Final Year Project Report

MOBILE PHONE USER DETECTION USING COMPUTER VISION

BS(CSIT) (Batch: 2017-18)

Project Advisor

Dr. Waseemullah

Assistant Professor
NEDUET

Submitted by

Muhammad Zaid Raza	CT-039
Muhammad Uzair Khan	CT-050

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
NED University of Engineering and Technology, Karachi.

PREFACE

In modern society, there is a lot of problems that could be reduced using Artificial Intelligence. We have picked one problem from our society that there are many places where mobile phone usage is restricted but people use a mobile phones. In petrol pumps mobile phone is restricted due to safety purposes, banks do not allow the customers to use mobile phone in bank premises for security reasons.

This project works to reduce mobile phone usage where its usage is restricted. We have developed this project to restrict the people to follow rules and regulations related to mobile phones.

This project detects the mobile phone users and output from the project i.e images can be used to take action against the rule breaker. We can use it in multiple scenarios, for instance, in-office meetings, it detects the person who is busy using a mobile phone rather than focusing on the meeting.

ACKNOWLEDGMENT

Beginning with the praise of Almighty Allah who is indeed the seer of all that happens and who has strengthened us and bestowed upon us the ability to think, decide, create and accomplish. It is due to His blessings what we have accomplished in this project thus far.

Secondly, we would like to thank our project supervisor, Dr. Waseemullah who has helped us with our project. He has been a very compliant and helpful project advisor for us in this journey. It is due to his knowledge that we were able to apply modern-day software techniques in our project.

We would also like to acknowledge our teachers who invested their time in teaching us and helped us during our study period. Learning the courses made us aware of the procedures followed in Software Engineering and Computer Science.

We also acknowledge the help of department and university teachers, lab attendants, and guides, who cooperated during this tenure.

Finally, we are extremely grateful to our parents for their constant support and encouragement throughout our course of study.

GROUP MEMBERS

Muhammad Zaid Raza CT-039

Immediate Contact: (0331-2399280, zaidraza22@gmail.com)

Muhammad Uzair Khan CT-050

Immediate Contact: (0341-0235923, mohammaduzairkhan97@gmail.com)

NED University of Engineering and Technology, Karachi

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

CERTIFICATE OF COMPLETION

This is to certify that the following students

Muhammad Zaid Raza	CT-039
Muhammad Uzair Khan	CT- 050

have successfully completed their final year project titled

MOBILE PHONE USER DETECTION USING COMPUTER VISION

in the partial fulfillment for the requirements of the Degree of Bachelor of Computer Science & Information Technology during the academic session 2017-2021.

Dr. Waseemullah
Assistant Professor

Dr. Najmi Ghani Haider
Chairman(CSIT Department)
NED University

ABSTRACT

In the modern-day, technology is being used in our surroundings to reduce human efforts. Technology is used to solve a lot of real-life problems in our society. Nowadays, people use the mobile phone a lot due to digitalization in almost every department. People use mobile phones where their usage is restricted. In petrol pumps, mobile phone usage is restricted because it can ignite fire but people use it. In banks, mobile phone usage is restricted due to security reasons.

We solved the issue by creating system that detects mobile phone user and save image of the user into their respective hardware, so that strict action could be taken against the rule breaker.

Object detection is used nowadays to solve a lot of problems. We also used object detection technique to create our system. We use yolov3 algorithm for object detection. The code is written in python and OpenCV is used for performing computer vision tasks on video.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter # 1
INTRODUCTION

1. Introduction

As a citizen of society, we observe lot of problem in our surrounding that needs to be solved in order to make our society better place. We have observed that people use the mobile phones where its usage is prohibited such as in petrol pumps and banks etc. In petrol pumps it is prohibited to use the mobile phone, using it can ignite fire and human life could be lost.

To overcome the problem, we created a system that detects a mobile phone user in real-time. This system takes a video input that be recorded or live video, it is then given to system as input, the system then process the video and give the image of the person using mobile phone.

This project can be used in multiple scenarios like in the examination hall, it detects the students using mobile phones for cheating purposes.

1.1 Goals and objectives

There is a very common problem that is being faced in our society that people use the mobile phone where its usage is prohibited. For instance, in banks, it is prohibited to use mobile phones due to security reasons, in petrol pumps people are not allowed to use mobile phones for human safety purposes. Our project tends to reduce this problem by detecting the mobile phone users and then work on the output for further action to be taken against them.

1.2 System statement of scope

Mobile phone user detection using computer vision is a system that works in real-time. It takes live video as input and gives us the images of the person using the mobile phone asoutput. In this system, we detect the person using a mobile phone and save its image into the system so that further action could be taken against them.

1.3 System context

Object detection is a technique that detects objects from the image or video. It draws the bounding box around the detected object which allow us to locate where the specific objects are in image or video.

1.4 Theoretical Background (of the project)

We see a lot of problems in our surroundings like people breaking signals, not following rules and regulations etc.

From many problems, we found that people use the mobile phone at petrol stations where the risk of incendive sparking is very high due to explosive atmosphere temporarily created during refueling. People also use the mobile phone in banks but it is prohibited to use mobile phones due to security reasons.

We as students tried to contribute to solve this problem by creating a project that will restrict people mobile phone usage in sensitive places.

1.5 Technology & Tools/hardware components (used in the Project)

The tools and technologies we used are as following:

1.5.1 Python

Python is dynamic, high-level scripting language. It supports object-oriented as well as procedural-oriented programming.

We can create almost all types of application in python such as web application, analytical application etc. It has larger number packages that assist you in developing application so that you don’t have to write each functionality from scratch. It is easier to code and understand it. It is interpreted programming language that enables debugging error easily.

First, when the .py file is executed by an IDE, it is compiled and converted into bytecode. The bytecode is loaded and interpreted by python virtual machine (PVM), which is a piece of code that reads each instruction from the bytecode and perform the operation indicated in the byte code.

We have choosen this language due to large number of packages available for the machine learning tasks. It has also a huge community support that helped us a lot during project.

1.5.2 OpenCV

OpenCV is an open-source library used for computer vision, machine learning, and image processing. It can process images and videos to identify objects, faces, or even handwriting of humans.

It supports languages such as python, java, C++. It support almost all operating systems such as Windows, MacOS, Linux. It is used for real-time operations which is very helpful in solving real life problems. Using OpenCV we can do object detection, image processing, video analysis, 3D-reconstruction, feature extraction, machine learning, computational photography, shape analysis, face and object recognition, text detection and recognition, surface matching and many more.

We have used OpenCV in our project for object detection purpose.

1.5.3 Visual Studio Code

Visual studio code is a Microsoft create powerful IDE with support of many languages like python, javascript, C++ etc. It is open-source. It is easily customizable according to our needs and has many useful extensions.

It has intellisense which help us in excellent code completion and also guide us about function arguments on the go. It has built-in terminal which perform exactly the same way in which you normal operating system terminal works and we can also open nested terminal sessions. It also provide git integration for our project. It has built-in powerful debugger for nodejs which can be very helpful for developers working on backend.

We have used visual studio code in our project because it is light weight, easy to use and its powerful intellisense.

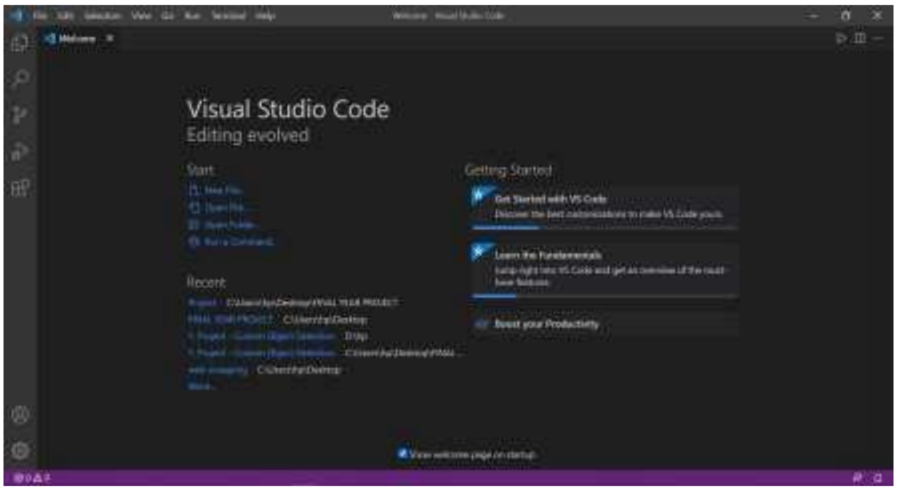


Figure 1.1: Visual Studio Code

1.5.4 Github

GitHub is a cloud based service that host the git repositories which help the developers around the globe to manage their code more efficiently especially in team. It provide service for both public and private repositories. It work in collaboration with the git that is version control system built to keep track of the files locally.

In github, developers simply create a branch to work on updates, commit changes to save them, open a pull request to propose and discuss changes, and merge pull requests once everyone is one the same page. We have used the github so that once our project is completed, it is available for all the people to user for their own benefits.

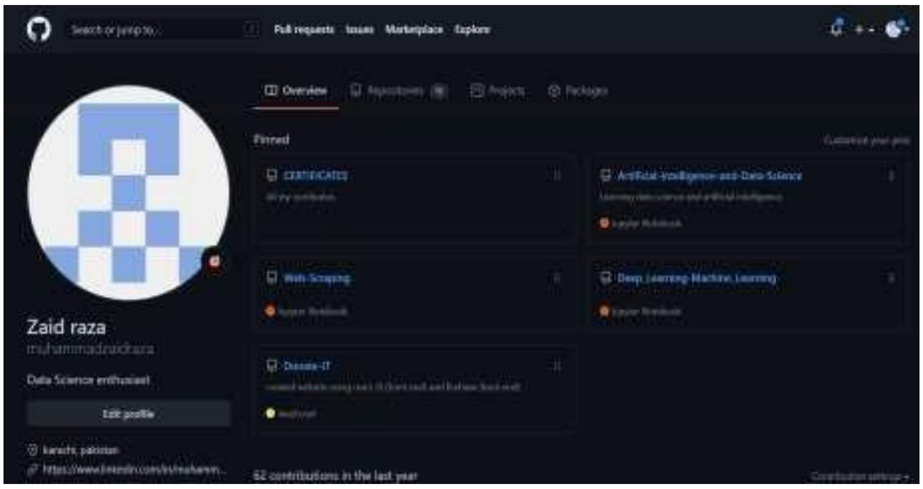
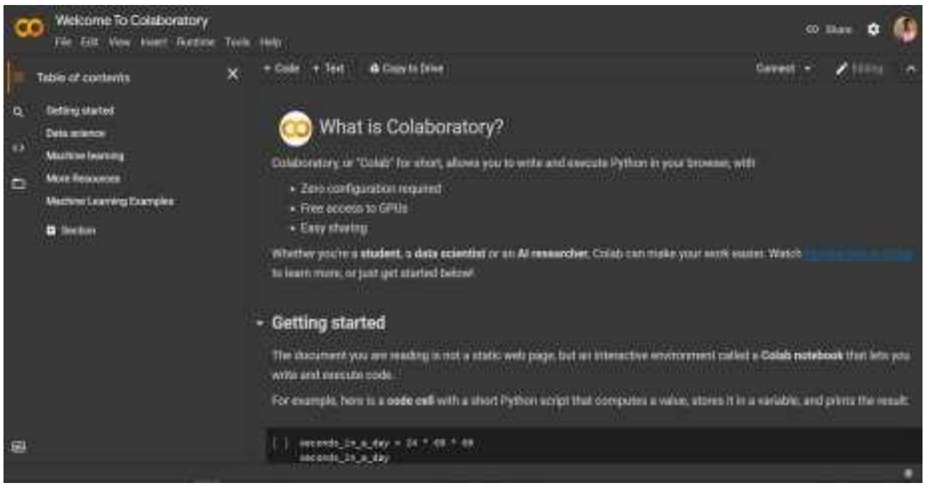


Figure 1.2: Github

1.5.5 Google Colab

Google Colab is a jupyter notebook environment that runs on the cloud. We don't need to setup the libraries and configuration by our own, it does all by itself. The most important feature of the google colab is its GPU and TPU service. It provides free GPU and TPU services to its users. Those who don't have access to GPU or TPU locally on their pc can use google colab GPU and TPU service.

We wanted to train our model using GPU so that it could be train faster and efficiently but we were not having access to GPU on our local machine. To overcome the GPU problem we used google colab to train our model.



What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [illegible] to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

Figure 1.3: Google Colab

1.5.6 Labelimg

It is free, open-source tool for annotation of the images. Labelimg support labelling in VOC XML or YOLO text file format.

We needed our own custom dataset according to our needs of project therefore, we used labelimg to label our dataset.

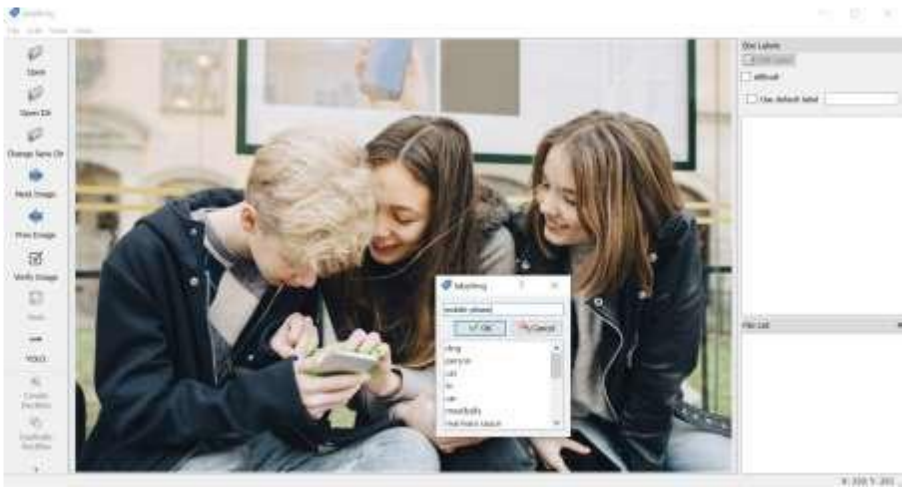


Figure 1.4: Labelimg

1.6 Chapter Conclusion

From above all, we conclude that we as a student developed a system that could restrict the mobile phone usage in sensitive places like banks, petrol pumps etc. It could also be implemented in different scenarios such as in examination hall to identify students using mobile phone for cheating purpose. We have used different tools and technologies such as python, OpenCV, VS code, Google Colab, Github.

Chapter # 2
Usage Scenario/User Interaction

2. Usage Scenario / User Interaction

This usage scenario for the system is explained below.

2.1 User profiles

The user profiles for our system are as following.

2.1.1 Management

Our system is designed to be used by only the authorized person from the institution who want to use our system. He/She can be from bank management, petrol pump management etc. The individual using our system has to provide either a recorded video or live video footage. Once the video is inserted into the system, the system does all the work by its own and provide the output to the user in the form of images. The images will be of the people using mobile phones in the restricted area. Then the management can take action against that individual according to their rules and regulation.

2.2 Use-cases

A use-case diagram is a diagram to represent the behavior of the system. It contain actor(management in our case), system and the interactions between them. Use-case diagram displays the behviour of the system in a very pleasing manner which makes the reader understand the system very easily.

Figure 2.1 shows the use-case diagram of our system. First of all, the user of our system that can be from management of any institution or any individual using it for its personal purpose has to provide video. The input provided by the user can be either recorded or live video footage. Once the video footage is provided to the system, the rest will be handled by the system on its own. The system takes the video footage and uses the yolov3 algorithm to detect mobile phone user, using mobile phone in restricted area such as in examination hall, petrol stations etc. When the system detects the mobile phone user, it automatically saves the image of the mobile phone user in the respective hardware of the user.

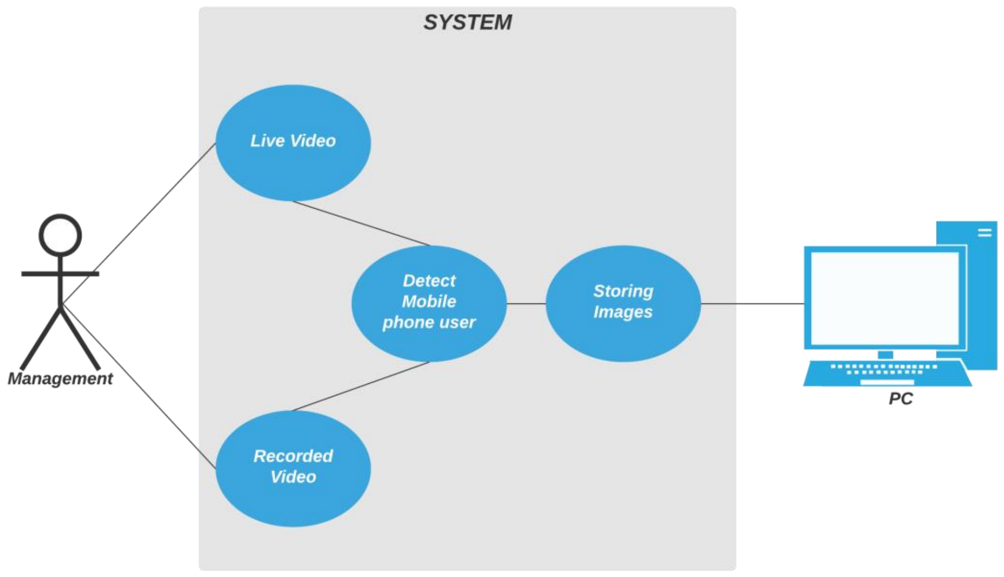


Figure 2.1: Use-case diagram

2.3 Chapter Conclusion

From this chapter we conclude that our system has only one user profile i.e management from departments who want to use our system. The management person, give the input video that could be recorded or live, the system process that video and return the output in the form of images. Those images contain the pictures of the person using mobile phone in the restricted area.

Chapter # 3
Functional and Data Description

3. Functional and Data Description

In this section, it is explained how our project handles the data and on which models it has been implemented upon. Every software has data needs and layered models each having their own module and sub modules. Architectural designs and design patterns help in the sequential development of the project with room for changes. Data objects are transformed as the execution continues to take place and this model describes what relationships these data objects have with the modules and sub modules and what kind of transfer of data takes place in between.

The Functional and Data Description component would come up with relationships allying main data objects and its essential parts that flow in our system. Data is basically all related to the videos gathered from different cameras placed in different places or captured from different users.

3.1 System Architecture

MOBILE PHONE USER DETECTION USING COMPUTER VISION is system that detects the mobile phone user using mobile phone in the restricted area such as petrol pumps, banks etc.

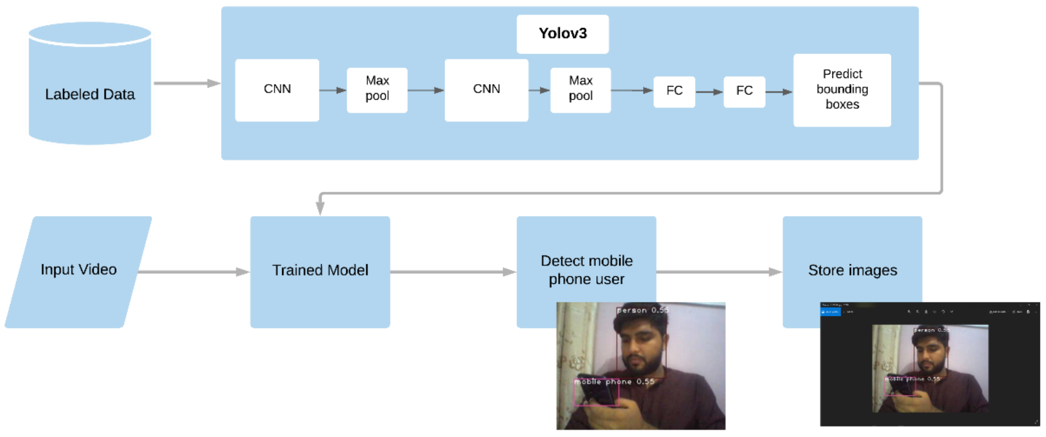


Figure 3.1: System Architecture Diagram

First, the project started with collection of data from various resources for instance website, images in our mobile phone gallery etc. The data collected was of images of mobile phone and person. Once the data is collected, it was labelled using labelimg tool freely available to use and also very easy to use. The labelled data contain the original image and its annotation coordinates in the .txt file.

The labeled data is given to YOLOv3 algorithm for model training. The training is conducted on google colab due to free access of gpu that can train model faster and efficiently. Without gpu the time taken to train the model was high and was less efficient. YOLOv3 is an algorithm in which series of CNN and maxpool layers are present that apply filters and extract feature from the images. The images are passed through the fully connected layers and then predict the bounding boxes and class probabilities of the images. The detailed architecture of YOLOv3 is discussed in section 3.1.1.

After the model training, it is executed using python programming language on local machine. The input video is given to the trained model. The video can be recorded or live. The trained model predict the bounding boxes and class probabilities from the input video. When the model detects any mobile phone user in the video, it captures the image and store it in the respective hardware.

3.1.1 YOLO Architecture

All the previous algorithms called region-based algorithms related to object detection have used only that region which has the high possibilities of containing the object. All algorithms do not look at the complete image. While YOLO algorithm does not look only the desired region but by taking the complete image in a single instance, it predicts the bounding box coordinates and class probabilities for these boxes. Therefore, YOLO is much different algorithm from region-based algorithms. After predicting bounding boxes and class probabilities, YOLO split the input image into (S X S) grid. Then image classification and localization are applied on each grid. Then for each bounding box, YOLO gives the class probability and offset values that are used to localize the desired objects within an image.

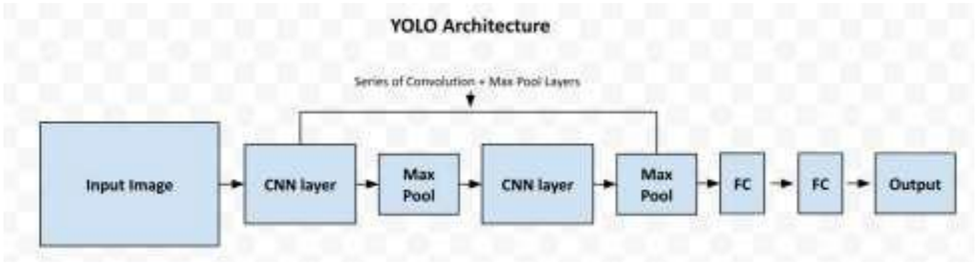


Figure 3.2: YOLO Architecture diagram

The input image is passed through a deep convolutional neural network (CNN) which is a series of Convolution and maxpool layers, followed by two fully connected layers to get the output.

The first layer is convolutional layer. In this, a filter is applied to the input which is an array of numbers in order to create feature maps that summarize the presence of those features in the input. Convolutional layers prove very effective, and stacking convolutional layers in deep models allows layers close to the input to learn low-level features (e.g. lines) and layers deeper in the model to learn high-order or more abstract features, like shapes or specific objects. A limitation of the feature map output is that they only record the precise position of features in the input means that small movements in the position of the feature in the input image will result in a different feature map. Now, nonlinearity (e.g. ReLU) has been applied to the feature maps output by a convolutional layer.

After the convolutional layer, a nonlinear function takes place that means each value of the feature map is passed through this function. It enables complex mappings between input and output layers in the network.This linear function brings the relevant features on the image into focus. Relu is the most commonly used non-linear activation function. This activation function is applied per pixel and replaces all negative pixel values in the feature map by zero.

After the feature map is adjusted, it is passed through the pooling layer.The addition of a pooling layer after the convolutional layer is a common pattern used for ordering layers within a convolutional neural network that may be repeated one or more times in a given model. The pooling layer operates upon each feature map separately to create a new set of the same number of pooled feature maps.

Together, the convolutional layer, non-linear activation function and the pooling layer extract the useful features from an image. The result of using these layers is to detect objects in the given input and produce the output.

3.2 Data Description

In this system, real-time video is given to system as input. The system takes the input data in the form of video and then detect the person visible in the video using the mobile phone. If the person is detected using the mobile phone, that particular image is saved into our hardware.

3.2.1 Major data objects

Our project “Mobile phone user detection using computer vision” is simple project that does not use many data objects. Images are used as data for training our model on YOLOv3 algorithm. We also use video as data when executing the system. When the system is executed it takes input video which can be live or recorded, the inputvideo is then processed to give us outputs.

3.3 System Interface Description

In MOBILE PHONE USER DETECTION USING COMPUTER VISION when image is saved and stored in the hardware, you can view that image in the result directory which also contain all previous stored outputs.

3.3.1 Output interfaces

Figure 3.2 shows the output interface of our project. When the images are saved in the result directory, it can be viewed using the hardware built-in software to view images.

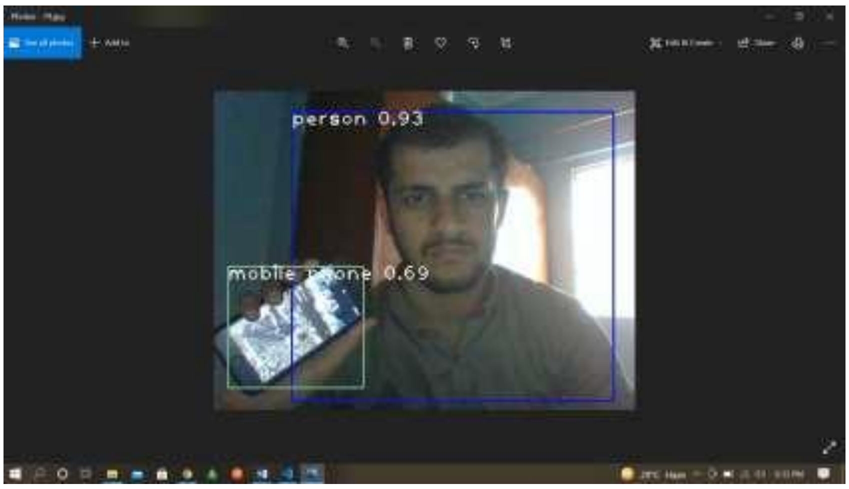


Figure 3.3: Output interface Screenshot

3.4 Chapter Conclusion

In this chapter, we have discussed the system architecture of our project. We have also briefly discussed the YOLO algorithm architecture. We have used YOLO algorithm in our system to perform object detection. We have choosen YOLO algorithm after doing research on various object detection algorithm like SSD, fast-RCNN and many more. Hence, we came to the conclusion that YOLOv3 is best performing object detection algorithm among all the other object detection algorithms. In this project major data was used in the form of images and videos. We used images to train our model. Video is used as input data inserted into the system to detect mobile phone users from the video.

Chapter # 4
Behavioral Model and Description

4. Behavioral Model and Description

Behavior of our system is explained here.

4.1 Description for system behavior

Our model works to restrict the people to follow rules and regulation regarding mobile phone usage. It detects the mobile phone user from the video and save the images of the people using the mobile phone. Video can be recorded or it can be live.

Any institution who want to use this model can access it, institution could be either from government sector or private sector.

4.1.1 Events/interrupts

This system includes a single event. The user must provide the video as input so that our model could process video and return the output. Video could be recorded or live feed.

4.1.2 States

We have the following states.

- Recorded Video
- Live Video
- Run Detection
- Save Frames

4.2 State Transition Diagrams

A state transition diagram represents an illustration of the various states that an object can attain during the life cycle of the program and also the transition between those states. The state diagram of the “mobile phone user detection using computer vision”is as follow.

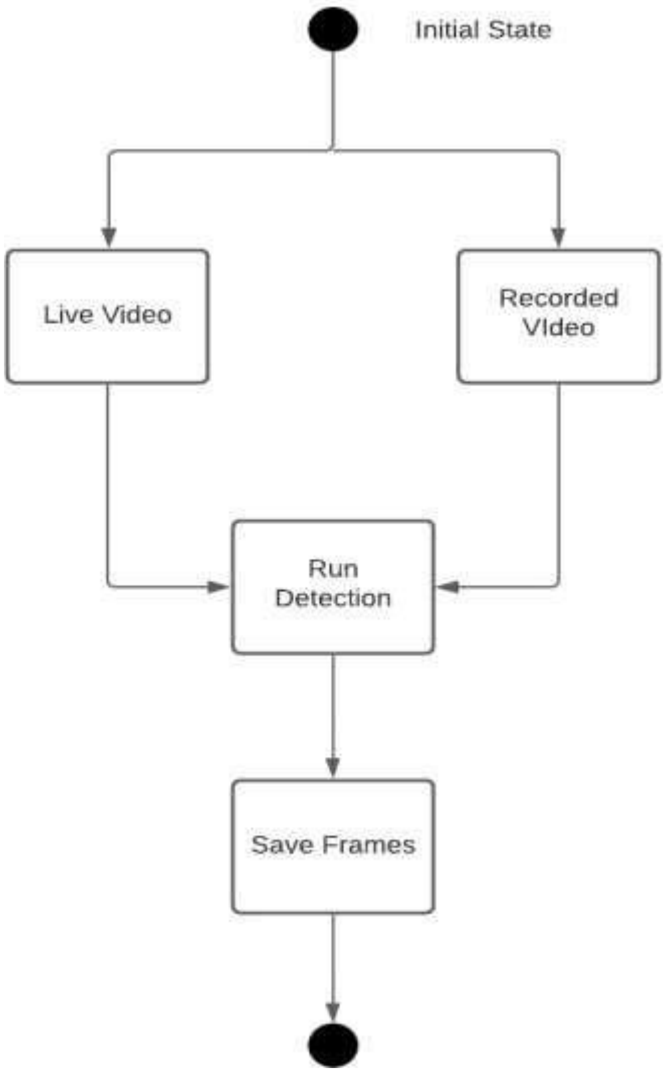


Figure 4.1: state transition diagram

In our system, the user will first provide the video as input. The video can be either in recorded video or live video. The video then enter into next phase i.e run detection. The frames of the video are processed by the system in this states using object detection technique. The detected mobile user image is passed to then next state. In the next state, images of the mobile user in restricted area are saved into the user hardware.

4.3 Chapter Conclusion

Our system is very simple, it just take the input from the user in form of video. Video can be either live or recorded, our system process the video and detects the mobile phone user and save the image into the respective user hardware. We have also included the state transition diagram in this chapter so that reader can easily understand our project.

Chapter # 5
System Prototype Modeling and Simulation Result

5. System Prototype Modeling and Simulation Results

Our main focus during the project was to have better accuracy so as to give better results.

5.1 Description of system modeling approach (if used)

At the beginning of our project, we used Google Colab as a cloud virtual machine for training our model on yolov3. As we did not have any GPU hardware on our local pc so we decided to use Google Colab for the faster and better training of our model.

We used images data from various resources, all the images were labeled and then given to the model as input for training. We used a supervised learning technique in which data labeled by a human is given to an algorithm for training the model.

We used the trained model and run it on our local machine. All the processing on frames and object detection was performed using python and OpenCV on video input given by the user. First, the results did not meet our expectations. The main reason behind the low accuracy result was the lack of data given to the algorithm for model training. We collected more data and again labeled images and the process was again repeated. After the increment in data for model training, the output results were satisfactory.

5.2 Simulation results

Before creating the actual project, we created the prototype of our project using a very small set of data to ensure that we are on the right path. The prototype was tested and the results are displayed in figure 5.1 and figure 5.2.

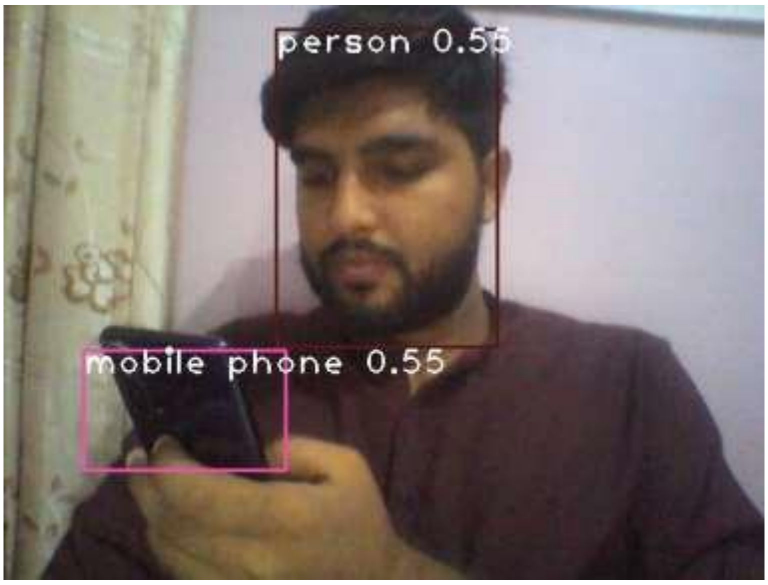


Figure 5.1: Simulation result

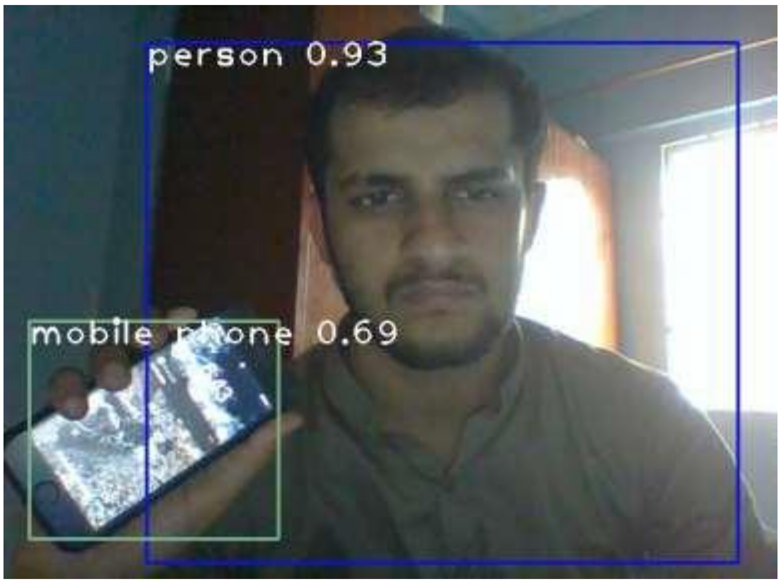


Figure 5.2: Simulation result

5.3 Special performance issues

Special performance issues are as following

5.3.1 GPU Problem

The first major problem we faced during the project completion was the lack of a graphics processing unit (GPU). GPU helps process video and images more efficiently and more frequently as compared to the Central processing unit (CPU). Due to lack of GPU, the first-time result was not according to our expectations.

5.3.2 Low Specification of Laptop

Low specification of our local system was also an issue. It was taking more than usual time to process the input video and give the results.

5.3.3 Camera Quality

Camera quality was also a significant issue. Due to low quality, the differentiation between mobile phones and other devices looking like mobile phones such as remote was not clear. As show in figure 5.3, due to low quality of picture it is difficult for our model to distinguish between mobile phone and remote.

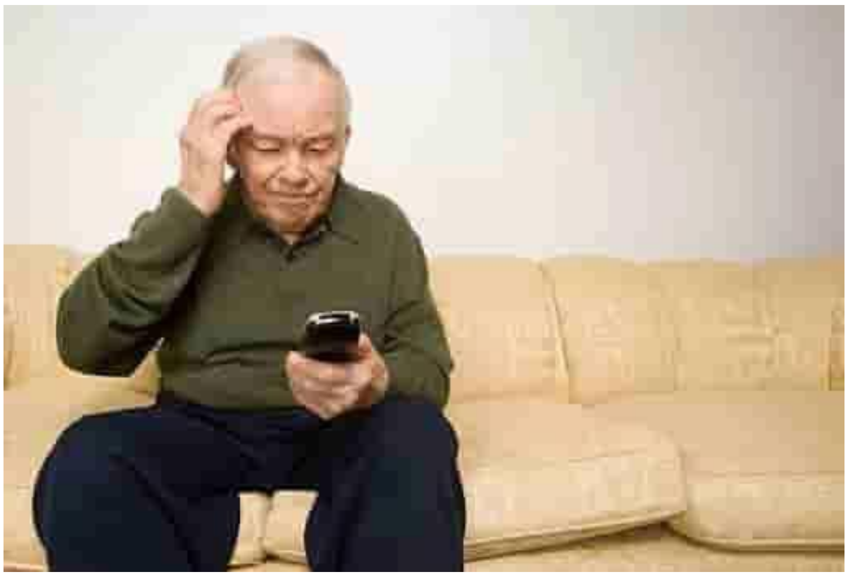


Figure 5.3: low quality image

5.4 Chapter Conclusion

In this chapter we have discussed the approach we used for creating the project. We have also attached some simulation results. Some of the performance issues like GPU problem, Low specifications of laptop and low quality camera are highlighted in this chapter.

Chapter # 6
System Estimates and Actual Outcomes

6. System Estimates and Actual Outcome

For every developer, the skill to evaluate estimated time to develop a given product or feature comes with experience and time. Cost estimation refers to the cost of resources and associated costs needed to execute the project. Both these disciplines are well-developed skills that help in improving the project objectives within the approved timeline and budget.

6.1 Historical data used for estimates

Our project "MOBILE PHONE USER DETECTION USING COMPUTER VISION" is a system that detects the mobile phone user from the input video and saves the images of the person using the mobile phone is stored in system. To find the cost and the time, we did our research on the process and details of the techniques used in the project. We read the articles and research papers.

6.2 Estimation techniques applied and results

We have used estimation technique called "COCOMO" to estimate the cost of our project "MOBILE PHONE USER DETECTION USING COMPUTER VISION". The COCOMO technique is briefly discussed here.

6.2.1 Estimation technique COCOMO

Constructive Cost Estimation Model (COCOMO) is a technique using which we can estimate the project price. It is procedural cost estimate model for software projects and often used reliably to predict the various parameters of software development process such as cost, size, effort, time and quality. It can be applied at three different classes of project that include Organic project, Semi-detached projects, Embedded projects.

Organic - A project is said to be of type organic if the team members required to complete the project are adequately low, the problem is well understood and is solved in the past.

Semi-detached – A project is said to be of type semi-detached if the various characteristics like team size, experience, knowledge of various programming environment lie between the organic and embedded. The project that lie in this category are difficult to build and require more creativity and experience as compare to organic.

Embedded – A project is said to be of type embedded if the project requires highest level of experience, creativity, complexity falls under this category. For these projects, developers need to be highly experiences and skilled.

CALCULATION - These are the following calculations used by the basic COCOMO to estimate efforts.

1. Effort Applied (E) = $a^{b}(KLOC)^{b}b$ [man-months]

2. Development time (D) = $c^{b}(Effort\ Applied)d^{b}$ [months]

3. People required (P) = *Effort Applied / Development Time*

The coefficients a^b, b^b, c^b, d^b are given in the following table.

Software Projects	**a**	**b**	**c**	**d**
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Table 6.1: COCOMO coefficients

6.2.1a Estimates for technique “COCOMO II”

We used online available tools to calculate the estimated cost of our project. The results of the COCOMO estimation model are given below.

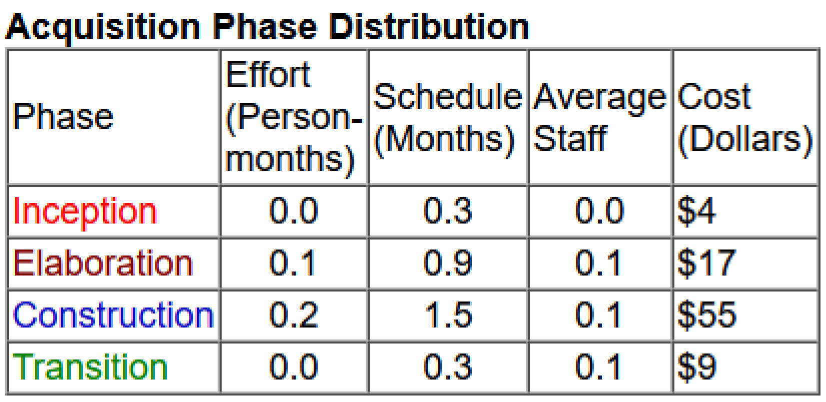
Effort = 0.2 person-months

Schedule = 2.3 months

Cost = $72

Total equivalent size (SLOC) = 103

Effort Adjustment Factor (EAF) = 1.00



Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.0	0.3	0.0	$4
Elaboration	0.1	0.9	0.1	$17
Construction	0.2	1.5	0.1	$55
Transition	0.0	0.3	0.1	$9

Table 6.2: Acquisition Phase Distribution

6.3 Actual Results and Deviation from Estimates

The deviation between the estimation and the actual resource utilization are display in table 6.3.

	COCOMO	**Actual**
Efforts applied	0.2	16
Development time	2.3	8
People required	0.1	2

Table 6.3: Estimation difference table

6.4 System Resources (Required and Used)

All the resources i.e software, hardware, tools used to build the project are listed below.

6.4.1 System Resources Required

Following resources were estimated before the development of the project was started.

- Operating system: Window 10
- Processor: intel i5 5th gen
- GPU (with basic specs)
- Development environment (IDE): Visual studio code
- Object detection: Yolov3
- Programming language: python

6.4.2 System Resources Used

Following resources were used in the development of the project.

- Operating system: Window 10
- Processor: intel i7 5th gen
- GPU
- Development environment (IDE): Visual studio code
- Object detection: Yolov3
- Programming language: python
- Google Colab

6.5 Chapter Conclusion

In this chapter we have distinguish between the estimates made for the project and the actual cost of this project. All the resources which were required before starting the project and all the resources used in this project are mentioned in this chapter.

Chapter # 7
Test Plan

7. Test Plan

Test plans refer to the quality and quantity based testing of different components, modules and subsystems inside the project separately and with inter-connectivity performance (integration) in between. Test plans specify the timing and complexity and execution capabilities of the project and provide a summary of how the application behaves in front of the user with extensive parameters to change from in order to achieve maximum efficiency.

7.1 System Test and Procedure

MOBILE PHONE USER DETECTION USING COMPUTER VISION is a system that is manually tested by applying different techniques to ensure its proper functioning. The main focus during the process was to ensure that it detect objects from various distances, angles.

7.2 Testing methods

We have used multiple methods to ensure that it is performing according to our expectations. Starting with placing the objects at different ranges from the camera, then mobile position testing by placing the mobile at different positions of the screen, the motion testing to ensure object is detected while it is in motion and at last we used multiple test objects to ensure that if multiple test objects appears in front of camera, it detects all of them.

7.2.1 Range testing

First range testing is conducted on our system. In this test, we placed test object at different ranges from the camera to ensure that it detect the test object from various ranges such as 2m, 5m and 7m.

7.2.1a Distance of 2m

First, the test object is placed at the distance of 2m from the camera. We wanted to check that when the test object is at 2m distance from the camera, does camera captures it or not. As seen in figure 7.1 and figure 7.2, the object is detected. This test is successful.

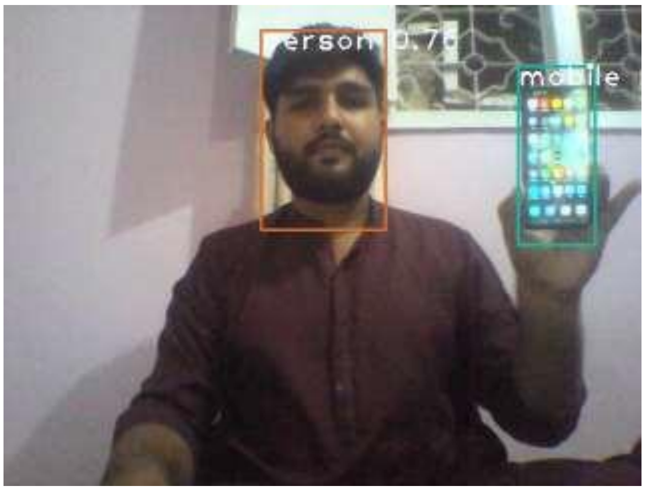


Figure 7.1: Test object at 2m

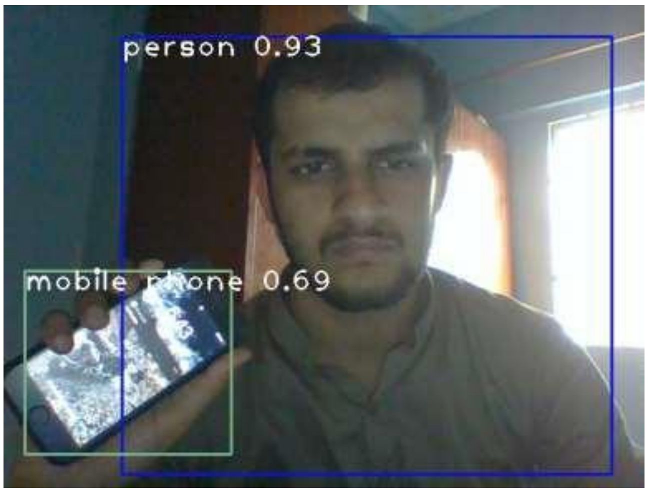


Figure 7.2: Test object at 2m

7.2.1b Distance of 5m

In this range test, we placed the test object at 5m distance from the camera. Using this test, we wanted to ensure that does the camera detects the object if it is 5m far from the camera. As seen in figure 7.3 and figure 7.4, the test object is detected. This test is successful.

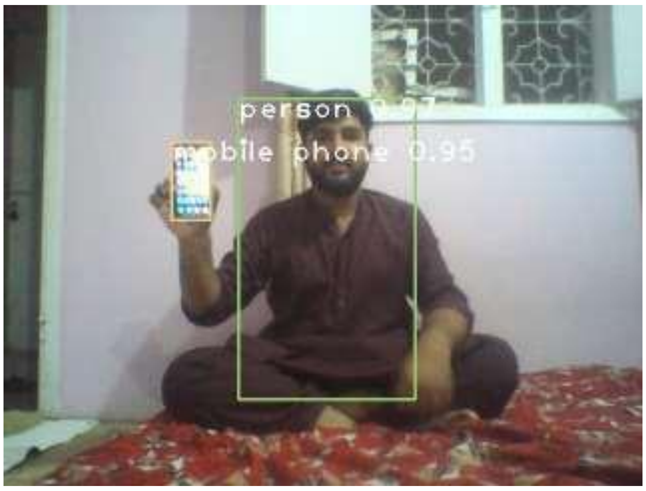


Figure 7.3: Test object at 5m

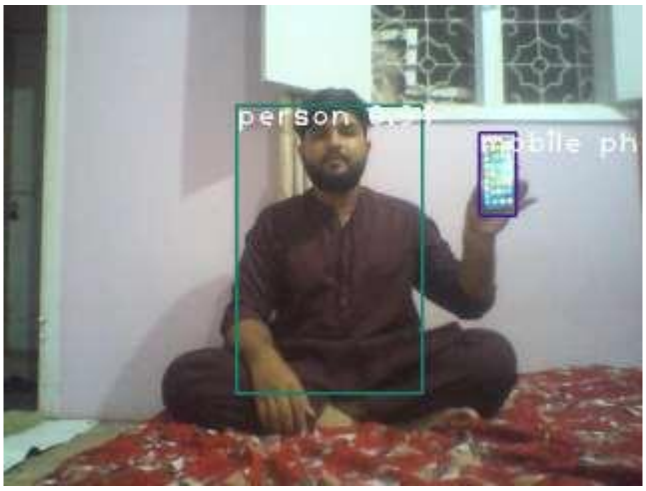


Figure 7.4: Test object at 5m

7.2.1 c Distance of 7m

In the last range test, the test object is placed at the distance of 7m from the camera. We want to check that does the system detect the test object if it is 7m far from the camera. As seen in figure 7.5, the test object is detected and hence, the test is successful.

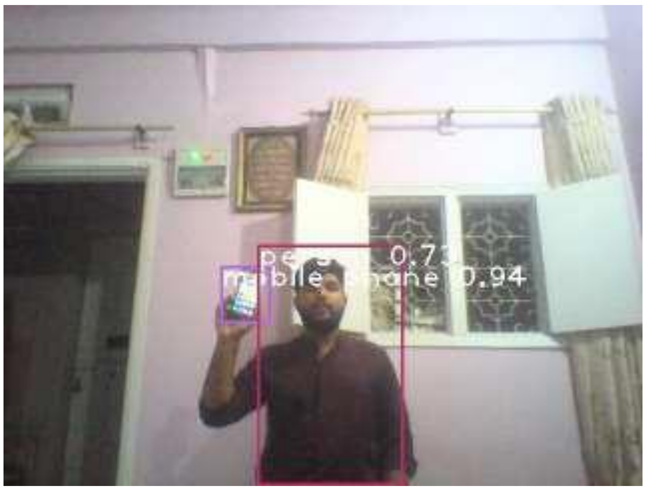


Figure 7.5: Test object at 7m

7.2.2 Mobile position testing

Secondly we conducted mobile position testing on system. In this test, we placed the mobile phone at different positions in the screen such as left, right, bottom etc. We conducted this test to ensure that if mobile phone appears on any side of the screen, it is detected.

7.2.2a Left side of screen

In mobile position testing, firstly the test object is placed at the left side of the screen. We conducted this test to ensure that if the test object appears at the left side of the screen, will the camera be able to detect it or not. As seen in figure 7.6 and figure 7.7, the test object is detected and the test is successful.

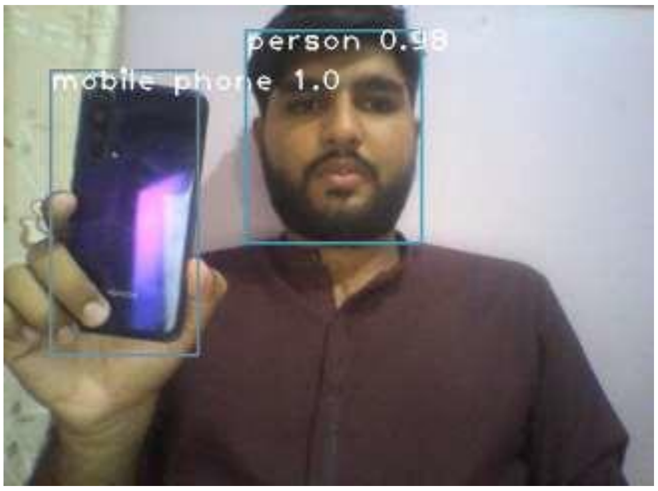


Figure 7.6: Mobile phone at left side of screen

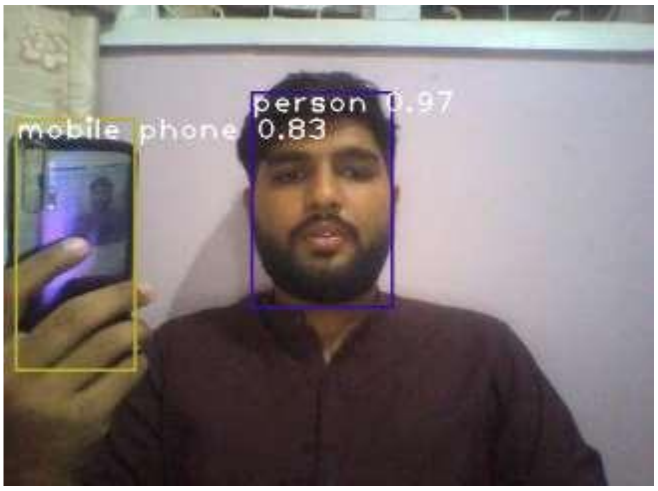


Figure 7.7: Mobile phone at left side of screen

7.2.2b Right side of the screen

Now, the test object is placed at the right side of the screen. We conducted this test to ensure that if the test object appears at the right side of the screen, will the camera be able to detect it or not. As seen in figure 7.8, the test object is detected and the test is successful.

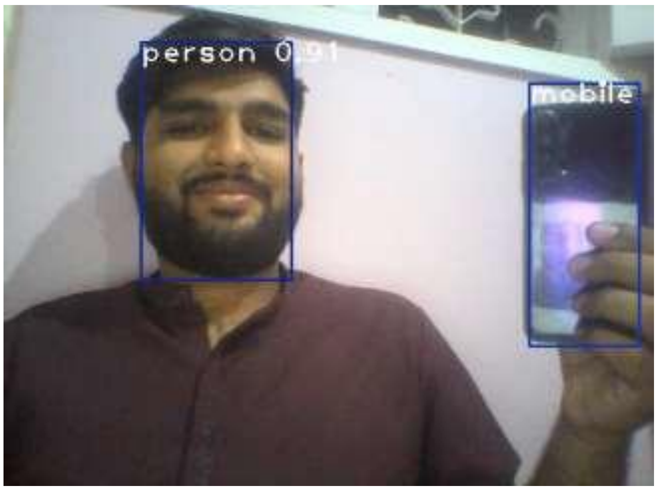


Figure 7.8: Mobile phone at right side of screen

7.2.2c Bottom of the screen

Now, the test object is placed at the bottom of the screen. We conducted this test to ensure that if the test object appears at the bottom of the screen, will the camera be able to detect it or not. As seen in figure 7.9, the test object is detected and the test is successful.

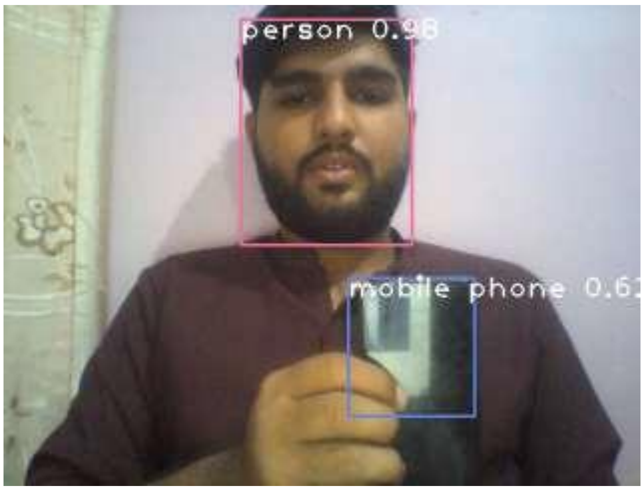


Figure 7.9: Mobile phone at bottom of screen

7.2.2d Top corners of the screen

Now, the test object is placed at the top-right and top-left corner of the screen. We conducted this test to ensure that if the test object appears at the top-right and top-left corner of the screen, will the camera be able to detect it or not. As seen in figure 7.10 and 7.11, the test object is detected and the test is successful.

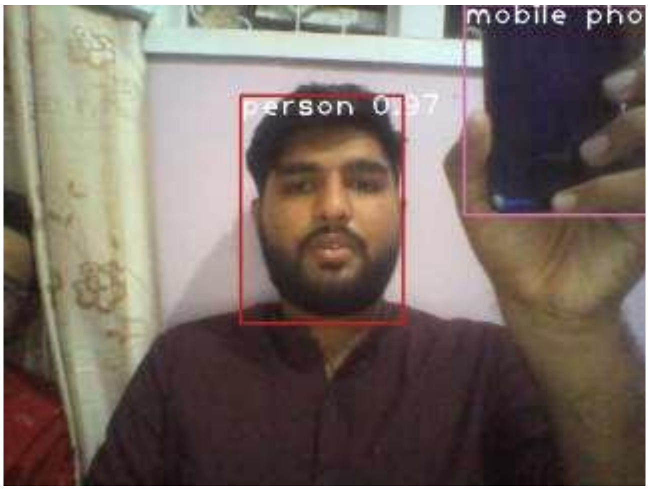


Figure 7.10: Mobile at top-right corner of screen

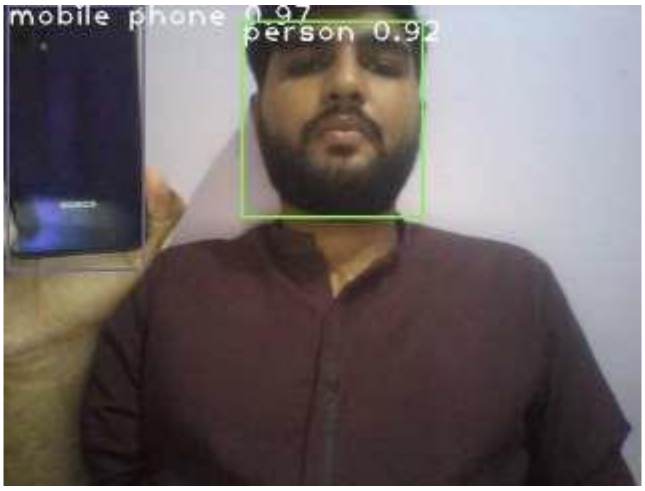


Figure 7.11: Mobile phone at top-left corner of screen

7.2.2e During the mobile usage with slight angle change

Now, the test object is placed with slightly tilted angle. We conducted this test to ensure that if the test object appears in slightly tilted position in front of the screen, will the camera be able to detect it or not. As seen in figure 7.12 and 7.13, the test object is detected and hence, the test is successfully conducted.

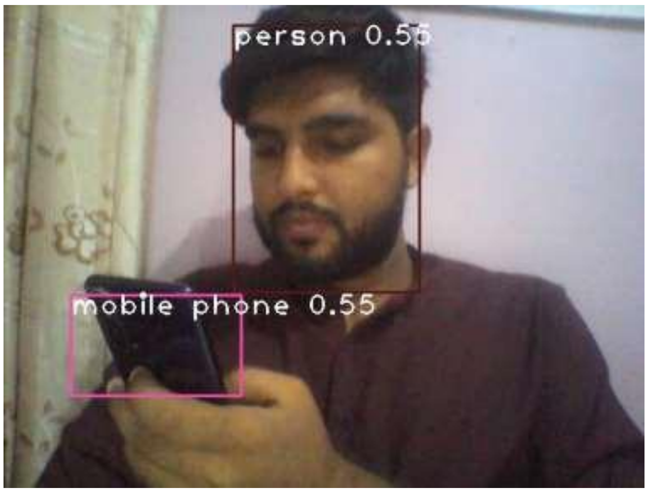


Figure 7.12: Mobile phone at slight angle change

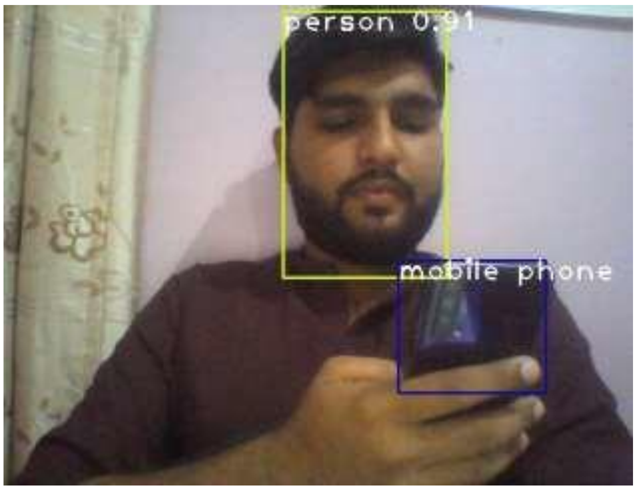


Figure 7.13: Mobile phone at slight angle change

7.2.2f When target is tilted (almost sideways)

Now, the test object is placed with tilted angle (almost sideways). We conducted this test to ensure that if the test object appears in tilted position (almost sideways) in front of the screen, will the camera be able to detect it or not. As seen in figure 7.14, the test object is detected and hence, the test is successfully conducted.

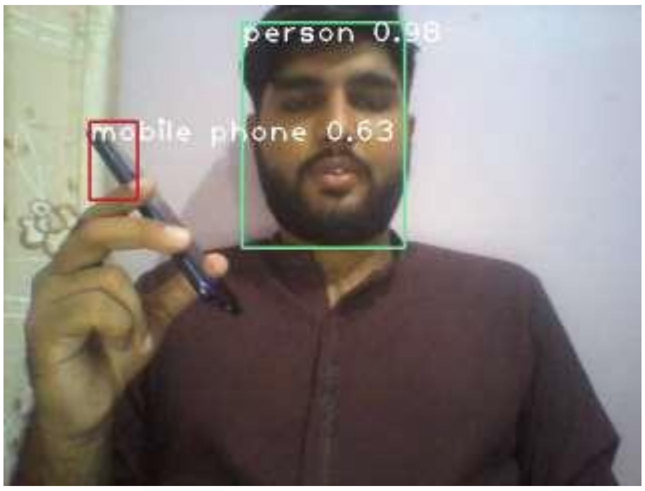


Figure 7.14: Mobile phone almost sideways

7.2.3 Motion testing

In motion testing, we test the object while it is in moving position. We conducted this test to ensure if in real-life the test object is moving, will the camera be able to detect it or not. In motion testing first the target was in motion while the camera was on rest and then camera was in motion while target was in rest.

7.2.3a Target in motion while camera on rest

In motion testing, firstly the mobile phone is in motion and the camera is at rest position. In real-life, usually the target is in motion so to ensure our system detects it we perform this testing. As seen in figure 7.15 and 7.16, the test object is detected and test is successful.

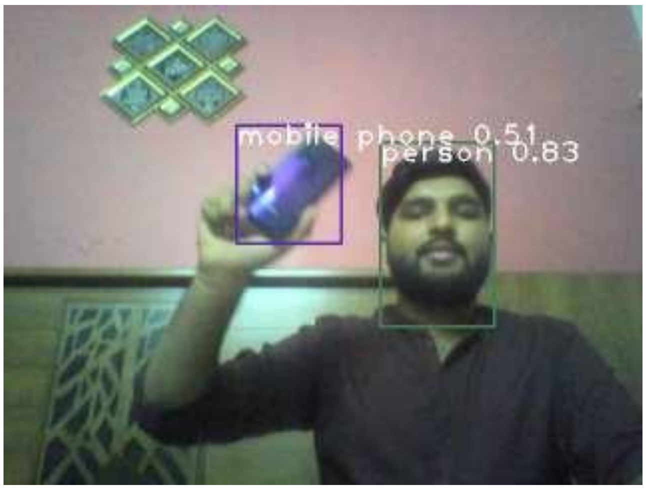


Figure 7.15: Test object in motion

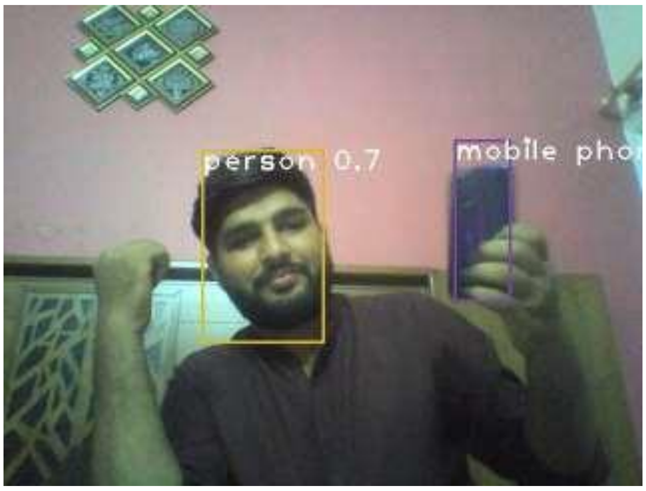


Figure 7.16: Test object in motion

7.2.3 b Camera on rest while target in motion

Now, the camera is in motion and the mobile phone is at rest position. In real-life, the camera is in motion so to ensure our system detects the test object we perform this testing. As seen in figure 7.17 and 7.18, the test object is detected and test is successful.

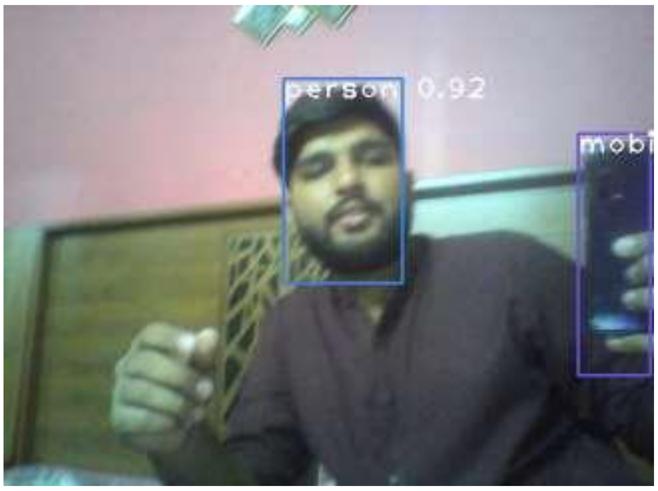


Figure 7.17: Camera in motion

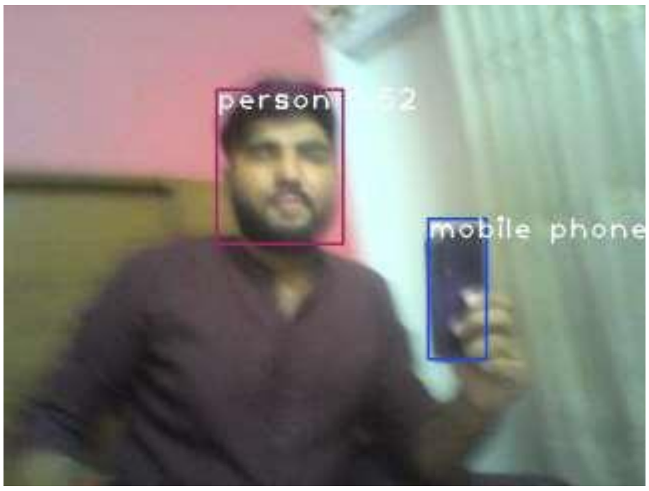


Figure 7.18: Camera in motion

7.2.4 Multiple target testing

This is the last test we conducted on our system. In this test, we used multiple targets such as multiple mobile devices and more than one people. In real-life, usually more than one test objects occur at single time, to ensure that it detect all test objects on the spot we perform this testing.

7.2.4a Multiple mobile phones

In this test, multiple mobile phone devices are being used by single person. In real-life, there are many scenarios when the person uses multiple mobile devices at single time. To ensure that our system detects multiple mobile devices we performed the test. As in figure 7.19, we see that multiple mobile phone devices are detected by our system and hence, the test is performed successfully.

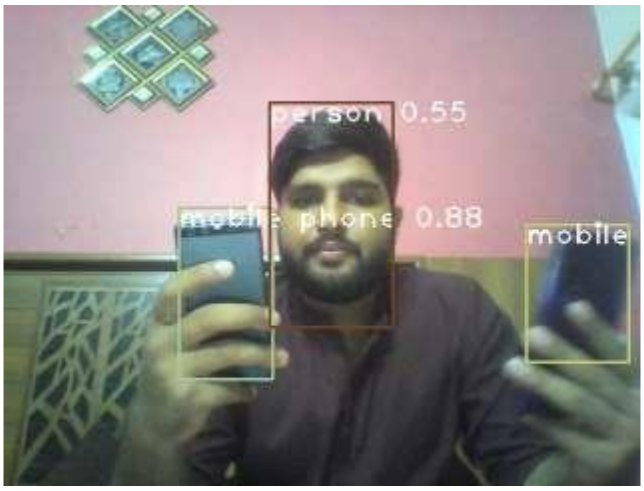


Figure 7.19: Multiple mobile phones

7.2.4b More than one people

In this test, more than one people appear in front of camera. In real-life, there are many scenarios when more than one person appears in front of the screen at single time. To ensure that our system detects more than one person we performed this test. As in figure 7.20 and figure 7.21, we see that more than one people are detected by our system and hence, the test is performed successfully.

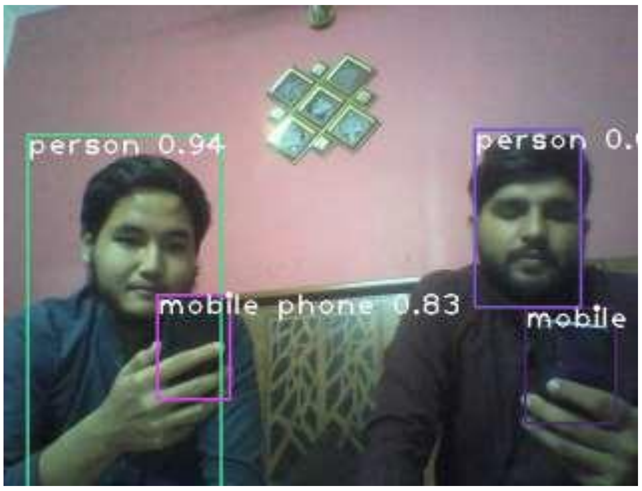


Figure 7.20: More than one person

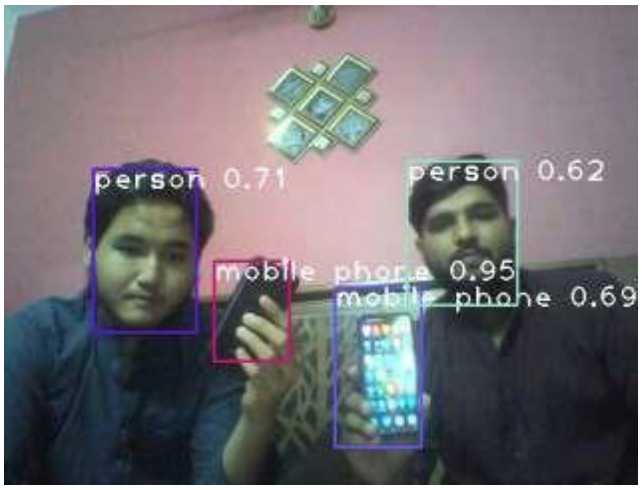


Figure 7.21: More than one person

7.3 Failed Cases

We tested our system using different methods to ensure its proper functioning, most of the time it detected object perfectly but sometimes it failed to detect the test object. In figure 7.22, we can see that person is using a mobile phone but it is not detecting the mobile phone user.

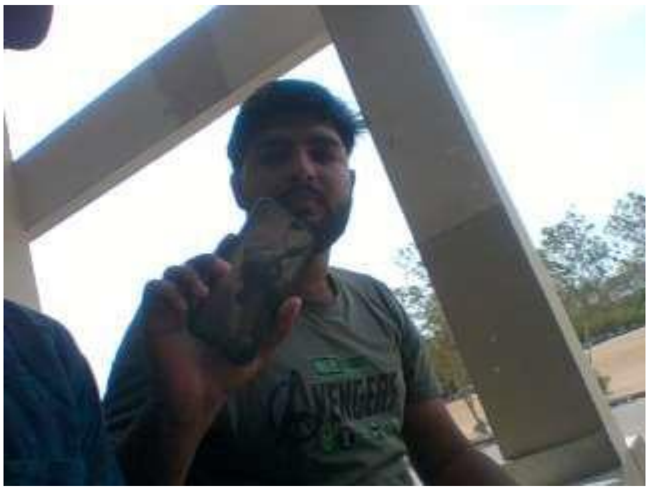


Figure 7.22: Failed case

7.4 System Accuracy

In this section, we have compared different algorithm used to perform the same task and what is the reason behind choosing YOLOv3 for the project. We have also highlighted the accuracy of our system.

7.4.1 Reason of choosing YOLO Algorithm

Object detection is technique that detect object from images and video based on the training data. It detects parts by differentiating the object based on its types in pictures and videos. Object detection works by checking the feature from the test relative to the training data, the output is classified and object name is displayed. Object detection can be performed using different algorithm which give different performance based on their architecture.

In R-CNN algorithm, it will first find regions in the image which are called region proposals that may contain an object. Then it will calculate CNN features from the region proposals. In the last step, it will classify the objects using the extracted characteristics. Regions with convolutional neural networks uses selective search for finding regions in an image and it creates

2000 region proposals for each image, i.e. we get the region of interest (RoI). During the classification of object in the image, R-CNN will only see small regions and regions having good output. The region proposals are cropped out of the image and all regions are reshaped into a fixed size. By applying bounding box regressor all regions are classified with special class specific linear support vector machines (SVMs). The region proposals are determined by checking total of positive and negative values. For each object area the bounding box regression is applied to the counted regions and then it is filtered with a non-maximum suppression (NMS) to create the bounding boxes. R-CNN has some limitations, it has to extract 2k regions for each image using selective search algorithm which is very time consuming, R-CNN model is costly and slow, R-CNN model is a multi-phase procedure.

The limitations of R-CNN were removed by introducing classification as well as bounding box regression and designed a unique CNN structure called fast R-CNN. It can find object better precisely than R-CNN. This method is one-step process with little loss of different tasks and this method changes the whole network altogether. Extra memory is not required for storing the calculation in Fast R-CNN. In the model fast R-CNN, the image is processed with deep convolutional network and max pooling layers to produce a convolutional image layer with different region proposal. Then, for each region proposal the pooling layer extracts a fixed-length object's characteristics from the convolutional layer. Each object's characteristics is placed into an order of fully connected layers that gives two common output layers. One output layer that gives softmax probability between the object and background values and second output layer produces four selection box positions values for each of the object. Although, fast R-CNN has region proposal, all its layers are trained with a multi-job loss in a one-step process. It improves the accuracy and testing time with the saving of extra expense in the storage space. However, the progress is not effective as the region proposals are created separately by additional process, which is expensive.

In both the method, R-CNN and fast R-CNN the object is detected by performing selective thorough search. This thorough search is a long process and takes larger time affecting the performance of the object detection method. Object detection algorithm called faster R-CNN removes the selective search process. In faster R-CNN, an additional region proposal network was introduced which takes convolutional characteristics from the convolutional network and computes it to find the object, instead searching again for the object in the image. Faster R-CNN consist of two sections. The first section is having a fully convolutional neural network that creates region proposal networks, and the second section is having the fast R-CNN detector that calculates the proposed regions for object classification. This whole algorithm is a combined process for finding the object.

Single Shot MultiBox Detector (SSD) is one among the leading object finding procedures. The accuracy of SSD for object finding method, came about 74.3% mean average accuracy at 59 frames per second making execution time 3 times faster and more accurate than faster R-CNN which is tested on typical datasets like common object in context (COCO). SSD takes on the idea of anchor in faster R-CNN. The bounding box is determined by the earlier boxes with varying values and measurement, then the characteristics is extracted by CNN and then classification and regression are done correctly. The whole method is executed only in a single process. SSD network selects the priors which contains the object of interest and find the coordinates that match exactly the object’s shape. These fully convolutional bounding box are same as anchor boxes used in Region Proposal Networks. The method is simplified by changing the 3*3 convolution layers to 1*1 layers to predict the values of the particular object and so an intermediate layer is not required. SSD design is the first work which joins different values from multiple features and with different resolutions classifies the object and improves quality of object detection. SSD detects multiple objects without sharing convolutional layers. SSD shares a same objective as MultiBox, yet, it can detect multiple categories in a single shot calculation unlike the two-stage method. After that, hard negative mining is performed where the highest prior values are taken which leads to faster performance of the network. Ending results are got by performing Non-Maxima Suppression (NMS) on multi-scale bounding boxes. Yet, SSD is not efficient in finding some little objects correctly.

YOLO is a combined architecture model and is extremely fast. YOLO algorithm detects objects in an image at 45 frames per second. It is better than other detection methods, including DPM and R-CNN. But YOLO v3 is faster than previous YOLO. YOLO v3 runs in 22 ms at 28.2 mean access precision as accurate as SSD but three times faster than SSD. YOLO v3 is also good at detecting small objects in an image. YOLO has YOLO-based convolutional neural network group of algorithms to find the object. YOLO is considered as fully convolutional network because it makes use of only convolutional layers. The YOLOv3 technique proposes finding the object a regression problem. The method is designed in DarkNet based on VGG which was previously developed in GoogleNet. The softmax loss in YOLO v2 is replaced by a logistic loss, which got better in finding small object. YOLO v3 removes region proposal method and joins every process in one network so as to form an accurate object detection algorithm. It finds the exact object from the different types of object. YOLO v3 method uses three anchors and so three bounding boxes for each cell is generated. The YOLO v3 method divides the image into a small net of cells and so each cell will give a selection box offsets and classify the objects through forward convolution. The bounding boxes are combined to detect the object after a post-processing process by the algorithm. If a grid cell is center of an object, then that grid cell is considered in finding the object. Each grid cell is used to find the exact area of the selection boxes.

Several object detection techniques like R-CNN, fast R-CNN, faster R-CNN, single shot detector (SSD), YOLO v3 etc. are being discussed and compared. From the discussions, it is found that as the model was developed the speed and accuracy has been improved and increased. Fast R-CNN is improved than R-CNN but Faster R-CNN is much improved than fast R-CNN. Also, single shot detector is better than faster R-CNN, while YOLO v3 is better than single shot detector. Earlier, till YOLO v3 was not developed SSD was the best. But now, the best technique found latest is YOLO v3 which is much better than SSD also and much faster than SSD. YOLOv3 is extremely fast and accurate. Hence, using YOLO v3 model, we can detect multiple objects faster and add our own images and labels in the datasets. This YOLO v3 model is beneficial as it can detect object directly and all objects are detected single time only in this model.

7.4.2 Model Accuracy

Confusion Matrix:

Confusion matrix is technique used to measure the effectiveness of a system based on machine learning classification where the output can be of 2 or more than 2 classes. It is a table containing the combination of predicted and actual values. The confusion matrix is very helpful in calculating accuracy, recall, precision and error-rate. The values used in the confusion matrix are TP, TN, FN, FP. These terms are defined below.

True Positive (TP): Model predicted the mobile phone user ant it's correct.

True Negative (TN): Model didn't predicted the mobile phone user and it's correct.

False Negative (FN): Model didn't predicted the mobile phone user and it's incorrect.

False Positive (FP): Model predicted the mobile phone user and it's incorrect.

For our system the confusion matrix is

n = 390	Predicted Yes	Predicted False
Actual Positive	200 (TP)	35 (FN)
Actual Negative	25 (FP)	130 (TN)

Table 7.1: Confusion Matrix

Accuracy:

Accuracy is a performance measure that is simply a ratio of correctly predicted observation to the total observation. The higher the accuracy, the better our model will be. For measuring the accuracy of our system, we tested on 390 sample images from which TP and TN are 330 images and FP and FN are 60 images.

$$
\begin{aligned}
Accuracy &= TP+TN \,/\, TP+FP+TN+FN \\
&= 330 \,/\, 390 \\
&= 0.84
\end{aligned}
$$

Error rate:

Error rate is a performance measure that is termed as inaccuracy of predicted output values. It is a proportion of the cases where the prediction is wrong. For measuring the error rate of our system, we tested on 390 sample images from which TP and TN were 330 images and FP and FN were 60 images.

Error rate = FP+FN / TP+FP+TN+FN

= 60 / 390

= 0.15

Precision:

Accuracy is a performance measure that is a ratio of correctly predicted positive observation to the total predictive positive observations. For measuring the precision of our system, we tested on sample images from which TP were 200 and FP were 25 images.

Precision = TP / TP+FP

= 200 / 225

= 0.88

Recall:

It is a ratio of correctly predicted positive observations to the total observation in the actual class predicted. We calculated the recall by using 200 TP images and 35 FN images.

Recall = TP / TP +FN

= 200 / 235

= 0.85

7.5 Testing resources and staffing

No external resources were hired for testing. All the work done by a group member is tested manually by another group member and vice versa for other group member.

7.6 Chapter Conclusion

In this chapter we have discussed the methods used to test our system. In most of the cases it performed well but in some cases it failed also. We choose YOLOv3 due to its high performance as compare to another algorithms like fast R-CNN, SSD and many more. In this testing no external resources were hired. All the testing was performed by our own teammates.

Chapter # 8
Future Enhancements and Recommendations

8. Future Enhancements and Recommendations

This MOBILE PHONE USER DETECTION USING COMPUTER VISION system can be applied where mobile phone usage is prohibited. For instance, in banks, it is prohibited to use mobile phones due to security reasons, in petrol pumps people are not allowed to use mobile phones for human safety purposes. The objective of our project is to detect that person which uses a mobile phone, save its image into the system so that further action could be taken against them.

As a scope for future enhancement,

- As our system's implementation will grow, we are planning to enhance it and will implement it to stop cheating in the examination hall using the mobile phone.
- We would also like to enhance our system so we could detect the usage of mobile phones while a person is driving a vehicle.
- We will enhance our system to detect mobile phone users among a large number of people at a conference.

Chapter # 9
Summary

9. Summary

We as a citizen of this society wanted to solve a problem that could also have a positive impact on our society. We have observed that there are a lot of people who use the mobile phone where its usage is prohibited such as petrol pumps because mobile phones have a high risk of fire ignition at petrol pumps. This system can reduce the usage of a mobile phone where mobile phone usage is prohibited. We can implement it in the examination hall, students using mobile phones in the examination hall for cheating purposes can be detected.

"Mobile Phone User Detection Using Computer Vision" is a real-time object detection system that takes live video as an input and captures images of the person using mobile in real-time as an output.

We have used the yolov3 algorithm for object detection in our project. We have choosen YOLOv3 algorithm due to its efficiency and accuracy as compare to other algorithms such as fast R-CNN, SDD and many others. All the computer vision task was performed using the computer vision library named OpenCV. All of the code is written in the python programming language. We have used visual studio code as IDE. We have used Google Colab for training our model and to label our dataset we have used Labelimg software.

We were not able to achieve the good fps speed because of the lack of hardware such as better GPU, if we improve this thing our system would be way more efficient.

REFERENCES

[1] Mamata S.Kalas, "REAL TIME FACE DETECTION AND TRACKING USING OPENCV", Proceedings of IRF International Conference, Pune India, 5th & 6th February 2014.

[2] V. Neethidevan, Dr. S.Anand, " A Real Time Object Detection System Using a Webcam with Yolo Algorithm", Annals of R.S.C.B, 05 May 2021.

[3] Chandan G, Ayush Jain, Harsh Jain, Mohana, "Real Time Object Detection and Tracking Using Deep Learning and OpenCV", 2018 International Conference of Inventive Research in Computing Application(ICIRCA), 2018, pp.1305-1308.

[4] Rakkshab Varadharajan Iyer, Priyansh Shashikant Ringe, Kevin Prabhulal Bhensdadiya, "Comaparison of yolov3, yolov5 and Mobile-Net SSD V2 for Real-Time Mask Detection", International Research Journal of Engineering and Technology (IRJET), 07 July 2021.

[5] R. Girshick, "Fast R-CNN", IEEE international conference on computer vision (ICCV), Santiago, USA, 2014

[6] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

[7] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers and A.W.M. Smeulders, "Selective Search for Object Recognition", International Journal of Computer Vision, 2013.

[8] Liu W. et al. (2016) "SSD: Single Shot MultiBox Detector". In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9905. Springer, Cham.

[9] Sang J, Wu Z, Guo P, Hu H, Xiang H, Zhang Q, Cai B. "An Improved YOLOv2 for Vehicle Detection". *Sensors*. 2018; 18(12):4272.

[10] J. Y. Lu, C. Ma, L. Li, X.Y. Xing, Y. Zhang, Z.G. Wang. J.W. Xu, "A Vehicle Detection Method for Aerial Image Based on YOLO", Journal of Computer and Communications, 2018.

[11] Zhao L, Li S. “Object Detection Algorithm Based on Improved YOLOv3”. *Electronics*. 2020; 9(3):537.

[12] N. Raviteja, M. Lavanya, S. Sangeetha, “An Overview on Object Detection and Recognition”, International Journal of Computer Sciences and Engineering, 2020.

[13] A. Kaur, D. Kaur, “Yolo Deep Learning Model Based Algorithm for Object Detection”, International Journal of Computer Sciences and Engineering, 2020.

APPENDIX

A. Project Schedule

This section presents an overview of project tasks.

a. Timeline chart / Gantt Chart

The complete schedule of “MOBILE USER OBJECT DETECTION” project is presented below.

	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Muhammad Uzair Khan	Muhammad Zaid Raza
Research	🟥								🟦	🟦
General Object Detection		🟥	🟥						🟦	🟦
Customizing Object Detection				🟥	🟥	🟥			🟦	🟦
Further Processing						🟥	🟥	🟥	🟦	🟦

Table A.1: Complete Gantt chat of project

b. Project Group Organization / Work Load Distribution

We are 2 group members who have completed the whole project. All the work was evenly distributed between both of us. Starting the project with research followed by data collection and its labelling and all other task were equally done by both of us.

Muhammad Zaid Raza

- research work
- data collection
- helped in data labeling
- model training
- mobile user detection

Muhammad Uzair Khan

- research work
- data collection
- data labeling
- helped in model training
- mobile user detection

GLOSSARY

Bounding boxes: Boxes which are drawn by the object detector for the reference is called bounding boxes.

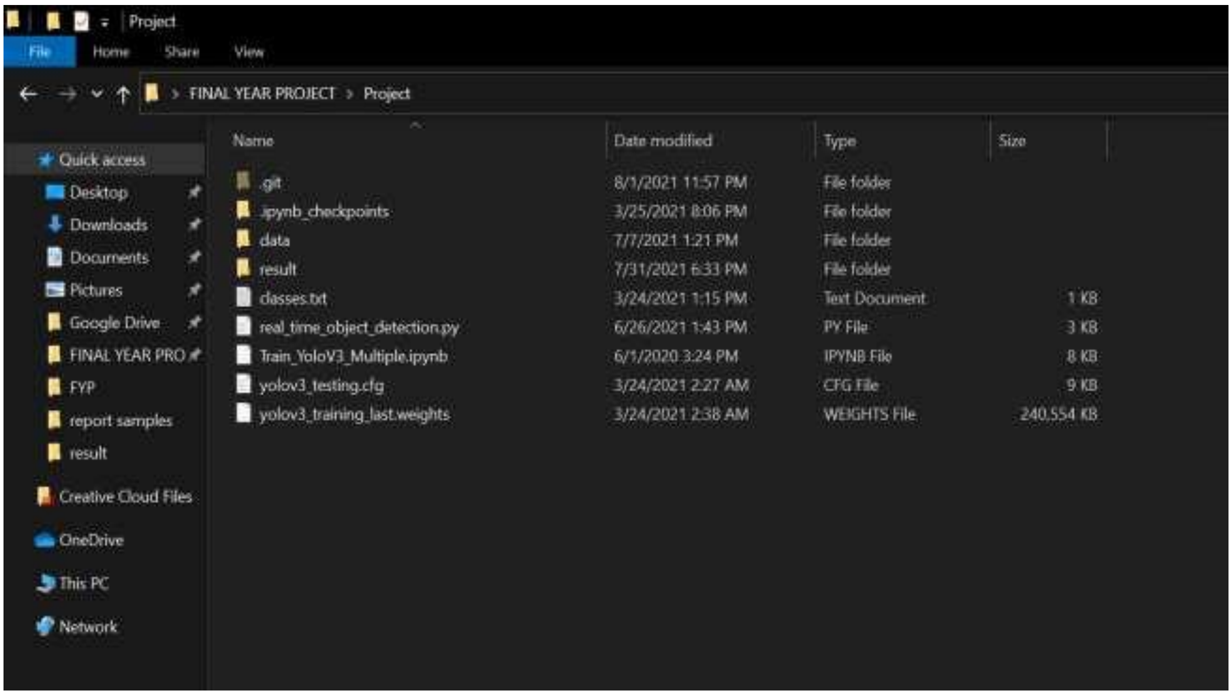
Computer vision: It is a field of AI that enable us to drive meaningful information from the digital images and videos.

Convolutional neural network: It is a class of artificial neural network which is used to analyze visual imagery.

Image-Processing: It is a method to perform operation on image.

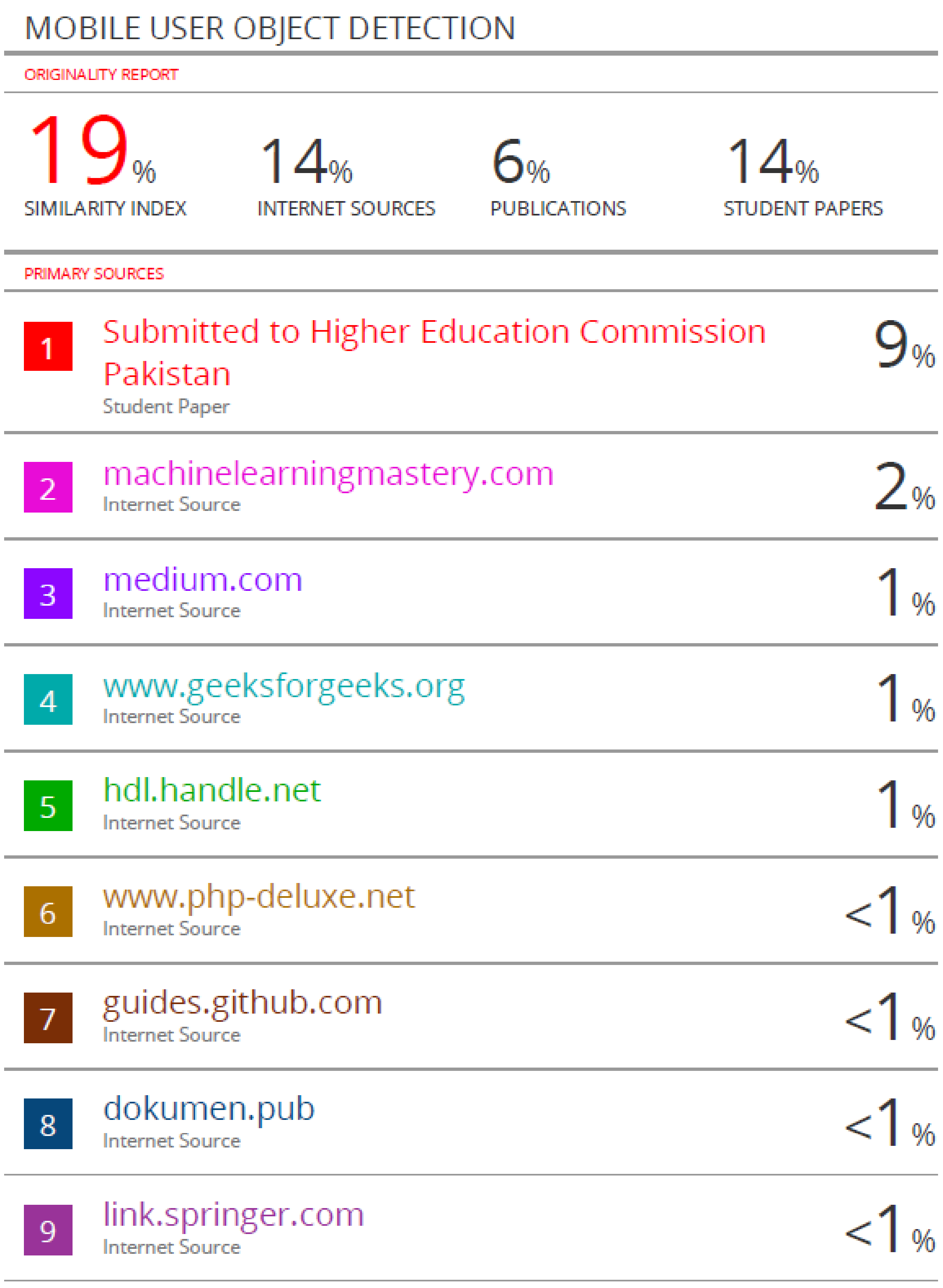
Object detection: It is computer vision technique using which we can locate and identify objects in image or video.

PROJECT DIRECTORY STRUCTURE



Name	Date modified	Type	Size
.git	8/1/2021 11:57 PM	File folder	
.ipynb_checkpoints	3/25/2021 8:06 PM	File folder	
data	7/7/2021 1:21 PM	File folder	
result	7/31/2021 6:33 PM	File folder	
classes.txt	3/24/2021 1:15 PM	Text Document	1 KB
real_time_object_detection.py	6/26/2021 1:43 PM	PY File	3 KB
Train_YoloV3_Multiple.ipynb	6/1/2020 3:24 PM	IPYNB File	8 KB
yolov3_testing.cfg	3/24/2021 2:27 AM	CFG File	9 KB
yolov3_training_last.weights	3/24/2021 2:38 AM	WEIGHTS File	240,554 KB

SIMILARITY REPORT



MOBILE USER OBJECT DETECTION

ORIGINALITY REPORT

19%	14%	6%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	9%
2	machinelearningmastery.com Internet Source	2%
3	medium.com Internet Source	1%
4	www.geeksforgeeks.org Internet Source	1%
5	hdl.handle.net Internet Source	1%
6	www.php-deluxe.net Internet Source	<1%
7	guides.github.com Internet Source	<1%
8	dokumen.pub Internet Source	<1%
9	link.springer.com Internet Source	<1%