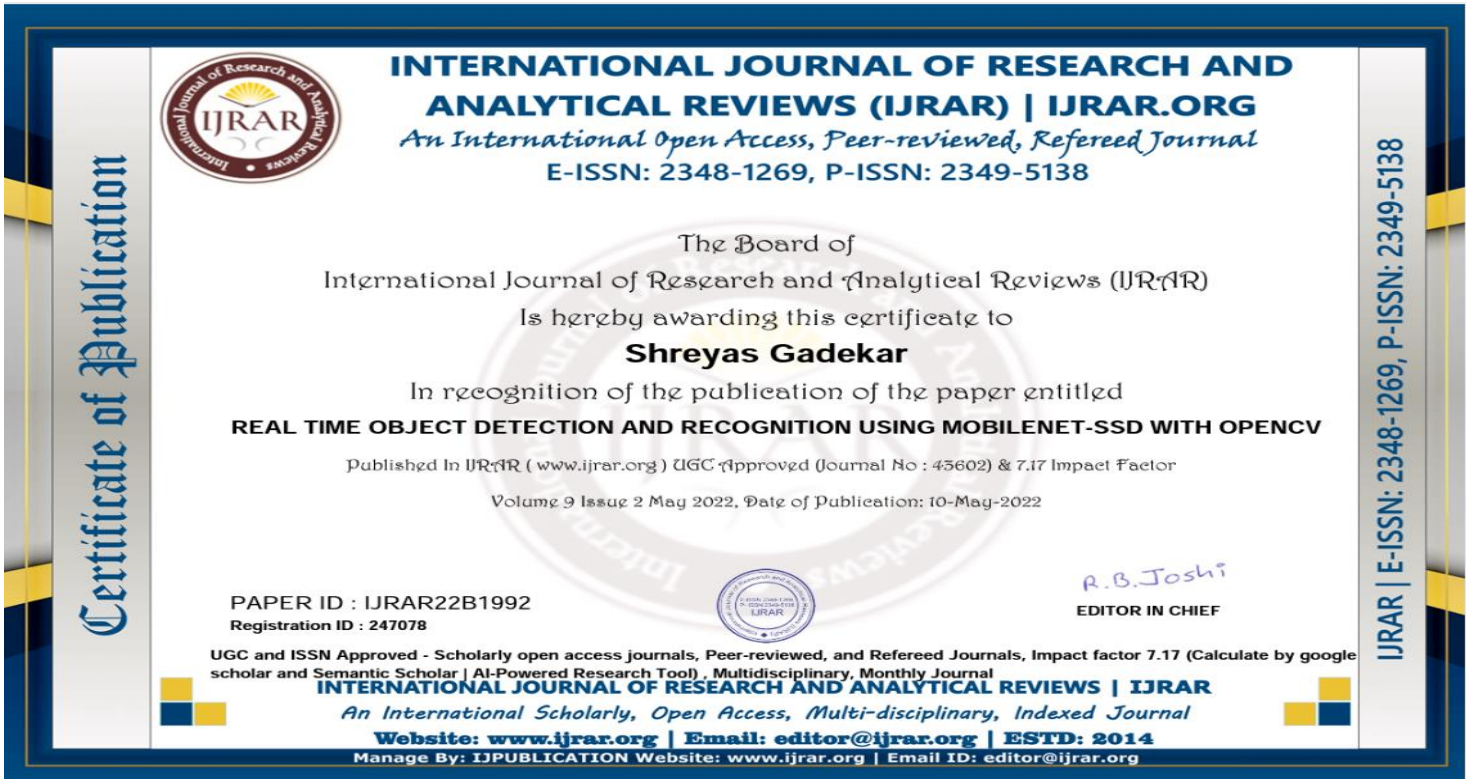
INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138



Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

Shreyas Gadekar

In recognition of the publication of the paper entitled

REAL TIME OBJECT DETECTION AND RECOGNITION USING MOBILENET-SSD WITH OPENCV

Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor

Volume 9 Issue 2 May 2022, Date of Publication: 10-May-2022

PAPER ID : IJRAR22B1992

Registration ID : 247078

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

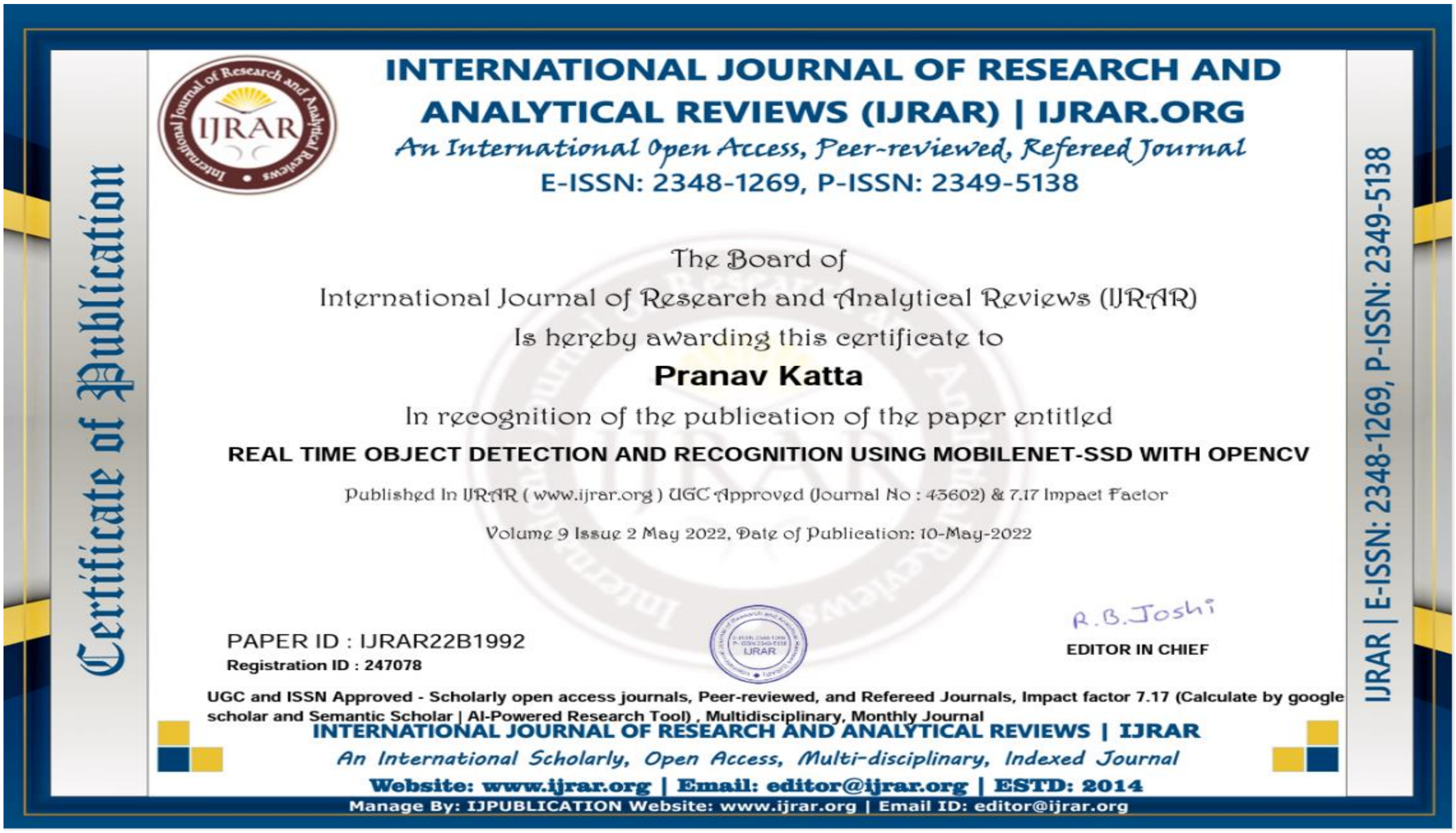
INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138



Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

Pranav Katta

In recognition of the publication of the paper entitled

REAL TIME OBJECT DETECTION AND RECOGNITION USING MOBILENET-SSD WITH OPENCV

Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor

Volume 9 Issue 2 May 2022, Date of Publication: 10-May-2022

PAPER ID : IJRAR22B1992

Registration ID : 247078

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138