ACKNOWLEDGEMENT

To list who all have helped us is difficult because they are so numerous and the depth is so enormous. However, we would like to acknowledge the following as being idealistic channels and fresh dimensions in the completion of this project. We would like to extend our sincere thanks to all of them. We are highly indebted to Mrs. Kinjal B. Mistree for her guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing the project. We would like to express our gratitude towards the Head of Department Dr. Devendra Thakor, the Director of Institute Dr. R. V. Patil, other faculties of the department, our families & friends for their kind co-operation and encouragement which helped us in completion of this project. We would like to express our special gratitude and thanks to industrial persons for giving us such attention and time. Our thanks and appreciations also go to our college in developing the project and people who have willingly helped me out with their abilities.

Rahul Sharma (201703100910060)
Priyam Shah (201703100910028)
Guruprasad Panda (201703100910022)
Neel Shah (201703100910099)

ABSTRACT

The physical movement of the human hand produces gestures, and hand gesture recognition leads to the advancement in automated system. In this project, the human hand gestures are detected and recognized using convolutional neural networks (CNN) classification approach. This process flow consists of hand region of interest segmentation using mask image, ROI Extraction, Colour Conversion, Contour Detection and gesture recognition using CNN classifier. The hand region of the image is segmented from the whole image using mask images. The adaptive histogram equalization method is used as enhancement method for improving the contrast of each pixel in an image. In this project, after extracting the Region Of Interest, The Colour Conversion and Contour Detection is done and then the ouput gesture from hand image are given to the CNN classification algorithm which classifies the image into various classes. The proposed hand gesture detection and recognition methodology using CNN classification approach with enhancement technique stated in this project achieves high performance with state-of-the-art methods.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1 Introduction

Conceptulization phase(Introduction phase) , starting with the seed of an idea, it covers identification of the product/service, Pre-feasibility, Feasibility studies and Appraisal and Approval. The project idea is conceptualized with initial considerations of all possible alternatives for achieving the project objectives. As the idea becomes established, a proposal is developed setting out method, estimated costs, benefits and other details for appraisal of the project managers. After reaching a broad consensus on the proposal the feasibility dimensions are analyzed in detail.

1.1 Background

This report aims to give a brief description about the hand gesture recognition system from frame sequence using deep learning. With the help of this report, the needs of the user and the solution that will be provided to that needy will be clearly presented. The hand gesture recognition from frame sequence using deep learning is a deep learning based solution for the humans to interact with machine naturally using hand gestures and avoiding the use of mechanical devices.

1.2 Problem Definition

At present, human–machine interaction (HMI) is very important for operating the machines in a remote manner by the commands which are received from humans. In this regard, gestures provide a way for computers to understand human body language. This system deals with a goal of interpreting human gestures via mathematical algorithms and enabling human to interact with machine (HMI) and interact naturally without any mechanical devices.

1.3 Motivation

The main motivation behind this problem definition selection is to provide a more efficient way of controlling devices that are not possible by conventional ways, to track hand movements and gestures through API (Application Programming Interface). This system also resolves the interaction problem with mouse.

- Have to locate Cursor
- Hard for some to control
- Limited forms of input from mouse

1.4 Objective

The main objective of this project is to evaluate different object detection techniques and filtering algorithms to perform hand gesture detection and using these algorithms, develop an algorithm for recognition of hand gestures with reasonable accuracy which further would help us to develop a software solution that will take detection results and convert them to interactive features.

1.5 Scope

Gesture recognition is an important topic in computer vision because of its wide range of applications, such as HCI, sign language interpretation, controlling devices and visual surveillance. This system can be used in many other areas like gaming consoles and medical machines, etc. This system would be an aid to Physically-Challenged people.

1.6 Applications

This System can have wide variety of applications in various fields like Image Controlling and Scaling, Controlling Mouse Inputs, Sign Language Recognition, Gaming Interface Controlling, Robot Control, Controlling Machines, System Control.

Chapter 2 System Planning

In this phase, the project structure is planned based on project appraisal and approvals. Detailed plans for activity, finance, and resources are developed and integrated to the quality parameters. In the process major tasks need to be performed in this phase are

• Identification of activities and their sequencing

• Time frame for execution

• Estimation and budgeting

2.1 Project Development Approach

Each project need to be developed with software model which makes the project with high quality, reliable and cost effective.

- Iterative Waterfall model.
- This application follows Iterative waterfall model. In this firstly simple implementation is done with small set of software requirements and slowly the complete system is implemented. At each iteration, the design modifications can be made and new functional capabilities can be added.
- We have selected this model because Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).
- Advantages of your software model:

1. Some working functionality can be developed quickly and early in the life cycle.
2. Results are obtained early and periodically.
3. Parallel development can be planned.
4. Progress can be measured.
5. Less costly to change the scope/requirements.
6. Testing and debugging during smaller iteration is easy.
7. Risks are identified and resolved during iteration; and each iteration is

an easily manageable milestone.

8. Easier to manage risk - High risk part is done first.
9. With every increment, operational product is delivered.
10. Issues, challenges & risks identified from each increment can be utilized/applied to the next increment.
11. Risk analysis is better.
12. It supports changing requirements.
13. Initial Operating time is less.
14. Better suited for large and mission-critical projects.
15. During life cycle, software is produced early which facilitates customer evaluation and feedback.

2.2 System Modules

2.2.1: Preprocessing Gesture Inputs

This module plays an important role in our system as it does the task of preprocessing the gesture inputs taken from the live video feed and convert it into a format which is easy for the model to process and perform mathematical calculations over the input .

2.2.2 : ROI/Frame extraction, Colour conversion, Contour detection.

This module is the main module of our system that does the task of frame extraction or finding the Region of Interest (ROI). After the ROI is extracted, the system converts the RGB Image to a HSV Image for better accurate recognition and then the contour formation around the hand image is done to detect the gesture.

2.2.3 : Training & Testing our Model on Static Images

This module is used to train a deep-learning based model with the images that we have in our dataset and then test the model with the input image that is processed by the system.

2.2.4 : Final Implementation of Recognition Modules using Live Feed.

Finally, all these above modules are combined into a single System via using the Computer Vision methods. The final system would then be accessible via a custom Tkinter Based GUI.

2.3 Functional Requirements

ID	Title & Description
FR1	***Title:*** *Hand Detection* ***Description:*** *This software shall utilize a Hand Detection system to filter out hand from the video capturing device. By applying hand detection, the system can ignore the region where the hand is located and thus reducing the amount of calculation needed to perform hand gesture detection. The hand detection unit will be implemented with the help of OpenCV.*
FR2	***Title:*** *Skin Detection Module* ***Description:*** *This software shall perform skin colour detection and filter out all objects that do not contain the colour of skin. By filtering object of non-skin coloured, the system can then use its remaining resources and focus on hand gesture detection and recognitions. This also allows the system to pinpoint possible locations of user's hands. The skin detection module can be achieved by using the RGB to HSV Color Conversion Module.*
FR3	***Title:*** *Filtered Object Detection* ***Description:*** *Once the program has filtered out most of the unwanted parts of the picture after using the skin detection module, the software shall read and recognize "clusters" skin coloured objects also known as "blobs".*
FR4	***Title:*** *Object Location* ***Description:*** *Upon detection, the system shall be able to compute the location of the object using simple trigonometry math.*
FR5	***Title:*** *Hand Calibration* ***Description:*** *Depending on user's preferences, the system shall perform adjustments according to user's dominant hand. This means that if the user is right*

	handed, the mouse control gesture mode should be recognized near the right side of the face instead of the whole field of view. This is achieved through trigonometry math conversions.
FR6	***Title:*** *Browsing Gesture Control Mode* ***Description:*** *The software shall allow the user to use the "Gesture Mode". In this mode, user's hand gesture will only be recognized for commands.*

2.4 Non Functional Requirements

- ***Portability:***
  The HGR software shall be portable to almost all operating platforms that support Python. Therefore, this software would not depend on the different operating systems.

- ***Security:***
  The HGR software shall comprise of a simple invoking command to prevent any unauthorized access.

- ***Performance:***
  This software shall minimize the number of calculations needed to perform image processing and hand gesture detection. Each captured video frame shall be processed within 350 milliseconds to achieve 3 frames per second performance.
- ***Reliability:***
  The HGR software shall be operable in all lighting conditions. Regardless of the brightness level in user's operating environment, the program shall always detect user's hands.

- ***Reusability:***
  The HGR software should be versatile in nature so that the block of code can be reused with other extensions and platform for further scope of development.

- ***Serviceability:***
  The HGR provides better service than previous input devices as it is a more cost efficient and easy to use solution to control devices and perform tasks.

2.5 Hardware and Software Requirements

i. Hardware Requirement

HGR software does not require special equipment except for a personal computer (PC) and a webcam.

- *Webcam - 320 x 260*
- *RAM - 512Mb or above*
- *Storage - at least 256 Mb free*
- *Processor - 400Mhz or above*

ii. Software Requirement

- *Windows 7 or Above.*

2.6 Timeline Chart

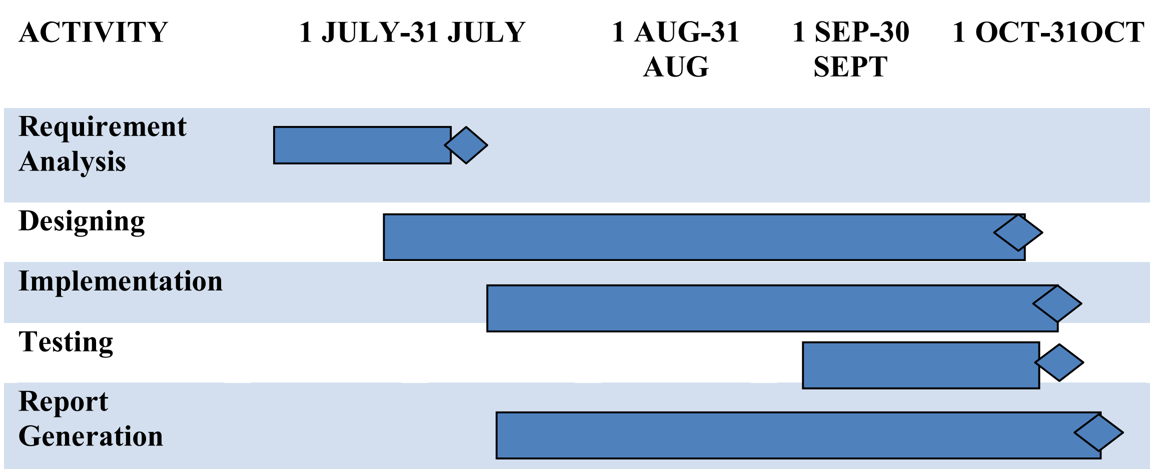


Figure 2.1:Timeline Chart

Chapter 3 System Design

This phase of the project witnesses the concentrated activity where the plans are put into operation. Each activity is monitored, controlled and coordinated to achieve project objectives. Important activities in this phase are

• Communicating with Guide/Project Managers

• Implementation of the project and Reviewing the progress

• Monitoring cost and time

• Controlling quality

• Managing changes

3.1 Database Schema

We are using Our Own Dataset consisting of approximately 10000+ images out of which 7500+ images are for training purpose and 2500+ images are for testing and validation purpose for the model. The live video feed is used for controlling the system via gestures.

The images consists of different types of numeric gestures of 0, 1, 2, 3, 4, 5 in different conditions and different manner.

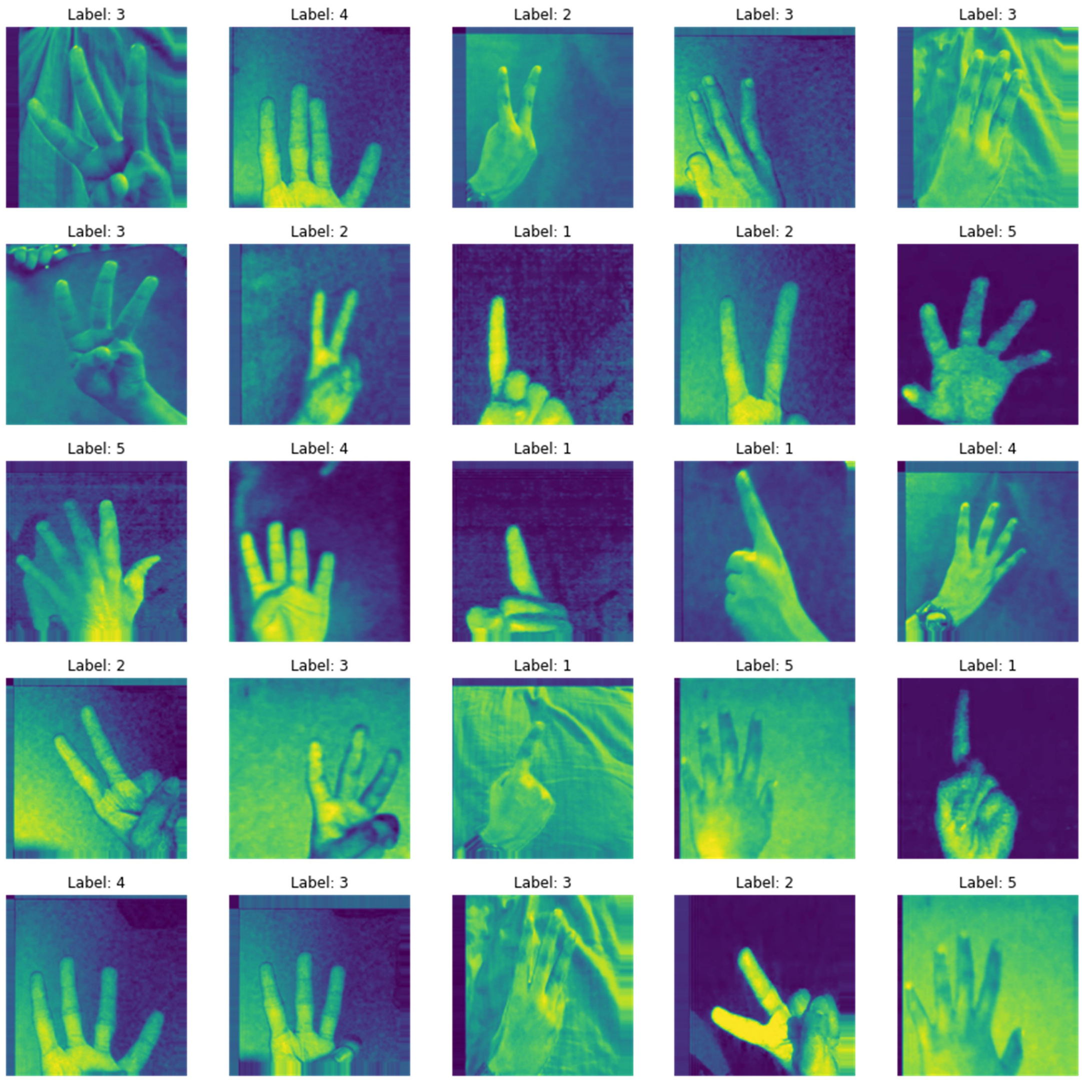


Figure 3.1: Dataset Overview

3.2 Use Case Diagram

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

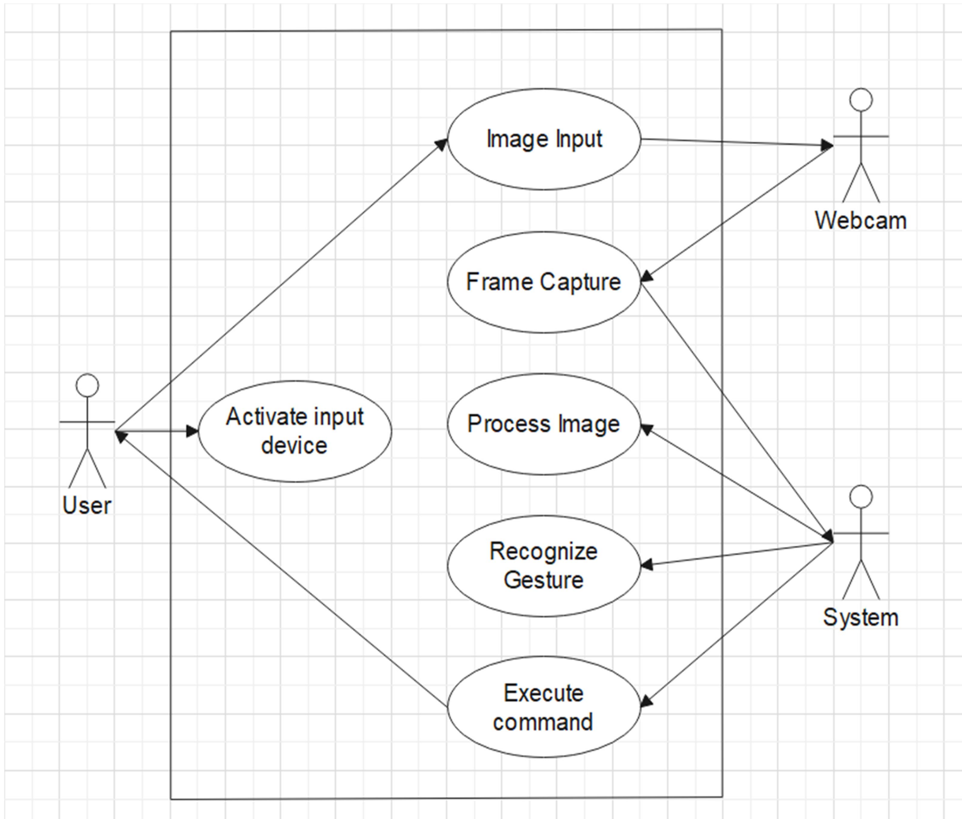


Figure 3.2:Use Case Diagram

3.3 Sequence Diagram

Describe the flow of messages, events, and actions between objects. Show concurrent processes and activations. Show time sequences that are not easily depicted in other diagrams. Typically used during analysis and design to document and understand the logical flow of your system.

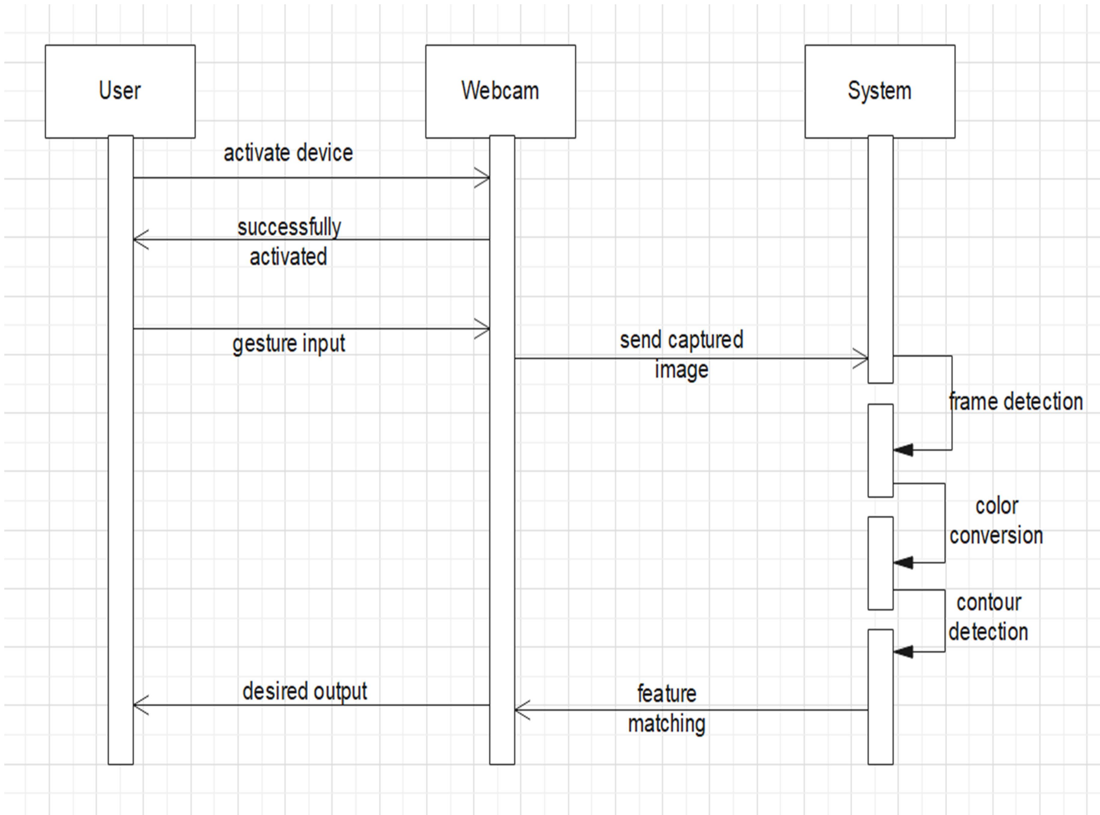


Figure 3.3: Sequence Diagram

3.4 Activity Diagram

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part.

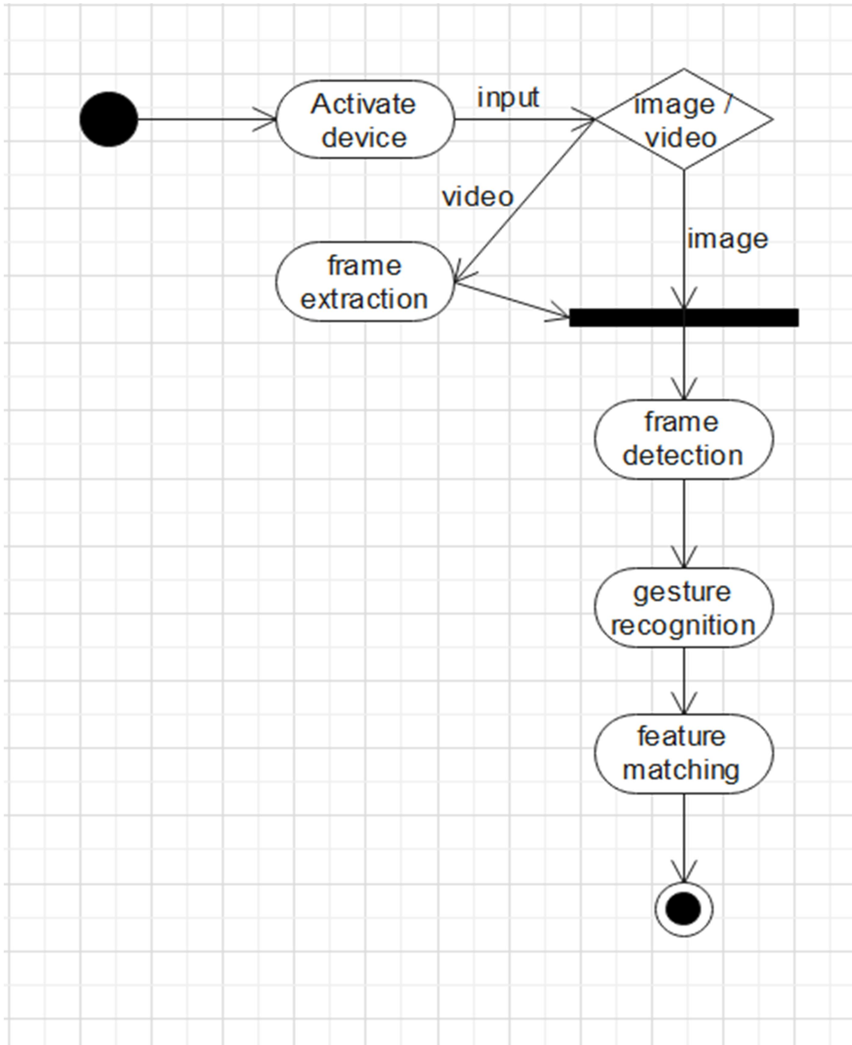
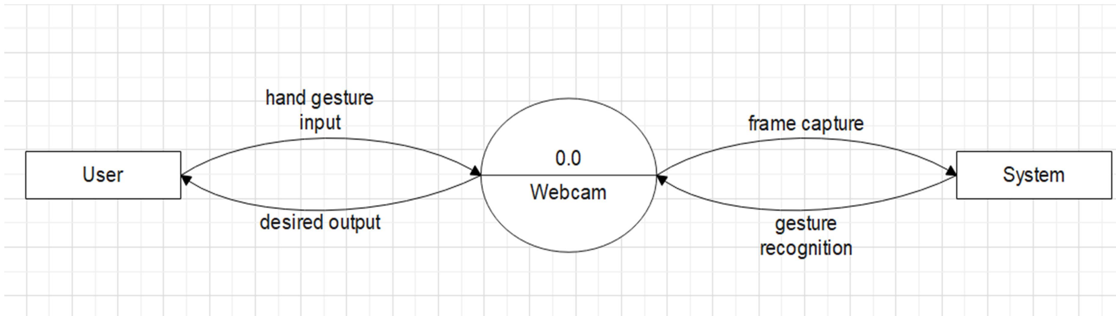


Figure 3.4: Activity Diagram

3.5 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).



DFD For LEVEL 0

Figure 3.5: Data Flow Diagram

Chapter 4 Implementation and Testing

This phase marks the completion of the project wherein the agreed requirements are fulfilled and project is put in to operation with arrangements for follow-up and evaluation.

4.1 Snapshots

- **Training & Testing Results**

```
▾ Training Our Model

history = model.fit_generator(train_batches, steps_per_epoch=64,validation_data=test_batches, validation_steps=28,epochs=10,callback

/usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/training.py:1844: UserWarning: `Model.fit_generator` is depreca
  warnings.warn('`Model.fit_generator` is deprecated and '
Epoch 1/10
64/64 [==============================] - 207s 3s/step - loss: 0.6337 - accuracy: 0.7759 - val_loss: 6.7940 - val_accuracy: 0.2645
Epoch 2/10
64/64 [==============================] - 203s 3s/step - loss: 0.0652 - accuracy: 0.9768 - val_loss: 3.2422 - val_accuracy: 0.5949
Epoch 3/10
64/64 [==============================] - 206s 3s/step - loss: 0.0565 - accuracy: 0.9838 - val_loss: 1.1764 - val_accuracy: 0.6741
Epoch 4/10
64/64 [==============================] - 203s 3s/step - loss: 0.0315 - accuracy: 0.9868 - val_loss: 1.8637 - val_accuracy: 0.6574
Epoch 5/10
64/64 [==============================] - 203s 3s/step - loss: 0.1235 - accuracy: 0.9721 - val_loss: 1.2703 - val_accuracy: 0.6920
Epoch 6/10
64/64 [==============================] - 205s 3s/step - loss: 0.0184 - accuracy: 0.9968 - val_loss: 0.8049 - val_accuracy: 0.8393
Epoch 7/10
64/64 [==============================] - 200s 3s/step - loss: 0.0133 - accuracy: 0.9964 - val_loss: 0.5792 - val_accuracy: 0.8404
Epoch 8/10
64/64 [==============================] - 206s 3s/step - loss: 0.0176 - accuracy: 0.9955 - val_loss: 1.1803 - val_accuracy: 0.7188
Epoch 9/10
64/64 [==============================] - 206s 3s/step - loss: 0.0112 - accuracy: 0.9980 - val_loss: 1.3854 - val_accuracy: 0.7411
Epoch 10/10
64/64 [==============================] - 202s 3s/step - loss: 0.0278 - accuracy: 0.9919 - val_loss: 1.0393 - val_accuracy: 0.7946
```

Figure 4.1: Training Result

```
Testing Accuracy = 80.23%
```

Figure 4.2: Testing Result

➢ Home Page

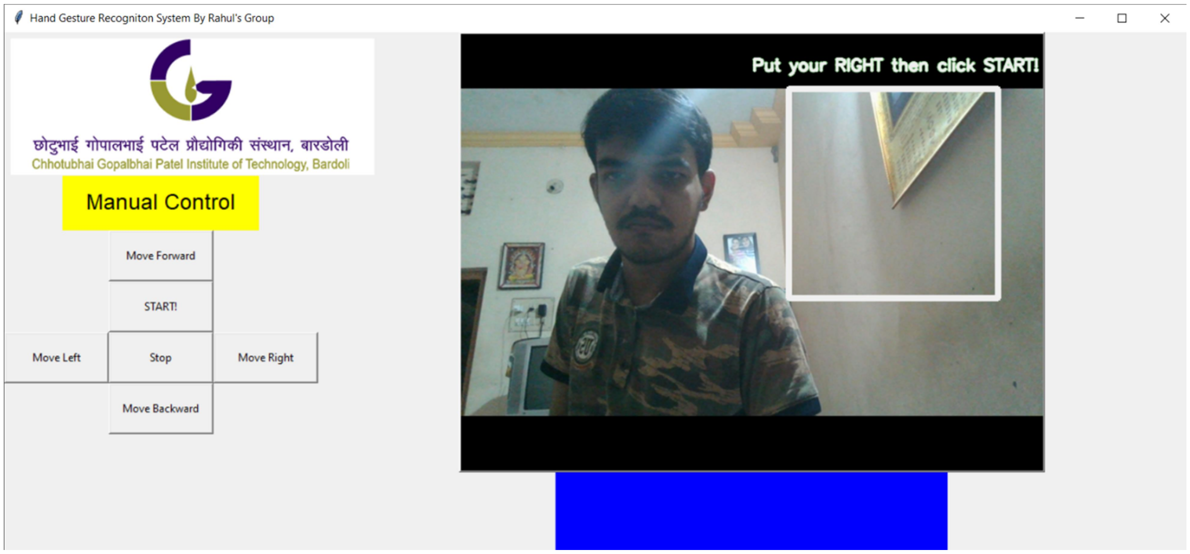


Figure 4.3: Home Page

➢ Test Case 1

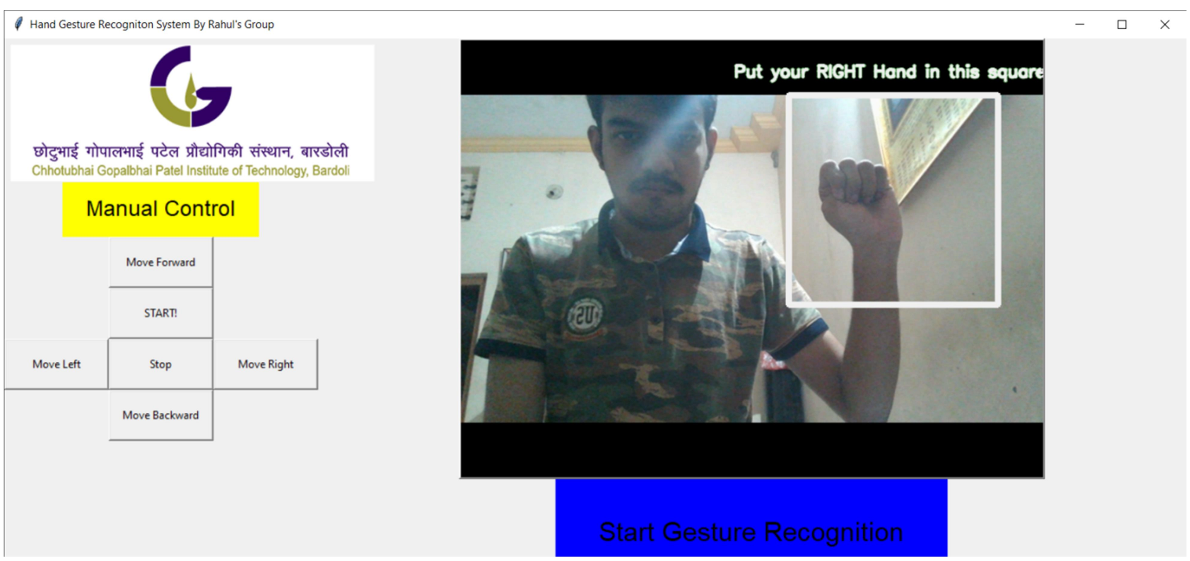


Figure 4.4: Test Case 1

➢ **Test Case 2**

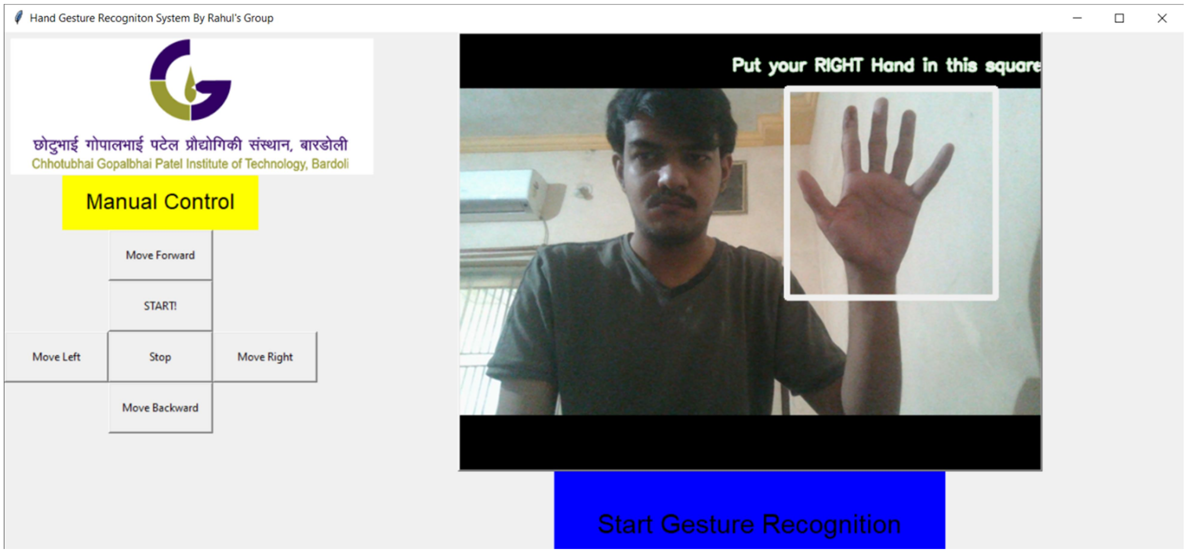


Figure 4.4: Test Case 2

➢ **Test Case 6**

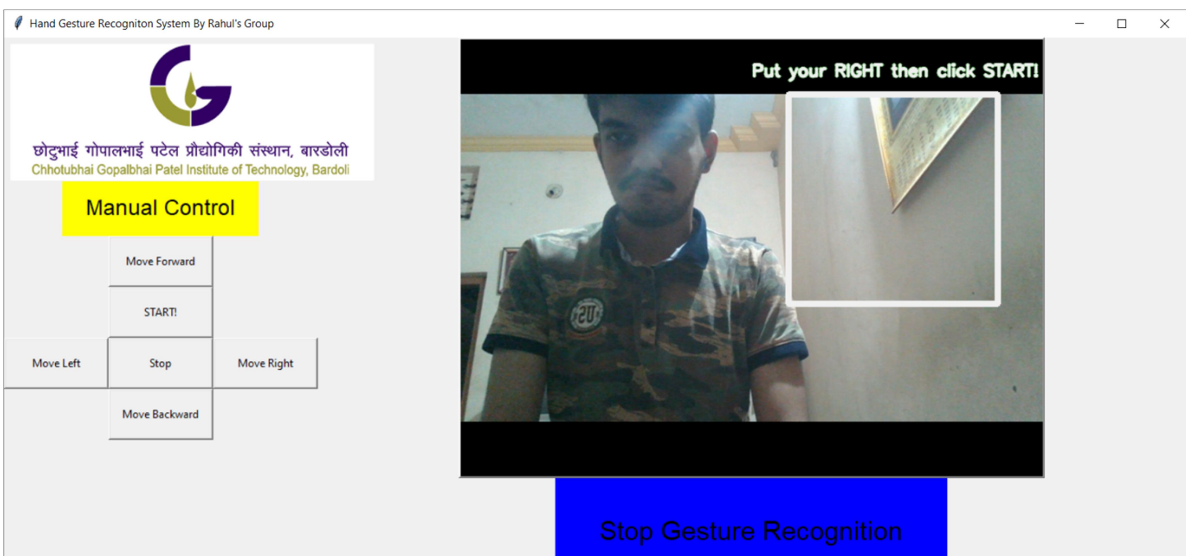


Figure 4.5: Test Case 6

4.2 Test Cases

Sample test cases are given as below:

Test ID	Case	Test Data	ExpectedResult	ActualResult	Pass/ Fail
1	Start	Press Start Button to initate Gesture Recognition Mode.	First time Recognition Module Initiated. When Start Button pressed, Gesture Recognition Control will initiate.	First time Recognition Module Initiated. When Start Button pressed, Gesture Recognition Control will initiate.	Pass
2	Move Forward/Scroll Down.	Show Number Sign 1 To Move Forward/Scroll Down.	When Number Sign 1 shown, Move Forward/Scroll Down Command will initiate.	When Number Sign 1 shown, Move Forward/Scroll Down Command will initiate.	Pass
3	Move Backward/Scroll Up.	Show Number Sign 2 To Move Backward/Scroll Up.	When Number Sign 2 shown, Move Backward/Scroll Up Command will initiate.	When Number Sign 2 shown, Move Backward/Scroll Up Command will initiate.	Pass
4	Move Left/Switch To Previous Tab.	Show Number Sign 3 To Move Left/Switch To Previous Tab.	When Number Sign 3 shown, Move Left/Switch To Previous Tab	When Number Sign 3 shown, Move Left/Switch To Previous Tab	Pass

			Command will initiate.	Command will initiate.	
5	Move Right/Switch To Next Tab	Show Number Sign 4 To Move Right/Switch To Next Tab.	When Number Sign 4 shown, Move Right/Switch To Next Tab Command will initiate.	When Number Sign 4 shown, Move Right/Switch To Next Tab Command will initiate.	Pass
6	Stop	Press Start Button again to stop Gesture Recognition Control.	When Start Button pressed, Gesture Recognition Control will terminate.	When Start Button pressed, Gesture Recognition Control will terminate.	Pass
7	Background Color same as Skin Color.	Coarse Recognition will be done.	When Background Color will be same as Skin Color, then Proper Recognition should be done.	When Background Color will be same as Skin Color, then Proper Recognition should be done.	Fail
8	Excessive Light coming from the background.	Coarse Recognition will be done.	When excess light would be coming from the background, then Proper Recognition should be done.	When excess light would be coming from the background, then Proper Recognition should be done.	Fail
9	Multiple	Coarse	When Multiple	When Multiple	Fail

	Gestures shown.	Recognition will be done.	Gestures will be shown, then Proper Recognition of atleast one gesture should be done.	Gestures will be shown, then Proper Recognition of atleast one gesture should be done.	
10	Inverted Gesture shown.	No Recognition will be done.	If Inverted Gesture is shown to the system, then Proper Recognition would be done.	If Inverted Gesture is shown to the system, then Proper Recognition would be done.	Fail

Conclusion and Future Scope

In this report, deep learning convolutional neural network based hand gesture recognition and gesture based system controlling methodology is proposed. This proposed method segments the Hand Gestures from the live video feed, and then, these Hand Gestures are given as input to the CNN classifier. The CNN classification approach trains and classifies the test hand gesture image class which is obtained from pre-trained Mobilenet v2 Model. The performance of the proposed hand gesture recognition and gesture based system controlling methodology is analyzed in terms of accuracy and recognition rate. The proposed hand gesture recognition and gesture based system controlling methodology using CNN classification approach stated in this report achieves 85% of accuracy and 95% of recognition rate.

This system has a wide future scope. This system can be used in many other areas like gaming consoles and medical machines, etc. This system would be an aid to Physically-Challenged people. This system can have more number of gestures implementation for various controls, can be used to access internet applications.

References

Paper references

- Ying Wu and Thomas S. Huang , "Vision-Based Gesture Recognition: A Review" , In: Gesture-Based Communication in Human-Computer Interaction, Vol 1739 of Springer Lecture Notes in Computer Science , pages 103-115.
- Matthias Rehm, Nikolaus Bee, Elisabeth Andre, "Wave Like an Egyptian – Accelerometer Based Gesture Recognition for Culture specific Interactions " , British Computer Society, 2007.
- Pavlovic. V, Sharma. R & Huang. T. (1997) , "Visual interpretation of hand gestures for human-computer interaction: A Review", IEEE Trans. Pattern Analysis and Machine Intelligence, July 1997. Vol. 19(7), pp. 677-695.

Web references

[1] [Online] https://link.springer.com/article/10.1007/s00500-020-04860-5. [06/09/2020]

[2] [Online] https://www.researchgate.net/publication/307174663_Deep_Learning-Based_Fast_Hand_Gesture_Recognition_Using_Representative_Frames. [06/09/2020]

[3] [Online] https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2015/W15/papers/Molchanov_Hand_Gesture_Recognition_2015_CVPR_paper.pdf [06/09/2020]

[4] [Online] https://medium.com/@aggirma/keras-convolutional-neural-network-cnn-implementation-for-hand-gesture-recognition-d7dd11958af6 [06/09/2020]

[5] [Online] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6928637/ [06/09/2020]

[6] [Online] https://keras.io/api/applications/ [01/10/2020]

[7] [Online] https://keras.io/api/applications/mobilenet/#mobilenetv2-function [01/10/2020]

[8] [Online] https://keras.io/api/applications/vgg/#vgg16-function [01/10/2020]

[9] [Online] https://github.com/Abeni18/Keras-TensorFlow-backend-CNN-Finger-Counting-deep_learning [06/09/2020]

[10] [Online] https://github.com/Jacob12138xieyuan/Hand-gesture-controled-Arduino-car-using-machine-learning-model [11/10/2020]