Human Detection and Counting

UCS503 Software Engineering Project Report

End-Semester Evaluation

Submitted by:

(101803414) RHYTHM BANSAL

(101853024) NEETIKA

(101803160) HIMMAT CHAHAL

BE Third Year, COE19

Submitted to

Ms. Arwinder Dhillon

Computer Science and Engineering Department TIET,
Patiala

November 2020

TABLE OF CONTENTS

1. Project Overview

The steady increase in population and overcrowding has become an unavoidable factor in any public gathering or on the street. Intelligent monitoring technology is being developed in recent years, and human tracking has made a lot of progress. In this project, we are developing a system using Deep learning using the HOG Descriptor that detects and provides a count of humans in the region using OpenCV-Python. Human tracking is achieved by indicating the direction of movement of the person. The results of the project will be helpful in managing any area with a high density of crowd. We aim at building an intelligent system that detects and captures moving targets for accurate object classification. Most existing digital video surveillance systems rely on human observers for detecting specific activities in a real-time video scene. However, there are limitations in the human capability to monitor simultaneous events in surveillance displays. Hence, human motion analysis in automated video surveillance has become one of the most active and attractive research topics in the area of computer vision and pattern recognition. Human detection is a difficult task from a machine vision perspective as it is influenced by a wide range of possible appearance due to changing articulated pose, clothing, lighting and background, but prior knowledge on these limitations can improve the detection performance.

2. **Software Requirement Specification**

Table of Contents

1. Introduction

The steady increase in population and overcrowding has become an unavoidable factor in any public gathering or on the street. Intelligent monitoring technology is being developed in recent years, and human tracking has made a lot of progress. In this project, we are developing a system using Deep learning using the HOG Descriptor that detects and provides a count of humans in the region using OpenCV-Python. Human tracking is achieved by indicating the direction of movement of the person. The results of the project will be helpful in managing any area with a high density of crowd.

1.1 Purpose

Public safety has become a major problem in areas like malls, railway stations, streets during festive seasons, etc. Massive disasters happen worldwide including numerous instances of fatality where people gather in crowds. An efficient automated system to manage the crowd count is essential. Extension of human tracking provides a way to detect the position, to obtain the motion trail and to maintain identities of persons in the scene**.**

1.2 Document Conventions

The document has used bold lettered and underlined words to highlight key words. The document has tried to maintain a priority of requirements. The priority has been determined by the judgement of the author and may be subject to change. The document has used short forms for some commonly abbreviated terms.

1.3 Project Scope

- Counting Military Men in an area
- COVID19 social distancing
- Detect locate people who got misplaced due to natural disasters in remote areas.
- Commercial fields for navigation

1.4 Intended Audience and Reading Suggestions

- This project is mainly intended for government authorities maintaining safety while natural calamities.
- Moreover, search teams can use extensions of this project to locate missing people in remote areas.
- Military officials can especially use this technology to track their men and army of the opposite party in a particular area.
- COVID-19 officials, for checking the norms of social distancing.

1.5 References

- IEEE 830-1998 standard for writing SRS document

- Deep learning Specialisation course by Coursera
- https://www.youtube.com/watch?v=4ESLTAd3IOM YouTube tutorial
- https://www.hindawi.com/journals/cin/2018/1639561/
- https://ieeexplore.ieee.org/document/5399555

2. Overall Description

This section will be focusing on describing the main functionalities and dependencies of the human detection and counting system, as we know that detection and tracking of humans in real time is a challenging and important field of research.

2.1 Product Perspective

We aim at building an intelligent system that detects and captures moving targets for accurate object classification. Most existing digital video surveillance systems rely on human observers for detecting specific activities in a real-time video scene. However, there are limitations in the human capability to monitor simultaneous events in surveillance displays. Hence, human motion analysis in automated video surveillance has become one of the most active and attractive research topics in the area of computer vision and pattern recognition. Human detection is a difficult task from a machine vision perspective as it is influenced by a wide range of possible appearance due to changing articulated pose, clothing, lighting and background, but prior knowledge on these limitations can improve the detection performance.

2.2 Product Features

Motion detection is the ability to recognize the presence of movements. There are many different ways to detect motion. detection. Most conventional approaches for object detection are background subtraction, optical flow and spatial-temporal filtering method.

Using webcam: Basically, the idea is to take a picture from a webcam every period of time (make it the current picture) and compare it with a previous picture and if we find a useful difference between them, we will save both pictures else free the memory from the old picture and make the new picture the current picture.

Using stored file: Similar steps can be performed if the captured video is stored in a memory location.

2.3User Classes and Characteristics

Our goal is to make a human detection and counting system. This system will have a varied application for law enforcers and anywhere where there is a gathering of people.

The main users will be-

1.Law enforcers

2.Search teams

3.Wardens

4.Army personnel

5.Detectives

6.Companies with large number of employees

Some of the highlighted user cases are:

1. The most obvious application of detecting humans in surveillance video is to early detect an event that is not normal. Abnormal events are single-person loitering, multiple-person interactions (e.g. fighting and personal attacks), person-vehicle interactions (e.g. vehicle vandalism), and person-facility/location interactions (e.g. object left behind and trespassing).

2. A person in a visual surveillance system can be identified using face recognition and gait recognition techniques. The detection and tracking of multiple people in cluttered scenes at public places is difficult due to a partial or full occlusion problem for either a short or long period of time.

3. Gender classification is another application of human detection in surveillance cameras. The classification could be carried out by fusion of similarity measures from multi-view gait sequences, exploiting separability of features from different views.

4. Detecting and counting persons in a dense crowd is also something which can be done using this software.

5. In hostels, this model can be used to detect, identify and count students, thus making the job easier for wardens.

2.4 Operating Environment

The human detection and counting system is hugely dependent on the environment in which it is being operated. Firstly, the camera should be at a wide angle so that the entire area is under focus and motion can be detected. Also, the camera can mistake laser for human movement so either the camera should be hidden or at a particular angle where laser cannot reach. It is also recommended to use the system in a localized area based on the area, so that motions can be detected easily.

2.5 Design and Implementation Constraints

1)The user will have to access the service from any computer that has internet browsing capabilities and an internet connection.

2)The service should be used in such an environment so that the existing datasets actually identify human movements.

3)The biggest problem encountered is in the angle and the placement of the source of video capture. The camera should be placed in such a way that it has a wide view of the surroundings so that human movement can be detected easily.

4) If the service is connected with an input device, it will have an internal hard drive. Software and the database cannot exceed the capacity of its hard drive. The software must be able to read and write to the SD card slot, if available.

5) The speed of the process will be very limited because of the web service.

2.6 User Documentation

The user is going to upload the video on our web page. Which will be stored in a PHP file at the backend. The output will be shown after detecting the count from the video to the user through Email. In real time, a web cam will be used for input and live detection of humans will be done and displayed on the screen itself for the user output.

3. System Features

The product will be available through web-based service. The following are the features available in the system that will be working on the front-end and back-end:

3.1 Registering with the web service

3.1.1Description and Priority

The user will register with the website through his/her email id. This step is of high priority because the web- service is available for multiple users.

3.1.2 Response Sequences

The provided information will be stored on the back-end and the resultant video with counting statistics will be sent on the user's email id for discretion.

3.1.3 Functional Requirements

The input from the user is taken and its storage is done by using PHP in Web Development.

3.2 Input video

3.2.1 Description and Priority

In this step, the input video has to be uploaded by the user for detecting and counting humans in it. This is a high priority step because all the operations will be performed on this input.

3.2.2 Response Sequence

The individual frames from the uploaded video will be extracted for detection.

3.2.3 Functional Requirements

The video is read using the imread() function of the OpenCV module. Each frame is detected using the while loop. A DetectByPath() function will be called to read each frame successfully till the last one.

3.3 Detection and Counting

3.3.1 Description and Priority
This is the main function where the detection and counting of humans is taking place. This is a high priority function.

3.3.2 Response Sequence
Rectangular boxes will appear on every passing human body to indicate detection. The count at a particular time will be displayed continuously.

3.3.3 Functional Requirements
Each frame will be processed using the function HOGCV. detectMultiScale() of the HOG descriptor. The HOG algorithm is run through the SVM detector. The count is displayed through the puttext() function of the OpenCV module.

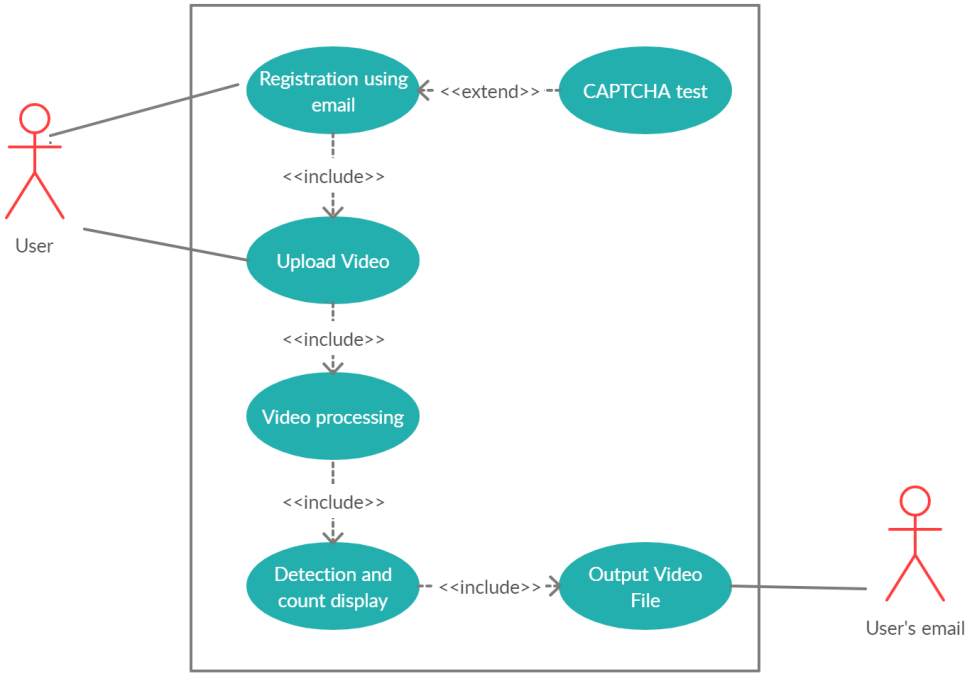
3.4 Output Video

3.4.1 Description and Priority
Each output frame is combined to make the resultant video with marked detection and count display. This is a high priority process.

3.4.2 Response Sequence
The output is sent to the user's e-mail.

3.4.3 Functional Requirements
The frames are combined and displayed using the imshow() function of the OpenCV module. The output video is then sent to the user using the mail() function from PHP.

Use Case Diagram:



4. External Interface Requirement

4.1 User Interface

Human Detection and counting system will take images form the webcam or the storage file in our computer. A video combines a sequence of images to form a moving picture. We call these images as Frame. So, in general we will detect the person in the frame. And show it one after another that it looks like a video. It will take a frame to detect a person in it. Make a box around a person and show the frame, and return the frame with person bounded by a green box, system will count that as an one human and store it to the memory.

4.2 Hardware Requirements

4.2.1 Webcam: A webcam is a camera that connects to a computer. It captures either still pictures or motion video, and with the aid of software, can transmit its video on the Internet in real-time

4.3 Software Requirements

4.3.1 OpenCV: OpenCV is open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects.

4.3.2 Imutils: A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV.

4.3.3 NumPy: NumPy is a python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

4.3.4 Argparse: The argparse module makes it easy to write user-friendly command-line interfaces. The program defines what arguments it requires, and argparse will figure out how to parse those out of sys.argv. The argparse module also automatically generates help and usage messages and issues errors when users give the program invalid arguments.

4.3.5 HOG: The HOG is used in computer vision and image processing applications as a

feature descriptor for the purpose of detecting the objects. This technique counts the occurrences of gradient orientation in localized portions of an image. This method of HOG is similar to that of edge orientation histograms, shape contexts, and scale-invariant feature transform descriptors, but differs in that as it computes over a dense grid of uniformly spaced cells. For improving the accuracy, it uses over- lapping local contrast normalization.
HOG descriptor algorithm implementation is as follows:
• **Gradient computation**: Divide the image into cells which are small connected regions, and compute histogram of gradients or orientation of edges for each cell.
• **Orientation binning**: Each cell is discretized into angular bins according to orientation of cells. Pixel of each cell contributes a weighted gradient to its corresponding angular bin.
• **Descriptor blocks**: Adjacent cells are grouped into blocks. Further normalization process is carried out based on the blocks formed by arrangement of cells.
• **Block normalization**: The block histogram is represented with the help of group of normalized histograms. The descriptor is represented by the normalized group of histograms.

4.3.6 Web page design: We are going to use HTML, CSS for designing the frontend of the webpage and backend will be handled by JavaScript. For inputting all the user data in the webpage, we are going to use PHP and later we are going use that for getting the human detection and count from the video.

4.4 Communication Requirements

Through the GUI interface, users will upload their video to our webpage and store in the php file. The PHP file will be linked to the machine learning system. The communication standard we are going to use in this project are HTTP and FTP.

5. Other Non-functional Requirement

5.1 Performance Requirements

- The actual use of the software can only be exploited if the service provides immediate results. It will only be possible through proper hardware like webcams with suitable machineries.
- This project uses web service to illustrate the meeting of the same requirements but without the actual equipment.
- The user details are stored locally for peak performance.

5.2 Software Quality Attributes

- **Accessibility:** The web service should be accessible to multiple users at the same time.
- **Flexibility:** The length of the video file that can be uploaded should have high range.
- **Correctness:** All suitable extensions like mp4, avi etc. should be taken as input and the correct result should be provided.

5.3 Security Requirements

- The resultant should only be accessed by the particular user. Hence, email support.
- To check and confirm that the user is human, captcha should be used.
- All the data to be sent via HTTP protocol for better encryption.
- All the data to be sent via HTTP protocol for better encryption

6. Other requirements

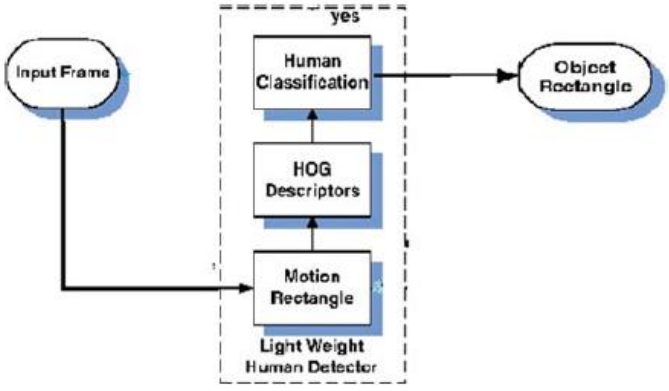
- The web service will be created using Web Development (CCS with HTML).
- Moving objects shall also be detected.

Appendix A: Glossary

HOG	Histogram of Oriented Gradient
SVM	Support Vector Machine
Open CV	Open Source Computer Vision
HTTP	HyperText Transfer Protocol
HTML	HyperText Markup Language
FTP	File Transfer Protocol
CSS	Cascading Style Sheet
Numpy	Numerical Python

Appendix B: Analysis Models

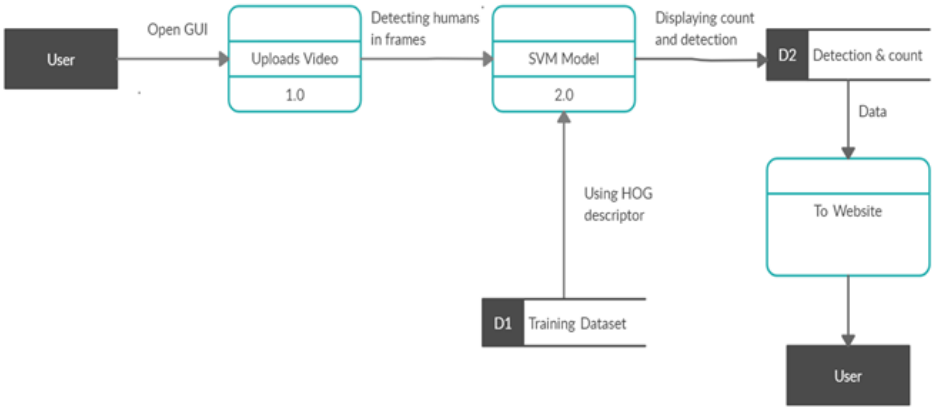
Methodology using HOG descriptor:



Data flow Diagrams:

Context Level DFD:

Level 1 DFD:



Appendix C: Issues list

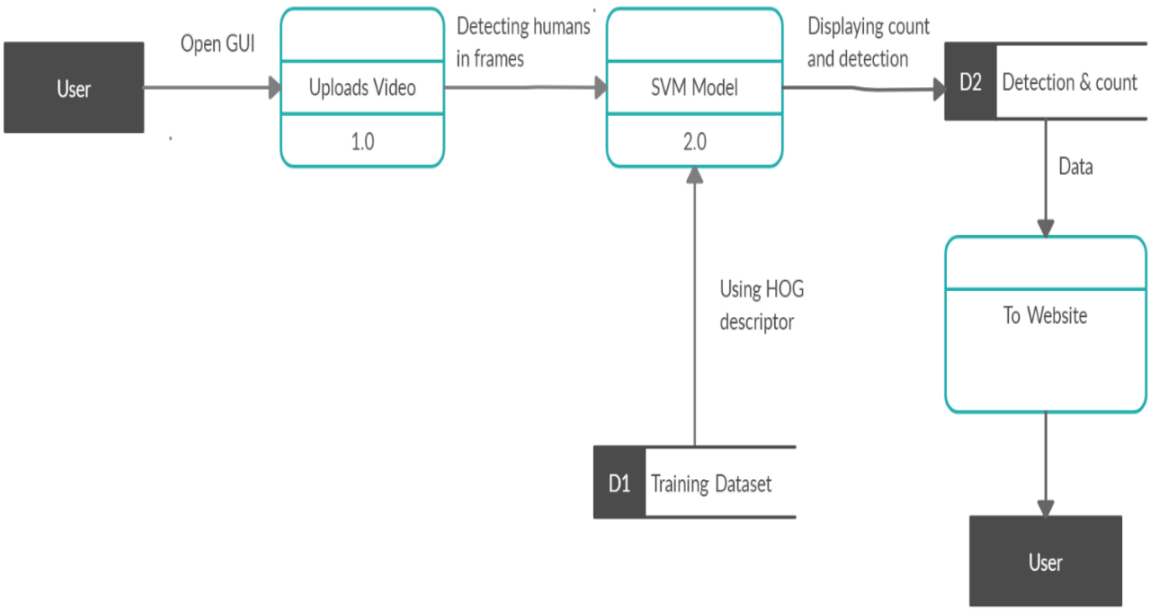
- The service should ideally be provided using live webcams.
- For locating humans stuck in places due to natural calamities, quadcopters should be used as an extension of the project.
- Raspberry Pi Board has to be used for image processing using external equipment.

3. Structured Analysis

3.1 Data Flow Diagrams

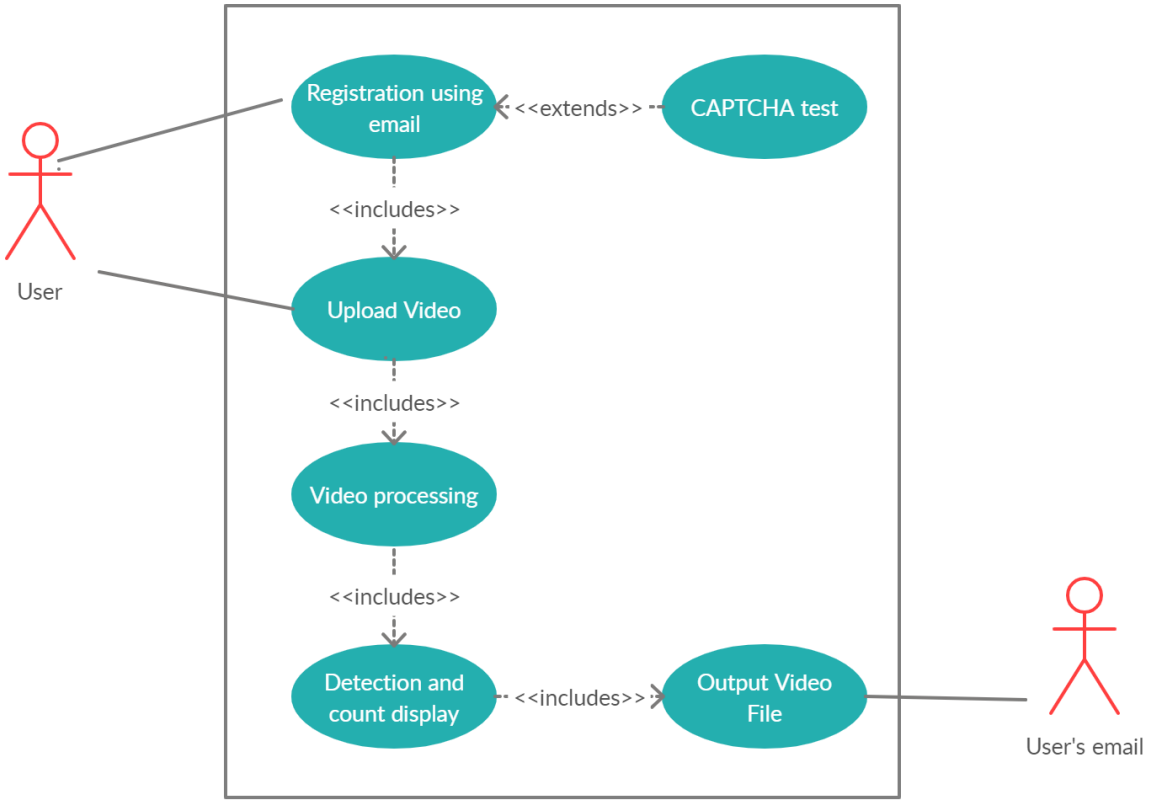
3.1.1 Context level DFD (Level 0)

3.1.2 Level 1 DFD



4. Object Oriented Analysis

4.1 Use Case Diagram



4.2 Use Case Templates

1. Use Case Title	Human Detection and Counting
2. Use Case Id	1
3. Actors	User

4. Description

The web service is developed so that a video file is fed and it returns an output file with humans detected and counted.

4.1 Pre-Conditions:

User has to upload a video or an image file.

4.2 Task Sequence:

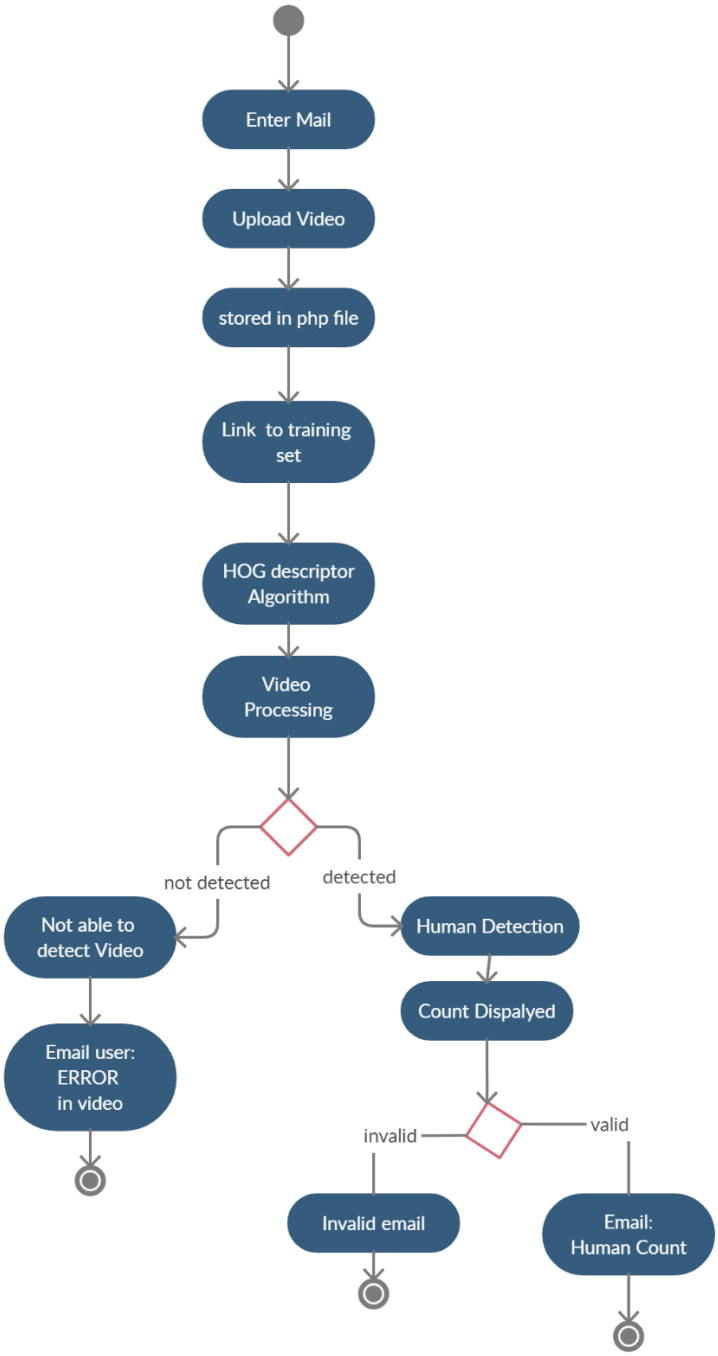
1. A website is shown by the system.
2. User enters email and uploads video.
3. User clicks on submit button.
4. The system will display the output video file with resultant human detection and counting.

4.3 Post Conditions:

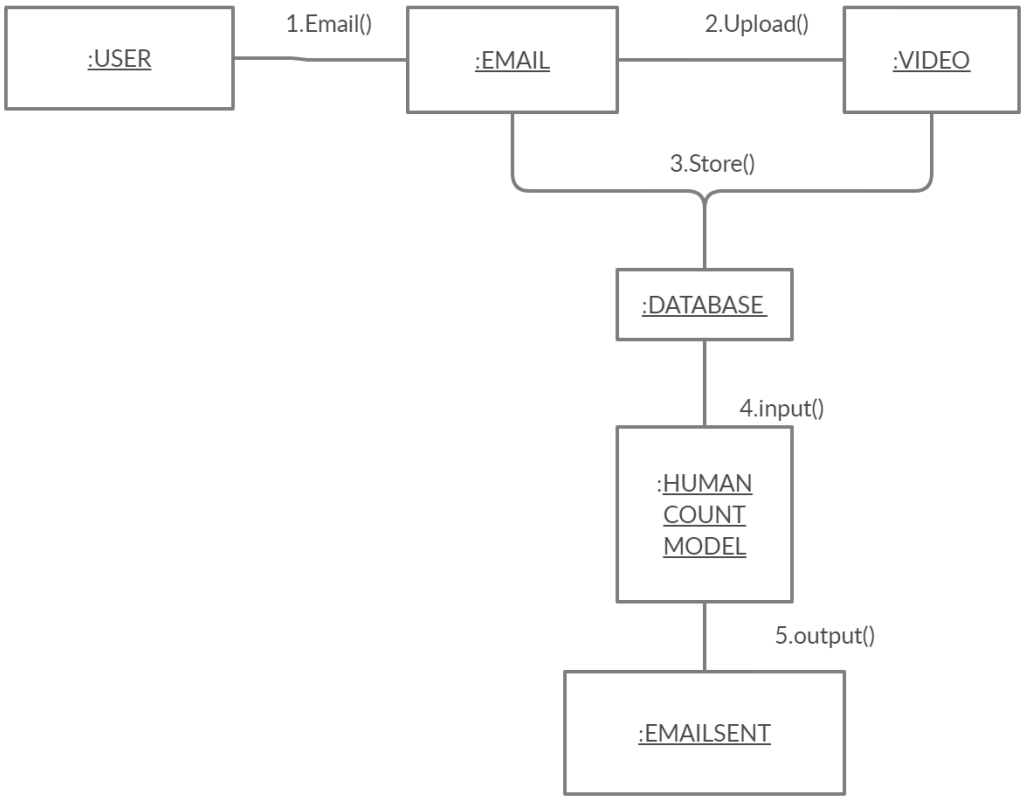
1. User can view count at any particular instant of the video.
2. User can upload another image.

5. Authors: Rhythm Bansal, Neetika, Himmat Chahal

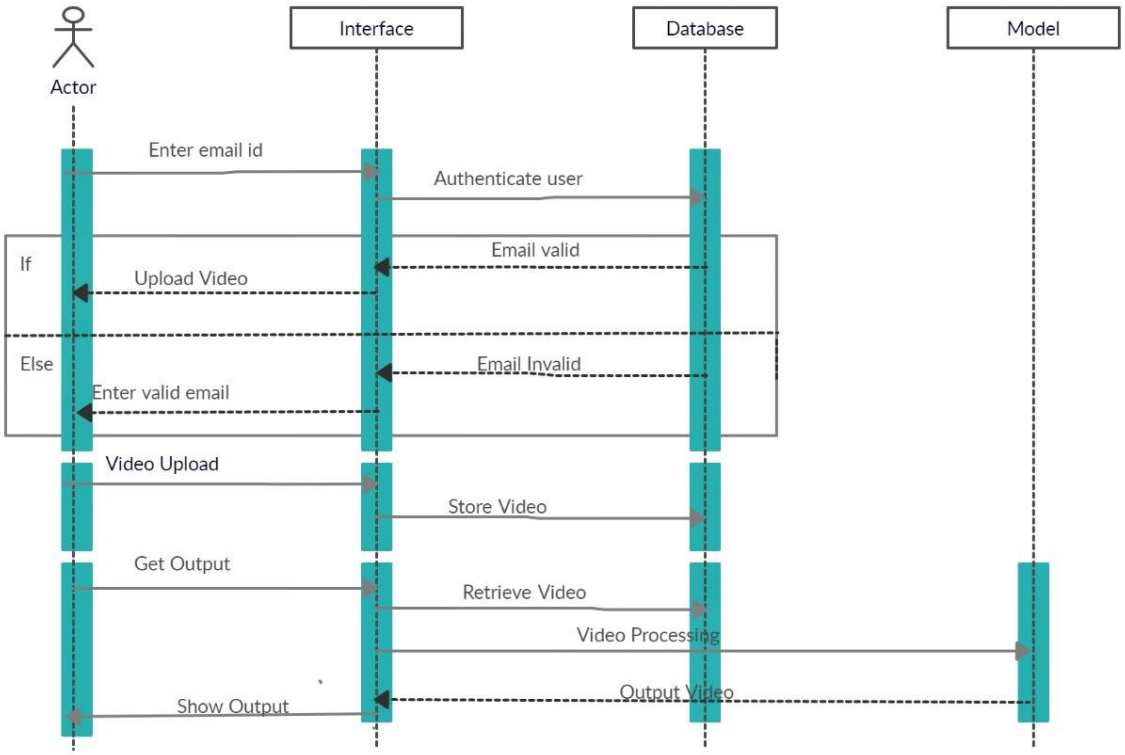
4.3 Activity Diagram



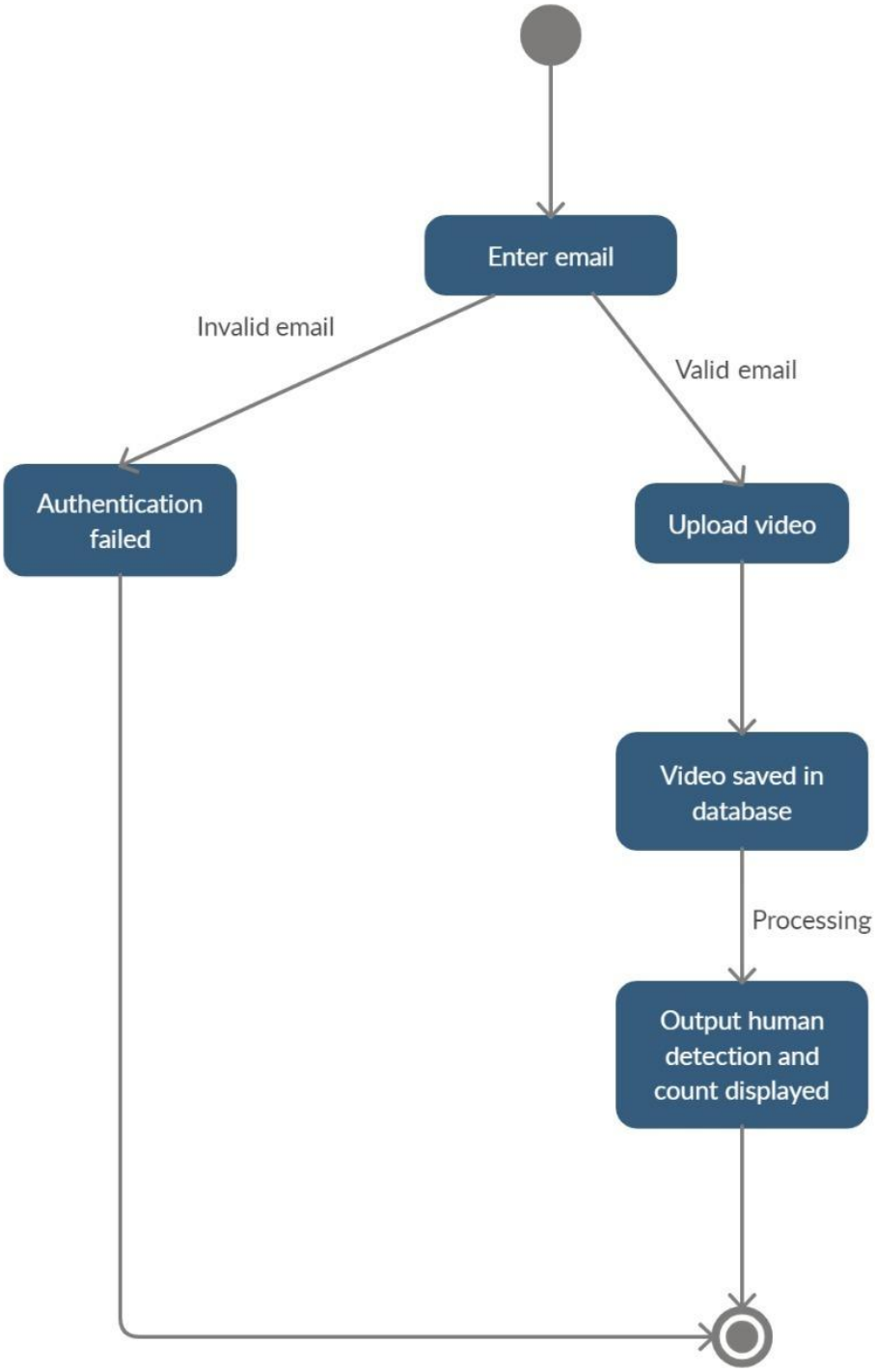
4.4 Collaboration Diagram



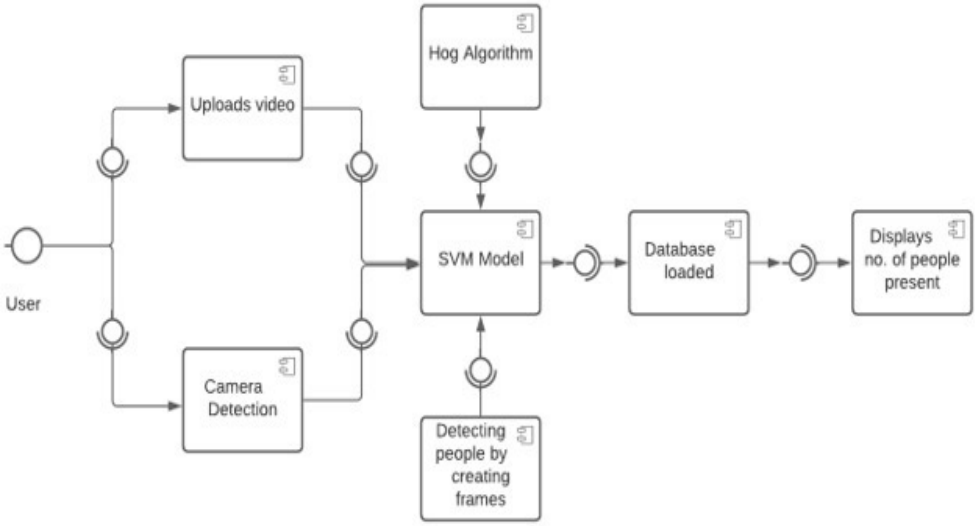
4.5 Sequence Diagram



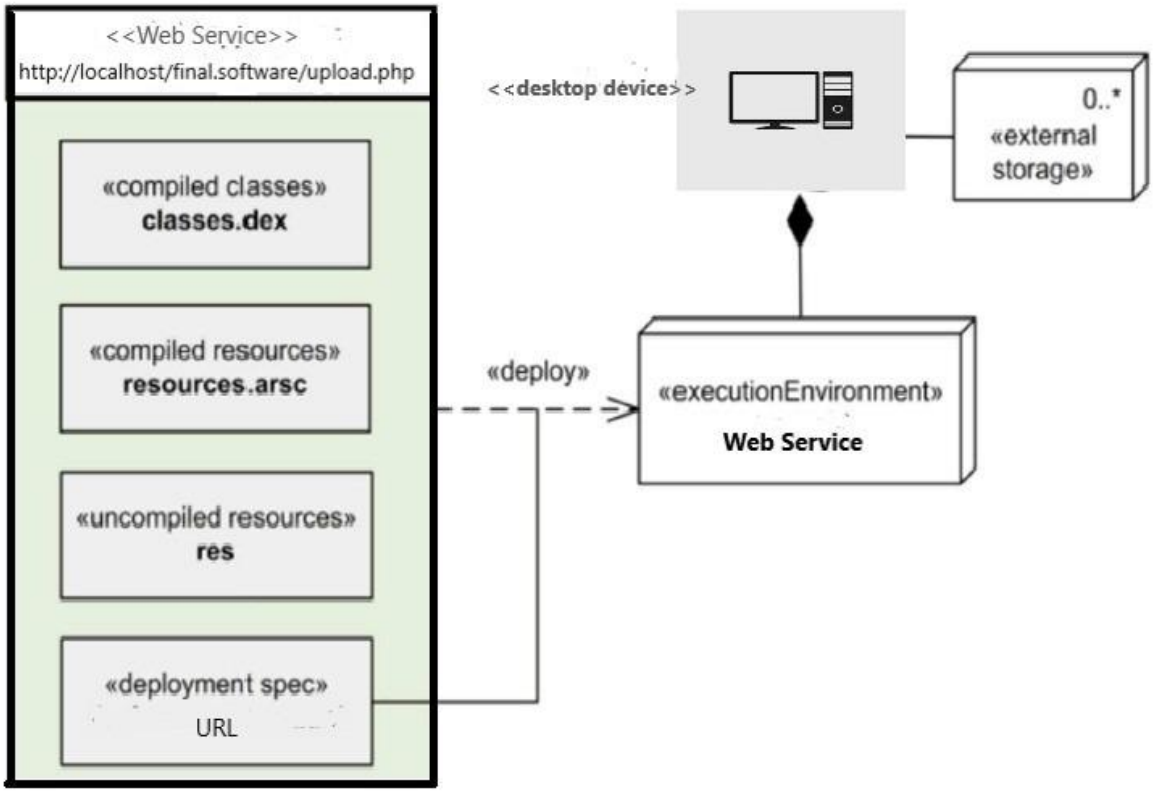
4.6 State Chart Diagram



4.7 Component Diagram



4.8 Deployment Diagram



5. Testing

5.1 Test Plan

Test cases were run on the news application on 2 different laptops. If the website did not crash and it produced the desired result without the freezing, the test case is considered to be passed. The testing involved the documentation of average time taken by the application to respond to an activity i.e. taking care of the time lag between query and response. It was made sure that the databases are reliable and robust against all cases. Security issues are also handled like the personal information of the user.

Scope:

To test the listed functional requirements of the product along with speed, robustness and accuracy of the product.

Approach: Manual testing of application on two different Windows/Linux/Mac devices for any bugs or non compliance with the requirements.

Features to be tested (various functionalities):

- Email authentication
- File upload
- Output video

Item pass fail criteria: If the testing produces the desired result for every activity without any freezing or taking very long time to execute, it is considered pass, otherwise fail.

5.2 Test Case Report

Test Case #: 1	**Test Case Name:Display Portal**
System: Human detection and counting	**Test**
Designed by: Thapar University	**Subsystem: Check user interface**
Executed by: Rhythm Bansal	**Design Date: 2 Dec 2020**
Short Description: Test the display portal	**Execution Date: 2 Dec 2020**

Pre-conditions
The user has a valid email id.
The system displays the user interface initially.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Open the Web Service	The system displays a message asking the user to enter the email.	Pass	Displayed as expected
2	Enter email	The system displays a message asking for a valid email id.	Pass	Displayed as expected
3	Enter video	The system displays a message to browse the video.	Pass	Prompt to browse video appears.
4	Upload video	The video is successfully uploaded.	Pass	Video is fed to the server.
5	Check post-condition 1			

Post-conditions
1. You have successfully uploaded the video.

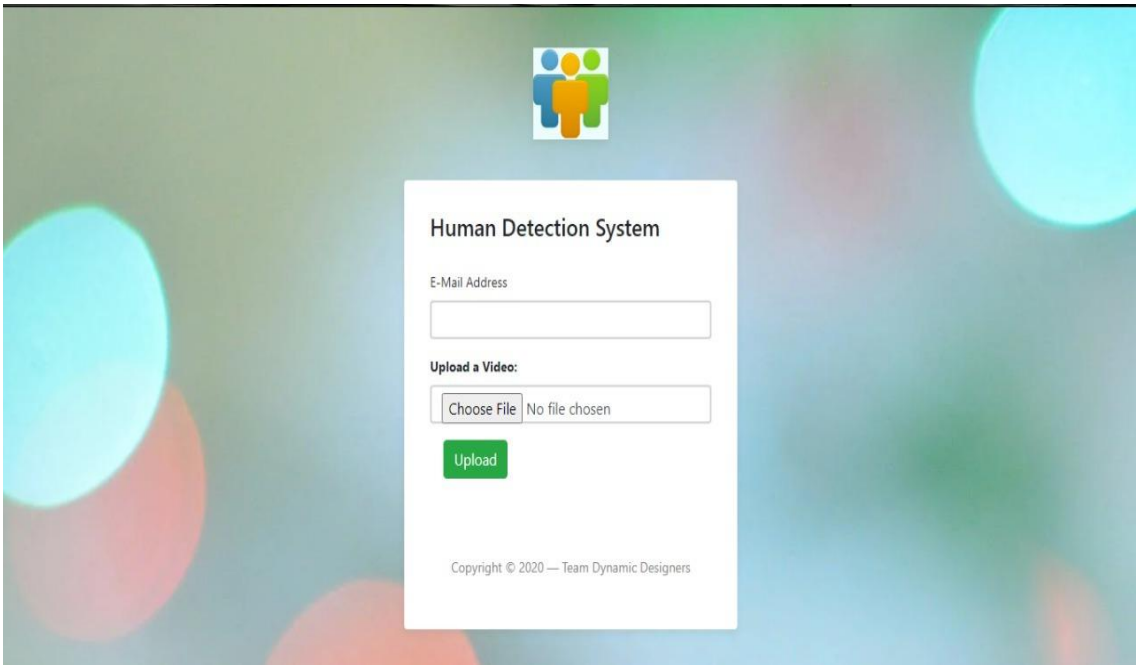
Test Case #: 2	**Test Case Name: Video Output**
System: Human detection and counting	**Subsystem: Check user interface**
Designed by: Thapar University	**Design Date: 2 Dec 2020**
Executed by: Rhythm Bansal	**Execution Date: 2 Dec 2020**
Short Description: Test the display portal	

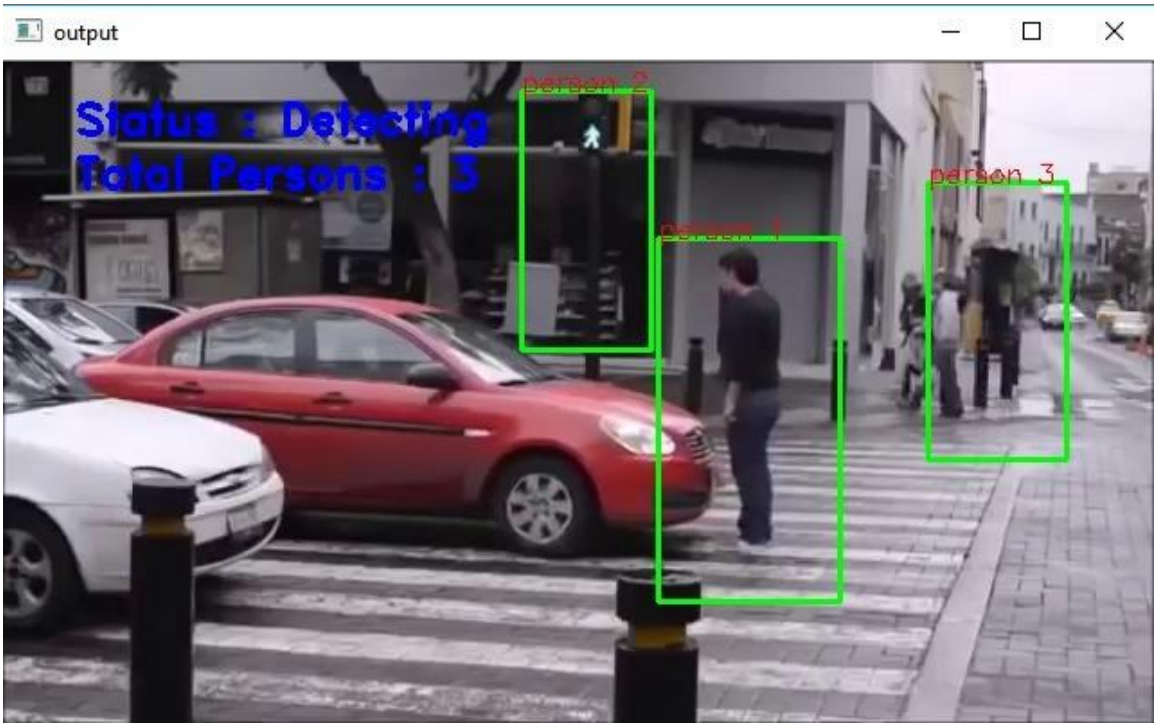
Pre-conditions
The user has successfully uploaded the video.
The web service is still running.

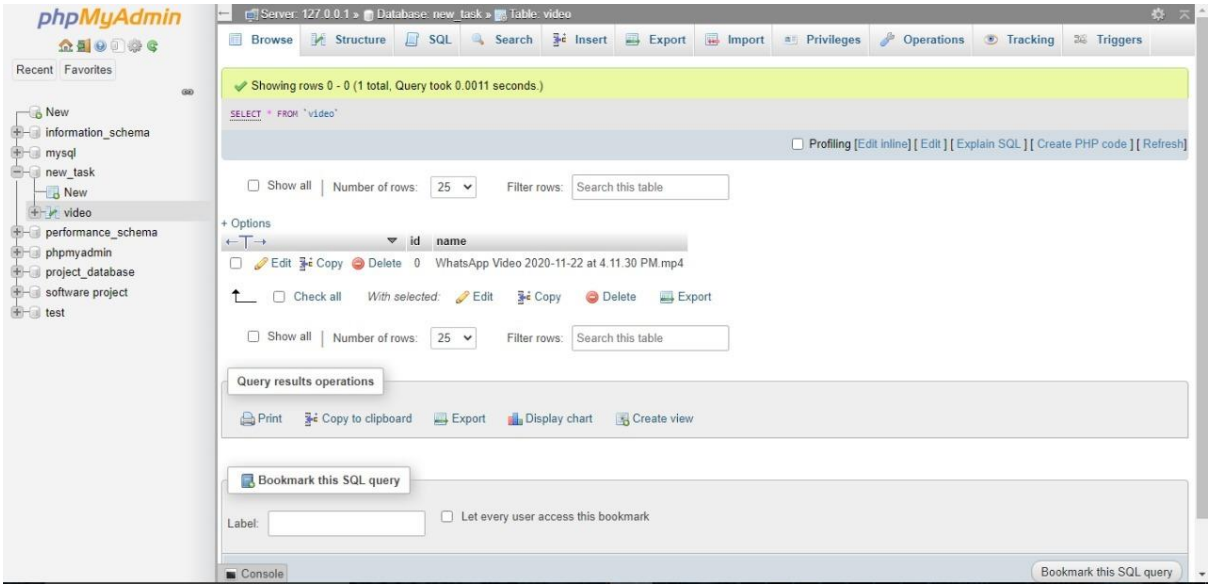
Step	Action	Expected System Response	Pass/ Fail	Comment
1	Display output video file	The output video file is displayed.	Pass	NA
2	Human count displayed	The video displays ongoing human count in the video	Pass	NA

Post-conditions
1. Option for uploading more videos.

5.3 Screenshots






phpMyAdmin
Recent
Favorites
New
information_segment
mysql
new_task
New
video
performance_schema
phpmyadmin
project_database
software project
test
Server: 127.0.0.1 » Database: new_task » Table: video
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Showing rows 0 - 0 (1 total, Query took 0.0011 seconds.)
SELECT * FROM `video`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows:
25
Filter rows:
Search this table
+ Options
id
name
Edit
Copy
Delete
0
WhatsApp Video 2020-11-22 at 4.11.30 PM.mp4
Check all
With selected:
Edit
Copy
Delete
Export
Show all
Number of rows:
25
Filter rows:
Search this table
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Console
Bookmark this SQL query