## VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590 014



# PROJECT WORK PHASE-I (18CSP77)

on

# Unusual Crowd Activity Detection And Localization Using Open CV And Deep Learning

Submitted in partial fulfillment for the award of the degree of Bachelor of Engineering

in

# COMPUTER SCIENCE AND ENGINEERING Academic year 2022-2023

Submitted By

Jaikumar K(1GD19CS014) Karthik S(1GD19CS018) Mukesh R(1GD19CS023) N Krishna Vamsi (1GD19CS028)

Under the guidance of

Mrs.Shushma R Assistant Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GOPALAN COLLEGE OF ENGINEERING & MANAGEMENT BANGALORE-560 048

## GOPALAN COLLEGE OF ENGINEERING AND MANAGEMENT

[ISO Certified 9002:2008, Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi] 181/1,182/1, Hoodi Village, Sonnenahalli, K.R. Puram, Bangalore - 560 048 DEPARTMENT OF COMPUTER SCIENCE ANDENGINEERING



## **CERTIFICATE**

Certified that the project work entitled "UNSUAL CROWD ACTIVITY DETECTION USING OPEN CV AND DEEP LEARNING" carried out by Jaikumar K (1GD19CS014), Karthik S (1GD18CS018), Mukesh R (1GD19CS023), N Krishna Vamsi (1GD19CS028), bonafide students of GOPALAN COLLEGE OF ENGINEERNG AND MANAGEMENT in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year, 2022- 2023. It is certified that all suggestions indicated for Internal Assessment have been incorporated in the Project Report deposited in the Department library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said degree.

Signature of the Guide Signature of the HOD

Mrs.Shushma R Dr. J. Somasekar

Signature of the Principal

Dr. Basavaraju.C

Name of the Examiners

Signature with date

1.

2.

#### **DECLARATION**

We, students Jaikumar(1GD19CS014), Karthik S (1GD19CS018), Mukesh R (1GD19CS023), N Krishna Vamsi (1GD19CS028) of VIII semester B.E in Computer Science and Engineering, Gopalan College Of Engineering & Management, Bangalore-560048 hereby declare that the project entitled "Unusual Crowd Activity Detection And Localization Using Open CV And Deep Learning" submitted to the Visvesvaraya Technological University during the academic year 2022-2023, is a record of a project work done by us under the guidance of Dr. J. Somasekar, HOD and Professor, Department of Computer Science and Engineering, Gopalan College of Engineering and Management, Bangalore. This project work is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree.

Date: Jaikumar K(1GD19CS014)
Karthik S(1GD19CS018)

Place: Bangalore Mukesh R(1GD19CS023)

N Krishna Vamsi(1GD19CS028)

#### **ACKNOWLEDGEMENT**

The project report was the result of the encouragement of many people who helped in shaping it and provided feedback, direction and valuable support. It is with heartly gratitude that we acknowledge their contribution.

We express our gratitude towards **Dr. C. Prabhakar,** Director, Gopalan Foundation and **Dr. Basavaraju.**C, Principal, Gopalan College of Engineering and Management, for providing us with the infrastructure and resources which helped in completing the project.

We are thankful to **Dr. J. Somasekar,** Professor and Head of Computer Science and Engineering, GCEM for constant support extended towards us during the course of this project. His help and advice instilled the drive in us to complete the project on time.

We are really thankful to **Mrs.Shushma R**, Assistant professor and Guide, Computer Science and Engineering, GCEM for the constant support extended towards us during the course of the project. Her help and advice instilled the drive in us to complete the project on time.

Finally, we would like to extend our thanks to Almighty, friends and family who supported us to complete the project.

Jaikumar K(1GD19CS014) Karthik S(1GD19CS018) Mukesh R(1GD19CS023) N Krishna Vamsi(1GD19CS028)

### **ABSTRACT**

Suspicious behavior is dangerous in public areas that may cause heavy causalities. There are various systems developed on the basis of video frame acquisition where motion or pedestrian detection occur but those systems are not intelligent enough to identify the unusual activities even at real time. It is required to recognized scamper situation at real time from video surveillance for quick and immediate management before any casualties. Proposed system focuses on recognizing suspicious activities and target to achieve a technique which is able to detect suspicious activity automatically using computer vision. Here system uses OpenCV library for classifying different kind of actions at real time. The motion influence map has been used to represent the motion analysis that frequently changes the position from one place to another. System uses pixel level presentation for making it easy to understand or identify the actual situation.

# **TABLE OF CONTENTS**

CHAPTER 11
INTRODUCTION 1
1.1 PROBLEM STATEMENT2
1.2 EXISTING SYSTEM
1.3 PROPOSED SYSTEM
CHAPTER 24
LITERATURE SURVEY5
2.1 SUMMARY 5
C HAPTER 36
SYSTEM REQUIREMENT SPECIFICATION7
3.1 FUNCTIONAL REQUIREMENT
3.1.1 PRODUCT REQUIREMENT8
3.1.2 ORGANIZATIONAL REQUIREMENTS8
3.1.3 USER REQUIREMENTS 8
3.1.4 BASIC OPERATIONAL REQUIREMENTS9
3.2 NON FUNCTIONAL REQUIREMENTS9
2.2 DECOLIDE DECLUDEMENT
3.3 RESOURCE REQUIREMENT 10
3.4 HARDWARE REQUIREMENTS 10

CHAPTER 4	11
SYSTEM ANALYSIS	11
4.1 FEASIBILITY STUDY	11
4.2 PERFORMANCE ANALYSIS	11
4.3 TECHNICAL ANALYSIS	11
4.4 ECONOMICAL ANALYSIS	11
CHAPTER 5	12
SYSTEM DESIGN	12
5.1 FUNDAMENTAL DESIGN CONCEPTS	12
5.2 INPUT DESIGN	12
5.3 OUTPUT DESIGN	12
5.4 SYSTEM DEVELOPMENT METHODOLOGY	12
5.5 MODEL PHASES	12
CHAPTER 6	12
IMPLEMENTATION	12
6.1 LANGUAGE USED FOR IMPLEMENTATION	12
6.2 PLATFORM USED FOR IMPLEMENTATION	12
6.3 PROPOSED METHODOLOGY	12
CHAPTER 7	20
TESTING	20
7.1 TYPES OF TESTS	12
7.2 UNIT TESTING	12

7.3 INTEGRATION TESTING
7.4 VALIDATION TESTING
7.5 TEST CASES
CHAPTER 820
INTERPRETATION OF RESULT20
CHAPTER 920
CONCLUSION20
REFERENCES21

## CHAPTER 1

#### INTRODUCTION

the increasing importance of security, a great number of surveillance cameras have been installed in private and public places. However, the plethora of video sequences available is overwhelming the human resources monitoring them. To this end, there has been significant interest in a smart surveillance system that can automatically detect unusual or abnormal activities. Over the last decades, many researchers in computer vision and pattern recognition have devoted their efforts toward human action and human-human interaction recognition in video sequences. Recently, abnormal or unusual activity detection in crowded scenes has gained more interest from researchers. Unlike human action or interaction recognition, conventional methods are not applicable to the detection and/or tracking of human subjects in a crowded scene owing to the presence of occlusions, small objects sizes, and other factors. For unusual activity detection in a crowded scene, texture information such as a spatio-temporal gradient, mixture of dynamic textures, and spatio-temporal frequency has been considered an efficient means of detection. In the meantime, other groups have used optical flows that directly characterize motion features in a sequence, e.g., a motion heat map, clustered motion patterns, spatial saliency of the motion feature, crowd prediction using a force field model, optical flow fields, particle trajectory, a social force model, and a local motion histogram. Although motion flow based approaches have shown their efficacy in previous works, we believe it is still important to consider the information on the size of the objects and their interactions. For example, in Fig. 1b, where riding a bicycle is considered an unusual activity, the size of the object and its effect to the nearby pedestrians' moving directions are important information along with the movement speed. To the best of our knowledge, none of the previous methods has explicitly considered this information, the use of which can be helpful in enhancing the performance.

characteristics of moving objects by considering their motion flows, sizes, and interactions, simultaneously. Specifically, we define a "motion influence map" that efficiently depicts the underlying motion patterns in a crowded scene.

#### **Problem Statement**

There are various researches have been done in the field of human activity detection but fewer researches found for unusual human activities detection at real time using camera. Computer vision is a challenging approach that can acquire real time human activity.

The objective of the system is to recognize unusual human activity from crowd using motion influence map and OpenCV for prior appraisal against crime in public places using camera. It will help to build or install a system that can work 24x7 for real time surveillance and decrease the crowd based criminal activities and saves us from fatal results.

#### **Existing System**

The Existing framework which is Based on Conventional Neural Network that trains for human facial acknowledgment. Framework can be prepared with various outward appearances and track exercises w.r.t. to sentenced articulations. The CNNbased AU location uncovered a comparable change in discoveries concerning newborn child quality between assignments.

#### **Disadvantages**

- CNN approach needs more time for training and classifying.
- It requires dataset in huge number.
- It has less efficiency and accuracy

#### **Proposed System**

We describe a method for representing motion characteristics for the detection and localization of unusual activities within a crowded scene. Here, we should note that, we considered two types of unusual activities: local and global. The activities such as the unique appearance of non-human objects or the fast movement of a person when most of the other pedestrians are walking slowly considered as local. Global unusual activities occur across the frame, for example, when every pedestrian within a scene starts to run suddenly to escape from the scene.

Proposed work is able to recognize human activity in crowd and analyze whether the action is usual or unusual. System purely debates with crowd based activities that ensure situations. System uses OpenCV library along with python IDE that deals with best precision. System proposes motion influence map that comprises for correct recognition rate. Recognizing unusual activity from crowd is difficult task especially for sensor networks; computer vision is an effective approach that can acquire real time human activities and later analyzes for uncommon frames.

Also Enhanced the Work with Improvising the Detection for Violence Using CNN-LSTM Model

#### Advantages

- With the opency we can easily work in real time and no dataset required for training separately.
- The "motion influence map" that efficiently depicts the underlying motion patterns in a crowded scene.
- Proposed work is cost-effective for a smart surveillance system to detect both local and global unusual activities in a unified framework.

### CHAPTER 2

#### LITERATURE SURVEY

Suspicious exercises on open zones and individual security are in genuine threat. In open territories, a huge number of video reconnaissance frameworks are utilized, for example, streets, detainment facilities, blessed locales, air terminals and grocery stores. Video reconnaissance cameras are not astute enough to perceive irregular exercises even at ongoing. It is important to screen the recognition of suspicious exercises and to check the legitimacy of reconnaissance video. It is required to perceived hurry circumstance at constant from video observation for fast and quick administration.

Zakia Hammal et al. [4] proposed a framework which is Based on Conventional Neural Network that trains for human facial acknowledgment. Framework can be prepared with various outward appearances and track exercises w.r.t. to sentenced articulations. The CNNbased AU location uncovered a comparable change in discoveries concerning newborn child quality between assignments. The exactness rate for acknowledgment right activity or articulation runs between 79 to 93 %.

He Xu et al. [5] proposed a framework which depends on RFID which is a physical sensor. The RFID framework can be isolated into the accompanying three segments: Reader, Tag and Back-end PC framework. Can convey through the peruser and label receiving wires.

The means of crafted by RFID framework are as per the following:

- (1) The perusers send radio recurrence flag in the encompassing condition, and check whether there is any tag;
- (2) When the tag in the peruser's reception apparatus perusing range is enacted by its very own receiving wire to speak with the peruser and send its chip electronic code or other information;
- (3) RFID Reader gets an electronic item code (EPC) or information sign of the tag by reception apparatus; Then the information is decoded and prepared, and it will be sent to the back-end PC framework.

Varsha Shrirang Nanaware et al. [6] made an overview over different executed framework over activity acknowledgment. Various specialists have chipped away at location procedures of different human pursue and activity acknowledgment in an undeniable time moving video, exhaustive writing overview of the ongoing works done by various writers is being given during this energizing and application disapproved of handy examination field. Truth be told, the study/audit paper is finished by U.S. as this can be the spot to start for our investigation deal with "location systems of different human pursue and activity acknowledgment in an undeniable time moving video observation".

Jiahao Li et al. [7] proposed a framework which depends on pyramid vitality map as highlight descriptor for a grouping of casings, it can spare and present the activity history that spatially contrasts and the activities perceived. It depends on bidirectional neural system which can back track the concealed layers and present the most pertinent outcomes. It is likewise powerful for single objective or skeleton however mistakes for different targets.

Nour El Din Elmadany et al. [8] proposed a framework which depends on Biset Globality Locality Preserving Canonical Correlation Analysis, which means to get familiar with the normal component subspace between two sets. The subsequent strategy is Multiset Globality Locality Preserving Canonical Correlation Analysis, which expects to manage at least three sets. It make arrangements of skeletons as informational indexes. The exactness for right acknowledgment rate is 90.1%.

Soumalya Sen et al. [9] proposed a framework which depends on picture parsing procedure. Picture parsing relates various kinds of activities which are performed by human that can be perceived in grouping of casings. Activity arranges as – strolling, running, applauding, running, cycling, surfing, and so forth. It depends on frontal area and foundation connection through which framework improves the closer view item and stores these edges for future correlation. Picture parsing brings together picture division, object discovery or acknowledgment.

## 2.1 Summary

This chapter mainly discusses about the papers, websites that are referred while making this dissertation report. All these papers and websites provide information related to learning of collective behavior, their existing solutions..

# **Chapter 3**

## SYSTEM REQUIREMENT SPECIFICATION

Software requirement Specification is a fundamental document, which forms the foundation of the software development process. It not only lists the requirements of a system but also has a description of its major feature. An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. It is important to note that an SRS contains functional and nonfunctional requirements only; it doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements to be.

# 3.1 Functional Requirement

The functionalities to be implemented are

- 1. Train and build a model
- 2. Load pretrained cnn\_lstm model
- 3. Test for video and classify it has unusual or violence or normal.

## 3.2 Non-functional Requirement

Non functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Non functional requirements arise through the

user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:-

1. Product Requirements 2. Organizational Requirements 3. User Requirements 4. Basic Operational Requirements

## 3.2.1 Product Requirements

**Portability:** Since the software is developed in python it can be executed on any platform for which the Python is available with minor or no modifications.

**Correctness:** It followed a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data.

**Ease of Use:** The front end is designed in such a way that it provides an interface which allows the user to interact in an easy manner.

**Modularity:** The complete product is broken up into many modules and well-defined interfaces are developed to explore the benefit of flexibility of the product.

**Robustness:** This software is being developed in such a way that the over all performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness. Python itself possesses the feature of robustness, which implies the failure of the system is negligible.

Non functional requirements are also called the qualities of a system. These qualities can be divided into execution quality & evolution quality. Execution qualities are security & usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

## **3.2.2 Organizational Requirements**

**Process Standards:** IEEE standards are used to develop the application which is the standard used by the most of the standard software developers all over the world.

**Design Methods:** Design is one of the important stages in the software engineering process. This stage is the first step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs.

The design of the system is perhaps the most critical factor affecting the quality of the software

and has a major impact on the later phases, particularly testing and maintenance. We have to design the

product with the standards which has been understood by the developers of the team.

3.2.3 User Requirements

The user must be able to visualize Graphical User Interface Window.

The user must be able to configure all the parameters.

3.2.4 Basic Operational Requirements

The customers are those that perform the eight primary functions of systems engineering, with

special emphasis on the operator as the key customer. Operational requirements will define the basic

need and, at a minimum, will be related to these following points:-

**Mission profile or scenario:** It describes about the procedures used to accomplish mission objective. It

also finds out the effectiveness or efficiency of the system.

**Performance and related parameters:** It points out the critical system parameters to accomplish the

mission

Utilization environments: It gives a brief outline of system usage. Finds out appropriate environments

for effective system operation.

**Operational life cycle:** It defines the system lifetime.

3.3 Resource Requirement

**SPYDER** 

Spyder, the Scientific Python Development Environment, is a free integrated development environment

(IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and

introspection features. After you have installed Anaconda, start Spyder on Windows, macOS, or Linux

by running the command spyder. Spyder is also pre-installed in Anaconda Navigator, which is included

in Anaconda. On the Navigator **Home** tab, click the Spyder icon. For more information about Spyder, see the Spyder web page or the Spyder documentation.

**Anaconda** command **prompt** is just like command **prompt**, but it makes sure that you are able to use **anaconda** and conda commands from the **prompt**, without having to change directories or your path. These locations contain commands and scripts that you can run.

#### **MATLAB**

MATLAB® combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly. It includes the <u>Live</u> <u>Editor</u> for creating scripts that combine code, output, and formatted text in an executable notebook

#### **Professionally Built**

MATLAB toolboxes are professionally developed, rigorously tested, and fully documented.

#### **With Interactive Apps**

MATLAB apps let you see how different algorithms work with your data. Iterate until you've got the results you want, then automatically generate a MATLAB program to reproduce or automate your work.

#### And the Ability to Scale

Scale your analyses to run on clusters, GPUs, and clouds with only minor code changes. There's no need to rewrite your code or learn big data programming and out-of-memory techniques

#### **Python Programming**

<u>Python</u> is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

#### 1) Finding an Interpreter:

Before we start Python programming, we need to have an interpreter to interpret and run our programs. There are certain online interpreters like **https://ide.geeksforgeeks.org/**, http://ideone.com/ or

http://codepad.org/that can be used to run Python programs without installing an interpreter.

*Windows*: There are many interpreters available freely to run Python scripts like IDLE (Integrated Development Environment) that comes bundled with the Python software.

## 3.4 Hardware Requirements

CPU: Intel 2.1 GHZ

Memory : 4GB

Disk: 100GB

Display : 15 inch color

# 3.5 Software (Tools & Technologies) Requirements

Coding : Python

Platform: python 3.7 and above

Tool : Spyder

OS : Windows 7

Libraries : tensorflow, keras, opency

# **Summary**

This chapter gives details of the functional requirements, non-functional requirements, resource requirements, hardware requirements, software requirements etc. Again the non-functional requirements turn contain product requirements, organizational requirements, user requirements, basic operational requirements etc.

## **Chapter 4**

#### SYSTEM ANALYSIS

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

## 4.1 Feasibility Study

All systems are feasible when provided with unlimited resource and infinite time. But unfortunately this condition does not prevail in practical world. So it is both necessary and prudent to evaluate the feasibility of the system at the earliest possible time. Months or years of effort, thousands of rupees and untold professional embarrassment can be averted if an ill-conceived system is recognized early in the definition phase. Feasibility & risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. In this case there are three primary areas of interest:-

## 4.1.1 Performance Analysis

For the complete functionality of the project work, the project is run with the help of healthy networking environment. Normally, the OS is windows XP. Performance analysis is done to find out whether our algorithm is more efficient. It is essential that the process of performance analysis and definition must be conducted in parallel.

#### 4.1.2 Technical Analysis

System is only beneficial only if it can be turned into information systems that will meet the organization's technical requirement. Simply stated this test of feasibility asks whether the system will work or not when developed & installed, whether there are any major barriers to implementation. Regarding all these issues in technical analysis there are several points to focus on:-

Changes to bring in the system: All changes should be in positive direction, there will be increased

level of efficiency and better customer service.

**Required skills:** Platforms & tools used in this project are widely used. So the skilled manpower is readily available in the industry.

**Acceptability:** The structure of the system is kept feasible enough so that there should not be any problem from the user's point of view.

### 4.1.3 Economical Analysis

Economical analysis is performed to evaluate the development cost weighed against the ultimate income or benefits derived from the developed system. For running this system, we need not have high performance servers. All the functions of implemented through software modules. In this system we are not using any physical devices for connection. So the system is economically feasible enough.

## **Summary**

The main aim of this chapter is to find out whether the system is feasible enough or not. For these reasons different kinds of analysis, such as performance analysis, technical analysis, economical analysis etc is performed.

# **Chapter 5**

#### SYSTEM DESIGN

Design is a creative process; a good design is the key to effective system. The system Design is defined as "The process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". Various design features are followed to develop the system. The design specification describes the features of the system, the components or elements of the system and their appearance to end-users.

## **5.1 Fundamental Design Concepts**

A set of fundamental design concepts has evolved over the past three decades. Although the degree of interest in each concept has varied over the years, each has stood the test of time. Each provides the software designer with a foundation from which more sophisticated design methods can be applied. The fundamental design concepts provide the necessary framework for "getting it right". The fundamental design concepts such as abstraction, refinement, modularity, software architecture, control hierarchy, structural partitioning, data structure, software procedure and information hiding are applied in this project to getting it right as per the specification.

## 5.1.1 Input Design

The input Design is the process of converting the user-oriented inputs in to the computer-based format. The goal of designing input data is to make the automation as easy and free from errors as possible. Providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right message and help for the user at right time are also considered for the development of the project. Input design is a part of overall system design which requires very careful attention. Often the collection of input data is the most expensive part of the system, which needs to be route through number of modules.

### 5.1.2 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other systems through outputs. It is most important and direct source information to the user. Efficient and intelligent output improves the systems relationship with source and destination machine.

## 5.2 System development methodology

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is waterfall model.

### **5.2.1 Model phases**

The waterfall model is a <u>sequential</u> software development process, in which progress is seen as flowing steadily downwards (like a <u>waterfall</u>) through the phases of Requirement initiation, <u>Analysis</u>, <u>Design</u>, Implementation, <u>Testing</u> and <u>maintenance</u>.

**Requirement Analysis:** This phase is concerned about collection of requirement of the system. This process involves generating document and requirement review.

**System Design:** Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on:-algorithm, data structure, software architecture etc.

**Coding:** In this phase programmer starts his coding in order to give a full sketch of product. In other words system specifications are only converted in to machine readable compute code.

**Implementation:** The implementation phase involves the actual coding or programming of the software. The output of this phase is typically the library, executables, user manuals and additional software documentation

**Testing:** In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation.

**Maintenance:** The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer need, adapt to accommodate change in the external environment, correct errors and oversights previously undetected in the testing phase, enhance the efficiency of the software.

## 5.2.2 Reason for choosing waterfall model as development method

- Clear project objectives.
- Stable project requirements.
- Progress of system is measurable.
- Strict sign-off requirements.
- Helps you to be perfect.
- Logic of software development is clearly understood.
- Production of a formal specification
- Better resource allocation.
- Improves quality. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage.
- Less human resources required as once one phase is finished those people can start working on to the next phase.

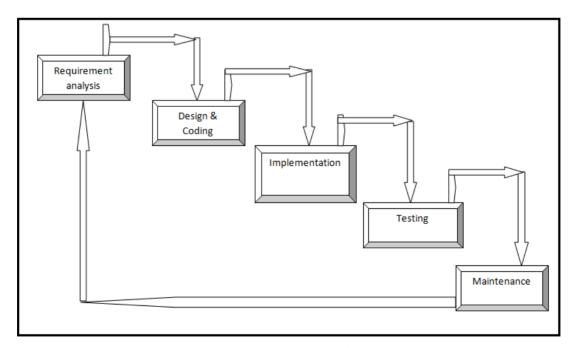
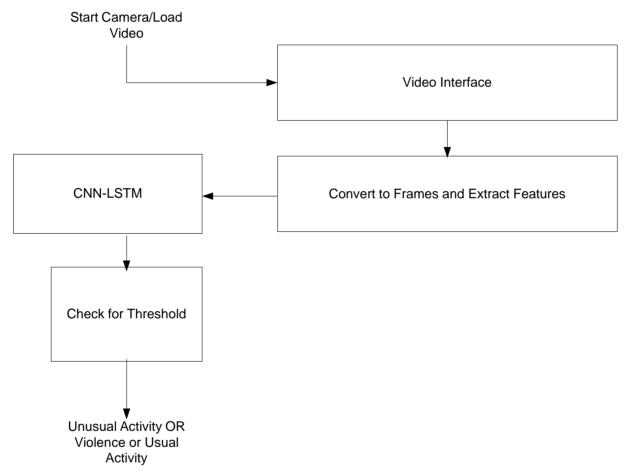


Fig 5.1:- Waterfall model

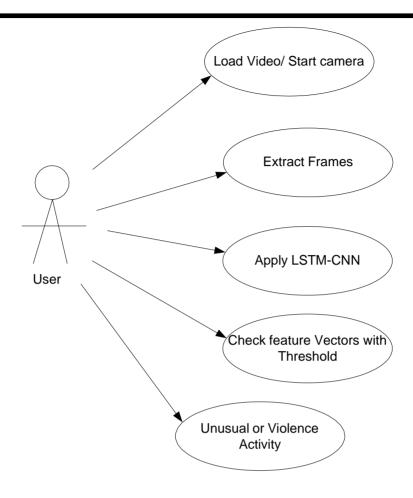
## **System Architecture**

System architecture is the conceptual design that defines the structure and behavior of a system. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system.



## Use case Diagram of the system

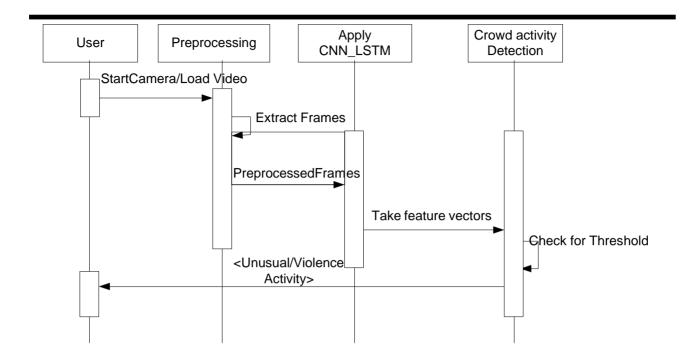
A use case diagram is a type of behavioral diagram created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.



# Sequence diagram of system operation

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.

The sequence diagrams shows below.

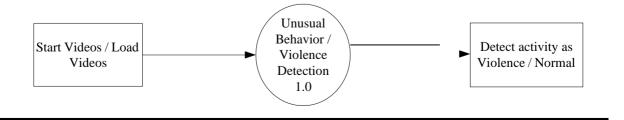


## **Data Flow Diagram of the system**

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an <u>information system</u>. DFDs can also be used for the <u>visualization</u> of <u>data processing</u> (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

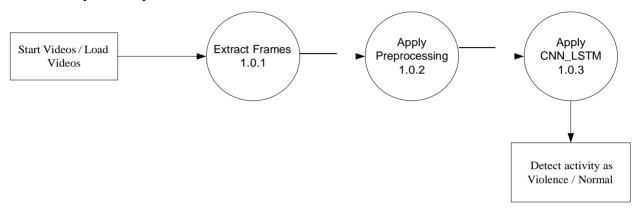
#### Level 0 Data flow diagram

A <u>context-level or level 0 data flow diagram</u> shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the Level 0 DFD) the system's interactions with the outside world are modeled purely in terms of data flows across the system boundary. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization



#### Level 1 Data flow diagram

The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole.



# **Summary**

This chapter mainly concentrates on system architecture, sequence diagram, use-case diagram, data flow diagram etc.

# Chapter 6

#### **IMPLEMENTATION**

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main workload and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled, it can cause chaos and confusion.

The implementation stage requires the following tasks.

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Evaluation of the changeover method.
- Correct decisions regarding selection of the platform
- Appropriate selection of the language for application development

## .

## 6.1 Language used for implementation

Implementation phase should perfectly map the design document in a suitable programming language in order to achieve the necessary final and correct product. Often the product contains flaws and gets ruined due to incorrect programming language chosen for implementation.

In this project, for implementation purpose mat lab is chosen as the programming language. Few reasons for which mat lab is selected as a programming language can be outlined as follows:-

MATLAB has several advantages over other methods or languages:

- Its basic data element is the matrix. A simple integer is considered an matrix of one row and one column. Several mathematical operations that work on arrays or matrices are built-in to the Matlab environment. For example, cross-products, dot-products, determinants, inverse matrices.
- Vectorized operations. Adding two arrays together needs only one command, instead of a for or while loop.
- The graphical output is optimized for interaction. You can plot your data very easily, and then change colors, sizes, scales, etc, by using the graphical interactive tools.
- Matlab's functionality can be greatly expanded by the addition of toolboxes. These are sets of specific functions that provided more specialized functionality. Ex: Excel link allows data to be written in a format recognized by Excel, Statistics Toolbox allows more specialized statistical manipulation of data (Anova, Basic Fits, etc)

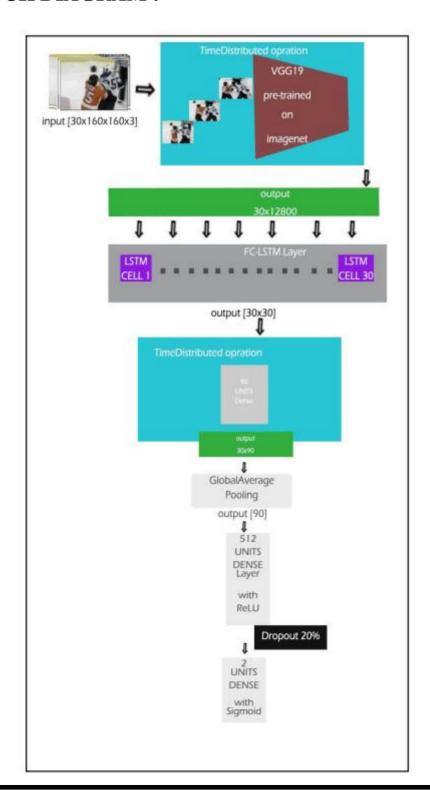
Matlab is an interpreted language for numerical computation. It allows one to perform numerical calculations, and visualize the results without the need for complicated and time consuming programming. Matlab allows its users to accurately solve problems, produce graphics easily and produce code efficiently.

# **6.2 Platform used for implementation**

A platform is a crucial element in software development. A platform might be simply defined as "a place to launch software". In this project, for implementation purpose Windows XP platform is used & reasons for choosing this platform are Integrated Networking support, More stable and secure than previous version, Contain remote desktop connection and restore option, Enhanced device driver verifier, Dramatically reduced reboot scenarios, Improved code protection, Side-by-side DLL support, Windows File Protection, Preemptive multitasking architecture, Scalable memory and processor support, Encrypting File System (EFS) with multi-user support, IP Security (IPSec), Kerberos support, Smart card support, Internet Explorer Add-on Manager, Windows Firewall, Windows Security Center, Fresh visual design.

# **6.3 Proposed Methodology**

# **BLOCK DIAGRAM:**



## **Working of Algorithm Steps**

In general, the steps we followed are the following:

- Read sequence of frames in 4d tensor (frame, H, W, RGB)
- Apply pre-trained CNN for each frame
- Group the result from the previous step and flatten the tensor to be a 2d shape (frames, SP) where SP is (H\*W\*RGB) and represent a spatial feature vector for one frame.
- Use the previous step output as feature vector input to LSTM where SP represent input and Frame represent time step ex for 30 frame input we have (SP1, SP2 .. SP30) each goes in a time step of LSTM.
- Take full sequence prediction from LSTM and feed it to a dense layer in a time distributed manner.
- Take the global average of the previous step output to get the result as a 1d tensor.
- Feed the output of the previous step into the output layer (dense layer with sigmoid activation which represents the probability of violence existence in the given video).

#### PSeudoCode:

fromfuture import absolute_import
fromfuture import division
fromfuture import print_function
import tensorflow as tf
import numpy as np
from skimage.io import imread
from skimage.transform import resize
import cv2
import numpy as np
import os
from violencemodel import *
from flask import Flask , request , jsonify , Response
from PIL import Image
from io import BytesIO
import time

```
from skimage.transform import resize
from tensorflow.keras.models import Sequential, Model, load_model
from tensorflow.keras.layers import Dropout, Dense, Flatten, Input
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import backend
from werkzeug.utils import secure_filename
from werkzeug.datastructures import FileStorage
from collections import deque
import numpy as np
import argparse
import pickle
import cv2
from twilio.rest import Client
model1 = cnnlstm\_model(tf)
# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--input", required=True,
       help="path to our input video")
ap.add_argument("-o", "--output", required=True,
       help="path to our output video")
ap.add_argument("-s", "--size", type=int, default=128,
       help="size of queue for averaging")
args = vars(ap.parse_args())
,,,
# load the trained model and label binarizer from disk
print("[INFO] loading model and label binarizer...")
model = model1
```

```
# initialize the image mean for mean subtraction along with the
# predictions queue
#mean = np.array([123.68, 116.779, 103.939][::1], dtype="float32")
#Q = deque(maxlen=args["size"])
Q = deque(maxlen=128)
# initialize the video stream, pointer to output video file, and
# frame dimensions
#vs = cv2.VideoCapture(args["input"])
vs = cv2.VideoCapture('fight.avi')
vc=cv2.VideoCapture('fight.avi')
fps = vs.get(cv2.CAP_PROP_FPS)
writer = None
(W, H) = (None, None)
client
                                               Client("ACea4cecca40ebb1bf4594098d5cef4541",
"32789639585561088d5937514694e115") #update from twilio
prelabel = "
ok = 'Normal'
okk='violence'
i=0
frames = np.zeros((30, 160, 160, 3), dtype=np.float)
datav = np.zeros((1, 30, 160, 160, 3), dtype=np.float)
frame_counter=0
# loop over frames from the video file stream
while True:
  # read the next frame from the file
  (grabbed, frm) = vc.read()
  frame_counter=vs.get(cv2.CAP_PROP_POS_FRAMES);
```

```
# if the frame was not grabbed, then we have reached the end
  # of the stream
if not grabbed:
    break
  # if the frame dimensions are empty, grab them
  if W is None or H is None:
    (H, W) = frm.shape[:2]
  #framecount = framecount+1
  # clone the output frame, then convert it from BGR to RGB
  # ordering, resize the frame to a fixed 224x224, and then
  # perform mean subtraction
  output = frm.copy()
  while i < 30:
    rval, frame = vs.read()
    frame_counter +=1
    if frame_counter == vs.get(cv2.CAP_PROP_FRAME_COUNT):
       frame_counter = 0 #Or whatever as long as it is the same as next line
       vs.set(cv2.CAP_PROP_POS_FRAMES, frame_counter)
    frame = resize(frame,(160,160,3))
    frame = np.expand_dims(frame,axis=0)
    if(np.max(frame)>1):
       frame = frame/255.0
    frames[i][:] = frame
    i +=1
  datav[0][:][:] = frames
  #frames -= mean
       # make predictions on the frame and then update the predictions
       # queue
```

```
preds = model1.predict(datav)
#
       print('Preds = :', preds)
#
       total = (preds[0] + preds[1] + preds[2] + preds[3] + preds[4] + preds[5])
#
       maximum = max(preds)
#
       rest = total - maximum
#
       diff = (.8*maximum) - (.1*rest)
       print('Difference of prob ', diff)
#
#
       th = 100
       if diff > .60:
#
#
               th = diff
#
       print('Old threshold = ', th)
  prediction = preds.argmax(axis=0)
  Q.append(preds)
       # perform prediction averaging over the current history of
       # previous predictions
  results = np.array(Q).mean(axis=0)
  print('Results = ', results)
  maxprob = np.max(results)
  print('Maximun Probability = ', maxprob)
  i = np.argmax(results)
  rest = 1 - maxprob
  diff = (maxprob) - (rest)
  print('Difference of prob ', diff)
  th = 100
  if diff > .80:
     th = diff
```

```
if (preds[0][1]) < th:
    text = "Alert : {} - {:.2f} % ".format((ok), 100 - (maxprob * 100))
    cv2.putText(output, text, (35, 50), cv2.FONT HERSHEY SIMPLEX, 1.25, (0, 255, 0), 5)
  else:
    text = "Alert : {} - {:.2f}%".format((okk), maxprob * 100)
    cv2.putText(output, text, (35, 50), cv2.FONT_HERSHEY_SIMPLEX, 1.25, (0, 0, 255), 5)
#
              if label != prelabel:
#
                     client.messages.create(to="<+country code>< receiver mobile number>",
#for example +91825555555
#
               from_="+180840084XX", #sender number can be coped from twilo
#
               body=\n'+ str(text) +\n Satellite: ' + str(camid) + \n Orbit: ' + location)
       # check if the video writer is None
  if writer is None:
         # initialize our video writer
    fourcc = cv2.VideoWriter_fourcc(*"MJPG")
    writer = cv2. VideoWriter('new.avi', fource, 27.0,
                     (W, H), True)
       # write the output frame to disk
  writer.write(output)
       # show the output image
  cv2.imshow("Output", output)
  key = cv2.waitKey(1) & 0xFF
```

# if the `q` key was pressed, break from the loop

```
if key == ord("q"):
    break

#print('Frame count', framecount)
# release the file pointers
print("[INFO] cleaning up...")

#writer.release()
vs.release()
vc.release()
```

# **Chapter 7**

#### **TESTING**

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. Testing is the final verification and validation activity within the organization itself. In the testing stage following goals are tried to achieve:-

- > To affirm the quality of the project.
- To find and eliminate any residual errors from previous stages.
- ➤ To validate the software as a solution to the original problem.
- > To provide operational reliability of the system.

During testing the major activities are concentrated on the examination and modification of the source code.

### 7.1 Types of tests

#### **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 its invasion. 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 perform accurately to the documented specification and contains clearly defined inputs and expected results.

#### **Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing its event driven and it's 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 specially aimed at exposing the problems that arise from the combination of components.

It is a testing in which the software under test is treated, as a black box, you cannot "see" into it. The test provides inputs and responds output without considering how the software works

#### **Unit testing**

Unit testing is usually conducted as part of a combine's code and unit test phase of the software lifecycle, although it's not uncommon for coding and unit testing to be conducted as two distinct phases.

#### 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 of the correct format
- No duplicate entries should be allowed

All links should take the user to 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 software system or-one step up-software applications at the company level-interact without error.

#### **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 much be exercised.
- Output: identified classes of application outputs must be exercised.
- Systems/procedures: interfacing systems or procedures must be invoked.

Organizations 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

#### **System Testing**

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 description and flows, emphasizing pre-driven process links and integration points.

#### White Box Testing

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

#### **Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner working, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document.

### **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.

☐ Testing the application from physically challenged person point of view as per (American Disability Act).

#### **Performance testing:**

Testing the stability and response time of application by applying load is known a Performance testing.

## 7.2 Unit Testing

Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing. In this step each module is found to work satisfactorily as regard to the expected output from the module. This testing is done to check for the individual block codes for their working. It is done so that when we carry out functional testing then the units which are part of these functionalities should have been tested for working.

# 7.3 Integration Testing

Classes are integrated to check whether interfaces between classes are working. The interactions are per sequence diagram of system

Classes Interated	Functions	Tests done	Remarks
Class: Main	loadPretrainedModel() DetectVideo() Alert()	The operations are Worked Correctly.	Success
Class: Detection	TraintheDatasetUsingCNN _LSTM() ClassifyTheInputvideo() AlertPopup()	The operations are Worked Correctly.	Success

# 7.4 Validation Testing

At the culmination of integration testing, software is completed and assembled as a package. Interfacing errors are uncovered and corrected. Validation testing can be defined in many ways. Here the testing validates the software function in a manner that is reasonably expected by the customer.

Functionality to be tested	Input	Tests done	Remarks
Working of Training and Build Model	Load the pretrained model and build	Pretrained model are trained and build a model	Success
		separately	
Working of Detection	Apply CNN_LSTM to detect the unusual or violence activity in the test video.	Test video is detected with CNN_LSTM.  Video belongs to which class has been find.	Success

**Table 7.2:- Validation testing table** 

### 7.5 Test Cases:

#### TABLE 7.3 Extract Dataset.

Test Case No.	1	
Name of Test Case	Extract Dataset	
Feature Being Tested	Checking leafs in dataset	
Expected Output	Should be Registered in the dataset	
Actual Output	Verified and profile page is displayed	
Remarks	Pass	

#### TABLE 7.4 Classification of Pest

Test Case No.	2
Name of Test Case	Verification of Leaf
Feature Being Tested	Checking Leaf format
Expected Output	Valid Leaf format is to be entered
Actual Output	Verify the Leaf format and check in the repository
Remarks	Pass

# **Chapter 8**

## INTERPRETATION OF RESULT

The following snapshots define the results or outputs that we will get after step by step execution of all the modules of the system.

**Interpretation:** 

**Add Snapshots HERE** 

# **Chapter 9**

### **CONCLUSION**

The systems which have been proposed till now are intended to recognize simple human action such as walking, running and many more but not suitable for crowded area. System which has been proposed is able to recognize unusual human action from crowd and action accordingly using motion influence map and OpenCV. The precision rate is bit higher than other and less researches have been made over this concept. Proposed system is able to work for Prior Appraisal against Crime. The accuracy is 96.42 % which is good enough for recognizing unusual activity in complex backgrounds. The proposed system is capable enough to efficiently recognize the unusual human activity from crowd by using OpenCV and Motion Influence Map, which enhances the accuracy and proficiency of the system up to a great extent. The Unusual Crowd Activity Detection can be implemented in various public places for prior and crime notification that enhances the casualty management. But accuracy is often important which requires enhancing for developing an ideal system that can be implemented practically.

#### **Future Work**

Further we can improvise with *Detection of a violence event in surveillance systems is playing a significant role in law enforcement and city safety.* 

## **References**

- [1] D. Lee, H. Suk, S. Park and S. Lee, "Motion Influence Map for Unusual Human Activity Detection and Localization in Crowded Scenes," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 10, pp. 1612-1623, Oct. 2015.
- [2] Donghui Song, "https://www.youtube.com/watch?v=y7Jo 9Z9z0U", Published on Jun 13, 2018.
- [3] D. Lee, H. Suk, S. Park and S. Lee, "Motion Influence Map for Unusual Human Activity Detection and Localization in Crowded Scenes," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 10, pp. 1612-1623, Oct. 2015.
- [4] Zakia Hammall, Wen-Sheng Chul, Jeffrey F. Cohnl, 2, Carrie Heike 3, and Matthew L. Speltz 4, "Automatic Action Unit Detection in Infants Using Convolutional Neural Network", 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII), IEEE.
- [5] He Xu1,2, Chengcheng Yuan1, Peng Li1,2, Yizhuo Wang3, "Design and Implementation of Action Recognition System Based on RFID Sensor", 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD 2017), IEEE.
- [6] Mrs. Varsha Shrirang Nanaware, Dr. Mohan Harihar Nerkar and Dr. C.M. Patil, "A Review of the Detection Methodologies of Multiple Human Tracking & Action Recognition in a Real Time Video Surveillance", IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI2017).
- [7] Jiahao Li<sup>†</sup>, Hejun Wu\* and Xinrui Zhou<sup>‡</sup>, "PeMapNet: Action Recognition from Depth Videos Using Pyramid Energy Maps on Neural Networks", 2017 International Conference on Tools with Artificial Intelligence, IEEE.
- [8] Nour El Din Elmadany, Yifeng He and Ling Guan, "Information Fusion for Human Action Recognition via Biset/Multiset Globality Locality Preserving Canonical Correlation Analysis", IEEE TRANSACTIONS ON IMAGE PROCESSING, 2018.
- [9] Soumalya Sen, Moloy Dhar and Susrut Banerjee, "Implementation of Human Action Recognition using Image Parsing Techniques", 2018 Emerging Trends in Electronic Devices and Computational Techniques
- [10] D. Aranki, G. Kurillo, P. Yan, D.M. Liebovitz, and R. Bajcsy, "Continuous, real-time, telemonitoring of patients with chronic heart-failure: lessons learned from a pilot study," in International Conference on Body Area Networks (Bodynets), 2014.