

MOVEMENT DETECTION AND TRACKING USING IMAGE PROCESSING

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering
by

PENUBOINA VENKAT (37110564)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

**INSTITUTE OF SCIENCE AND
TECHNOLOGY (DEEMED TO BE
UNIVERSITY)**

Accredited with Grade "A" by NAA

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600 119

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Penuboina Venkat (37110564)** who carried out the project entitled as **Movement Detection and Tracking Using Image Processing** under my supervision from November 2020 to March 2021.

Internal Guide

Dr. L. SUJHELEN, M.E., Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **Penuboina Venkat (37110564)** hereby declare that the Project Report entitled **“Movement Detection and Tracking Using Image Processing”** done by me under the guidance of **Dr. L. SUJHELEN, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

SIGNATURE OF THE CANDIDATE

PLACE:

ACKNOWLEDGEMENT

I am pleased to acknowledge our sincere thanks to **Board of Management of Sathyabama Institute Of Science And Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D.**, and **Dr. L. Lakshmanan, M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. L. SUJHELEN, M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

In this work we are introducing another model to recognize and follow the moving articles in a given video. Because of article location's cozy relationship with video examination and image understanding, it has pulled in much exploration consideration lately. Customary item identification techniques are based on carefully assembled highlights and shallow teachable designs. Recognizing and following of moving articles has become significant region of examination and is use for movement of different items on a given video. Their exhibition effectively deteriorates by developing complex troupes which consolidate numerous low-level image highlights with significant level setting from object finders and scene classifiers. With the fast advancement in profound learning, all the more useful assets, which can learn semantic, undeniable level, further highlights, are acquainted with address the issues existing in conventional designs. These models carry on distinctively in network design, preparing system and streamlining capacity, and so on In this paper, we give a survey on python based item location systems. We center around regular nonexclusive item recognition models alongside certain alterations and valuable stunts to improve discovery execution further. As particular explicit discovery undertakings show various attributes, we additionally momentarily overview a few explicit assignments, including notable article recognition, face identification and walker location. Test investigations are likewise given to analyze different strategies and reach some important inferences. At last, a few promising headings and errands are given to fill in as rules to future work in both article identification and we have attempted to catch the primary image/casing of the video yet straightforwardly catching the videowe have shown how this python program will permit us to recognize movement and furthermore store the time timespan movement.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	IV
	LIST OF FIGURES	IX
	LIST OF TABLES	X
1	INTRODUCTION	01
	1.1 OVERVIEW	01
	1.2 PURPOSE AND OBJECTIVE	02
	1.3 PROBLEM STATEMENT	03
	1.4 OBJECTIVES	03
2	LITERATURE SURVEY	04
3	SYSTEM ANALYSIS	06
	3.1 DRAWBACKS OF EXISTING SYSTEM	06
	3.2 PROPOSED SYSTEM	06
	3.3 SYSTEM REQUIREMENTS	06
	3.3.1 HARDWARE REQUIREMENTS	07
	3.3.2 SOFTWARE REQUIREMENTS	07
4	SYSTEM DESIGN	08
	4.1 PROPOSED WORK	08
	4.2 ARCHITECTURE DIAGRAM	09
	4.2.1 IMAGE ACQUISITION	09
	4.2.2 IMAGE SUBTRACTION	10

	4.2.3 IMAGE PROCESSING	11
	4.2.4 CONTOUR FINDING	11
	4.2.5 BINARY IMAGE PROCESSING	12
	4.2.6 AREA MAPPING	13
	4.3 METHODS USED	14
	4.3.1 FOUNDATION IMAGE INSTATEMENT	14
	4.3.2 MOVING OBJECT EXTRACTION	14
	4.3.3 EXTRACTION OF MOVING BODY	15
	4.3.4 CRAWLING THROUGH URL ORDERING	15
	4.3.5 REPROCESSING AS THE INTRICACY OF THE FOUNDATION	15
5	SYSTEM IMPLEMENTATION	17
	5.1 MODULES	17
	5.1.1 PANDAS	17
	5.1.2 BOKEH	17
	5.1.3 TIME	19
	5.1.4 CV2	21
	5.1.5 PYTHON	21
	5.2 RESULTS AND DISCUSSION	22
6	CONCLUSION AND FUTURE WORK	28
	6.1 CONCLUSION	28
	6.2 FUTURE WORK	28
	REFERENCES	29
	APPENDIX	31
	A. SOURCE CODE	31

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.2	ARCHITECTURE DIAGRAM	09
5.1	MOVEMENT DETECTION OF HEAD	22
5.2	MOVEMENT DETECTION OF HAND	22
5.3	TIME INTERVALS	23
5.4	GRAPH	23

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Image processing is a technique to play out some procedure on a image, to get an upgraded image or to separate some valuable data from it. It is a kind of sign preparing in which info is a image and yield might be image or attributes/highlights related with that image. Investigation of item movement is one of the new and mainstream research themes in computerized image preparing. In which the development of items is the significant piece of article location and movement investigation. Recordings are really arrangement of images, every one of which called a casing, shown in quick enough recurrence so natural eyes can percept the coherence of its content. It is clear that all image processing methods can be applied to singular edges. Furthermore, the substance of two continuous edges are generally intently related. Object discovery in recordings includes checking the presence of an article in image groupings and perhaps finding it unequivocally for recognition. Object following is the way toward finding an item or various items in a video file. In object location and following we need to target article and track that object in sequential casings of a video document.

Location of moving item in a image arrangements is significant issue of moving video. An exceptionally basic trouble in the discovery of moving article is the presence of self image movement. We address it by processing the prevailing movement . Visual observation is an extremely dynamic exploration territory in PC vision. The utilization of this idea in reconnaissance for security, battle against psychological oppression and wrongdoing. The fundamental assignment incorporates movement recognition , object arrangement ,following action. The recognition of moving items in video transfers is the main pertinent advance of data extraction in numerous PC vision applications. Numerous associations and foundations needs to get their offices subsequently need to utilize security and reconnaissance frameworks that are outfitted with most recent innovation. Shrewd video sensors were created to help security frameworks to distinguish startling

development without human mediation. The significant data of move , area , speed and any ideal data of focus from the caught edges can be taken from the camera

and can be moved to the investigation part of the framework. Development identification is one of these canny frameworks to which recognize and tracks moving targets.

There are various techniques to distinguish moving articles yet these strategies having a few impediments for constant application. Because of this explanation, in this paper we use foundation deduction technique, which is more appropriate for ongoing application which gives precise outcome. In this paper we use strategy in which single static camera to catch video for that caught video we convert in to outline and apply foundation deduction technique with refreshing foundation. The between outline distinction strategy demonstrating by combination of Laplacian conveyance and Gibbs arbitrary field is utilized for portraying the mark set [1]. Representing moving images as set of covering layers. A image coding includes three sections encoder, portrayal and decoder . The most widely recognized image coding framework utilized today dependent on Low level image processing idea like DCT's [2]. Following moving item in jumbled scene is likewise done by foundation demonstrating. Two cycles engaged with this strategy are procedure to separate an underlying foundation model when foundation circumstance changes at later time [3]. The information division and edge recognition is utilized in this Modeling. Various sorts of moving articles share likenesses yet additionally express contrasts regarding their dynamic conduct and the idea of development [4,5].

Diverse strategy utilized for development location .outline deduction technique is appropriate for assortment of dynamic climate however by and large it is hard to get total layout of moving article, as result the identification of moving item isn't exact. Huge amount of estimation, affectability to commotion, helpless enemy of clamor execution makes Optical Flow strategy not appropriate for constant requesting events. The foundation deduction technique is to utilize the distinction strategy for the current image and foundation image to identify moving items, with straightforward calculation, yet exceptionally delicate to the adjustments in the outside climate and has helpless enemy of obstruction capacity. The Background strategy is touchy to the adjustments in the outside climate and

has helpless enemy of impedance capacity. This Python program will permit us to identify movement and

furthermore store the time frame movement. Recordings can be treated as heap of images called outlines. Here we are contrasting various edges with the primary casing which ought to be static. We look at two images by contrasting the power estimation of every pixels utilizing python we can do it without any problem.

In this work consistently needed a product dependent on the webcam that can identify development and record in a video document just if something is moving. To be sure, it isn't actually possible to record from the beginning on the grounds that the hard circle drive would be immediately filled if the product needs to run a day for example. The principle objective is to save hard plate memory however much as could be expected. Surely, it isn't actually possible to record from the start in light of the fact that the hard plate drive would be immediately filled if the product needs to run a day for example.

1.2 PURPOSE AND OBJECTIVE

As proposed earlier, this project is to be linked with another project to come up with the final system called Human Motion Detection System. This project would be focused on the Video Motion Detection module where we would perform research on the techniques and methodology to detect motion and to develop a module for a technique that we prefer to use in this project. This module would record down motion and pass it into the next module that would be on object classification where it classify human and non-human object. Thus, this project is to come up with a solution that detects motion effectively and record it down with one or more objects that are moving and causing motions.

The purpose of this project is to help new researchers learn and further research on their topic of interest, which in this case is the human motion detection system. The question to be addressed here in this module is, given a sequence of images, how do we detect motion or track a moving object? The project is to mainly answer this particular question addressed by providing a prototype to emulate or prove the algorithms or techniques that are available to perform motion detection by an input of images in a number of frames.

1.3 PROBLEM STATEMENT

To automate the process of checking the updates or changes of a webpage in its contents or by any visual change occurred on the webpage so that it could help an individual who wants to monitor a particular webpage for a period of time in a while.

1.4 OBJECTIVE

Motion Detection of human has a very wide area of applications. Thomas and Erik [18] classified them into three major areas which are surveillance, control and analysis. Milan, Vaclav and Roger [16] separate three types of motion-related problems into motion detection which is usually used for security purposes, motion object detection and location which is linked to the 14th chapter discussed by Ramesh, Rangachar and Brian [7] where they further separate the category into 4 possibilities of set ups being (SCSO, SCMO, MCSO and MCMO) and finally, the third group is related to derivation of 3D object properties. The application for this field of research is clearly seen here as being very wide and broad. This research would thus focus on mainly the 2nd type of motion-related problems discussed by Milan, Vaclav and Roger [16] whereby not only detecting motion is sufficient, further steps are taken to obtain the object that causes the motion and using an recognition engine instead of just image-matching as suggested in Milan, Vaclav and Roger [16] to classify the object to check whether if it is a human or not.

With this, therefore, this project is not fixed to only the first application area discussed by Thomas and Erik [18]. It can also be applied to applications in control areas or analysis areas where human motion is to be classified to distinguish them from other objects that cause motion. However, in this paper, we would mainly focus on the application of the algorithm in surveillance systems only as this would be a basis for human motion detection. Only some enhancement or addition needs to be added if the algorithm discussed were to be ported into other applications such as those in control application areas. For example providing inputs to Virtual reality or Augmented reality applications, we need to have extra recognition of the

notified human motion returned by the algorithms discussed here. As for the classification as presented by Ramesh, Rangachar and Brian [7], the project's focus would be based on the SCMO class meaning static camera, moving objects. The reason is

simply because of cost factors since we only have one camera and for simplicity of the algorithm with the assumption of a stationary camera being used.

CHAPTER 2

LITERATURE SURVEY

In this chapter, we will look at several motion detection techniques and methodologies that have been researched and implemented by other researchers. For further understanding on their methods and techniques, refer to the reference page at the end of this thesis report to search for the papers or text or even websites published.

2.1 Introduction to a Motion Detection Algorithm

The most popular way to implement motion detection algorithm is by implementing the image segmentation and image subtraction techniques of computer vision [8]. Most of the algorithm first segments the foreground moving objects from the background image. To do this, they would have to take a sequence of images with no motion by default to initialize the background image. This background image would be updated subsequently to provide a real-time environment where changes to the background are taken into considerations. For example, when a moving object stopped, it would then be a part of the background image. Clearly, the foreground objects can be acquired by using simple arithmetic using image subtraction. The result of the subtraction techniques where pixels belonging to the current image are subtracted by the corresponding pixels in the background image or vice versa would obtain the foreground moving objects. After obtaining the foreground objects, the focus for region of interest is set to these foreground objects instead of the whole image. Therefore further image processing is performed only on these regions of interest. One of the important steps here is to extract important features of the moving objects to recognize the object. However, recognizing the object is beyond this research topic as this module of the project is to provide the objects and the inputs to the object classification engine.

2.2 Image Segmentation & Image Subtraction

As mentioned, the first step of a motion detection algorithm would be the foreground-background segmentation step. To segment out the foreground from

background there's a lot of techniques available. Ming and Tim [5] presented a way to improve the usual image subtraction techniques which only uses the pixels intensity values by adding also the colour illumination invariance into the algorithm.

As a result, their work can also be implemented effectively on motion detection areas where the illumination changes rapidly.

As said above, a motion detection algorithm begins with the segmentation part where foreground or moving objects are segmented from the background. The simplest way to implement this is to take an image as background and take the frames obtained at the time t , denoted by $I(t)$ to compare with the background image denoted by B . Here using simple arithmetic calculations, we can segment out the objects simply by using image subtraction technique of computer vision meaning for each pixels in $I(t)$, take the pixel value denoted by $P[I(t)]$ and subtract it with the corresponding pixels at the same position on the background image denoted as $P[B]$.

2.3 Moving Edge Detection

Implementing edge detectors helps in obtaining a clearer picture of the moving objects. Ramesh, Rangachar and Brian [7] & Milan, Vaclav, Roger [16] both stated in their text that using edge detectors combining with the different

pictures will give a better result for motion detection by helping to overcome several limitations. There are many edge detectors algorithms being introduced in the image processing field. Mostly all text on image processing does cover the topics on edge detection. Examples of these image processing texts are [7], [8], [16] & [17]. There are also many online references such as [6] which describe convolution techniques which may be applied to edge detection algorithms as well. Ramesh, Rangachar and Brian [7] identified 3 steps in edge detection algorithms which are:

- Filtering

- Enhancement

- Detection

“Many edge detectors have been developed in the last two decades.”
quoted from [7]. Some first derivatives operator masks used for edge detectors are Roberts operator, Sobel operator, Prewitt operator. Others use second derivatives operator

masks such as Laplacian operator. Some uses Gaussian edge detection methods, a popular known method which uses it is the Canny edge detector.

CHAPTER 3

SYSTEM ANALYSIS

3.1 DRAWBACKS OF EXISTING SYSTEM

Article following generally involves assessing the area of a specific district in progressive edges in a video sequence. Properly identifying items can be an especially difficult task, especially since articles can have rather confounded constructions and may change in shape, size, location and direction over ensuing video frames. Various calculation and plans have been presented in the couple of decades, that can follow objects in a specific video sequence, and every calculation has their own benefits and drawbacks. In object following the, important challenge that needs to consider while the working a video tracker are the point at which the foundation is show up which is like intrigued object or another article which are available in the scene.

There are very efficient existing systems are there in the theorized work but not more than ten percent of them come live to work on the real life to automate the process the following are the few reasons for that

- Ambient light
- Clamor
- Occlusions

3.2 PROPOSED SYSTEM

In this proposing work python will permit us to recognize movement and furthermore store the time timespan movement. The new depicted calculation was applied on three more video successions and it likewise showed excellent and promising outcomes. One of the hued successions had two moving articles. The proposed technique prevailing to totally effectively identify the both moving articles in the scene freely of the luminance conditions. We appeared in this paper the outcomes got from the one of the pre-owned successions that was the most helpful for the portrayal. The proposed calculation, invariant to outside luminance changes, has been tried under different lighting conditions, misleadingly recreated

on the PC and with the moving article more splendid and hazier than the foundation, and acceptable and promising outcomes have been accomplished.

3.3 SYSTEM REQUIREMENTS

Requirement analysis determines the requirements of a new system. This project analyses on product and resource requirement, which is required for this successful system. The product requirement includes input and output requirements it gives the wants in term of input to produce the required output. The resource requirements give in brief about the software and hardware that are needed to achieve the required functionality.

3.3.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

- ❖ System : Intel core
- ❖ Speed : 2.4GHZ
- ❖ Hard Disk : 500 GB and Above
- ❖ Ram : 4GB and Above
- ❖ Net Speed : 2Mbps and Above

3.3.2 SOFTWARE REQUIREMENTS

The software requirements are the specification of the system. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

- ❖ Operating System : Windows
- ❖ Coding language : Python
- ❖ IDE : Python IDLE
- ❖ Web framework : Django

CHAPTER 4

SYSTEM DESIGN

4.1 PROPOSED WORK

In this proposing work python will permit us to recognize movement and furthermore store the time timespan movement. The new depicted calculation was applied on three more video successions and it likewise showed excellent and promising outcomes. One of the hues successions had two moving articles. The proposed technique prevailing to totally effectively identify the both moving articles in the scene freely of the luminance conditions. We appeared in this paper the outcomes got from the one of the pre-owned successions that was the most helpful for the portrayal. The proposed calculation, invariant to outside luminance changes, has been tried under different lighting conditions, misleadingly recreated on the PC and with the moving article more splendid and hazier than the foundation, and acceptable and promising outcomes have been accomplished.

4.2 ARCHITECTURE DIAGRAM

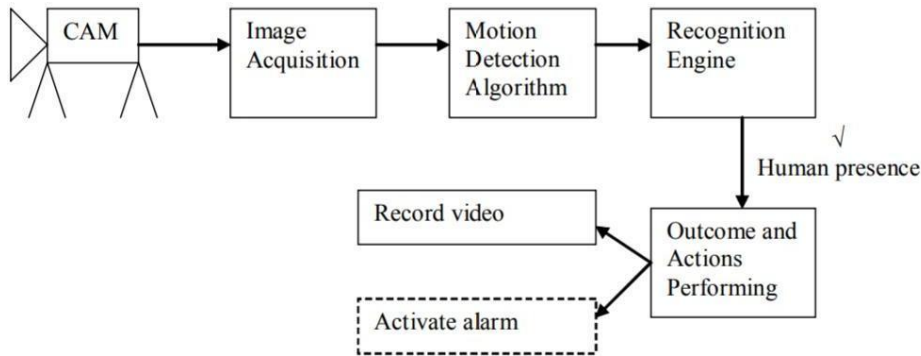


Fig 4.2 Architecture Diagram

4.2.1 Prototype Implementation using Intel OpenCV Libraries

In building the prototype, we'd implemented MFC (Microsoft Foundation Classes) libraries for some basic windows functions using some of the win32 APIs (Application Programming Interfaces) provided by Microsoft. For the image processing side, we'd incorporated the Intel OpenCV libraries [19] to do the processes. For the recognition engine, the partner in this project had used some Torch libraries [20] for Artificial Neural Network programming. Anyhow, here we would discuss some of the important functions that are critical to implement such motion detection algorithm discussed in the previous chapter. We refer back to figure 5 shown in the previous chapter which is chapter 3. Here, we will thus follows the flow of the motion detection algorithm implemented as displayed in the figure.

1) Image Acquisition

Handling codes are used to distinguish the image capture options. First the program checks for input from camera devices. If there's no camera devices attached, then users will be given an option to initialise the "avi" version where input will be captured from a recorded video file.

2) Image Subtraction

As mentioned, a colour absolute subtraction approach is being implemented. Thus, we had developed a customised function for this purpose.

3) Image Processing

As described previously, this section has two distinct kinds of image processing techniques being implemented. The adaptive threshold function was performed using a function provided by the partner in this project since the purpose was to give a better result for the recognition engine and helps improve his feature extraction algorithm. To apply a threshold value returned by the function we can simply use the `cvThreshold(some parameters)` function provided by OpenCV library [19]. For the other processes, libraries function by OpenCV can be implemented directly since they provided Dilate & Erode function in `cvDilate(some parameters)` & `cvErode(some parameters)` for algorithm 2. To get the actual parameters for these functions one may read the OpenCV's documentation or manual provided once one installed the libraries. In the case of algorithm 2, functions such as `cvCanny(some parameters)` and `cvDilate(some parameters)` is also being implemented for the moving edge concept described in the 2nd and 3rd chapters. However, here the moving edges method being implemented is different from what was described in the 2nd chapter, thus, the function `cvAnd(some parameters)` were not implemented at all as described in

the 3rd chapter. Thus, the `cvCanny(some parameters)` function is performed directly on the resulted image of the subtraction after it was being threshold. In this case, it uses the image as resulted from the adaptive threshold method implemented instead of the dilated and eroded image.

4) Contour Finding

Similar to those performed in the previous stage, this section can be implemented by using functions provided by OpenCV libraries [19] directly.

5) Bounding Rectangle Calculations

An algorithm was developed to remove overlaying rectangles caused by disconnected contours components.

6) Binary Image Processing

We developed a way to better obtain a shape of a moving object from the threshold result of subtraction.

7) Area Mapping

This section is basically implemented by using the cvSetImageROI(some parameters) and the cvResetImageROI(some parameters) functions as provided by OpenCV [19].

4.3 METHODS USED

4.3.1 FOUNDATION IMAGE INSTANTMENT

There are numerous techniques to get starting foundation image. Time normal strategy cannot arrangement with image shadow issues. While the technique for taking the middle from ceaseless multiframe can resolve this issue basically and viably. So, the middle technique is chosen for foundation introduction.

4.3.2 FOUNDATION UPDATE FOR THE FOUNDATION MODEL

Here can all the more likely adjust to light changes, the foundation should be refreshed progressively, to precisely extricate the moving article. The camera is fixed; the foundation model can remain generally stable in the significant stretch of time. At this stage we can successfully maintain a strategic distance from the startling wonder of the foundation, for example, the unexpected appearance of something in the foundation which is excluded from the first foundation. Additionally by the update of pixel dark estimation of the foundation, the effect brought by light, climate and different changes in the outside climate can be adequately adjusted.

4.3.3 MOVING OBJECT EXTRACTION

To separate moving article dynamic edge strategy is reasonable by utilizing progressively changed limit esteems as per lighting changes of the two images .This technique adequately smothered the impact of light changes.

4.3.4 REPROCESSING AS THE INTRICACY OF THE FOUNDATION

The distinction image got contains the movement locale, moreover it additionally

contains enormous number of commotion. Subsequently, commotion should be eliminated. To eliminate this clamor we utilize middle channel with the 3×3 window and channels out some commotion. After the middle channel, moreover the

movement locale, incorporates body parts, yet additionally may incorporate moving vehicles, flying birds, streaming mists and influencing trees and other no one sections. Morphological strategies are utilized for additional processing. First and foremost, erosion activity is taken also viably to sift through non-human movement zones. Also, utilizing the development activity to sift through the greater part of the non-body movement districts while protecting the state of human movement without injury. After extension and erosion tasks, some disconnected spots of the image and some impedance of little pieces are wiped out, and we get more exact human movement locale.

4.3.5 EXTRACTION OF MOVING HUMAN BODY

After middle sifting and morphological tasks, some precise edge districts will be got, yet the locale has a place with the moving human body couldn't be resolved. Likewise it is seen that while moving item shows up shadow will likewise show up in some district of scene. It will influence to exactness of moving item identification. By utilizing vertical and flat projection to distinguish the tallness of movement area. This will assist with disposing of the effect of the shadow up to specific degree. Human body identification is to recognize the relating part of human from the moving locale. In any case, the extricated moving locale may compare to various moving items, like people on foot, vehicles and other such birds, drifting mists, the Swaying tree and other moving articles. Henceforth we utilize the shape highlights of movement areas to additionally decide if the moving item is an individual. Making a decision about models are as per the following: (1) The item zone is bigger than the set limit

(2) The perspective proportion of the article locale ought to adjust to the set proportion. In the event that these two conditions are met, the moving item is the moving human body, or is certifiably not a human body.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 MODULES

5.1.1 PANDAS

pandas is a Python bundle giving quick, adaptable, and expressive information structures intended to make working with "social" or "named" information both simple and natural. It intends to be the central significant level structure block for doing functional, certifiable information investigation in Python.

5.1.2 BOKEH

Bokeh is an information perception library in Python that gives elite intuitive outlines and plots. Bokeh yield can be acquired in different mediums like journal, html and worker. It is conceivable to insert bokeh plots in Django and jar applications.

5.1.3 TIME

The Python time module provides many ways of representing time in code, such as objects, [numbers](#), and strings. It also provides functionality other than representing time, like waiting during code execution and measuring the efficiency of your code. It additionally gives usefulness other than addressing time, such as holding up during code execution and estimating the effectiveness of your code. In this undertaking we utilized TIME module to store the length that entered by end client and up to that time that entered by client we will send the updates that occurred in site through mail to client.

5.1.4 CV2

CV2 is an open source library which is supported by multiple platforms including Windows, Linux, and MacOS, and is available for use in multiple other languages as well; however, it is most commonly used in Python for Machine Learning applications, specifically in the Computer Vision domain.

Apart from its cross-platform support and availability in multiple other computer languages, which allows applications developed in it to be used on different systems, CV2 is also, in comparison to other such libraries, fairly efficient in terms of computations, as it uses vector operations for most of its functions.

5.2 RESULTS AND DISCUSSION

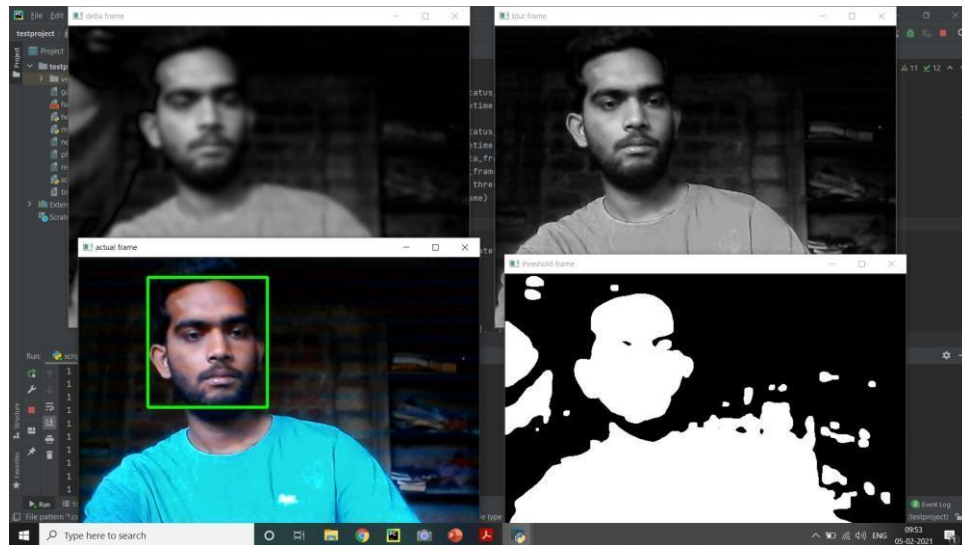


Fig 5.1 MOVEMENT DETECTION OF HEAD

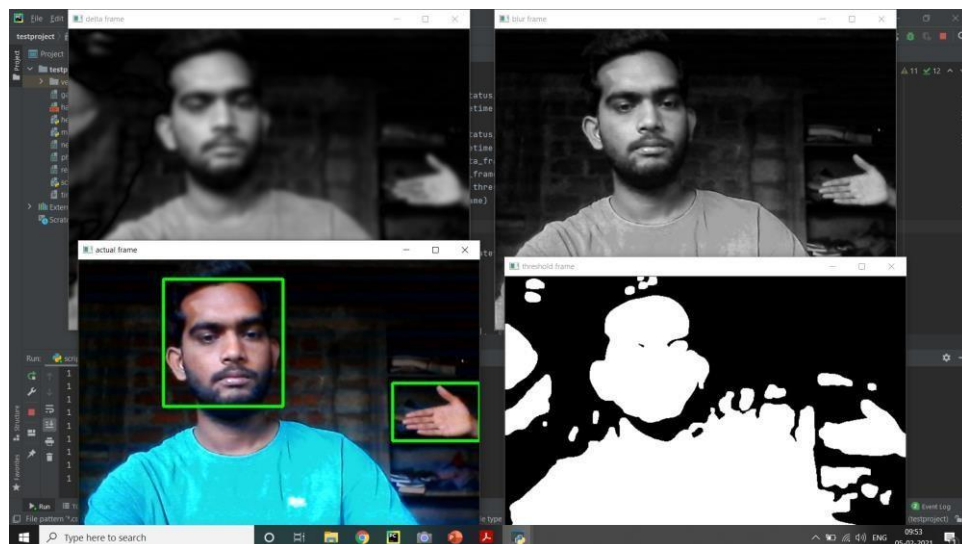


Fig 5.2 MOVEMENT DETECTION OF HAND

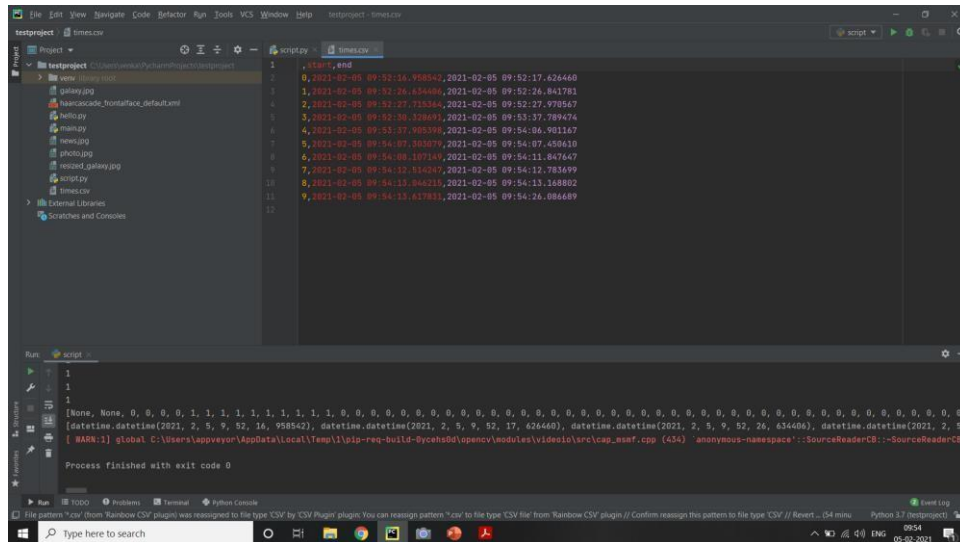


Fig 5.3 TIME INTERVALS

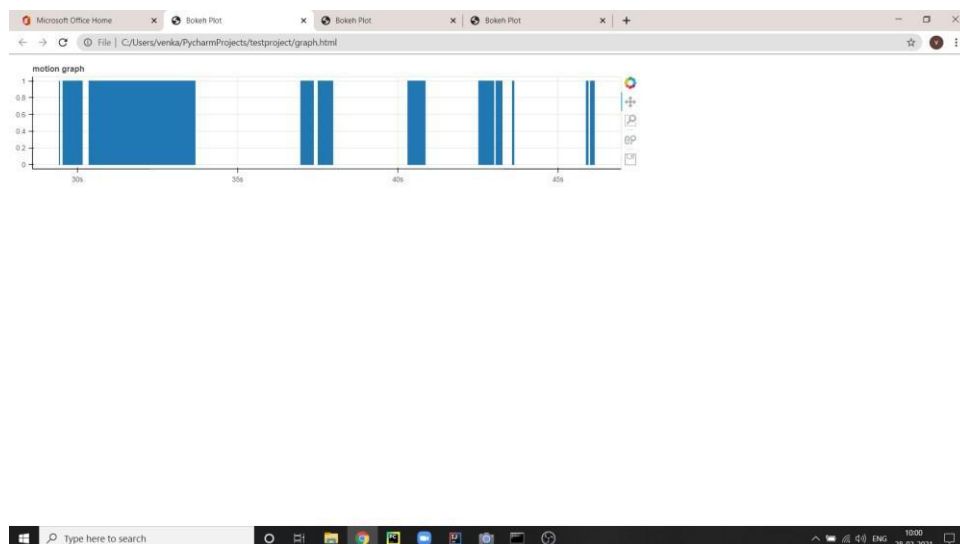


Fig 5.4 GRAPH

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In this work, we have exhibited how this python program will permit us to distinguish movement and furthermore store the time timespan movement. we have talked about the idea of item identification and following in a video utilizing OpenCV in Python, utilizing different strategies, for example, outline differencing, colorspace, foundation detachment, optical streams, and Haar falls classifiers. We have additionally examined in detail, a renowned edge recognition calculation. We utilized the rich library set of OpenCV for a powerful face recognition from an example video. We later stretched out our model to identify eyes and nose in a similar information video. We additionally actualized all ideas like edge differencing, colorspace, foundation partition, optical streams utilizing OpenCV in Python. • We actualized Canny's Edge Detection calculation additionally in OpenCV on an example video and distinguished all the edges in the video. Every one of our models could effectively distinguish all faces, eyes, and noses in the info image with 100% discovery exactness and continuously recognition speed.

6.2 FUTURE WORK

Here, we had presented a research on some image processing techniques implemented for motion detection algorithms and also some of the methodology and approaches of implementing a motion detection algorithm itself. The review includes research on a basic end-product implementation of a motion detection application. We had presented some ways for implementing the approaches researched and also shared some ideas for alternatives ways to implement the motion detection algorithm. An implementation and system design of a prototype system developed for testing purposes is reviewed in this report as well.

There are still many areas which can be further researched from this point onwards. For example, the techniques introduced may be enhanced to suit some problem specific applications or some domain specific applications. Also the techniques may be further enhanced by implementing more useful methods and algorithms such as those involving with tracking the object which causes the motion events such as those using optical flows or also known as image flows. Here, human motions are being detected. However, future works may also want to recognise the pose or gesture of the human body registered by the algorithm implemented here in the prototype system.

REFERENCES

- [1]. R. Sujeetha, Vaibhav Mishra , “Object Detection and SZ Tracking using Tensor Flow”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1, May 2019.
- [2] Abdul Muhsin M, Farah F. Alkhalid, Bashra Kadhim Oleiwi, “Online Blind Assistive System using Object Recognition”, International Research Journal of Innovations in Engineering and Technology (IRJIET) ,Volume 3, Issue 12, pp 47-51, December – 2019
- [3] 3. Bhumika Gupta, Ashish Chaube, Ashish Negi, Umang Goel, “Study on Object Detection using Open CV Python”, International Journal of Computer Applications Foundation of Computer Science (FCS), NY, USA, Volume 162, Number 8, 2017
- [4] Kartik Umesh Sharma and Nileshsingh V. Thakur, “A Review and an Approach for Object Detection in Images”, International Journal of Computational Vision and Robotics, Volume 7, Number 1/2, 2017.
- [5] Mukesh Tiwari, Dr. Rakesh Singhai, “A Review of Detection and Tracking of Object from Image and Video Sequences”, International Journal of Computational Intelligence Research, Volume 13, Number 5 (2017)
- [6] Aishwarya Sarkale, Kaiwant Shah, Anandji Chaudhary, Tatwadarshi P. N., “A Literature Survey: Neural Networks for Object Detection”, VIVA-Tech International Journal for Research and Innovation Volume 1, Issue 1 (2018) ISSN(Online): 2581-7280 Article No. 9.
- [7] Karanbir Chahal¹ and Kuntal Dey, “A Survey of Modern Object Detection Literature using Deep Learning”, International Research Journal of

- [8] T. Horprasert, D. Harwood, L.S. Davis, A statistical approach for real-time Robustbackground subtraction and shadow detection.
- [9] Du-Ming Tsai and Shia-Chih Lai, "Independent Component Analysis Based BackgroundSubtraction for Indoor Surveillance," Image processing IEEE Transaction a volume 18issue, 1Jan 2009
- [10] NiuLianqiang and Nan Jiang, "A moving objects detection algorithm based on improvedbackground subtraction,"Intelligent Systems Design and Applications, 2008.ISDA'08.Eighth International conference on volume3, 26-28 Nov 2008.
- [11]T. Aach, A. Kaup, and R. Mester.Statistical model-based change detection in moving video. Signal Processing, 31:165{180, 1993.
- [12]M. Abdel-Mottaleb and A. Elgammal. Face detection in complex environments from color images. In IEEE International Conference on Image Processing, Kobe, Japan, 1999.
- [13]D. Comaniciu, V. Ramesh, P. Meer, "Kernel-Based Object Tracking", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.25, No. 5, 2003.
- [14]D. Comaniciu, V. Ramesh, P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift", IEEE Conf. on Computer Vision and Pattern Recognition (CVPR „00), vol.2, 2000.
- [15]Th. Gevers, "Color in Image Search Engines", Survey on color for image retrieval from Multimedia Search. M. Lew, Springer Verlag, 2001.

Appendix

A. Source code

```
script.py
import cv2
import pandas
import datetime

first_frame = None
status_list = [None, None]
times = []
df = pandas.DataFrame(columns = ["start", "end"])
video = cv2.VideoCapture(0)

while True:
    check, frame = video.read()
    status = 0
    #print(check)
    #print(frame)
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    blur_frame = cv2.GaussianBlur(gray_frame, (21,21), 0)
    if first_frame is None:
        first_frame = blur_frame
        continue
    delta_frame = cv2.absdiff(first_frame, blur_frame)
    threshold_frame = cv2.threshold(delta_frame, 30, 255, cv2.THRESH_BINARY)[1]
    threshold_frame = cv2.dilate(threshold_frame, None, iterations = 2)
    (cnts, _) = cv2.findContours(threshold_frame.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    for contour in cnts:
        if (cv2.contourArea(contour) < 10000 or cv2.contourArea(contour) > 50000):
            continue
    status = 1
```

```
(x, y, w, h) = cv2.boundingRect(contour)
```

```

cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 3)

status_list.append(status)
status_list = status_list[-2:]
if status_list[-1] == 1 and status_list[-2] == 0:
    times.append(datetime.datetime.now())

if status_list[-1] == 0 and status_list[-2] == 1:
    times.append(datetime.datetime.now())
cv2.imshow("delta frame", delta_frame)
cv2.imshow("blur frame", gray_frame)
cv2.imshow("threshold frame", threshold_frame)
cv2.imshow("actual frame", frame)
key = cv2.waitKey(1)
if key == ord('q'):
    if status == 1:
        times.append(datetime.datetime.now())
    break
print(status)
print(status_list)
print(times)
for i in range(0, len(times), 2):
    df = df.append({"start":times[i], "end": times[i+1]}, ignore_index= True)
df.to_csv("times.csv")
video.release()
cv2.destroyAllWindows()

```

Plotting.py

```

from script import df
from bokeh.plotting import figure, output_file, show
from bokeh.models import HoverTool, ColumnDataSource

```

```
df["start_string"] = df["start"].dt.strftime("%Y-%m-%d %H:%M:%S")  
df["end_string"] = df["end"].dt.strftime("%Y-%m-%d %H:%M:%S")
```

```
cds = ColumnDataSource(df)

p = figure(x_axis_type="datetime", height=200, width=1000, title="motion graph")
p.yaxis.minor_tick_line_color = None

hover = HoverTool(tooltips = [("start", "@start_string"), ("end", "@end_string")])
p.add_tools(hover)

q = p.quad(left="start", right="end", top=1, bottom=0, source = cds)
output_file("graph.html")
show(p)
```