A Project Report

On

DETECTING THE MOVEMENT OF OBJECTS WITH WEBCAM

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY

Kukatpally, Hyderabad-500085, Telangana, India

In partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

By

TANGELLAMUDI SAI TEJA [19E31A1226]

SAIRI LAXMI PRASANNA [19E31A1221]

LINGALA NAVEEN [19E31A1213]

Under the guidance of

Mr. R. KESAVAIAH

Assistant Professor, MIST

Department of Information Technology

MAHAVEER INSTITUTE OF SCIENCE AND TECHNOLOGY

(Affiliated to JNTU Hyderabad, Approved by AICTE)

Accredited by NAAC 'A' Grade

Vyasapuri, Bandlaguda, Post: Keshavagiri, Hyderrabad-500-005

2022-2023

MAHAVEER INSTITUTE OF SCIENCE AND TECHNOLOGY

(Affiliated to JNTU Hyderabad, Approved by AICTE)

Accredited by NAAC 'A' Grade

Vyasapuri, Bandlaguda, Post: Keshavagiri, Hyderabad-500005

Department of Information Technology

ESTD : 2001

CERTIFICATE

This is to certify that this project work report entitled "**DETECTING THE MOVEMENT OF OBJECTS WITH WEBCAM**" which is being submitted by **TANGELLAMUDI SAI TEJA [19E31A1226], SAIRI LAXMI PRASANNA [19E31A1221], LINGALA NAVEEN [19E31A1213]** in partial fulfillment for the award of the **Degree of Bachelor of Technology in Information Technology**, affiliated to **Jawaharlal Nehru Technological University**, Hyderabad and is a record of the bonafide work carried out by them under our guidance during 2022-2023.

Signature of Project Guide

Mr. R. KESAVAIAH

(Assistant Professor)

Signature of Head of the department

Dr. A. NANDA GOPAL REDDY

(Head of the Department)

External Examiner

DR. B.V SANKER RAM

(principal)

ACKNOWLEDGMENT

We would like to express our deep-felt appreciation and gratitude to **Mr. R. KESAVAIAH (**Assistant Professor), Department of Information Technology our guide, for his skillful guidance, constant supervision, timely suggestion, keen interest, and encouragement in completing the individual seminar within the stipulated time.

We wish to express our gratitude to **Ms. K. PRIYANKA,** Project Coordinator, who has shown keen interest and even rendered his valuable guidance in terms of suggestions and encouragement extended to us with immense care and zeal.

We express our profound sense of gratitude to **Dr. A. NANDA GOPAL REDDY**, Head of Department of the Information Technology, who has served as a host of valuable corrections and for providing us time and amenities to complete this project.

We express our thanks to **Dr. B. V. SANKER RAM**, Principal of our college and the management of Mahaveer Institute of Science and Technology for providing an excellent academic environment in the college.

We wish to express our gratitude to the Members of staff and all others who helped us in more than one way. We would also like to thank the Lab Assistants and Programmers for helping us through our project.

TANGELLAMUDI SAI TEJA [19E31A1226]

SAIRI LAXMI PRASANNA [19E31A1221]

LINGALA NAVEEN [19E31A1213]

DECLARATION

We hereby declare that the project entitled "**DETECTING THE MOVEMENT OF OBJECTS WITH WEBCAM"** submitted to partial fulfillment of the requirements for award of the **DEGREE OF BACHELOR OF TECHNOLOGY** AT **MAHAVEER INSTITUTE OF SCIENCE & TECHNOLOGY**, affiliated to **JAWAHARLAL NEHRU TECHNOLOGY UNIVERSITY**, Hyderabad in authentic work and has not been submitted to any other university institute for award of any degree.

TANGELLAMUDI SAI TEJA [19E31A1226]

SAIRI LAXMI PRASANNA [19E31A1221]

LINGALA NAVEEN [19E31A1213]

ABSTRACT

Various Methods Are Used in Motion Detection of a Particular Interest. Each Algorithm Is Found Efficient in One Way. But There Exists Some Limitation in Each of Them. This Paper Proposes a Method for Detecting Motion in a Particular Region Being Observed. Motion Tracking Surveillance Has Gained a Lot of Interests Over the Past Few Years. This System Is Brought into Effect Providing Relief to The Normal Video Surveillance System Which Offers Time-Consuming Reviewing Process. Through The Study and Evaluation of Products, We Propose a Motion Tracking System Consisting of Its Method for Motion Detection. In Our Proposed System Those Disadvantages Are Omitted and Combining the Usage of Best Method We Are Creating a New Motion Detection Algorithm for Our Proposed Motion Tracking System.

Object tracking is the process of locating moving objects over time using the camera in video sequences. The objective of object tracking is to associate target objects in consecutive video frames. Object tracking requires location and shape or features of objects in the video frames. So, object detection and object classification are the preceding steps of object tracking in computer vision application. To detect or locate the moving object in frame, Object detection is the first stage in tracking. It is a challenging or difficult task in the image processing to track the objects into consecutive frames. Various challenges can arise due to complex object motion, irregular shape of object, occlusion of object to object and object to scene and real time processing requirements. This paper presents the various techniques of object tracking in video sequences.

Keywords— Moving object detection and tracking; Gaussian Mixture Model; Adaptive Background Modeling; color as a feature; video surveillance.

TABLE OF CONTENTS

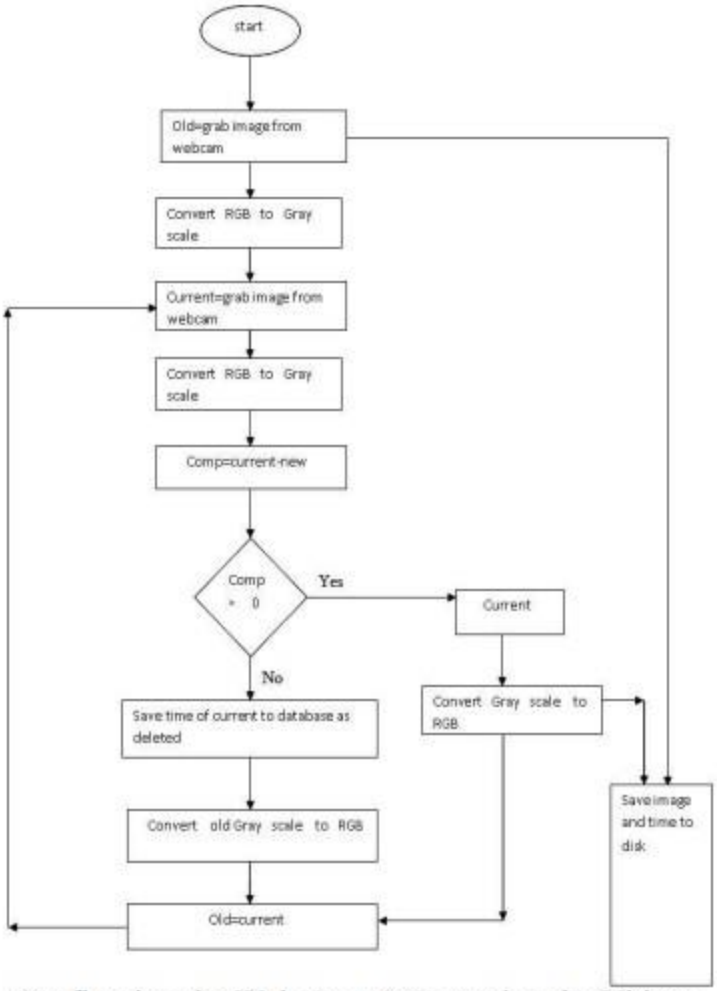
CHAPTER 1

INTRODUCTION

Motion detection is the process of detecting a change in the position of an object relative to its surroundings or a change in the surroundings relative to an object. Motion detection can be achieved by either mechanical or electronic methods. When motion detection is accomplished by natural organisms, it is called motion perception.

A motion detector is a device that detects moving objects, particularly people. Such a device is often integrated as a component of a system that automatically performs a task or alerts a user of motion in an area. An occupancy sensor detects the presence of an entity within a certain space. Motion controllers are also used for video game consoles as game controllers. A camera can also allow the body's movements to be used as an input device, such as in the Kinect system. In video editing motion estimation is a type of video compression scheme. The motion estimation process is done by the coder to find the motion vector pointing to the best prediction macroblock in a reference frame or field. For compression redundancy between adjacent frames can be exploited where a frame is selected as a reference and subsequent frames are predicted from the reference using motion estimation. The motion estimation process analyzes previous or future frames to identify blocks that have not changed, and motion vectors are stored in place of blocks. The process of video compression using motion estimation is also known as interframe coding.

Detection, tracking and identifying people in real time videos have become more and more important in the field of computer vision research. It has many applications, such as video-based surveillance and human–computer interaction. Its aim is to locate targets, retrieve their trajectories, and maintain their identities through a video sequence.



Configuring the Web Cam & Capturing the Videos

1.1 Detecting movement of objects with a webcam:

Detecting movement of objects with a webcam involves analyzing consecutive frames to identify changes in pixel values or patterns, indicating motion. Here's a theoretical overview of the process:

1. **Frame acquisition:** The webcam captures a series of frames (images) at a regular interval. Each frame is a two-dimensional grid of pixels, where the intensity or color of each pixel represents the visual information.

2. **Frame differencing:** To detect motion, the current frame is compared with the previous frame. This is achieved by taking the absolute difference between the pixel values of corresponding pixels in the two frames. The resulting image, known as the difference image, highlights areas where motion has occurred.
3. **Thresholding:** The difference image obtained in the previous step contains both significant motion regions and minor noise. Applying a threshold helps separate the two. Pixels with intensities above a certain threshold are considered part of the moving objects, while pixels below the threshold are considered background or noise.
4. **Morphological operations:** Morphological operations, such as dilation and erosion, are commonly applied to refine the binary image obtained from thresholding. Dilation expands the regions, which can help connect broken contours, while erosion shrinks the regions, removing small noise pixels.
5. **Contour detection:** Contours are the boundaries of connected components in the binary image. OpenCV provides functions to find contours based on the processed binary image obtained from thresholding and morphological operations. Each contour represents a potential moving object in the frame.
6. **Motion analysis:** Once the contours are identified, you can perform various analyses on them, such as calculating the area or centroid of each contour. By comparing the properties of different contours across frames, you can track and analyze the movement of objects over time.
7. **Object tracking and recognition (optional):** If you want to track specific objects or perform object recognition, you can employ additional techniques such as object tracking algorithms (e.g., Kalman filters, optical flow, or correlation-based methods) or machine learning-based object detection models (e.g., YOLO, SSD).

By combining these steps, you can develop a system that effectively detects and tracks motion in a webcam feed. Keep in mind that the actual implementation may require fine-tuning and adjusting parameters based on the specific environment and application requirements.

CHAPTER 2

LITERATURE SURVEY

1. Anguelov D. The task of object detection is to determine whether there are objects belonging to the specified category in the image. If it exists, then the subsequent task is to identify its category and location information. Traditional object detection algorithms are mainly devoted to the detection of a few types of targets, such as pedestrian detection and infrared target detection.
2. Cardarelli E. Detection of moving objects in roundabouts based on a monocular system. Improved the Haar features and trained the Adaboost cascade classifier to classify and identify the vehicle, and the reliability of vehicle object detection is improved, but the detection effect is reduced when the relative speed of the vehicle is too fast. The above artificial feature extraction algorithm first extracts candidate regions through sliding window filtering and other methods, then manually extracts features, and finally uses a classifier for classification and recognition.
3. Dumitru Erhan, "Scalable Object Detection using Deep Neural Networks". We used a real-time object detection algorithm, YOLO, to train our machine learning model for object detection.
4. Erhan, D et al., 2014 proposed research on object detection using neural networks. In the object detection method, a deep convolutional network was used, whereas region based conventional network increases the accuracy of the network and decreases the time.
5. Eric. K. W 2019 techniques have brought much more easiness to train large and deeper networks and shown enhanced performance. Newly, approaches have been established to identify vehicles and other objects from videos or static images using deep convolutional neural networks (DCNN).
6. Felzenszwalb. P. F. et al., "Object detection with discriminatively trained part-based models". At present there are Automated systems which use Object detection techniques to detect the object, in that object detecting techniques they have used the CNN older algorithms. But here we are using the Yolo algorithm which is a new algorithm to detect objects quickly with more accuracy so that detecting speed will be improved, and action will be done quickly.

7. Girshick. R., faster R-CNN proposes candidate regions and uses CNN to confirm candidates as valid objects. YOLO uses end-to-end unified, fully convolutional network structure that predicts the objectless assurance and the bounding boxes concurrently over the whole image.
8. Galvez R. L. et al Classification and detection of objects is now accurately possible with the recent advancements in the field of deep neural networks in image processing. In this paper, the authors have used CNN to detect objects in the living environment. Outputs very clearly show that the former model is ideal for applications in real-time, the reason being its speed.
9. Han, C., Liu et al., 2018 proposed research on a novel approach to identify the shape of the clustered image. The transformation from queried shape to cluttered image is the core idea in the transformation of the image. The detection of the moving objects can be daunting tasks and objects can be detected using several methods such as histogram-oriented gradients, background subtractions. A point based descriptively; PAD (Pyramid of arc length descriptor) identifies the pairs between the queried shape and local shape image.
10. Joseph Redmon et al “YOLOv3: An Incremental Improvement" by Ali Farhadi discusses how YOLOv3 is an incremental improvement over earlier versions of the algorithm. The deepest learning and YOLO structure were used in a deep convolution neural network. In this paper, the authors present some updates to the YOLO algorithm.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

Digital surveillance systems are mostly specifically designed for commercial use, and it has always been out of reach for other users. The cost for CCD cameras, networking devices and the software designed for this system has made it inaccessible and impractical for home users with moderate requirements. Also, not all the existing products have the motion detection function. In traditional systems for security operations, cameras are used to deliver analogue video images to monitors or time-lapse video cassette recorders (VCR). Although many local image processing functions are possible to improve the system application, this requires a lot of processing resources and high-power-consuming hardware. Although Digital video surveillance and security systems are widely used, analogue systems still serve as a cheaper alternative.

Drawbacks of existing systems:

1. Used for Commercial purposes.
2. Inaccessible to the other users (common people).
3. CCD cameras, networking devices are Expensive.
4. Absence of motion detection functionality.
5. Requires a lot of processing resources.
6. The archive space used to store videos is too high.
7. Manual monitoring of videos is Time consuming.
8. Requires high-power-consuming hardware.
9. Less accuracy
10. low Efficiency

3.2 Proposed System:

In this we are going to write a python program which is going to analyze the images taken from the webcam and try to detect the movement. Videos can be treated as a stack of pictures called frames. Here I am comparing different frames(pictures) to the first frame which should be static (No movements initially). We compare two images by comparing the intensity value of each pixel. In my project, I used Python Programming Language and its most important and specific libraries OpenCV which is most required for solving problems related to images and videos and this is an Open-Source Computer Vision based personal project to detect Human Faces and different objects coming in front of the webcam for a specific time frame. This python script detects movement on your webcam and outlines the moving object on your computer screen.

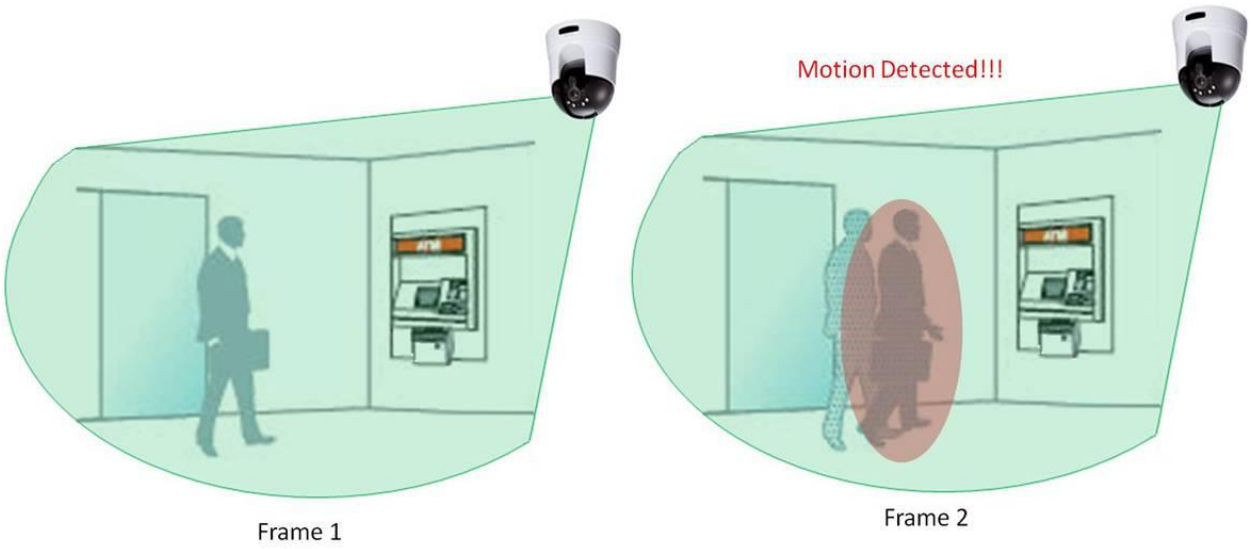
Advantages:

1. Requires less memory.
2. Analysis is done automatically.
3. Alert systems may be implemented automatically when the motion is detected.
4. High accuracy
5. High efficiency

3.3 System Design and Specification

MOTION DETECTORS:

Motion sensors and detectors have been around since the seventies and are designed to provide effective security to homeowners. Although technological advancements have improved the functioning of motion detectors, the basic purpose and system of this device remains the same. A motion detector is basically designed to detect motion or movement in a designated area and may instantly send a signal to the security alarm system.



A motion detector is a device that detects moving objects, particularly people. Such a device is often integrated as a component of a system that automatically performs a task or alerts a user of motion in an area. They form a vital component of security, automated lighting control, home control, energy efficiency, and other useful systems.

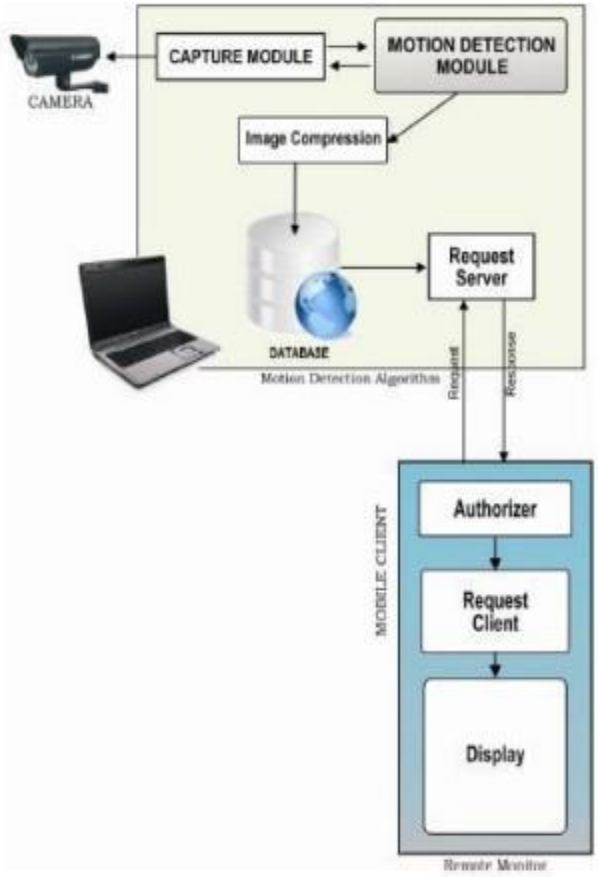
How motion detection cameras work:

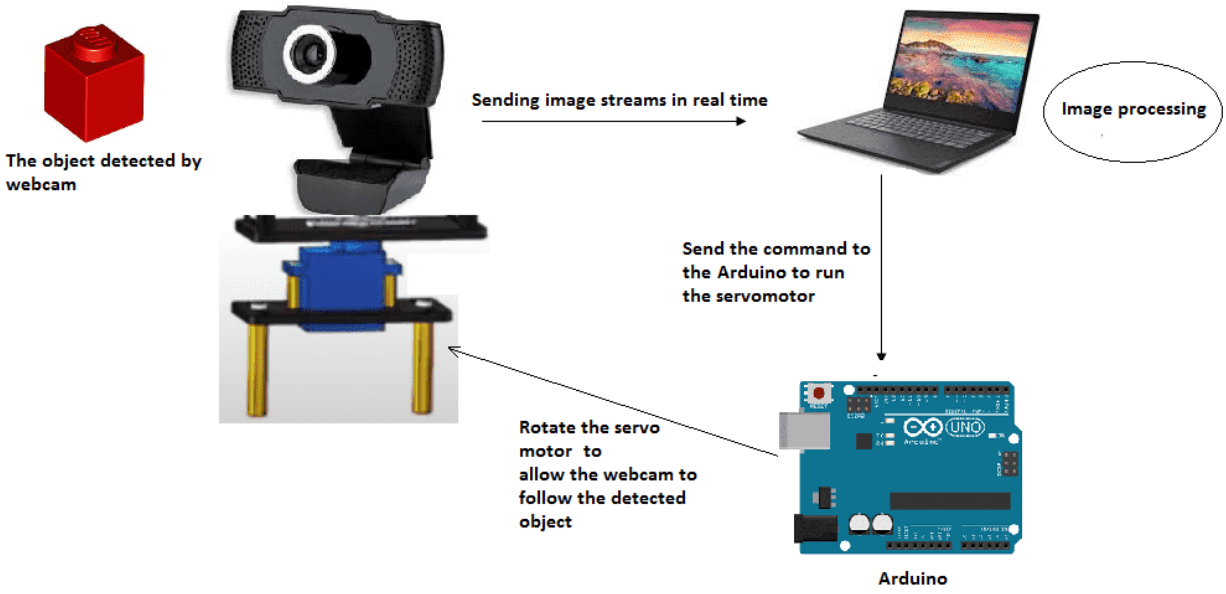
Monitoring your data centers and building access points with CCD cameras can represent a boon to your overall security, but you need to know a little about camera terminology to make the best use of them. Don't think about "aim and shoot" when you think about these cameras. Think, instead, about how you might go about detecting motion in a series of still images. First off, let's examine the term "CCD". This stands for "charge-coupled device". A CCD is a silicon chip with a surface that is divided into light-sensitive pixels. When light hits these pixels, tiny electric charges are generated. With enough of these pixels, you can get a high-resolution image. With adequate "sensitivity" you can detect motion even in a dimly lit room.

The difference between digital cameras and security cameras is in their basic way of working. Digital cameras store images when you tell them to. Security cameras only provide images (save and transmit them) when they detect motion.

And motion detection is basically the process of comparing sequential images and determining whether the differences between them represent motion. If there are significant differences between two consecutive images, the cameras "conclude" that there has been motion within the camera view.

3.4 SYSTEM ARCHITECTURE:





3.5 System Requirements:

HARDWARE REQUIREMENTS:

- **System** : MINIMUM i3.
- **Hard Disk** : 40 GB.
- **RAM** : 4 GB.

SOFTWARE REQUIREMENTS:

- **Operating System:** Windows 8,10 & 11
- **Coding Language**: Python 3.7

3.6 System Study

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and the business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

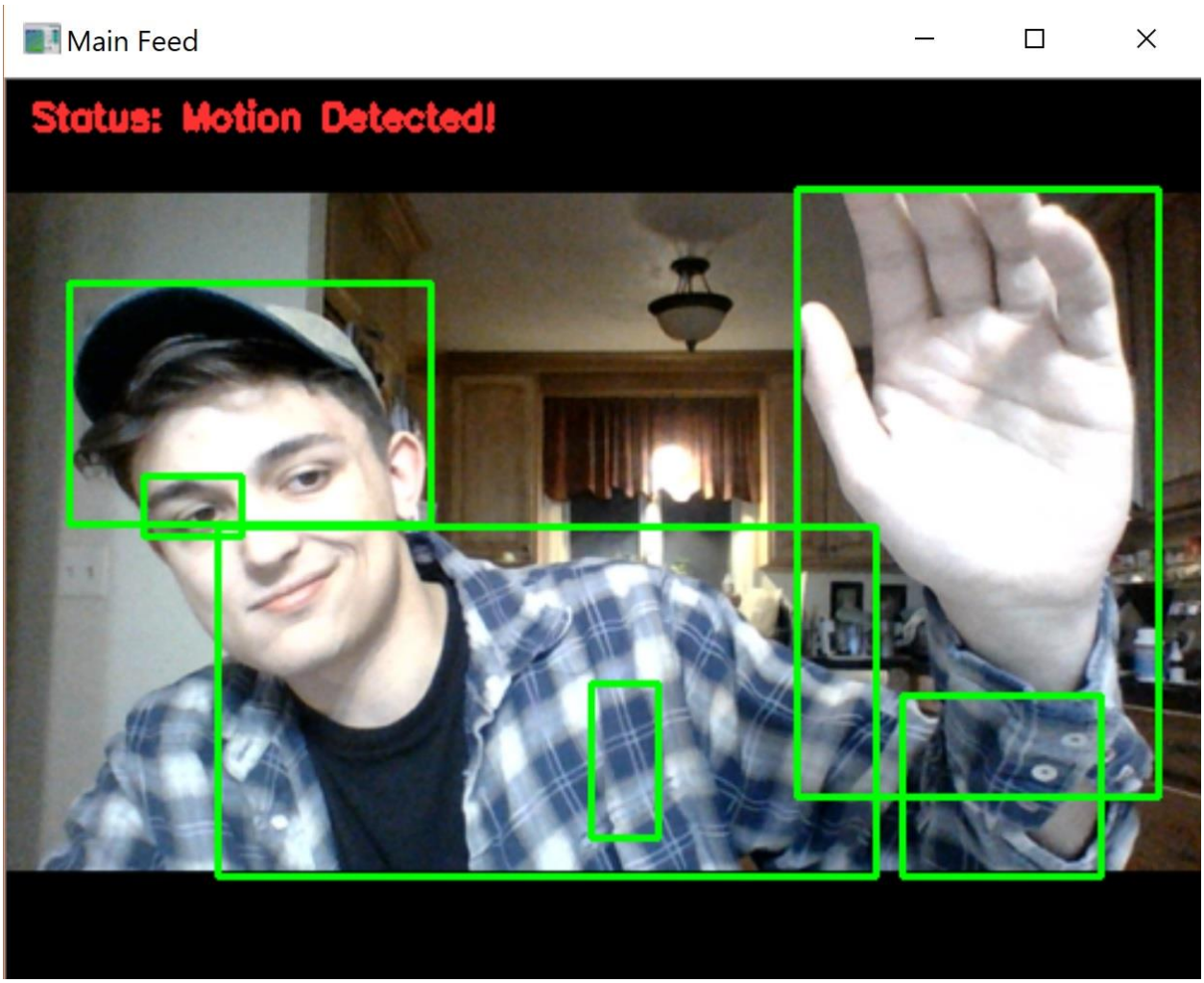
This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditure must be justified. Thus, the developed system is well within the budget, and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. No system developed must not have a high demand for the available technical resources. This will lead to high demand for the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.



CHAPTER 4

SYSTEM DESIGN

4.1 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

UML is a very important part of developing objects-oriented software and the software development process. UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.

6. Support higher level development concepts such as collaborations, frameworks, patterns, and components.
7. Integrate best practices.

4.1.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

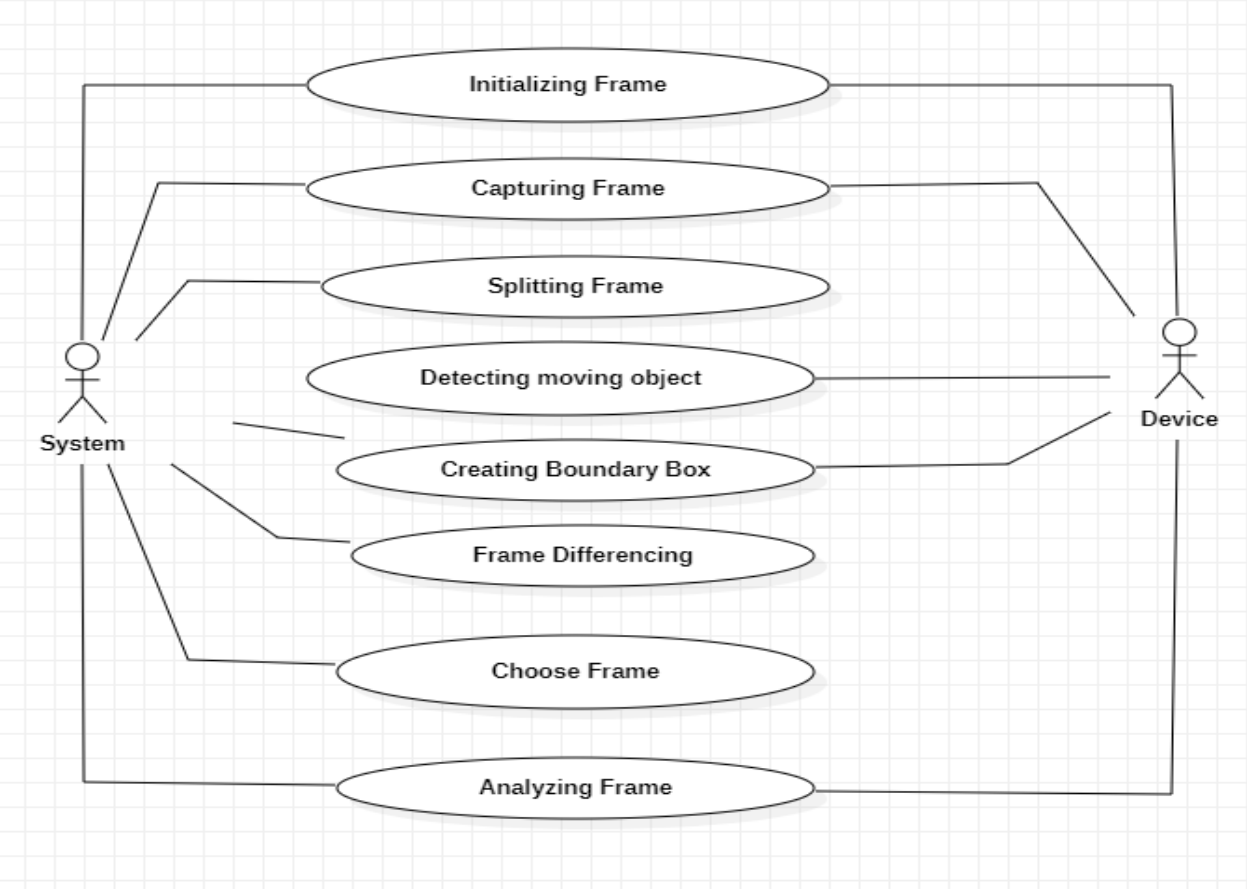


Fig no: 4.1.1 Use Case Diagram

4.1.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

In the diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centered, and the firstletter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned, and the firstletter is lowercase.
- The bottom compartment contains the operations the class can execute. They are also left-aligned,and the first letter is lowercase.

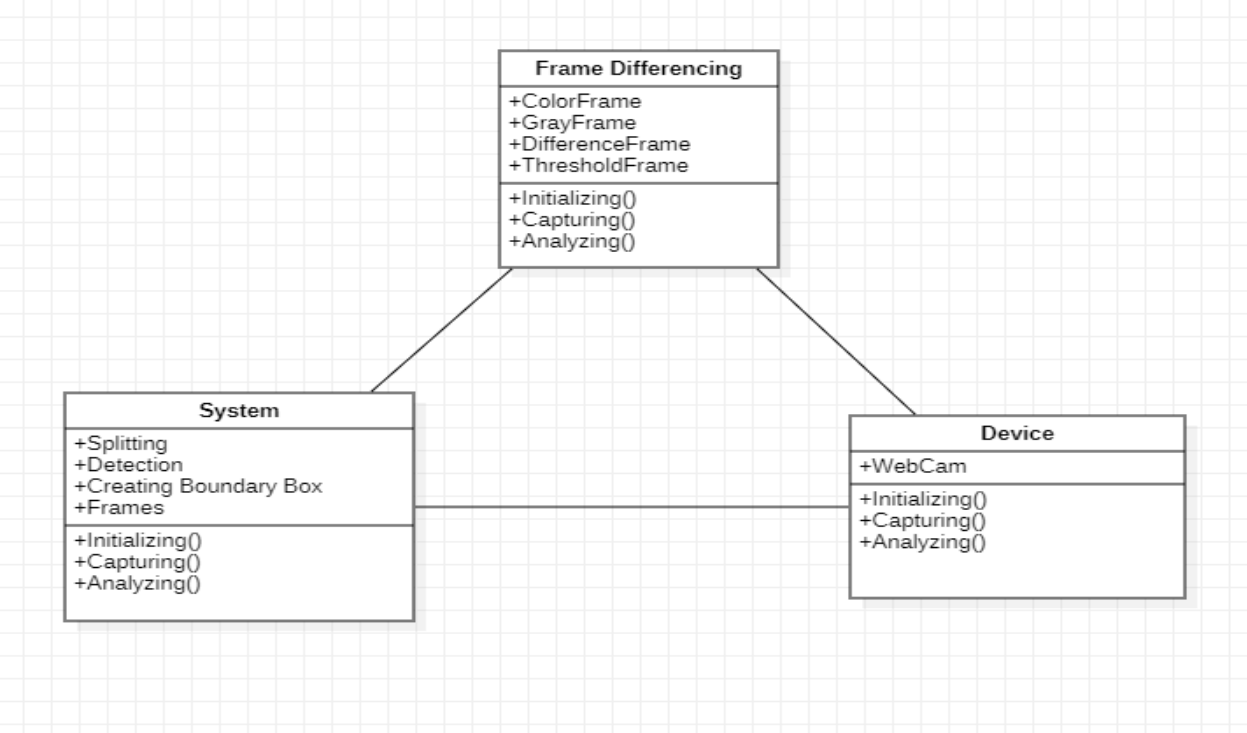


Fig no: 4.1.2 Class Diagram

4.1.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

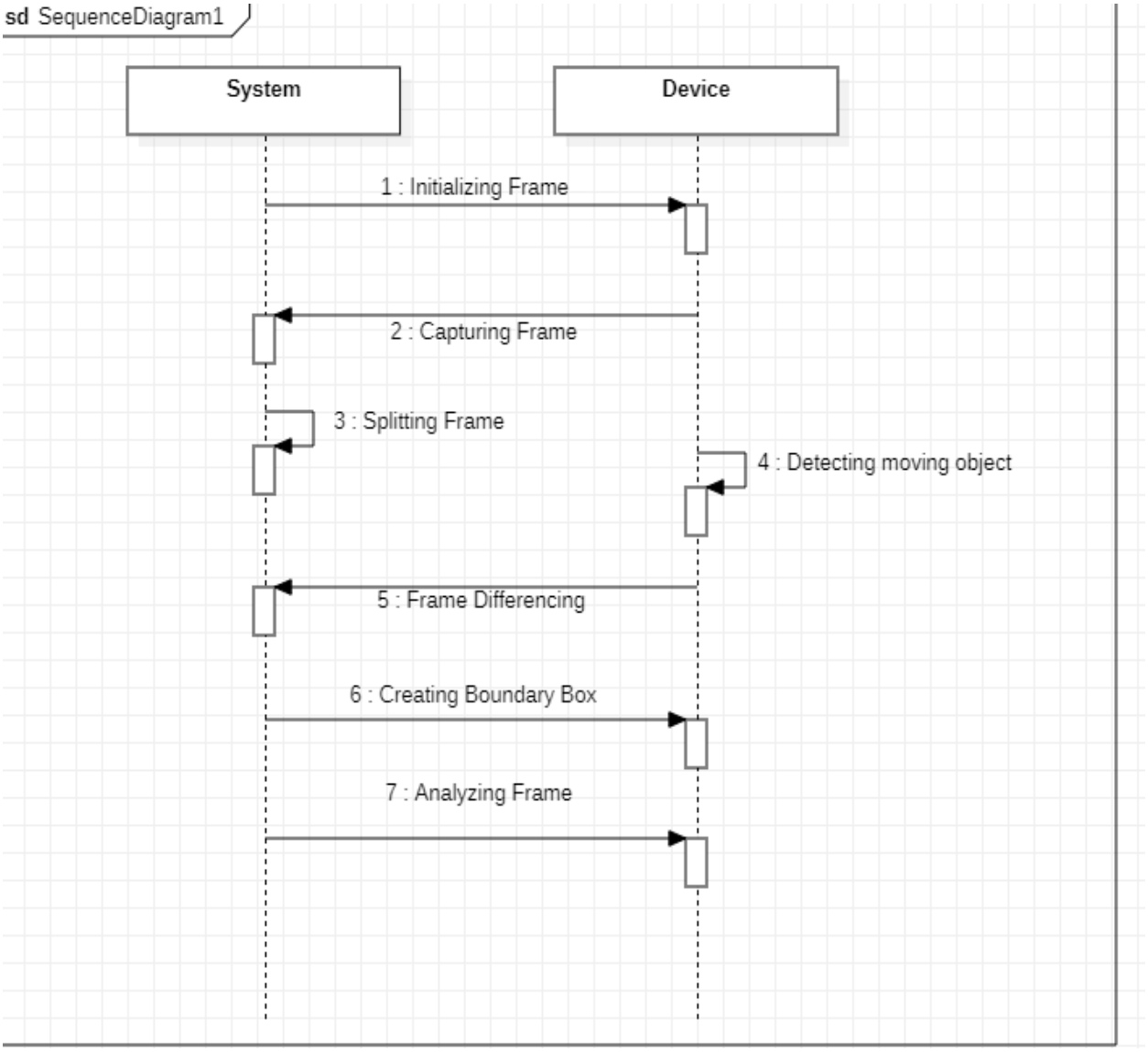
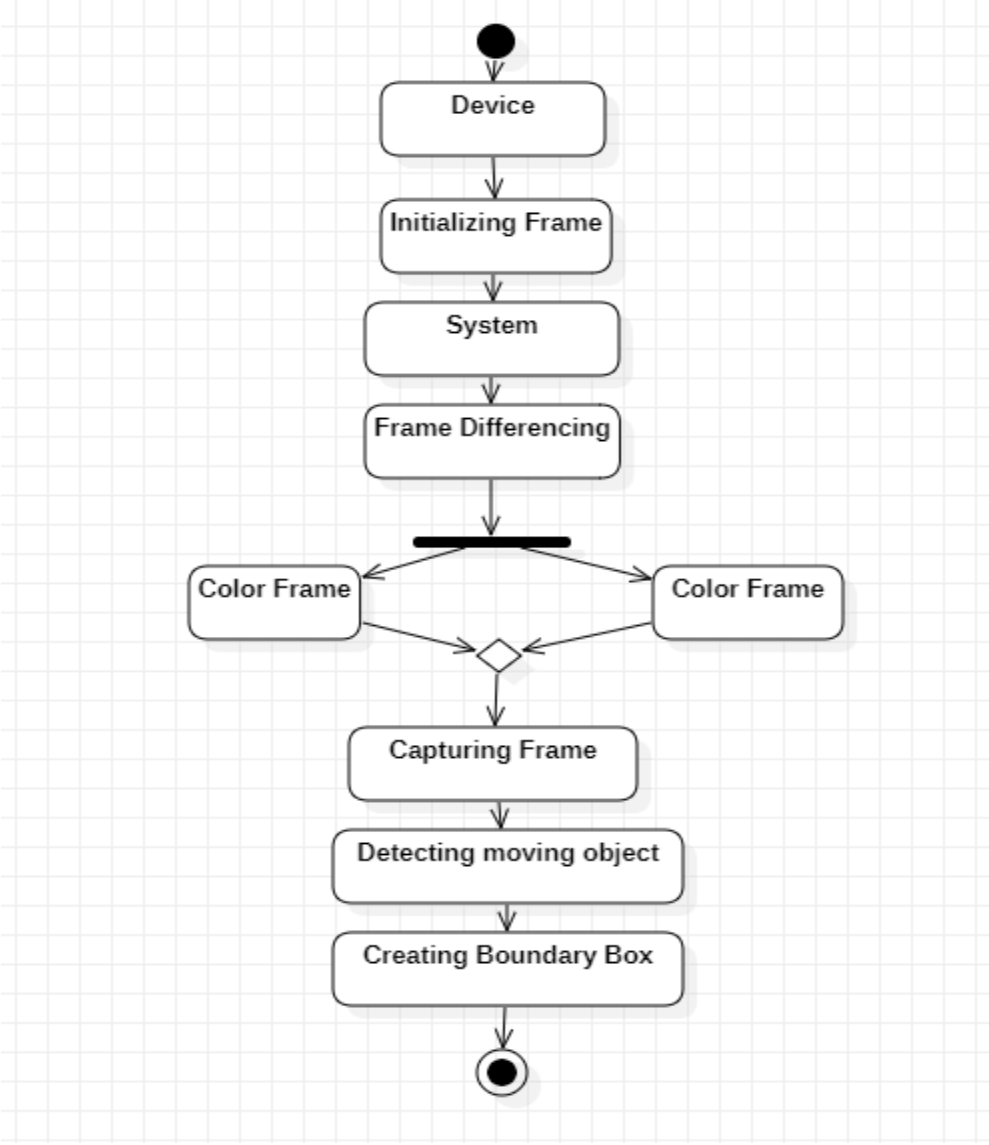


Fig no:4.1.3 Sequence Diagram

4.1.4 ACTIVITY DIAGRAM:

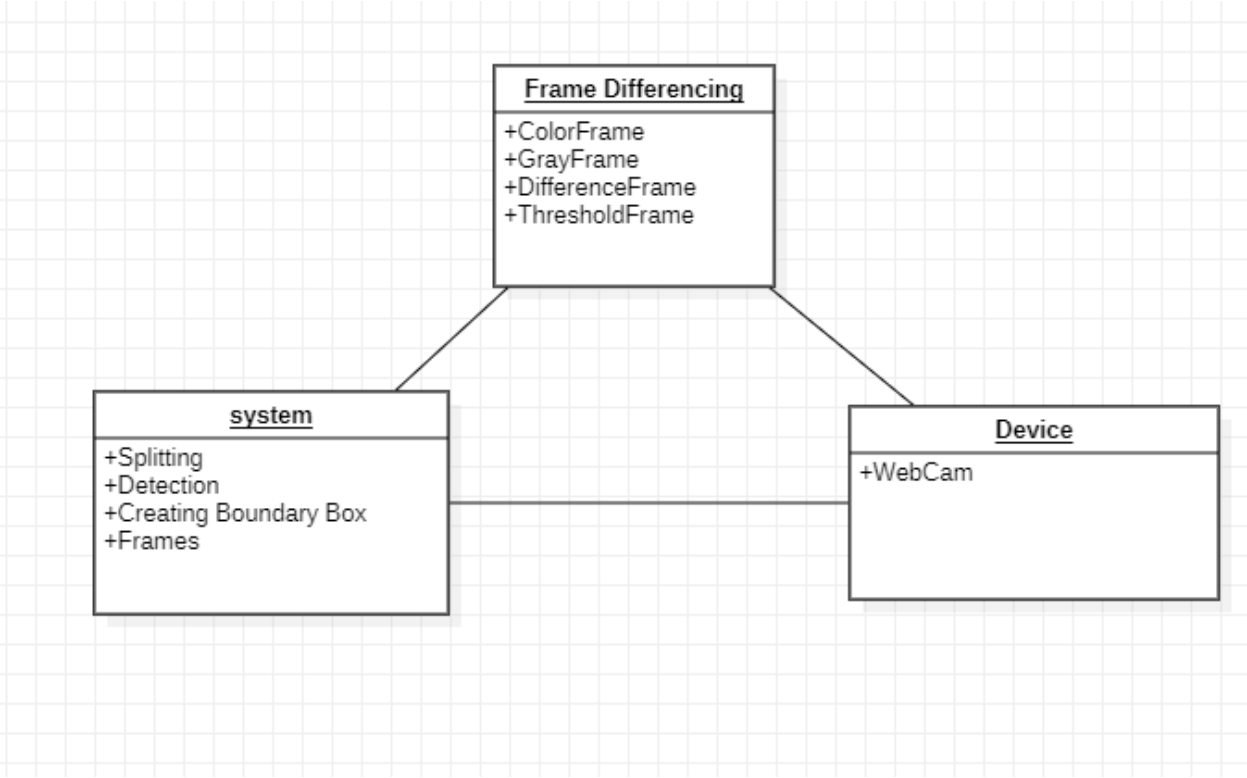
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



4.1.4 ACTIVITY DIAGRAM

4.1.5 Object Diagram:

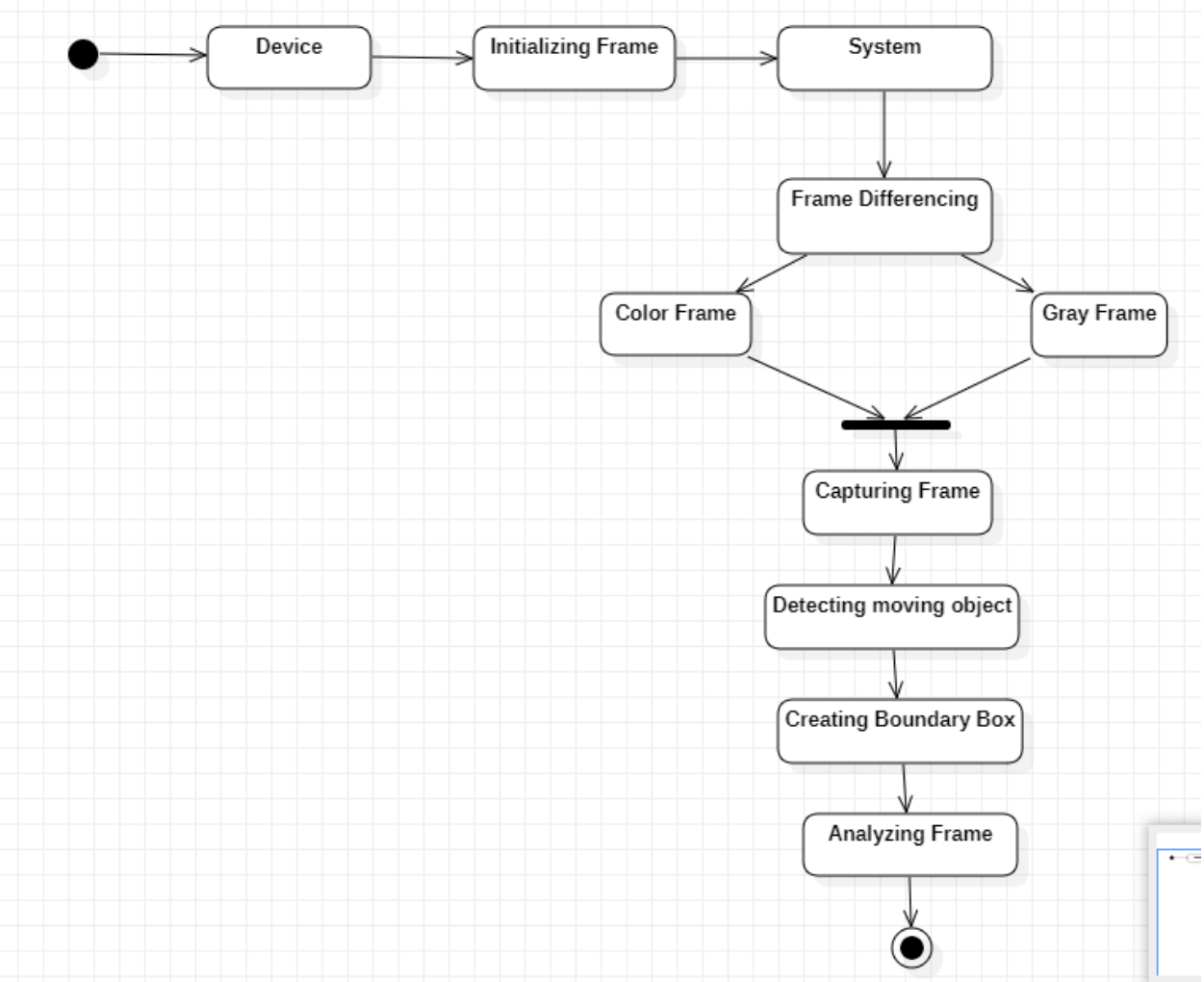
An object diagram is a UML structure diagram that shows the instance of the classifier in model. Object diagrams use notation that is like that used in class diagram. Class diagram shows the actual classifier and their relationship in a system.



4.1.5 Object Diagram

4.1.6 StateChart Diagram:

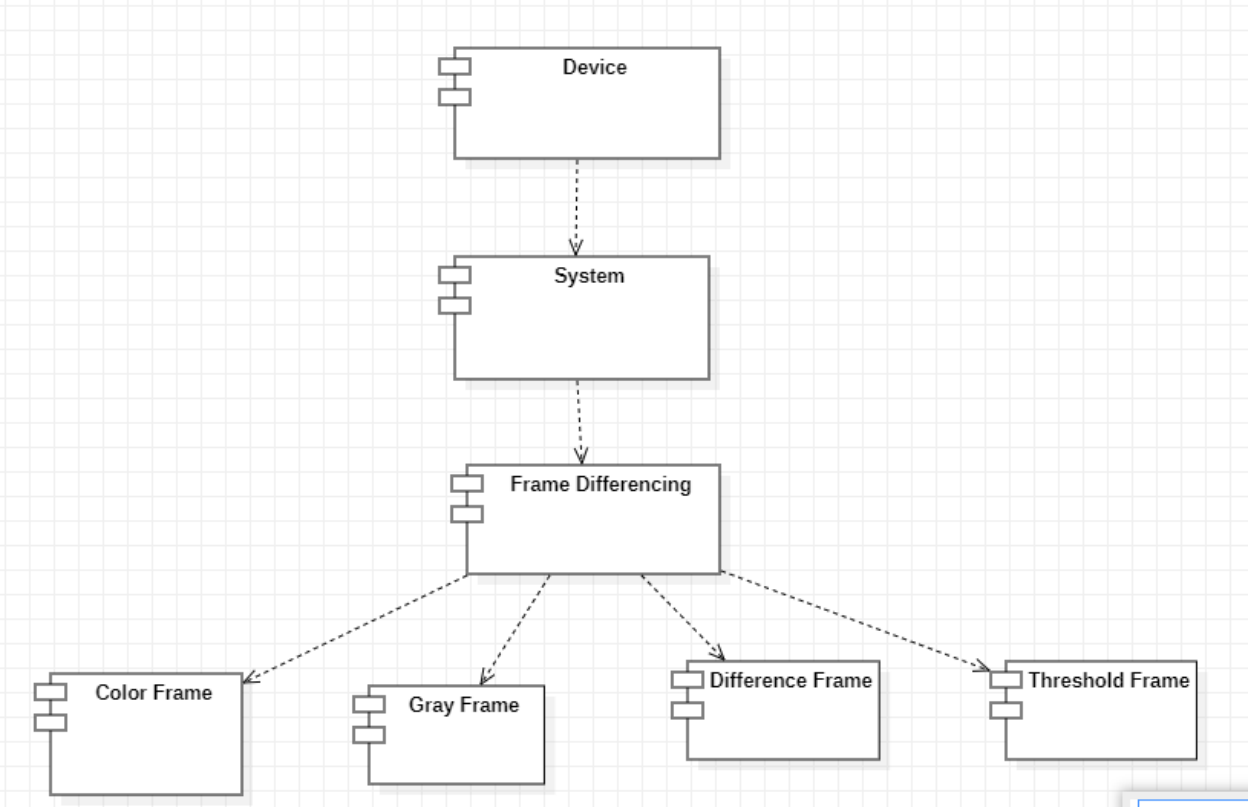
A Statechart diagram describes a state machine. A state machine can be defined as a machine which defines different states of an object, and these states are controlled by external or internal events.



4.1.6 StateChart Diagram

4.1.7 Component Diagram:

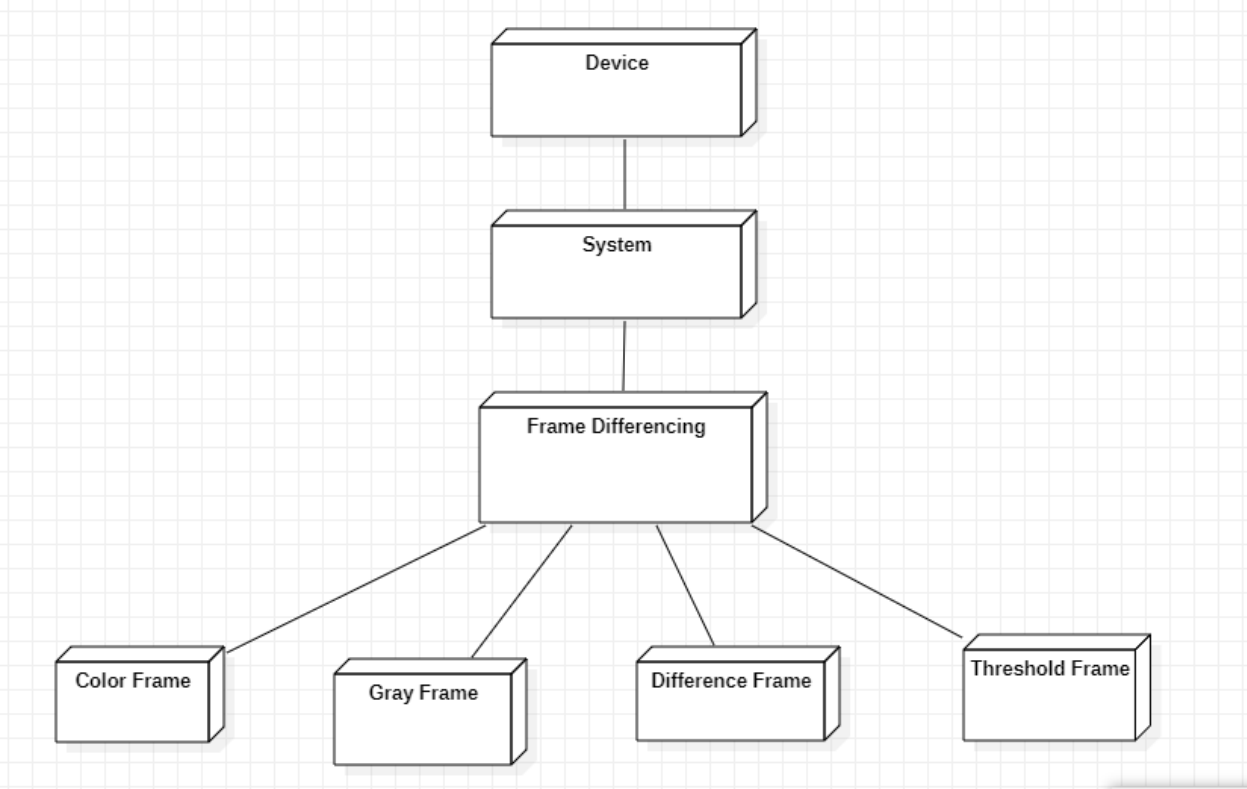
A component diagram, also known as a UML component diagram, describes the organization andwiring of the physical components in a system. Component diagrams are often drawn to help model implementation detail and double-check that every aspect of the system's required functions is covered by planned development.



4.1.7 Component Diagram

4.1.8 Deployment Diagram:

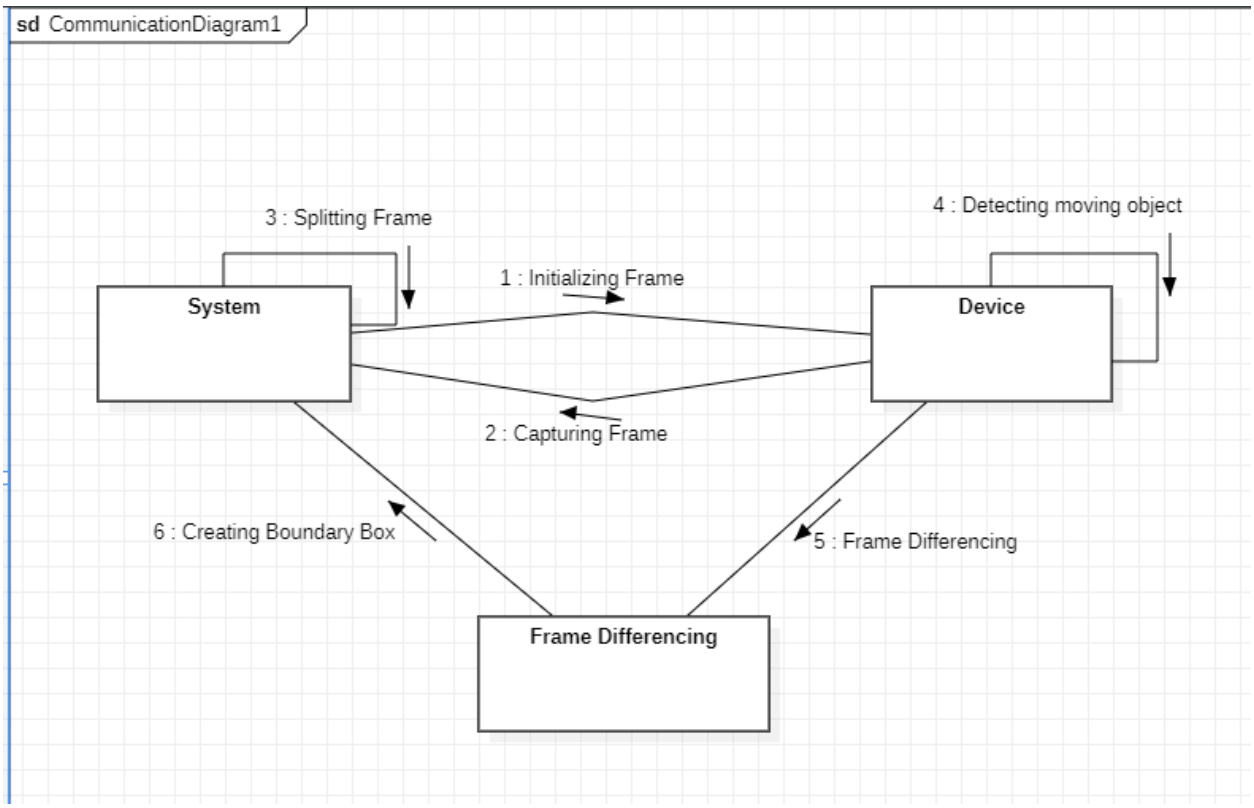
A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environment, and the middleware connecting/algorithm. Deployment diagrams are typically used to visualize the physical hardware and software of a system.



4.1.8 Deployment Diagram

4.1.9 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.



4.1.9 Collaboration Diagram

CHAPTER 5

SOFTWARE ENVIRONMENTS

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos, and other visual inputs — and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand. Computer vision works much the same as human vision, except humans have a head start. Human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving and whether there is something wrong in an image.

Object detection is a computer vision technique that works to identify and locate objects within an image or video. Specifically, object detection draws bounding boxes around these detected objects, which allow us to locate where said objects are in (or how they move through) a given scene. Object detection is commonly confused with image recognition, so before we proceed, it's important that we clarify the distinctions between them. In more traditional ML-based approaches, computer vision techniques are used to look at various features of an image, such as the color histogram or edges, to identify groups of pixels that may belong to an object.

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy. Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

Python is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is dynamically typed and garbage-collected programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python supports multiple programming paradigms, including Procedural, Object Oriented and Functional programming languages. Python design philosophy emphasizes code readability with the use of significant indentation. Python is consistently rated as one of the world's most popular programming languages. Python is easy to learn, so if you are starting to learn any programming language then Python could be your great choice.

5.1 Computer Vision

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos, and other visual inputs — and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand.

Fig no:5.1 Computer Vision

Computer vision works much the same as human vision, except humans have a head start. Human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving and whether there is something wrong in an image.

Computer vision trains machines to perform these functions, but it must do it in much less time with cameras, data, and algorithms rather than retinas, optic nerves, and a visual cortex. Because a system trained to inspect products or watch a production asset can analyze thousands of products or processes a minute, noticing imperceptible defects or issues, it can quickly surpass human capabilities.

5.1.1 Working of Computer Vision

Computer vision needs lots of data. It runs analyses of data over and over until it discerns distinctions and ultimately recognizes images. For example, to train a computer to recognize automobile tires, it needs to be fed vast quantities of tire images and tire-related items to learn the differences and recognize a tire, especially one with no defects.

Computer vision is like solving a jigsaw puzzle in the real world. Imagine that you have all these jigsaw pieces together and you need to assemble them to form a real image. That is exactly how the neural networks inside a computer vision work. Through a series of filtering and actions, computers can put all the parts of the image together and then think on their own.

However, the computer is not just given a puzzle of an image - rather, it is often fed with thousands of images that train it to recognize certain objects. Machine learning uses algorithmic models that enable a computer to teach itself about the context of visual data. If enough data is fed through the model, the computer will “look” at the data and teach itself to tell one image from another.

5.1.2 Evolution of Computer Vision

Before the advent of deep learning, the tasks that computer vision could perform were very limited and required a lot of manual coding and effort by developers and human operators.

For instance, if you wanted to perform facial recognition, you would have to perform the following steps:

- **Create a database**: You had to capture individual images of all the subjects you wanted to track in a specific format.
- **Annotate images**: Then for every individual image, you would have to enter several key data points, such as distance between the eyes, the width of nose bridge, distance between upper-lip and nose, and dozens of other measurements that define the unique characteristics of each person.
- **Capture new images**: Next, you would have to capture new images, whether from photographs or video content. And then you had to go through the measurement process again, marking the key points on the image. You also had to factor in the angle the image was taken.

After all this manual work, the application would finally be able to compare the measurements in the new image with the ones stored in its database and tell you whether it corresponded with any of the profiles it was tracking. In fact, there was very little automation involved and most of the work was being done manually.

5.1.3 Scope of Computer Vision

Computer vision is a multidisciplinary field that focuses on enabling computers to understand and interpret visual data from images or videos. It encompasses a wide range of tasks and applications, and its scope continues to expand as technology advances. Here are some key areas within the scope of computer vision:

- Object detection and recognition: This involves identifying and localizing specific objects or patterns within an image or video stream. Object detection algorithms can be used for various applications, such as autonomous driving, surveillance, and augmented reality.
- Image classification: Computer vision techniques can be employed to classify images into different categories or classes. This has numerous applications, including content-based image retrieval, medical imaging analysis, and quality control in manufacturing.
- Semantic segmentation: In semantic segmentation, the goal is to assign a semantic label to each pixel in an image, effectively dividing the image into meaningful regions. This is useful for tasks such as image understanding, scene understanding, and autonomous navigation.
- Pose estimation: Pose estimation involves estimating the spatial position and orientation of objects or humans within an image or video. This can be applied in areas like robotics, motion tracking, and virtual reality.
- Tracking and motion analysis: Computer vision techniques can track the movement of objects or people over time in videos or image sequences. This has applications in video surveillance, sports analysis, and visual effects.
- Facial analysis and recognition: Computer vision algorithms can be employed to detect and analyze faces in images or videos, performing tasks like facial recognition, emotion analysis, and age estimation. These applications find use in security systems, user authentication, and social robotics.
- 3D reconstruction: Computer vision can be used to reconstruct the three-dimensional structure of objects or scenes from multiple images or depth sensors. This has applications in 3D modeling, virtual reality, and cultural heritage preservation.
- Visual understanding and scene understanding: The goal of visual understanding is to extract high-level meaning and context from visual data, enabling computers to comprehend scenes, objects, and actions depicted in images or videos. Scene understanding involves inferring the relationship between objects and their interactions within a scene.
- Medical image analysis: Computer vision plays a vital role in analyzing medical images, such as X-rays, MRIs, and histopathology slides. It aids in diagnosis, disease detection, treatment planning, and monitoring.

- Autonomous systems: Computer vision is crucial for enabling autonomous systems like self-driving cars and drones. It helps these systems perceive and interpret the surrounding environment to make informed decisions and navigate safely.

The scope of computer vision is vast and continually evolving with advancements in deep learning, sensor technologies, and computational power. These applications demonstrate the wide-ranging impact and potential of computer vision across various industries and domains.

Computer vision is a rapidly growing field in research and applications. Advances in computer vision research are now more directly and immediately applicable to the commercial world. AI developers are implementing computer vision solutions that identify and classify objects and even react to them in real time. Image classification, face detection, pose estimation, and optical flow are some of the typical tasks. Computer vision engineers are a subset of deep learning (DL) or machine learning (ML) engineers that program computer vision algorithms to accomplish these tasks.

5.1.4 Computer Vision Tasks

The computer vision tasks necessary for understanding cellular dynamics include cell segmentation and cell behavior understanding, involving cell migration tracking, cell division detection, cell death detection, and cell differentiation detection.

- **Image classification** sees an image and can classify it (a dog, an apple, a person's face). More precisely, it can accurately predict that a given image belongs to a certain class. For example, a social media company might want to use it to automatically identify, and segregate objectionable images uploaded by users.
- **Object detection** can use image classification to identify a certain class of image and then detect and tabulate their appearance in an image or video. Examples include detecting damage on an assembly line or identifying machinery that requires maintenance.
- **Object tracking** follows or tracks an object once it is detected. This task is often executed with images captured in sequence or real-time video feeds. Autonomous vehicles, for example, need to not only classify and detect objects such as pedestrians, other cars, and

road infrastructure, they need to track them in motion to avoid collisions and obey traffic laws.

- **Content-based image retrieval** uses computer vision to browse, search and retrieve images from large data stores, based on the content of the images rather than metadata tags associated with them. This task can incorporate automatic image annotation that replaces manual image tagging. These tasks can be used for digital asset management systems and can increase the accuracy of search and retrieval.

5.1.5 Computer Vision Benefits

Computer vision can automate several tasks without the need for human intervention. As a result, it provides organizations with a few benefits:

- **Faster and simpler process** - Computer vision systems can carry out repetitive and monotonous tasks at a faster rate, which simplifies the work for humans.
- **Better products and services** - Computer vision systems that have been trained very well will make zero mistakes. This will result in faster delivery of high-quality products and services.
- **Cost-reduction** - Companies do not have to spend money on fixing their flawed processes because computer vision will leave no room for faulty products and services.

5.1.6 Advantages of Computer Vision

The use of Computer Imagining grows rapidly thanks to the discovery of advantages for industries. There are five main advantages of computer vision:

- **Process in a simpler and faster way**: it allows the clients and industries to check. Also, it gives them access to their products. It's possible thanks to the existence of Computer Vision in fast computers.
- **Reliability**: computers and cameras don't have the human factor of tiredness, which is eliminated in them. The efficiency is usually the same, it doesn't depend on external factors such as illness or sentimental status.

- **Accuracy**: the precision of Computer Imagining, and Computer Vision will ensure better accuracy on the final product.
- **A wide range of uses**: We can see the same computer system in several different fields and activities. Also, in factories with warehouse tracking and shipping of supplies, and in the medical industry through scanned images, among other multiple options.
- **The reduction of costs**: time and error rate are reduced in the process of Computer Imagining. It reduces the cost of hiring and training special staff to do the activities that computers will do, as hundreds of workers.

5.1.7 Disadvantages of Computer Vision

Despite all the advantages of computer vision thanks to the capacity of Machine Learning, we must consider some disadvantages:

- **Necessity of specialists**: there is a huge necessity of specialists related to the field of Machine Learning and Artificial Intelligence. A professional that knows how those devices work and takes full advantage of Computer Vision. Also, the person can repair them when necessary. There are a lot of work opportunities after doing a Master in Artificial Intelligences. However, companies still wait for those specialists.
- **Spoiling**: eliminating the human factor may be good in some cases. But when the machine or device fails, it doesn't announce or anticipate that problem. Whereas a human person can tell in advance when the person won't come.
- **Failing in image processing**: when the device fails because of a virus or other software issues, it is highly probable that Computer Vision and image processing will fail. But if we do not solve the problem, the functions of the device can disappear. It can freeze the entire production in the case of warehouses.

5.1.8 Applications

As per the increasing demand for AI and Machine Learning technologies, computer vision also has a great demand among different sectors. It has a massive impact on different industries, including retail, security, healthcare, automotive, agriculture, etc. Below are some most popular applications of computer vision:

- Defect detection using Computer Vision
- OCR using Computer vision.
- Crop Monitoring
- Analysis of X-rays, MRI, and CT scans using Computer Vision
- Road Condition Monitoring
- 3D model Building using Computer vision.
- Cancer Detection using Computer Vision
- Plant Disease Detection using Computer Vision
- Traffic Flow Analysis

5.2 Detecting movement of objects:

Detecting movement of objects with a webcam refers to the process of analyzing the video stream captured by a webcam to identify regions or objects that are undergoing motion. It involves comparing consecutive frames and detecting changes in pixel values, patterns, or optical flow to determine the presence and location of moving objects within the scene.

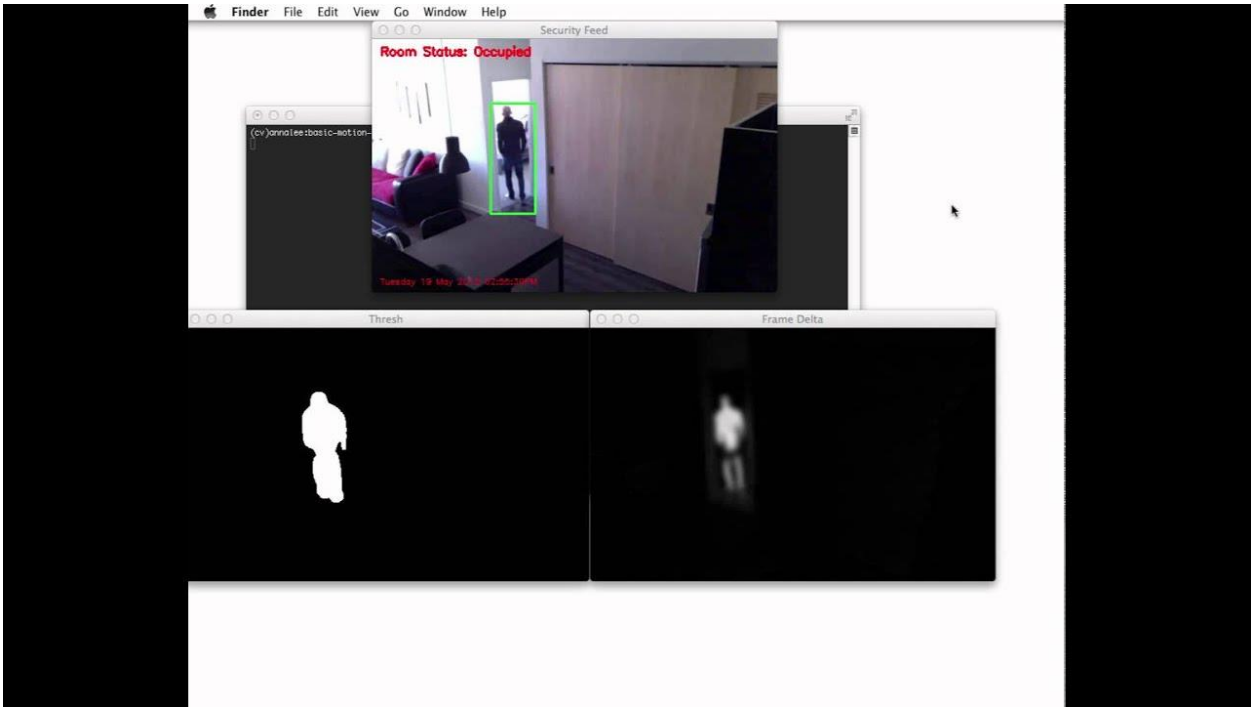
The objective is to differentiate between the static background and the moving objects within the video feed. Various techniques can be employed for motion detection, including frame differencing, background subtraction, and optical flow analysis. These methods utilize algorithms and image processing techniques to highlight regions where motion occurs, typically by generating a binary mask or highlighting the moving areas within the video frames.

By detecting and tracking movement, applications can be developed for various purposes, such as surveillance systems, gesture recognition, human-computer interaction, activity monitoring, and more. The accuracy and effectiveness of motion detection algorithms depend on factors like lighting conditions, camera quality, object speed, and the complexity of the scene. Therefore, appropriate adjustments, parameter tuning, and algorithm selection are essential to achieve reliable and precise motion detection with a webcam.

There are several approaches and techniques for detecting movement of objects with a webcam. Here are three commonly used methods:

1. **Frame differencing:** This method involves calculating the pixel-wise absolute difference between consecutive frames. The resulting difference image highlights areas with significant changes in pixel values, indicating motion. By applying thresholding and morphological operations, you can identify and extract moving objects. However, this method is sensitive to changes in lighting conditions and can produce false positives due to noise or small variations.
2. **Background subtraction:** This technique assumes that the background in a scene remains relatively static while objects in motion stand out. It involves creating a background model from a set of initial frames or an adaptive approach that updates the background over time. The current frame is then compared to the background model, and the differences indicate moving objects. Background subtraction is robust to lighting changes and can handle gradual variations in the scene. However, it may struggle with complex scenes or persistent moving backgrounds.
3. **Optical flow:** Optical flow methods estimate the motion of pixels between consecutive frames by analyzing the displacement of image features. These methods calculate the direction and speed of motion for each pixel, providing a dense motion field. By thresholding or analyzing the optical flow vectors, you can detect moving objects. Optical flow is effective for tracking smooth motion and can handle camera movements, but it may struggle with occlusions or fast-moving objects.

It's important to note that these methods can be combined or extended with additional techniques to improve accuracy and handle specific scenarios. For example, you can incorporate object tracking algorithms, machine learning-based object detection models, or deep learning approaches for more advanced and robust motion detection and object recognition. The choice of method depends on the specific requirements, environment, and complexity of the motion detection task.



5.2.1 Importance of Detecting movement of objects:

Detecting movement of objects with a webcam holds significant importance in various domains and applications. Here are some key reasons why it is important:

1. **Security and Surveillance:** Webcam-based motion detection is widely used in security and surveillance systems. By monitoring and detecting movement in specific areas, it helps

identify potential intrusions, suspicious activities, or unauthorized access. Motion detection alerts can be sent to security personnel, triggering immediate response and ensuring the safety of premises.

2. **Intrusion Detection:** Motion detection is crucial for identifying and detecting intrusions in restricted areas. It can be used to secure sensitive locations, such as laboratories, data centers, or high-security facilities, by triggering alarms or notifying security personnel when unauthorized individuals or objects enter the protected area.
3. **Object Tracking:** By detecting and tracking the movement of objects, webcams can assist in tracking valuable assets, monitoring inventory, or observing the movement of goods in logistics and warehousing. This enables better control over the flow of items, prevents theft or loss, and streamlines inventory management processes.
4. **Activity Monitoring and Analysis:** Motion detection with webcams finds applications in activity monitoring and analysis. It can be used in various contexts, such as monitoring patient movements in healthcare facilities, analyzing customer behavior in retail environments, or tracking user gestures for interactive applications. By understanding patterns of movement, valuable insights can be gained for improving processes, enhancing user experiences, and ensuring safety.
5. **Home Automation and Smart Environments:** Motion detection is utilized in home automation systems to trigger actions based on movement. For example, lights can be automatically turned on when a person enters a room, or heating and cooling systems can be adjusted based on occupancy. This enhances energy efficiency, convenience, and comfort in smart homes.
6. **Video Conferencing and Collaboration:** Motion detection can be employed in video conferencing and collaboration tools to enhance the user experience. It can be used to automatically focus on the active speaker or adjust the display layout based on the presence and movement of participants, improving the overall engagement and efficiency of remote meetings.

Overall, the ability to detect movement with a webcam opens a range of applications that contribute to security, efficiency, automation, and enhanced user experiences across various domains. It enables real-time monitoring, analysis, and automation based on the detected motion, providing valuable insights, and facilitating informed decision-making.

5.2.2 Scope of Detecting movement of objects:

The scope of detecting movement of objects with a webcam is quite broad and encompasses various domains and applications. Here are some areas where motion detection with a webcam is commonly applied:

1. **Surveillance and security:** Webcam-based motion detection is widely used in security systems for detecting intruders or suspicious activities. It can trigger alarms, send notifications, or initiate recording when motion is detected within the camera's field of view.
2. **Human-computer interaction:** Motion detection with a webcam enables gesture recognition and tracking, allowing users to interact with computers or devices using hand movements, body gestures, or facial expressions. This technology is utilized in applications such as gaming, virtual reality, augmented reality, and touchless interfaces.
3. **Activity monitoring and automation:** By detecting movement within a monitored area, webcams can be employed to monitor human activities or specific object interactions. This is useful in applications like smart homes, elderly care, industrial automation, or retail analytics, where motion detection can trigger actions, provide insights, or assist in managing resources efficiently.
4. **Video analytics and object tracking:** Webcam-based motion detection forms the foundation for more advanced video analytics tasks, such as object tracking, behavior analysis, and anomaly detection. By combining motion detection with object recognition and tracking algorithms, webcams can track specific objects or individuals within a scene.

5. **Augmenting video conferencing:** Motion detection in webcams can enhance video conferencing experiences by automatically adjusting camera angles, zoom levels, or screen layouts based on the participants' movements. This helps keep the focus on active speakers or improves the framing of the video feed.
6. **Education and research:** Motion detection with webcams finds applications in educational settings for tracking movements in sports, analyzing body movements in dance or fitness training, or studying human behavior in research projects.

The scope of detecting movement with webcams is constantly expanding, and new applications continue to emerge as technology advances. From security to interactive experiences, automation, and beyond, motion detection with webcams offers a versatile range of possibilities across various fields.

5.2.3 Advantages of Detecting movement of objects:

Detecting movement of objects with a webcam offers several advantages, making it a valuable technology in various applications. Here are some of the key advantages:

1. **Cost-effective:** Webcams are widely available and relatively inexpensive compared to specialized motion detection equipment or sensors. This makes webcam-based motion detection an affordable solution for many individuals, businesses, and organizations.
2. **Accessibility and ease of use:** Webcams are easy to set up and use, requiring minimal technical expertise. They are plug-and-play devices that can be connected to a computer or other devices with ease, allowing for quick deployment and integration into existing systems.
3. **Non-intrusive:** Webcam-based motion detection is non-intrusive, as it does not require physical contact with the objects being monitored. This makes it suitable for applications where maintaining the integrity or appearance of the environment is essential, such as in homes, offices, or public spaces.
4. **Flexibility and adaptability:** Webcams offer flexibility in terms of placement and coverage. They can be positioned at various angles and heights, adjusted or moved to capture specific areas of interest, and even integrated into different devices or environments. This adaptability makes webcam-based motion detection suitable for diverse applications and scenarios.
5. **Real-time monitoring and responsiveness:** Webcam-based motion detection provides real-time monitoring, enabling immediate detection and response to motion events. This is crucial in applications such as security systems, where timely alerts or actions are required.
6. **Integration with other technologies:** Webcam-based motion detection can be easily integrated with other technologies and systems. It can be combined with image processing algorithms, object recognition, machine learning, or data analytics techniques to enhance the capabilities and accuracy of motion detection, enabling advanced functionalities such as object tracking, behavior analysis, or automation.
7. **Wide range of applications:** Webcam-based motion detection finds applications in various domains, including security, human-computer interaction, automation, video

analytics, education, and research. Its versatility allows it to be utilized in different fields and industries, catering to a wide range of needs and requirements.

Overall, the advantages of detecting movement with a webcam make it a practical and accessible solution for numerous applications, providing cost-effective, flexible, and reliable motion detection capabilities.

5.2.4 Disadvantages of Detecting movement of objects:

While detecting movement of objects with a webcam offers several advantages, there are also some limitations and potential disadvantages to consider:

1. **Limited field of view:** Webcams typically have a fixed field of view, which means they can only capture a specific area at a given time. If the monitored scene is large or requires coverage of multiple angles, multiple webcams or additional equipment may be needed, increasing the complexity and cost of the setup.
2. **Sensitivity to lighting conditions:** Webcams are sensitive to variations in lighting conditions, such as changes in ambient light, shadows, or glare. Sudden changes in lighting can result in false positives or false negatives in motion detection, affecting the accuracy and reliability of the system.
3. **False alarms:** Webcam-based motion detection can produce false alarms due to various factors. Minor movements or noise in the scene, such as moving leaves, shadows, or changes in lighting, can trigger false detections, leading to unnecessary alerts or actions. Fine-tuning of threshold values and implementing additional filtering techniques can help mitigate false alarms, but it requires careful optimization.
4. **Limited accuracy in complex scenes:** In scenes with complex or crowded backgrounds, distinguishing between background motion and the movement of specific objects can be challenging. The effectiveness of webcam-based motion detection may be reduced in such scenarios, potentially leading to inaccurate detection or tracking.

5. **Processing power and computational requirements:** Analyzing video frames in real-time for motion detection can be computationally intensive, especially when dealing with high-resolution video or complex algorithms. This may require a powerful computer system or dedicated hardware to handle the processing load efficiently.
6. **Privacy concerns:** Webcam-based motion detection involves capturing and processing video data, which raises privacy concerns, especially in applications involving surveillance or monitoring of private spaces. It is important to ensure that appropriate privacy measures and protocols are in place to protect sensitive information and comply with applicable regulations.
7. **Limited depth perception:** Webcams typically provide a 2D representation of the scene, lacking depth perception. This limitation can affect the accuracy of motion detection in scenarios where depth information is crucial, such as tracking objects at different distances or dealing with overlapping objects.

It's important to consider these disadvantages and limitations when implementing webcam-based motion detection systems, and make informed decisions based on the specific requirements and constraints of the application.

5.3 Python

5.3.1 Python Overview

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.

- **Python is Interactive:** You can sit at a Python prompt and interact with the interpreter directly to write your programs.

- **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python Features

Python's features include:

- **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.

- **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.

- **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

- **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

- **Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

- **Databases:** Python provides interfaces to all major commercial databases.

Some important points in python:

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.
- Programmers must type relatively less and indentation requirement of the language, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.
- The biggest strength of Python is huge collection of standard libraries which can be used for the following –

  - Machine Learning
  - GUI Applications (like Kivy, Tkinter, PyQt etc.)
  - Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  - Image processing (like Opencv, Pillow)
  - Web scraping (like Scrapy, BeautifulSoup, Selenium)
  - Test frameworks
  - Multimedia

5.3.2 Advantages of Python

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it contains code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be extended to other languages**.** You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn**,** understand**,** and code**.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code.
It also does not need curly braces to define blocks, and **indentation is mandatory.** These further aids the readability of the code.

8. Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once**,** and you can run it anywhere. This is called Write Once Run Anywhere (WORA)**.** However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

5.3.3 Advantages of Python Over Other Languages

1. Less Coding

Almost all the tasks done in Python require less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 GitHub annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac, or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning**,** automate things, do web scraping and also build games and powerful visualizations. It is an all-round programming language.

5.3.4 Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is rarely seen on the client-side**.** Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle.

The reason it is not so famous despite the existence of Python is that it isn't that secure.

3. Design Restrictions

As you know, Python is dynamically typed. This means that you don't need to declare the type of variable while writing the code. It uses duck-typing. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors**.**

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open Database Connectivity)**,** Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

5.3.5 History of Python

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked at that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python.

I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So, I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

5.4 Machine Learning

Before we look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is like the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

5.4.1 Categories of Machine Leaning:

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

5.4.2 Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate, and solve complex problems. On the other side, AI is still in its initial stage and hasn’t surpassed human intelligence in many aspects. Then the question is what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process.

These data-driven decisions can be used, instead of using programing logic, in problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but another aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

5.4.3 Challenges in Machines Learning:

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome a number of challenges. The challenges that ML is facing currently are −

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

5.4.4 Applications of Machines Learning

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

5.4.5 How to Start Learning Machine Learning?

Arthur Samuel coined the term “Machine Learning” in 1959 and defined it as a “Field of study that gives computers the capability to learn without being explicitly programmed”.
And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is The Best Job of 2019 with a *344%* growth and an average base salary of $146,085 per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it. So, this article deals with the Basics of Machine Learning and the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let's get started!!!

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

So, if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as Fork Python available Free on GeeksforGeeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

- **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
- **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- **Training –** The idea is to give a set of inputs(features) and its expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained in.
- **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- **Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
- **Unsupervised Learning –** This involves using unlabeled data and then finding the underlying structure in the data to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning –** This involves using unlabeled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning –** This involves learning optimal actions through trial and error. So, the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

5.4.6 Advantages of Machine learning

1. Easily identifies trends and patterns

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

2. No human intervention needed (automation)

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

3. Continuous Improvement

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

4. Handling multi-dimensional and multi-variety data

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

5. Wide Applications

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

5.4.7 Disadvantages of Machine Learning

1. Data Acquisition

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

2. Time and Resources

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

3. Interpretation of Results

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

4. High error-susceptibility

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

Python Development Steps: -

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release already included exception handling, functions, and the core data types of lists, dict, str and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting Unicode. Python flourished for another 8 years in versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it. Some changes in Python 7.3:

- Print is now a function.
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g., a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e., int. long is int as well.

- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behavior.
- Text Vs. Data Instead of Unicode Vs. 8-bit

Purpose: -

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type of system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional, and procedural, and has a large and comprehensive standard library.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.
- **Python is Interactive** − you can sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you must scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels.

All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

5.5 Modules Used in Project: -

5.5.1 Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

The most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all its products to improve the search engine, translation, image captioning or recommendations.

To give a concrete example, Google users can experience a faster and more refined search with AI. If the user types a keyword the search bar, Google provides a recommendation about what could be the next word.

Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

- Researchers
- Data scientists
- Programmers.

They can all use the same tool set to collaborate with each other and improve their efficiency.

Google does not just have any data; they have the world's most massive computer, so TensorFlow was built to scale. TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research.

It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++, or Java.

5.5.2 TensorFlow Architecture

Tensor flow architecture works in three parts:

- Pre-processing the data
- Build the model.
- Train and estimate the model.

It is called Tensor flow because it takes input as a multi-dimensional array, also known as tensors. You can construct a sort of flowchart of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output.

This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

Where can Tensor flow run?

TensorFlow can hardware, and software requirements can be classified into

1. Development Phase: This is when you train the mode. Training is usually done on your Desktop or laptop.
2. Run Phase or Inference Phase: Once training is done Tensorflow can be run on many different platforms. You can run it on

- Desktop running Windows, macOS or Linux.
- Cloud as a web service
- Mobile devices like iOS and Android

You can train it on multiple machines, then you can run it on a different machine, once you have the trained model.

The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python.

Finally, a significant feature of Tensor Flow is the Tensor Board. The Tensor Board enables to monitor graphically and visually what TensorFlow is doing.

5.5.3 List of Prominent Algorithms supported by TensorFlow.

- Linear regression: tf. estimator. Linear Regressor
- Classification: tf. Estimator. Linear Classifier
- Deep learning classification: tf. estimator. DNN Classifier
- Booster tree regression: tf. estimator. Boosted Trees Regressor
- Boosted tree classification: tf. estimator. Boosted Trees Classifier

5.6 Python's standard library

- Pandas
- Numpy
- Sklearn
- seaborn
- matplotlib

Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code.
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary datatypes can be defined using Numpy which allows Numpy to integrate with a wide variety of databases seamlessly and speedily.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tools using its powerful data structures. Python was majorly used for data munging and preparation. It made very little contribution towards data analysis.

Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object-oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn.

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type of system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional, and procedural, and has a large and comprehensive standard library.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.
- **Python is Interactive** − you can sit at a Python prompt and interact with the interpreter directly to write your programs.

  Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you must scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

5.7 Install Python Step-by-Step in Windows and Mac:

Python, a versatile programming language, doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac:

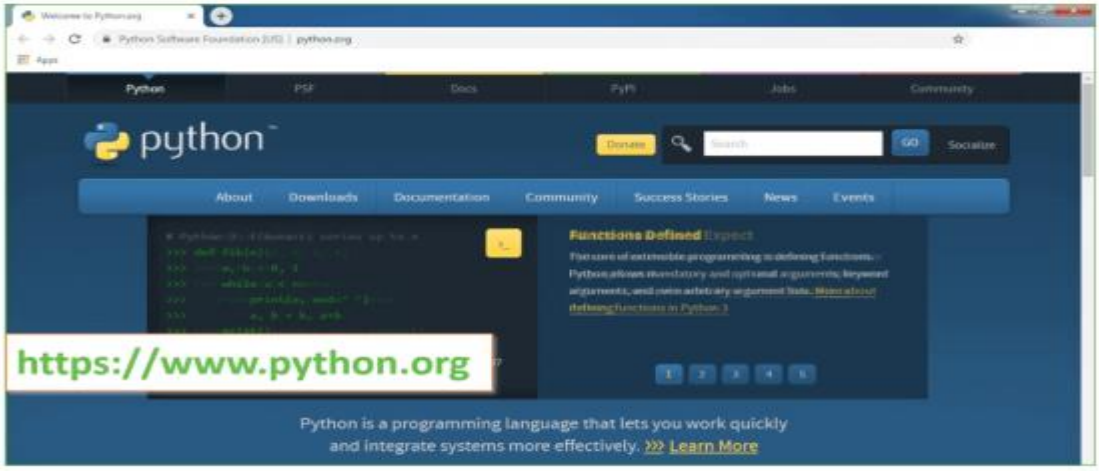
There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python, but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e., operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So, the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheat sheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system.

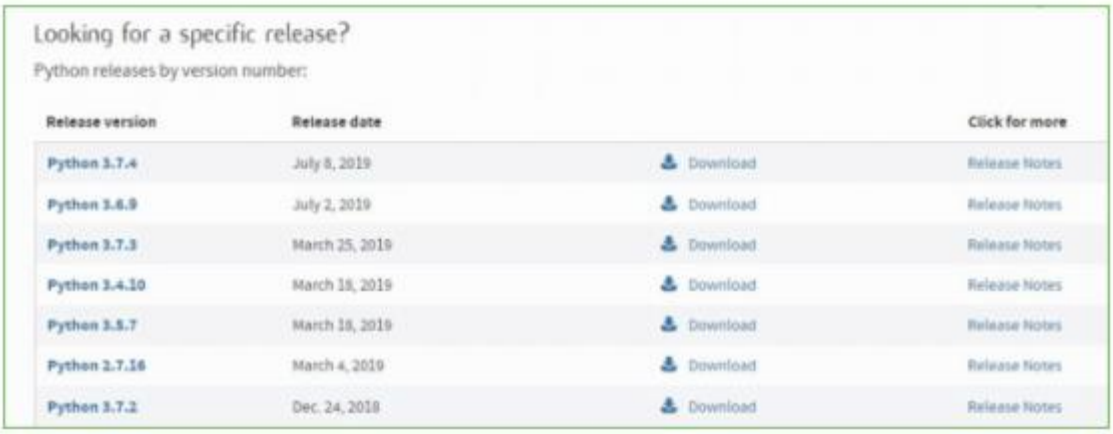
Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: **https://www.python.org**



Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cbabcee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a457738f5e4a936b241f	28082845	SIG
Windows help file	Windows		d639995573a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40726a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a383e563e00	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608b6d73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3bd1f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc802942a54446a3d8451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

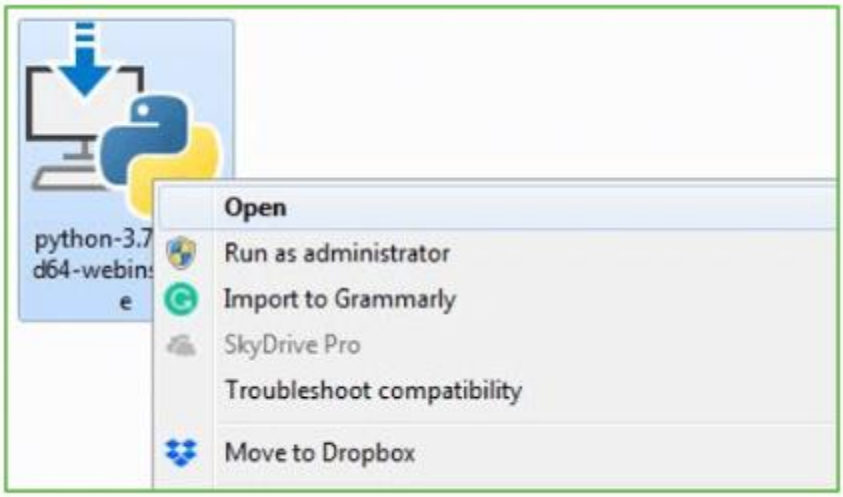
•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

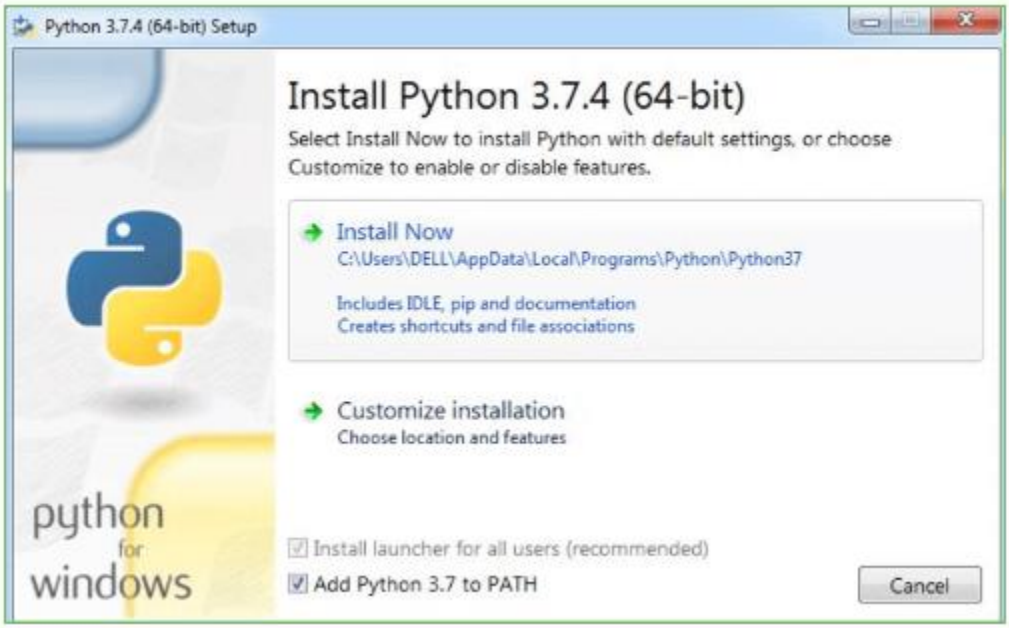
Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.



Step 3: Click on Install NOW After the installation is successful. Click on Close.

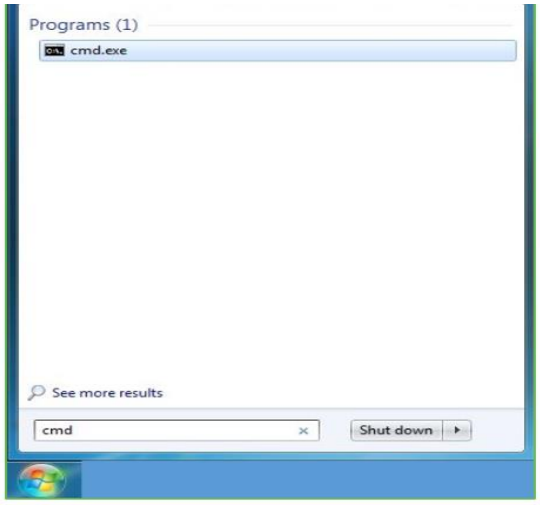
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

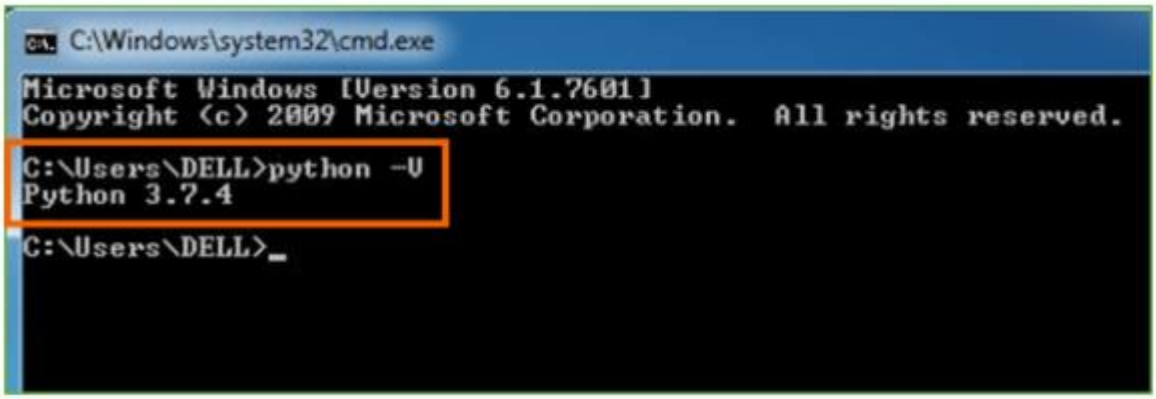
Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.



Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

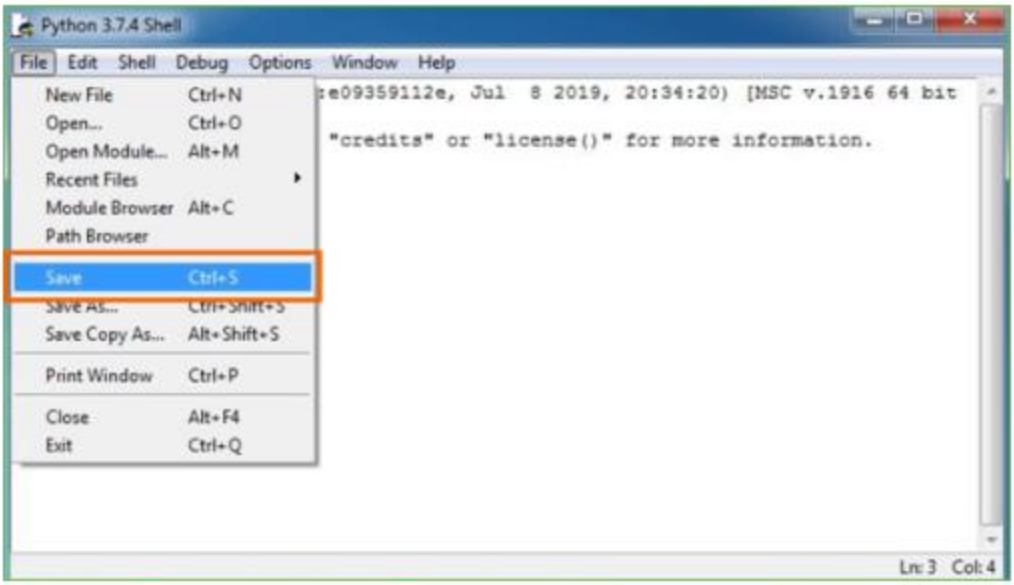
Check how the Python IDLE works.

Step 1: Click on Start

Step 2: In the Windows Run command, type "python idle".

Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program.

Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g., **enter print.**

5.8 How to Install OpenCV for Python on Windows?

OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy, python can process the OpenCV array structure for analysis. To Identify image patterns and its various features we use vector space and perform mathematical operations on these features.

To install OpenCV in Python, you can use the pip package manager. Here are the steps to install OpenCV in Python:

1. **Ensure that you have Python installed on your system:** Open a terminal or command prompt and run the following command to check if Python is installed:

python --version

If Python is not installed, visit the official Python website at https://www.python.org and download the appropriate installer for your operating system.

2. **Install OpenCV using pip:** Open a terminal or command prompt and run the following command:

pip install opencv-python

If you also need additional modules like OpenCV's contrib package, you can install it with the following command:

pip install opencv-contrib-python

This command will download and install the OpenCV package and its dependencies.

3. **Verify the installation:** You can verify that OpenCV is installed correctly by running a simple Python script. Create a new Python file (e.g., **opencv_test.py**) and add the following code:

import cv2 print(cv2.__version__)

Save the file and run it with the Python interpreter:

python opencv_test.py

If OpenCV is installed correctly, it will print the version number on the console without any errors.

That's it! OpenCV is now installed and ready to be used in your Python projects. You can import the cv2 module in your Python scripts and start utilizing the functionalities provided by OpenCV.

CHAPTER 6

ALGORITHMS

Motion Tracking System:

6.1 Introduction

A motion tracking system is a technology that detects and tracks the movement of objects or individuals in each environment. It is widely used in various fields, including computer vision, robotics, surveillance, sports analysis, and virtual reality. By analyzing the changes in position, shape, or appearance of objects over time, motion tracking systems provide valuable information about motion patterns, interactions, and behavior.

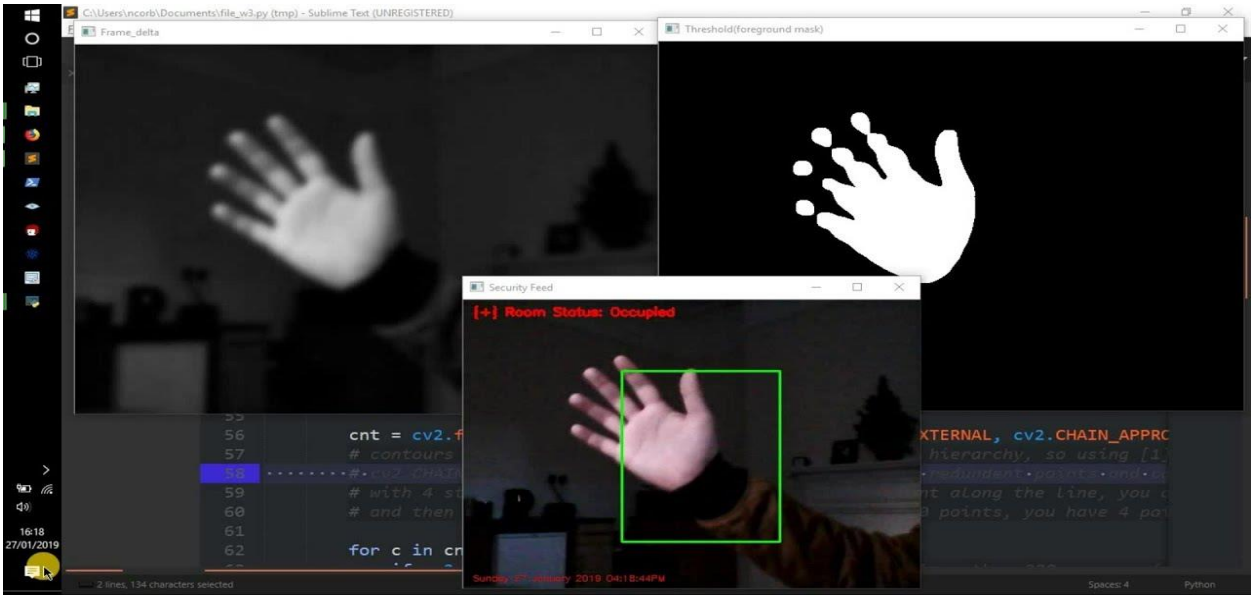
The primary goal of a motion tracking system is to track the movement of objects accurately and efficiently in real-time. This is typically achieved by utilizing cameras or other sensing devices to capture visual information or other relevant data. The captured data is then processed and analyzed using algorithms and techniques to identify and track the objects of interest.

In computer vision-based motion tracking systems, computer algorithms are employed to analyze the visual input from cameras. These algorithms can detect, and track objects based on various features such as color, shape, texture, or motion patterns. Some common techniques used in motion tracking systems include background subtraction, optical flow, feature tracking, and object recognition.

Motion tracking systems find applications in diverse fields. In sports analysis, they are used to track the movement of athletes, analyze their performance, and provide feedback for training and improvement. In virtual reality and augmented reality, motion tracking enables users to interact with virtual objects and environments by mapping their real-world movements. In robotics, motion tracking systems help robots navigate and interact with their surroundings, allowing for tasks such as object manipulation, obstacle avoidance, and human-robot interaction. Advancements in computer vision, machine learning, and sensor technologies have contributed to the development of more sophisticated and accurate motion tracking systems. These systems can

handle complex scenarios, track multiple objects simultaneously, and provide detailed information about motion dynamics, velocities, and trajectories.

Overall, motion tracking systems play a crucial role in various domains where the analysis of movement and motion-related information is essential. By capturing and interpreting motion data, these systems enable a wide range of applications that enhance understanding, enable interaction, and improve performance in numerous fields.



6.2 Advantages of Motion Tracking System:

1. **Realistic animation**: Motion capture can provide realistic and lifelike animations, as it captures the actual movements and nuances of human actors.
2. **Time-saving**: Motion capture can save time compared to traditional animation methods, as it can record a large amount of data quickly, which can be used to generate animations.
3. **Cost-effective**: Motion capture can be cost-effective in the long run, as it reduces the need for expensive animation equipment and personnel.
4. **Versatility**: Motion capture can be used for a wide range of applications, from film and video games to scientific research and sports analysis.

6.3 Disadvantages of Motion Tracking System:

1. **Limitations of technology:** The accuracy and quality of motion capture can be limited by the technology used, which can result in errors and glitches.
2. **Expensive setup**: Setting up a motion capture system can be expensive, as it requires specialized equipment and expertise.
3. **Not suitable for all animations**: Motion capture may not be suitable for all types of animations, such as those that require exaggerated or stylized movements.
4. **Dependency on actors**: Motion capture is dependent on the skill and availability of human actors, which can affect the quality and availability of data.

6.4 Applications of Motion Tracking System:

Motion tracking systems have a wide range of applications across various industries and domains. Here are some common applications:

1. **Surveillance and Security:** Motion tracking systems are extensively used in surveillance and security applications. They enable real-time monitoring of areas, automatically detect and track suspicious activities, and provide alerts or trigger alarms when unauthorized movement is detected.
2. **Robotics and Automation:** Motion tracking plays a crucial role in robotics and automation. It allows robots to perceive and interact with their environment by tracking objects or individuals, enabling tasks such as object manipulation, navigation, and human-robot interaction.
3. **Sports Analysis and Training:** Motion tracking is widely used in sports analysis to capture and analyze the movements of athletes. It helps coaches and trainers assess performance, analyze techniques, and provide feedback to improve skills and optimize training regimens.

4. **Virtual Reality and Augmented Reality:** Motion tracking is fundamental to creating immersive virtual reality (VR) and augmented reality (AR) experiences. It enables users to interact with virtual objects and environments, providing realistic and intuitive interactions.
5. **Human-Computer Interaction:** Motion tracking systems are employed in human-computer interaction scenarios. They enable gesture recognition, tracking user movements, and allowing natural and intuitive control of user interfaces. This technology finds applications in gaming, virtual assistants, smart TVs, and interactive displays.
6. **Medical and Rehabilitation:** Motion tracking has applications in medical fields, including rehabilitation. It enables precise tracking of patient movements, aiding in the assessment of mobility, range of motion, and progress in rehabilitation exercises. Motion tracking systems assist in developing personalized rehabilitation programs.
7. **Film and Animation:** Motion tracking is widely used in the entertainment industry for motion capture. It allows the realistic transfer of human movements onto digital characters, enabling lifelike animations for movies, video games, and visual effects.
8. **Gesture Recognition:** Motion tracking systems are employed in gesture recognition applications. They analyze and interpret hand or body movements to recognize specific gestures, enabling touchless control of devices, sign language recognition, and other interactive applications.
9. **Biomechanics and Ergonomics:** Motion tracking is used in biomechanics and ergonomics research to analyze human movement patterns and assess ergonomic factors. It helps in understanding body mechanics, improving workplace ergonomics, and designing products with optimal user comfort and safety.
10. **Automotive and Driver Assistance:** Motion tracking systems find applications in the automotive industry for driver assistance and safety systems. They can track objects and pedestrians around the vehicle, assist in collision avoidance, and enable features like adaptive cruise control or lane-keeping assistance.

These are just a few examples of the numerous applications of motion tracking systems. As technology advances, motion tracking continues to find new applications and contributes to advancements in various fields.

CHAPTER 7
IMPLEMENTATION AND CODING

7.1 SAMPLE CODE

```
# Python program to implement

# Webcam Motion Detector


# Importing OpenCV, time and Pandas library

import cv2, time, pandas

# importing datetime class from datetime library

from datetime import datetime


# Assigning our static_back to None

static_back = None


# List when any moving object appear

motion_list = [ None, None ]


# Time of movement
```

```
time = []

# Initializing DataFrame, one column is start
# time and other column is end time
df = pandas.DataFrame(columns = ["Start", "End"])

# Capturing video
video = cv2.VideoCapture(0)

# Infinite while loop to treat stack of image as video
while True:
    # Reading frame(image) from video
    check, frame = video.read()

    # Initializing motion = 0(no motion)
    motion = 0

    # Converting color image to gray_scale image
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Converting gray scale image to GaussianBlur
```

```
# so that change can be find easily
gray = cv2.GaussianBlur(gray, (21, 21), 0)

# In first iteration we assign the value
# of static_back to our first frame
if static_back is None:
        static_back = gray
        continue

# Difference between static background
# and current frame(which is GaussianBlur)
diff_frame = cv2.absdiff(static_back, gray)

# If change in between static background and
# current frame is greater than 30 it will show white color(255)
thresh_frame = cv2.threshold(diff_frame, 30, 255, cv2.THRESH_BINARY)[1]
thresh_frame = cv2.dilate(thresh_frame, None, iterations = 2)

# Finding contour of moving object
cnts,_ = cv2.findContours(thresh_frame.copy(),
```

```
                                        cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    for contour in cnts:
        if cv2.contourArea(contour) < 10000:
            continue
        motion = 1

        (x, y, w, h) = cv2.boundingRect(contour)
        # making green rectangle around the moving object
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 3)

    # Appending status of motion
    motion_list.append(motion)

    motion_list = motion_list[-2:]

    # Appending Start time of motion
    if motion_list[-1] == 1 and motion_list[-2] == 0:
        time.append(datetime.now())
```

```
# Appending End time of motion
if motion_list[-1] == 0 and motion_list[-2] == 1:
        time.append(datetime.now())

# Displaying image in gray_scale
cv2.imshow("Gray Frame", gray)

# Displaying the difference in currentframe to
# the staticframe(very first_frame)
cv2.imshow("Difference Frame", diff_frame)

# Displaying the black and white image
# in which if intensity difference greater than 30 it will appear white
cv2.imshow("Threshold Frame", thresh_frame)

# Displaying color frame with contour of motion of object
cv2.imshow("Color Frame", frame)

key = cv2.waitKey(1)
# if q entered whole process will stop
if key == ord('q'):
```

```
            # if something is movingthen it append the end time of movement
            if motion == 1:
                time.append(datetime.now())
            break

# Appending time of motion in DataFrame
for i in range(0, len(time), 2):
    df = df.append({"Start":time[i], "End":time[i + 1]}, ignore_index = True)

# Creating a CSV file in which time of movements will be saved
df.to_csv("Time_of_movements.csv")

video.release()

# Destroying all the windows
cv2.destroyAllWindows()
```

CHAPTER 8

OUTPUT SCREENS

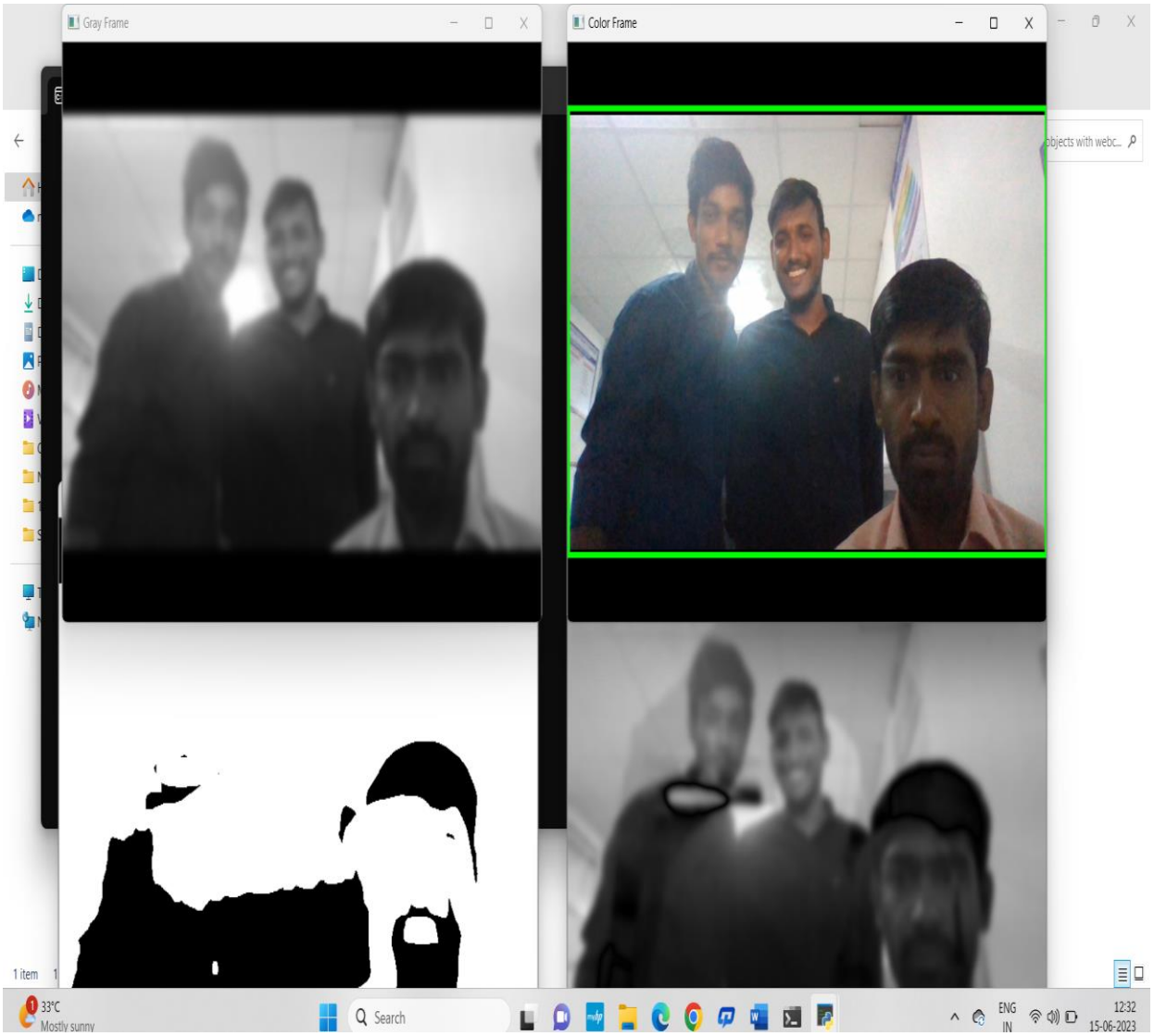


Fig 1. Output screen

1.Color Frame: In this frame, you can see the color images in color frame along with green contour around the moving objects.

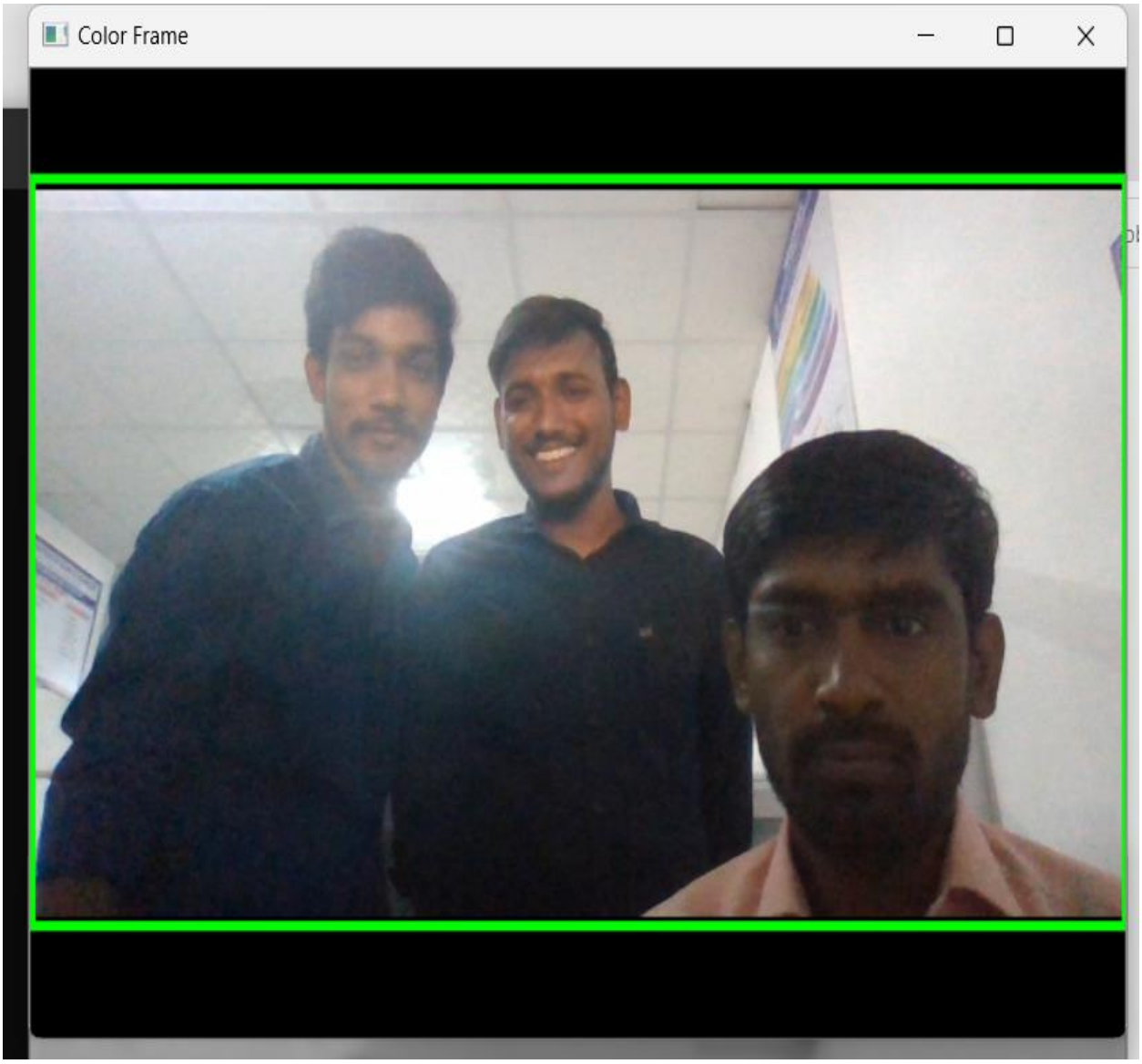


Fig 2. Color Frame

2.Gray Frame: In Gray frame the image is a bit blur and in grayscale we did so because, In gray pictures there is only one intensity value whereas in RGB (Red, Green and Blue) image there are three intensity values. So, it would be easy to calculate the intensity difference in grayscale.

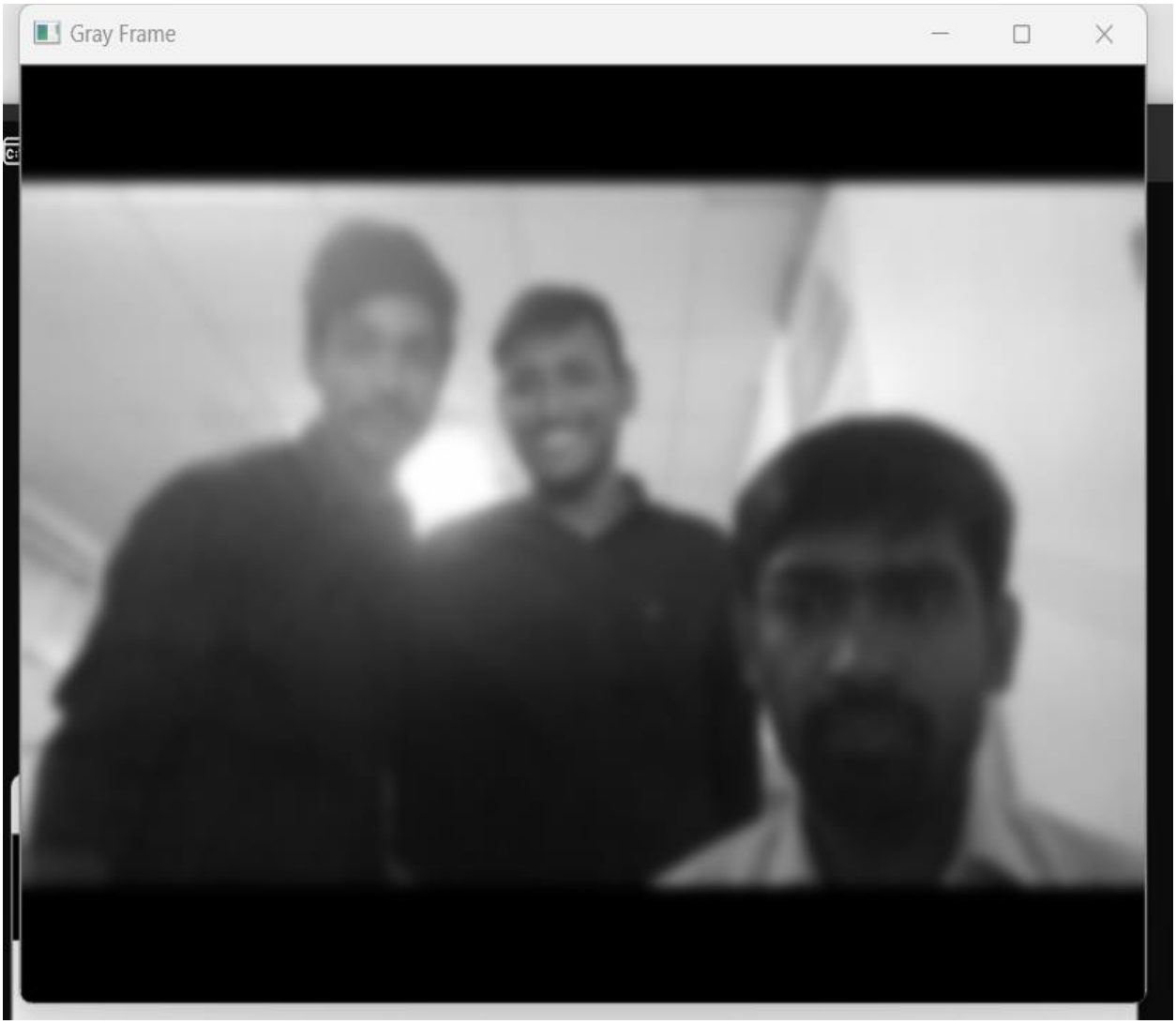


Fig 3. Gray Frame

3.Difference Frame: Difference frame shows the difference of intensities of first frame to the current frame.

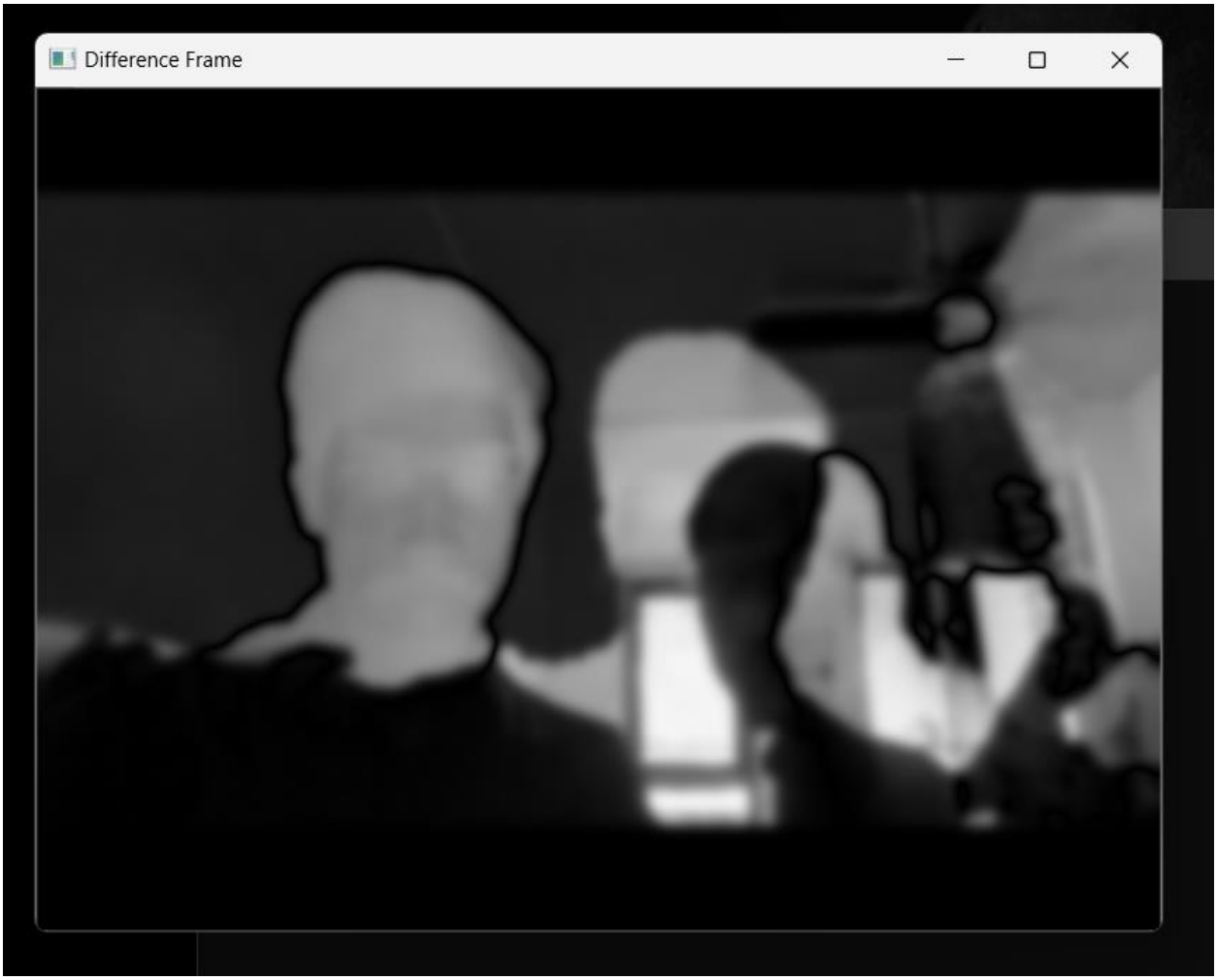


Fig 4. Difference Frame

4.Threshold Frame: If the intensity difference for a particular pixel is more than 30(in my case) then that pixel will be white and if the difference is less than 30 that pixel will be black.

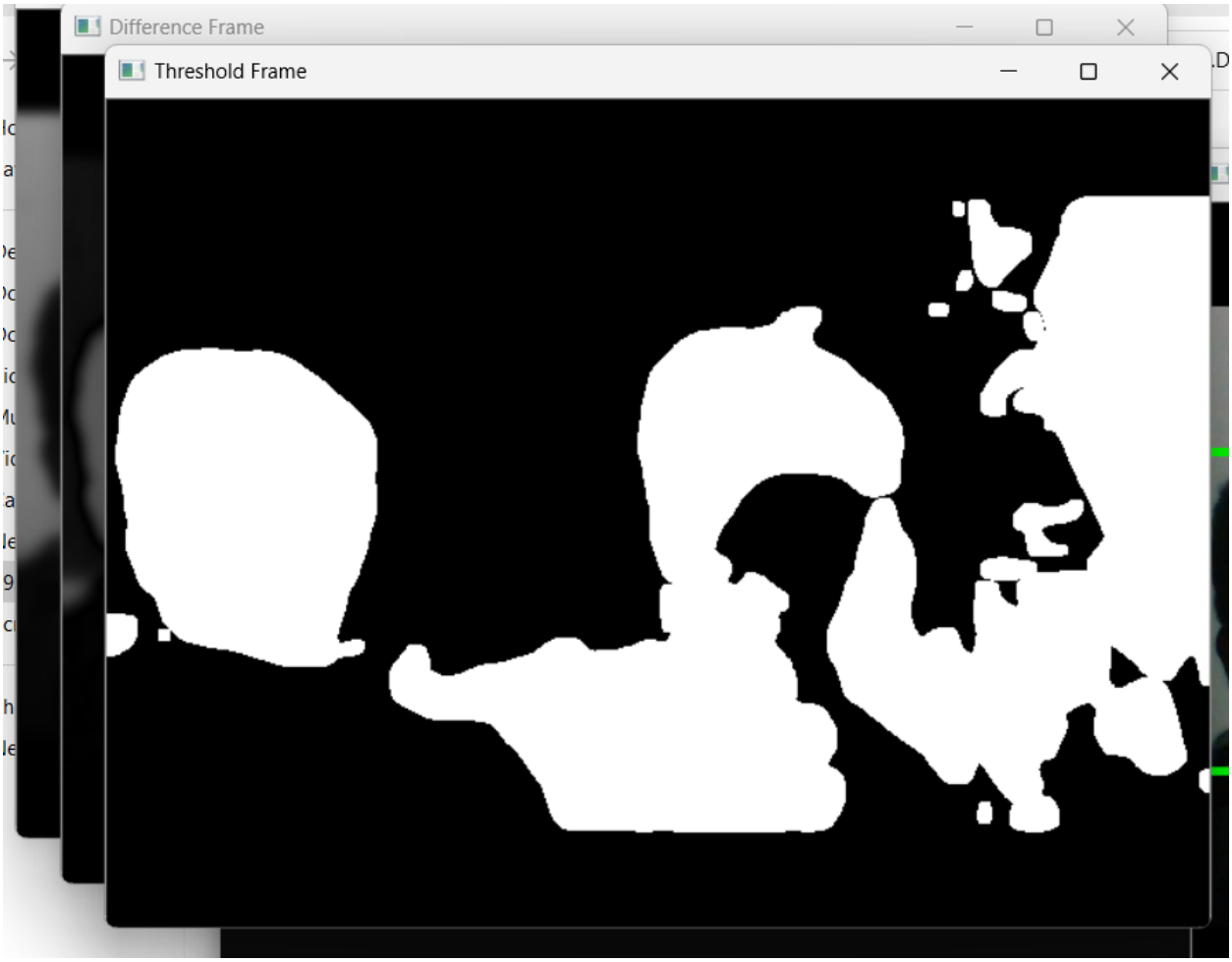


Fig 5. Threshold Frame

CHAPTER 9

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

9.1 TYPES OF TESTS

1. **Unit testing:** Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

2. **Integration testing:** Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

3. **Functional test:** Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

4. **System Test:** System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5. **White Box Testing:** White Box Testing is a test in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It is the purpose. It is used to test areas that cannot be reached from a black box level.

6. **Black Box Testing:** Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, like most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a test in which the software under the test is treated as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

7. **Unit Testing:** Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

9.2 Test strategy and approach

Field testing will be performed manually, and functional tests will be written in detail.

Test objectives:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested:

- Verify that the entries are in the correct format.
- No duplicate entries should be allowed.
- All links should take the user to the correct page.

Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 10

CONCLUSION

Although we have many motion detection algorithms previously implemented, the current algorithm implemented is much more efficient than that of before. It just takes the input frame of images and detects the difference considering it as the motion detection. Alert systems may be implemented automatically when the motion is detected. The outcome of the project is to detect motion through a webcam with much higher accuracy than any other previously implemented motion detection algorithms. Few things concluded from my project which are absent in many previously implemented algorithms are...

- Requires less memory
- Saves a lot of time
- Most reliable
- No need of manual monitoring
- Analysis is done automatically
- Archive space to store videos isn’t needed.
- Alert systems may be implemented automatically when the motion is detected.

CHAPTER 11

FUTURE SCOPE

We all can see that now we can perform multiple actions sitting at our home for which we had to run around many offices back then. Some of these tasks were performed using a webcam, like for the purpose of official documents, for online exams, for online proctoring, and many other tasks. Now, every laptop comes embedded with a camera; thus, giving us the option to perform all these tasks from sitting at our home. This feature enabled us to perform all these tasks without any hustle and without worrying about running from here to there. Apart from these benefits, the webcam feature also provides us the convenience of sitting at home and saves our time. In today's time, we can perform multiple actions using the webcam feature of our system, and we can utilize it to develop multiple projects, which we were unable to do back then. Today we can do many developments and programming-related work where we have to use the webcam and develop many projects using this feature of our system.

sOne of the most popular webcams uses a webcam as the motion detector sensor to sense the motion from a particular area. We can use the webcam of our system as a motion detector and use it to perform several actions where the use of a motion detector is required. The use of a webcam as a motion detector is very popular now, and one can use this feature sitting from their home too. Yes, it is true that we can convert our webcam into a motion detector and all that by ourselves. There are many options that we can use to convert our system's webcam into a motion detector sense and use it while performing several tasks that require a motion detector.

CHAPTER 12

BIBLIOGRAPHY

1. Anguelov D. (2014) Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441.

2. Cardarelli E, Detection of moving objects in roundabouts based on a monocular system. Expert Systems with Applications. 2015 Jun;42(9):4167–76.

3. Dumitru Erhan "Scalable Object Detection using Deep Neural Networks", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 2147-2154.

4. Erhan, D et al (2014). Scalable object detection using deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition vol 3, no 4, pp. 2147-2154.

5. Eric. K. W, "Deep fusion feature-based object detection method for YOLO and high resolution optical remote sensing images," Applied Science, vol. 34, 2019.

6. Felzenszwalb. P. F. et al., 2010 "Object detection with discriminatively trained part-based models," IEEE Trans. Pattern Anal. Mach. Intel., vol. 32, no. 9, pp. 1627–1645

7. Girshick. R., "Fast r-CNN," in Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448, Berlin, Germany, 2015.

8. Galvez. R L. (2018) Object Detection Using Convolutional Neural Networks. TENCON 2018-2018 IEEE Region 10 Conference

9. Han, C., Liu (2018). TransHist: Occlusion-robust shape detection in cluttered images. Computational Visual Media, vol4, no. 2, pp. 161-172.

10. Joseph Redmon, Ali Farhadi, "YOLO9000: Better, Faster, Stronger", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7263-7271.