

# **INTELLIGENT EMPLOYEE ACTIVITY DETECTOR IN RESTAURANT USING PRETRAINED RESNET DEEP LEARNING MODEL**

A UG Project Phase – I report submitted to  
**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD**

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

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

Submitted By

**MALYALA CHANDANAPRIYA**

**18UK1A0529**

**VIJAYAGIRI SAI HARSHA**

**18UK1A0560**

**SHAIK ABBAS**

**18UK1A0550**

**GUDURU SHIVA DHANUSH**

**18UK1A0575**

Under the guidance of

**Mr. P.ILANNA**

Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VAAGDEVI ENGINEERING COLLEGE**

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
VAAGDEVI ENGINEERING COLLEGE  
WARANGAL**



**CERTIFICATE  
UG PROJECT PHASE-1**

This is to certify that the UG Project Phase – I report entitled "**INTELLIGENT EMPLOYEE ACTIVITY DETECTOR IN RESTAURANT USING PRETRAINED RESNET DEEP LEARNING MODEL**" is being submitted by **MALYALA CHANDANAPRIYA (18UK1A0529), VIJAYAGIRI SAI HARSHA (18UK1A0560), SHAIK ABBAS (18UK1A0550), GUDURU SHIVA DHANUSH (18UK1A0575)** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2021- 2022.

**Project Guide**

**Mr. P.ILANNA**

**HOD**

**Dr. R. NAVEEN KUMAR**

**EXTERNAL**

## ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. Prasad Rao**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase - I in the institute.

We extend our heartfelt thanks to **Dr. R. Naveen Kumar**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase - I.

We express heartfelt thanks to the Major Project Coordinator, **Dr. G. Aruna Kranthi**, Assistant Professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase - I.

We express heartfelt thanks to the guide, **Mr. P. Ilanna**, Assistant Professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase - I.

Finally, we express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experiencing throughout thesis.

## **ABSTRACT**

Activity recognition has been an emerging field of research since the past few decades. Humans have the ability to recognize activities from a number of observations in their surroundings. These observations are used in several areas like video surveillance, health sectors, gesture detection, energy conservation, fall detection systems and many more. Intelligent employee activity detector in restaurant is used to understand and analyze the activities performed in an restaurant. A step-by-step procedure is followed in this paper to build an intelligent employee activity detector. A general architecture of the Resnet model is explained first along with a description of its workflow. Convolutional neural network which is capable of classifying different activities is trained using the kinetic dataset which includes more than 400 classes of activities. The videos last around tenth of a second. The Resnet-34 model is used for image classification of convolutional neural networks and it provides shortcut connections which resolves the problem of vanishing gradient. The model is trained and tested successfully giving a satisfactory result by recognizing over 400 human actions. Finally, some open problems are presented which should be addressed in future research.

***Keywords: Video Surveillance, Resnet, Convolutional Neural Network, Kinetic Dataset.***

# TABLE OF CONTENTS

## LIST OF FIGURES

LIST OF CHAPTERS	PAGE NO
1. INTRODUCTION.....	1
1.1. MOTIVATION .....	1
1.2. DEFINITION .....	2
1.3. OBJECTIVE OF PROJECT .....	3
1.4. PURPOSE.....	4
2. PROBLEM STATEMENT .....	5
3. LITERATURE SURVEY.....	6
3.1. EXISTING SYSTEM.....	6
3.2. PROPOSED SOLUTION.....	9
4. EXPERIMENTAL ANALYSIS.....	12
4.1. PROJECT ARCHITECTURE .....	12
4.2. BLOCK DIAGRAM .....	13
4.3. SOFTWARE REQUIREMENTS.....	14
4.4. PROJECT FLOW .....	15

<b>5. DESIGN .....</b>	<b>17</b>
<b>5.1.CLASS DIAGRAM .....</b>	<b>17</b>
<b>5.2.USE CASE DIAGRAM .....</b>	<b>17</b>
<b>5.3.SEQUENCE DIAGRAM.....</b>	<b>18</b>
<b>5.4.FLOWCHART .....</b>	<b>19</b>
<b>6. CONCLUSION .....</b>	<b>19</b>
<b>7. FUTURE SCOPE.....</b>	<b>19</b>

## **LIST OF FIGURES**

## **PAGE NO:**

Figure 1: Types of Human Activity Recognition	1
Figure 2: Recognition of Human Activities	3
Figure 3: Recognition Procedure	5
Figure 4: Using SVM Classifier	6
Figure 5: Using Context Based Activity Recognition	7
Figure 6: Using Deep Learning	8
Figure 7: ResNet Procedure	9
Figure 8: Open CV Analyzing	11
Figure 9: Project Architecture	12
Figure 10: Block diagram representing process of Human Activity Recognition	13
Figure 11: Logos of python and VSCode and the base environment location in Anaconda	14
Figure 12: Project Flow Representation	16
Figure 13: Class diagram	17
Figure 14: Use Case Diagram	18
Figure 15: Sequence Diagram	18
Figure 16: Flowchart	19

# 1. INTRODUCTION

## 1.1. MOTIVATION

Human activity recognition has gained a wide range of attention in the past few decades. The data collected through activity monitoring can be used in several places like safe driving, controlling the crime rates, taking suitable actions during medical treatment and many more. Activity recognition also assists elderly people to make their life easier and simpler. Human beings have the potential to identify other human's activities through observation and communication. However, machines need to go through a learning phase to be able to recognize activities. Activity recognizer is capable of recognizing a wide range of activities like walking, applauding, reading, washing hands, and many more. The activities like handshaking, hugging and more comes under the category of human-to-human interaction in which two humans are involved whereas reading a book or newspaper comes under the category of human-to-object interaction in which one human and one object is involved. Some of these activities can either be simple or very complex. Complex activities may be broken down into simpler activities which will make them easier to understand.

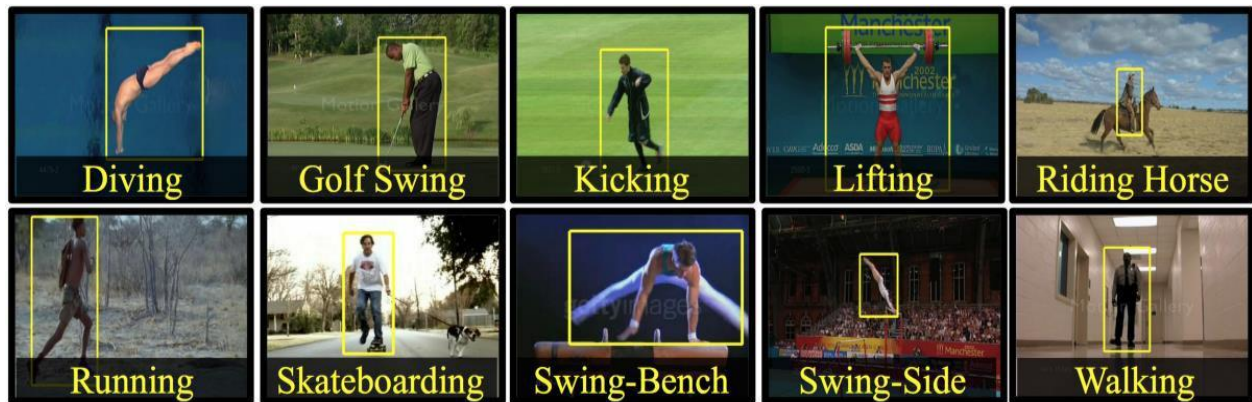


Figure 1: Types of Human Activity Recognition



## 1.2. DEFINITION

Human Activity Recognition (HAR) is the problem of identifying a physical activity carried out by an individual dependent on a trace of movement within a certain environment. Activities such as walking, laying, sitting, standing, and climbing stairs are classified as regular physical movements and form our class of activity which is to be recognized. To record movement or change in movement, sensors such as triaxial accelerometer and gyroscopes, capture data while the activity is being performed. A triaxial accelerometer data detects acceleration or movement along the three axes and a gyroscope measures rotation along the three axes to determine direction. Data recorded is along three dimensions of the X, Y and Z axis at the specified frequency. For example, a frequency of 20Hz would indicate that 20 data points are recorded each second of the action. Various other physiological signals such as heartbeat, respiration, etc. and environmental signals such as temperature, time, humidity, etc. can further augment the recognition process. Activity recognition can be achieved by exploiting the information retrieved from these sensors.

The challenge arises as there is no explicit approach to deduce human actions from sensor information in a general manner. The large volume of data produced from the sensors and use of these features to develop heuristics introduces the technical challenge. Storage, communication, computation, energy efficiency, and system flexibility are some of the aspects which need to be analyzed in detail to build a robust activity recognition system. Conventional pattern recognition methods have made tremendous progress in discovering significant information from scores of low-level readings. But such recognition models are successful for data collected in controlled environments, and for few activities only. Complex HAR tasks are hindered due to the naïve feature extraction techniques and limitation in domain knowledge. The shallow features extracted degrades the performance of unsupervised learning algorithms and connected activities. Deep learning models have the capabilities to learn features of the higher order. Advancement in such models makes it conceivable to learn and improve the performance of the predictive models and find deeper knowledge from human activities.

### 1.3.OBJECTIVE OF PROJECT:

Many researchers have contributed innovative algorithms and approaches in the area of human action recognition system and have conducted experiments on individual data sets by considering accuracy and computation. In spite of their efforts, this field requires high accuracy with less computational complexity. The existing techniques are inadequate in accuracy due to assumptions regarding clothing style, view angle and environment. Hence, the main objective of this thesis is to develop an efficient multi-view based human action recognition system using shape features. During the development phase, the following two objectives have been conceived in the proposed approach: Primary Objective – to develop an efficient human action recognition system using multiple views. Secondary Objective – to understand human behavior model using probabilistic action graph.

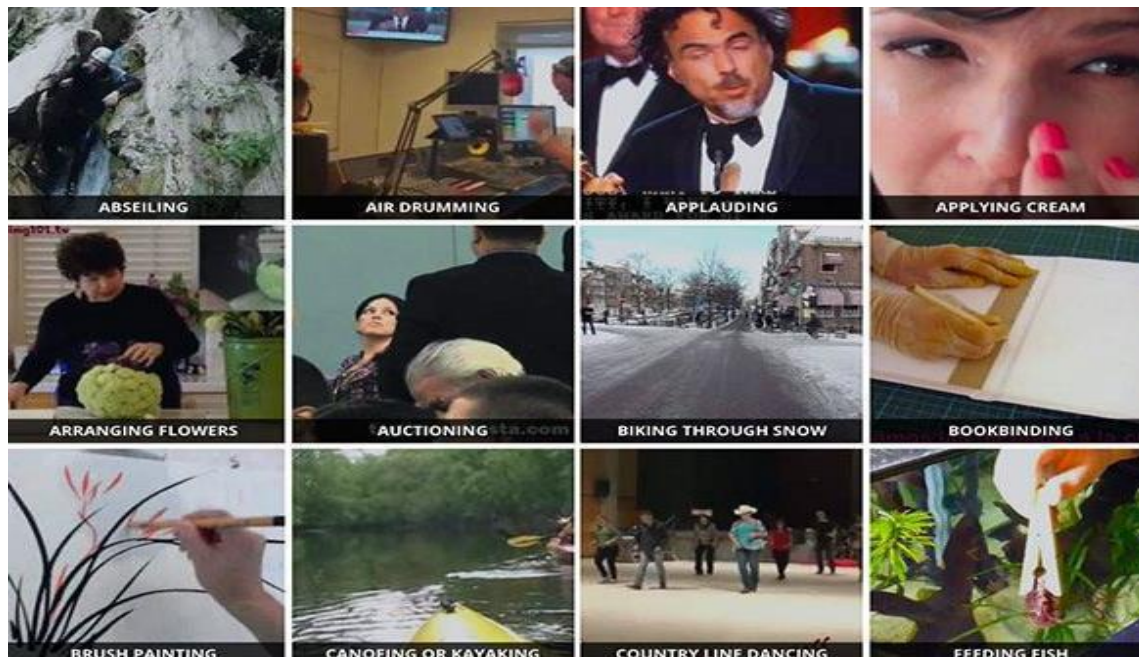


Figure 2: Recognition of Human Activities

## **1.4.PURPOSE:**

The purpose of this project is to create a model that can identify the human activities in the restaurant like washing hands, doughing pizza, cutting vegetables, serving food, eating, etc. The model will be predicting the activities that are performed in the video. The label of a video will be the action that is being performed in that particular video. The model will have to learn this relationship, and then it should be able to predict the label of an input (video) that it has never seen. Technically the model would have to learn to differentiate between various human actions in restaurant. Among various classification techniques two main questions arise: “What action?” (i.e., the recognition problem) and “Where in the video?” (i.e., the localization problem). When attempting to recognize human activities, one must determine the kinetic states of a person, so that the computer can efficiently recognize this activity.

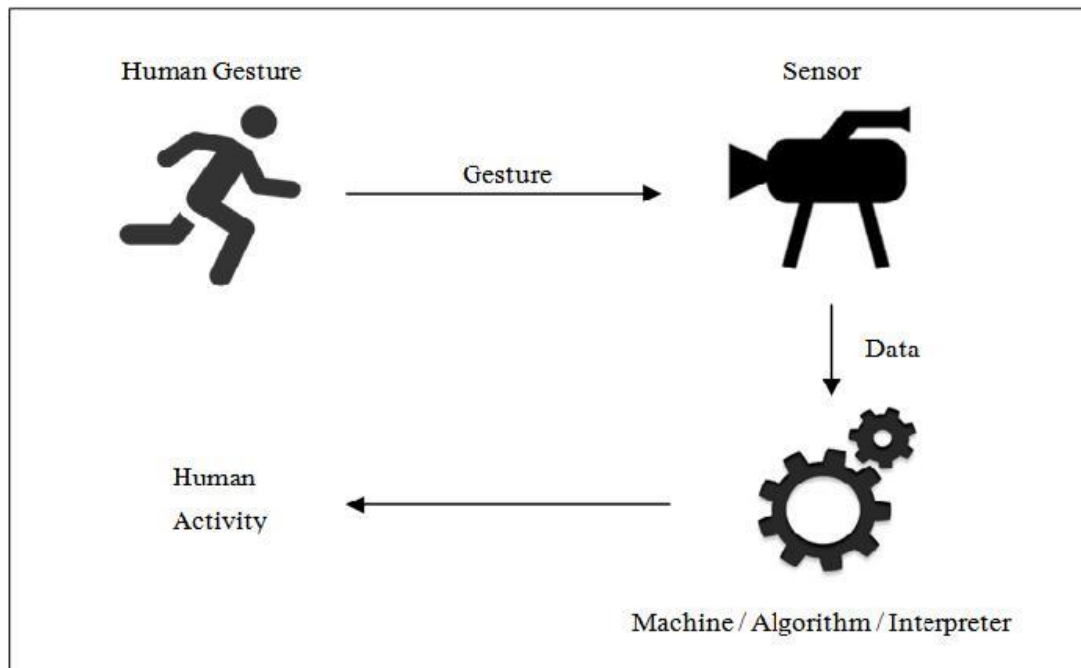
## 2. PROBLEM STATEMENT

Recognition of Human activities in a video scene of a surveillance system is attracting more attention due to its wide range of applications.

**Practical applications of human activity recognition include:**

- Automatically classifying/categorizing a dataset of videos on disk.
- Monitoring a new employee to correctly perform a task (ex., proper steps, and procedures when making a pizza, including rolling out the dough, heating the oven, putting on the sauce, cheese, toppings, etc.).
- Verifying that a food service worker has washed their hands after visiting the restroom or handling food that could cause cross-contamination (i.e. chicken and salmonella).
- Monitoring restaurant patrons and ensuring they are not over-served.

This project aims at making use of a pre-trained model to detect human activities in a video frame. We make use of the ResNet-34 pre-trained model which is trained on 400 human activities.

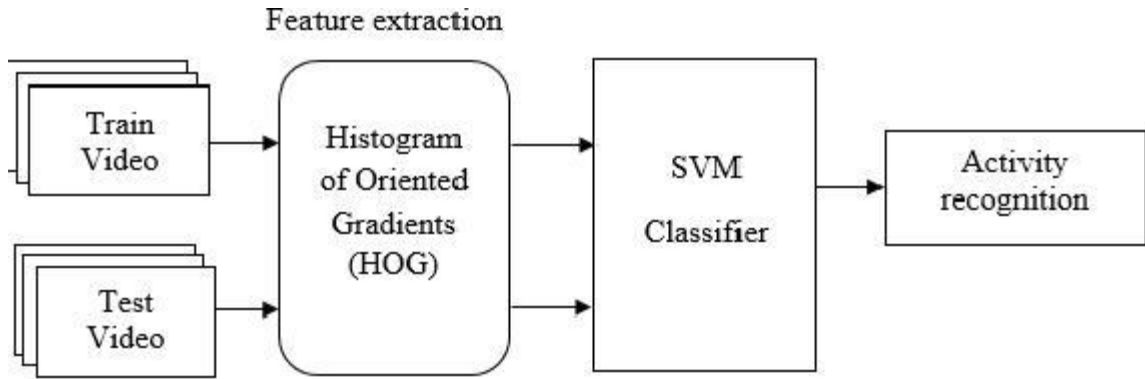


**Figure 3: Recognition Procedure**

### 3. LITERATURE SURVEY

#### 3.1. EXISTING SYSTEM

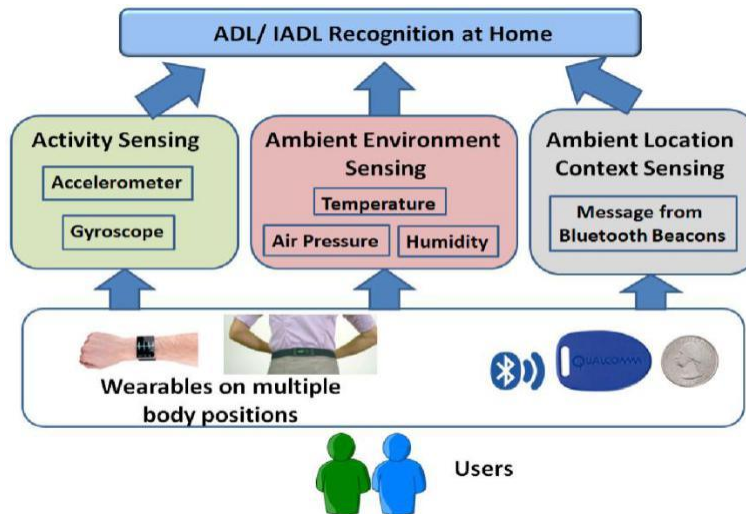
In human activity recognition systems, various lowlevel features are introduced to describe the activity observation. Schuldt et. al. (Schuldt et al., 2004) proposed a local space-time feature to represent the human movement observed in a video, and integrated such representations with SVM classification schemes for recognition. Laptev et. al. (Laptev et al., 2008a) proposed space-time feature point (STIP) and spatio-temporal bag-of-features as the descriptor for human motion. Tran et. al. (Tran et al., 2012) presented a framework for human action recognition based on modeling the motion of human body parts. They utilized a descriptor that combines both local and global representations of human motion, encoding the motion information as well as being robust to local appearance changes. The mentioned activity recognition methods mainly focus on recognizing the individual action. Their frameworks are difficult to scale to address real-world scenarios where multiple people activity and interaction are involved. Our approach represents the motion information using STIP feature similar to (Laptev et al., 2008a), but combines the rich context information that we extract from the video. By using the deep model, our method is able to: capture the extensive information about people motion and interactions; scale to recognize activity of each individual in the scene; and improve the accuracy of the overall activity recognition task.



**Figure 4: Using SVM Classifier**

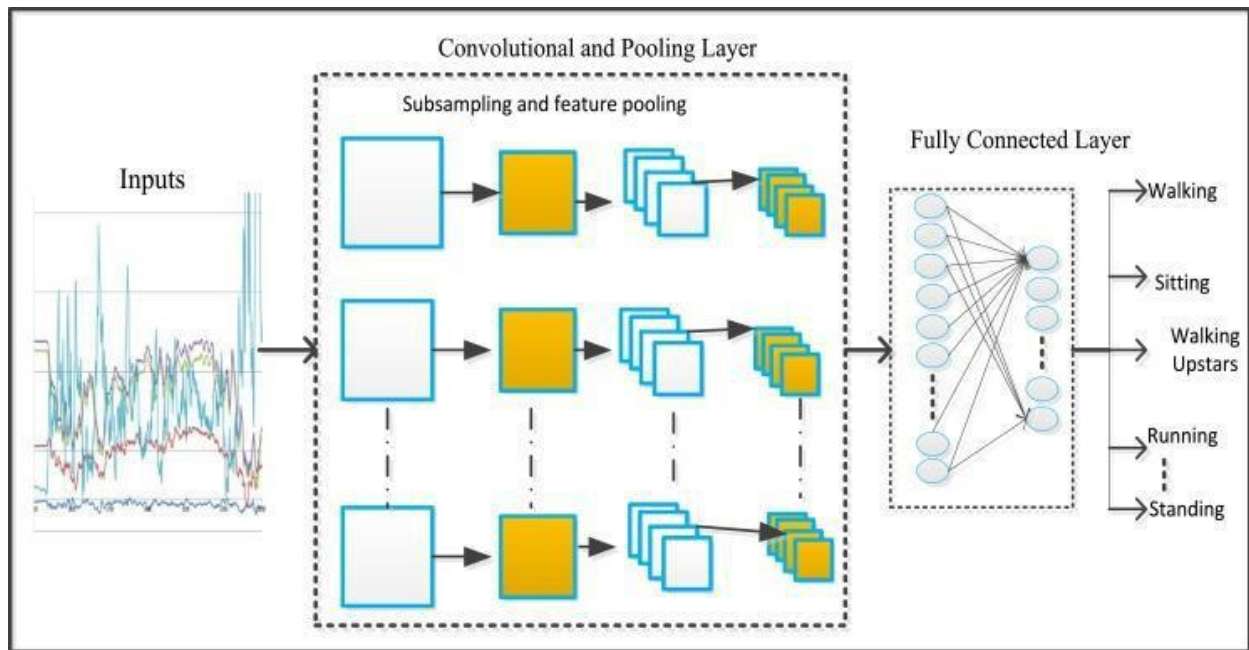
**Context based Activity Recognition.** Context information is widely utilized in many video analysis applications (Wei and Shah, 2015; Wei and Shah, 2016). In the topic of human activity recognition, many approaches integrate contextual information by proposing new feature

descriptors extracted from an individual and its surrounding area. Lan et. al. (Lan et al., 2012) proposed Action Context (AC) descriptor capturing the activity of the focal person and the behavior of other persons nearby. The AC descriptor is concatenating the focal person action probability vector with context action vectors that captures the nearby people action. Choi et. al. (Choi et al., 2009) propose Spatio-Temporal Volume (STV) descriptor, which captures spatial distribution of pose and motion of individuals in the scene to analyze group activity. STV descriptor centered on a person of interest is used to classify centered person's group activity. SVM with pyramid kernel is used for classification. The same descriptor is leveraged in (Choi et al., 2011), however, the random forest classification is used for group activity analysis. In (Lan et al., 2012; Choi et al., 2009; Choi et al., 2011), the nearby person that serves as context are selected according to the distance to the centered target. This does not necessarily ensure the existence of interactions among the selected persons. To address this issue, Tran et. al. (Tran et al., 2015) proposed group context activity descriptor similar to (Lan et al., 2012), but the people are first clustered into groups by modeling the social interaction among the persons. However, due to the noisy observation in videos, the group detection might not be robust or stable. Therefore, our approach utilizes the social interaction region to select the contextual people without a clustering process. Besides focusing on people as context, our approach also introduces scene information as context for the first time. The scene context describes the environment around the center target at the local and global levels. We utilize the existing place recognition method (Zhou et al., 2014) to provide scene context features that have semantic meanings.



**Figure 5: Using Context Based Activity Recognition**

**Deep Model for Activity Recognition.** Deep learning methods aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features. Automatically learning features at multiple levels of abstraction allow a system to learn complex functions mapping the input to the output directly from data, without depending completely on human-crafted features. Deep learning algorithms seek to exploit the unknown structure in the input distribution in order to discover good representations, often at multiple levels, with higher-level learned features defined in terms of lower-level features. The hierarchy of concepts allows the computer to learn complicated concepts by building them out of simpler ones. If we draw a graph showing how these concepts are built on top of each other, the graph is deep, with many layers. For this reason, we call this approach to AI deep learning. Deep learning excels on problem domains where the inputs (and even output) are analog. Meaning, they are not a few quantities in a tabular format but instead are images of pixel data, documents of text data or files of audio data. Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction.



**Figure 6: Using Deep Learning**

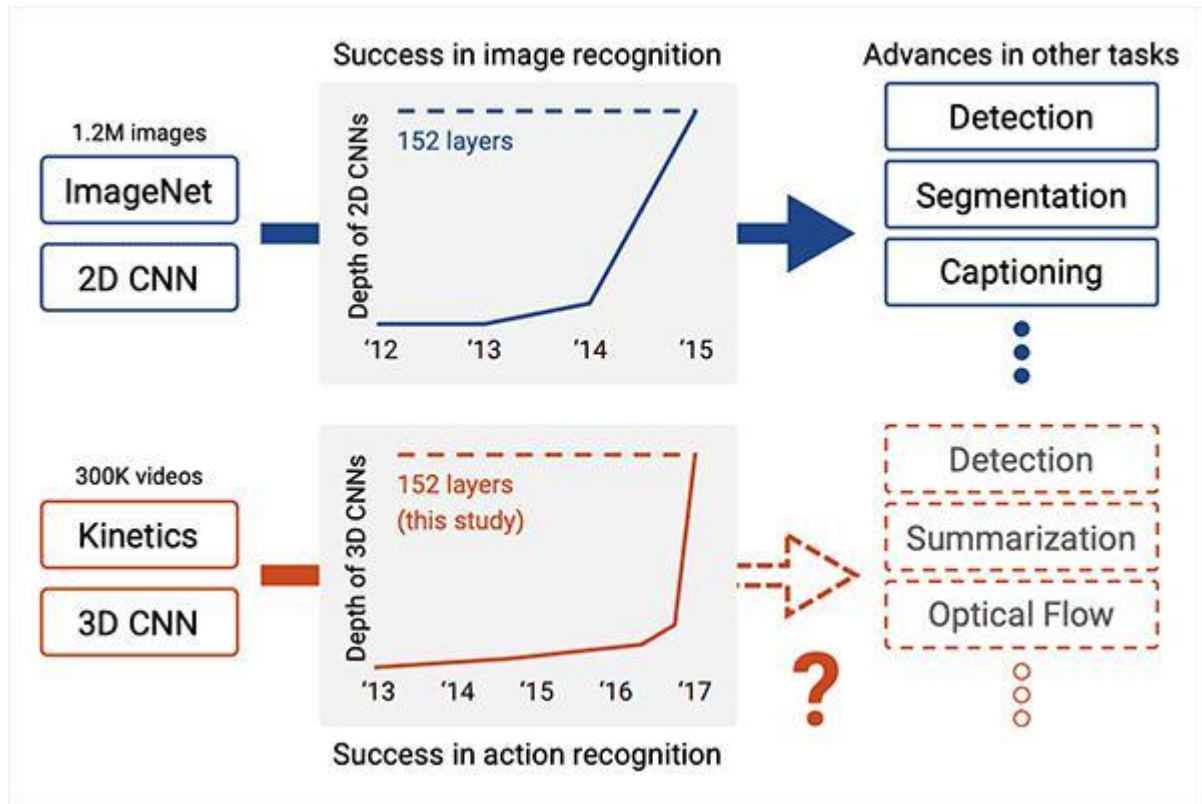


## 3.2. PROPOSED SOLUTION

### 3D ResNet for Human Activity Recognition

In this architecture shows how existing state-of-the-art 2D architectures (such as ResNet, ResNeXt, DenseNet, etc.) can be extended to video classification via 3D kernels. These architectures have been successfully applied to image classification.

- The large-scale ImageNet dataset allowed such models to be trained to such high accuracy.
- The Kinetics dataset is also sufficiently large.



**Figure 7: ResNet Procedure**

In order to determine whether current video datasets have sufficient data for training very deep convolution neural networks (CNNs) with spatio-temporal three-dimensional (3D) kernels. Recently, the performance levels of 3DCNNs in the field of action recognition have improved significantly. However, to date, conventional research has only explored relatively shallow 3D architectures. We examine the architectures of various 3D CNNs from relatively shallow to very



deep ones on current video datasets. Based on the results of those experiments, the following conclusions could be obtained:

- (i) ResNet-18 training resulted in significant over fitting for UCF-101, HMDB-51, and Activity Net but not for Kinetics.
- (ii) The Kinetics dataset has sufficient data for training of deep 3D CNNs, and enables training of up to 152 Res Nets layers, interestingly similar to 2D ResNets on ImageNet. ResNeXt-101 achieved 78.4% average accuracy on the Kinetics test set.
- (iii) Kinetics pretrained simple 3D architectures outperforms complex 2D architectures, and the pretrained ResNeXt-101 achieved 94.5% and 70.2% on UCF-101 and HMDB-51, respectively. The use of 2D CNNs trained on ImageNet has produced significant progress in various tasks in image. We believe that using deep 3D CNNs together with Kinetics will retrace the successful history of 2D CNNs and ImageNet, and stimulate advances in computer vision for videos

## **OpenCV**

Human activity Recognition can be done using one of the 2 techniques.

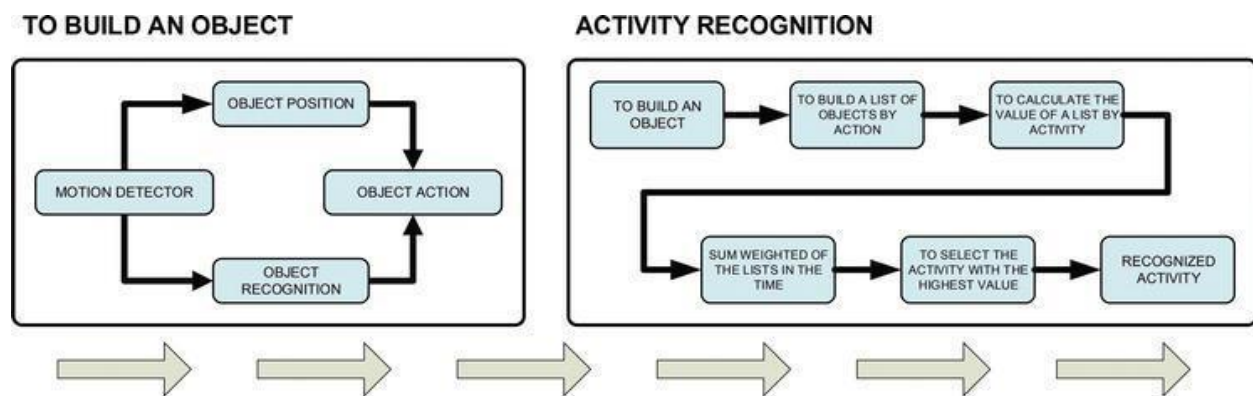
(i)Template Matching Technique: The template matching technique convert an picture(image) sequence into a static shape pattern here instead of using GMM we will use HMM(Hidden Markov Model and optical flow For defining the sequence of the data in the separated frames.) and then compare the value of the static picture with that of the values previously stored in the trained data-set, when the value of the data set matches the value of the data the blobs displays the derived result. The advantage of using the template matching procedure is that it takes less computational power of the system but it is still reactive to the temporal anomaly discussed above.

(ii)State-Space Model defines each Stationary static pose as a single state. This stationary pose is relevant to each frame formed by HMM These states are connected by certain Possibilities such as the activities will all have a predefined number and other activities surrounding that number will form a chain of events likely to happen and hence increasing the probability of recognition and also making prediction a reality. Any motion sequence taken into account as a tour going through these states. Joint expectation is to be calculated

through all these tours and the value cost maximum and closest to the values in the data-set is chosen as the criteria for classifying activities. In such a scenario, temporal anomaly of motion 14 does not raise any issue because each state on loop visits itself in repetition. Hence this method of state-space

The model is reliable against temporary anomalies. below are the broad steps of the projected technique :

- (1)Pre processing
- (2)Feature Extraction
- (3)Human activity Recognition



**Figure 8: Open CV Analyzing**

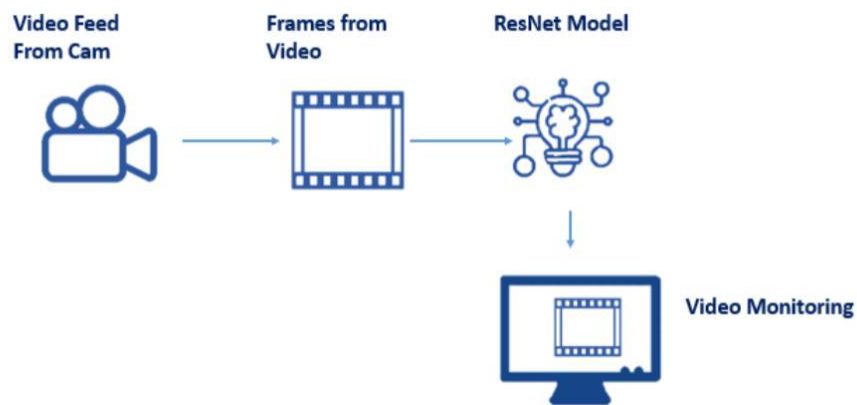
## 4. EXPERIMENTAL ANALYSIS

Human activity analysis is one of the most important problems that has received considerable attention from the computer vision community in recent years. It has various applications, spanning from activity understanding for intelligent surveillance systems to improving human-computer interactions. The project aims to develop a model to analyze the activities of humans in Restaurants.

### 4.1. PROJECT ARCHITECTURE:

The Project Architecture briefly explains the procedure involved:

- Grabbing Video
- Preprocessing
- Construction of frames
- testing frame (in accordance to data set)
- Feature Extraction
- Human Activity Recognition
- Classification of human activity(in acc. To data-set)

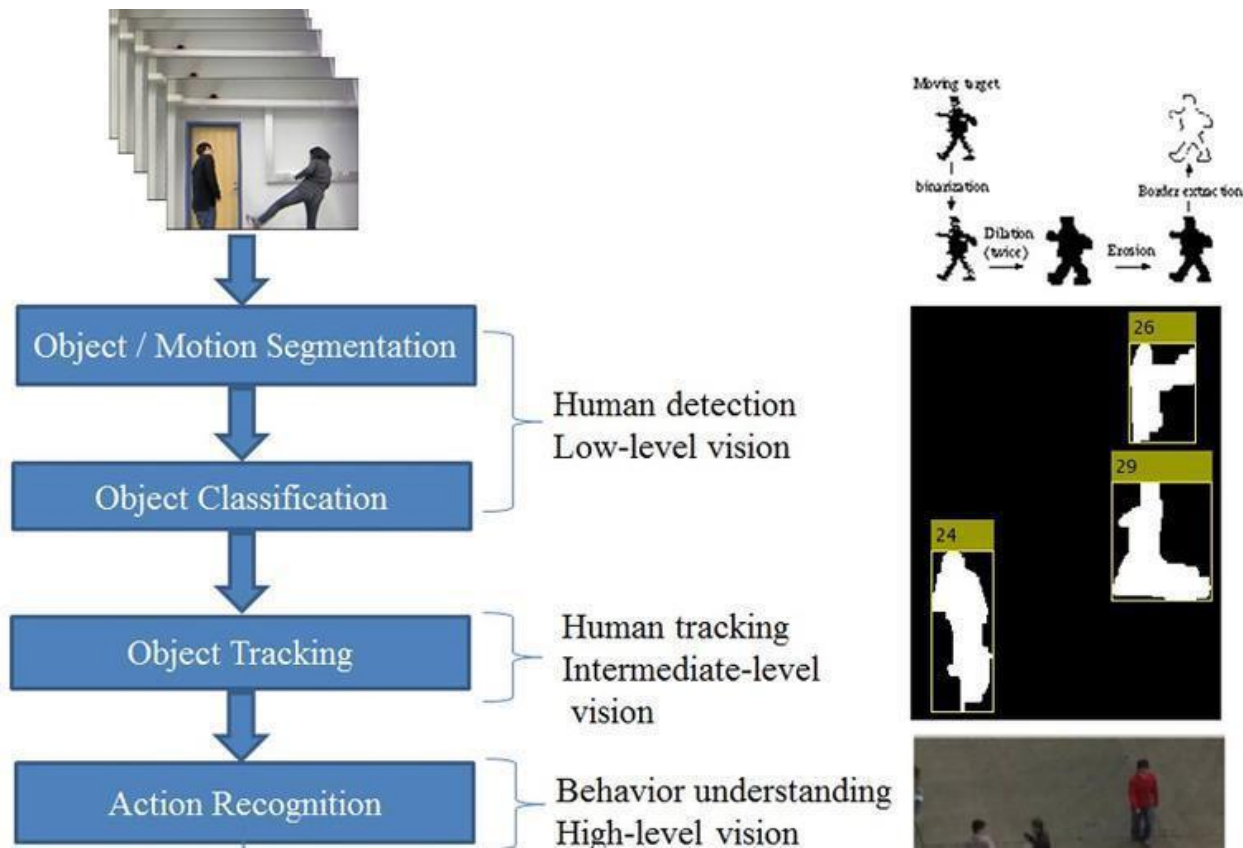


**Figure 9: Project Architecture**

## 4.2. BLOCK DIAGRAM

Block diagram represents the procedure in systematic and sequential manner with its blocks connected by lines that show the relationship of the blocks.

- Input Video
- Object/Motion Segmentation
- Object Classification
- Object Tracking
- Action Recognition



**Figure 10: Block diagram representing process of Human Activity Recognition**

### 4.3. SOFTWARE REQUIREMENTS

➤ Python 3.9:

- Python is an interpreted high-level general-purpose programming language.
- Python can be used on a server to create web applications.

➤ Visual Studio Code:

- Visual studio code is a source-code editor made by Microsoft for Windows, linux and macOS.
- Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

➤ Anaconda Environment

- The default environment base (path) is used because it consists of multiple libraries and modules.

➤ Keras Modules:

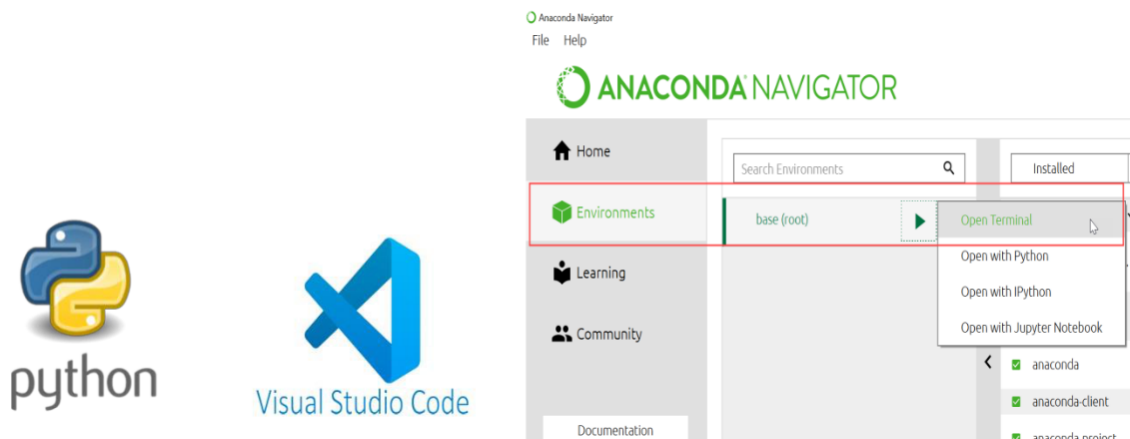
- Keras is used as a backend for the pre-trained model.

➤ Open CV

- For image processing and loading of images opencv is used.

➤ Imutils:

- Basic image processing functions such as translation, rotation, resizing etc



**Figure 11: Logos of python and VSCode and the base environment location in Anaconda**

## **4.4. PROJECT FLOW**

### **1) Video Grabbing**

The video data from the dataset or recorded surveillance videos is taken into consideration. It is a finding that if the data is supervised the results will be better than that of the unsupervised data(video).

### **2) Preprocessing**

The process leads on with the first step of importing necessary packages of numpy, argparse, imutils, sys, opencv2, after which the construction of the argument parser to parse the arguments takes place, using cvHMM version will eventually provide us with the preconfigured code settings for the dataset.

### **3) Construction of The Frames/blobs:**

2D blobs are the most commonly used feature (low level) for recognition of human activity, that is why we generally come across it as the first stage. The dilation in blob is for the enhancement of the frame, dilation can be done easily via masking or by applying a filter it is only after dilation that we obtain a 2D blob. Blob segments the frame(here we are taking one frame of the video set in consideration) into foreground and Background & the net median numerical video. Blobs are multidimensional arrays or data.

### **4) Testing Of The Frames**

After loading the contents of the class label, it is advisable to define the sample duration that is defining the number of frames for classification and sample size just to save the computational costs. loading it into human activity recognition model in order to test the data, after this it would provide a better gui experience for the user as well.

### **5) Feature Extraction**

After the classification of the segments in the blob the next stage in the sequel is of feature extraction, here the numerical median of the blob in motion is taken into consideration as the value for the recognition of activity is best described by the blob rather than the colour or the size of the actor. Here the feature of” Motion/Activity/Movement” of the actor in the blob is done. Here as previously mentioned to go from one video frame to another we use optical flow which is nothing but the usage of the HMM in between of the frames, following are the popular methods for finding optical flow (i)Horn-Schunck Technique (ii) Lucas-Kanade Technique 17

Horn Schunck technique is used for floating point input & Lucas-Kanade for otherwise (I.e for fixed point input. )

#### 6) Action Recognition:

This in Sequence is after the ‘Feature Extraction’ where the activity/Movement which was the median numerical number of the blob is extracted, here then by using the optical flow of the Lucas kanade Method & also for human activity recognition we use Hidden Markov Model.

#### 7) Action classification

The “Activity/Motion/Movement” is classified due to the median of the blob which is then compared to the already stored numerical values of the pre-trained data-set. each activity has a corresponding numerical value to it, which when matched with the value given by the blobs results in itself classifying the activity in Observation.

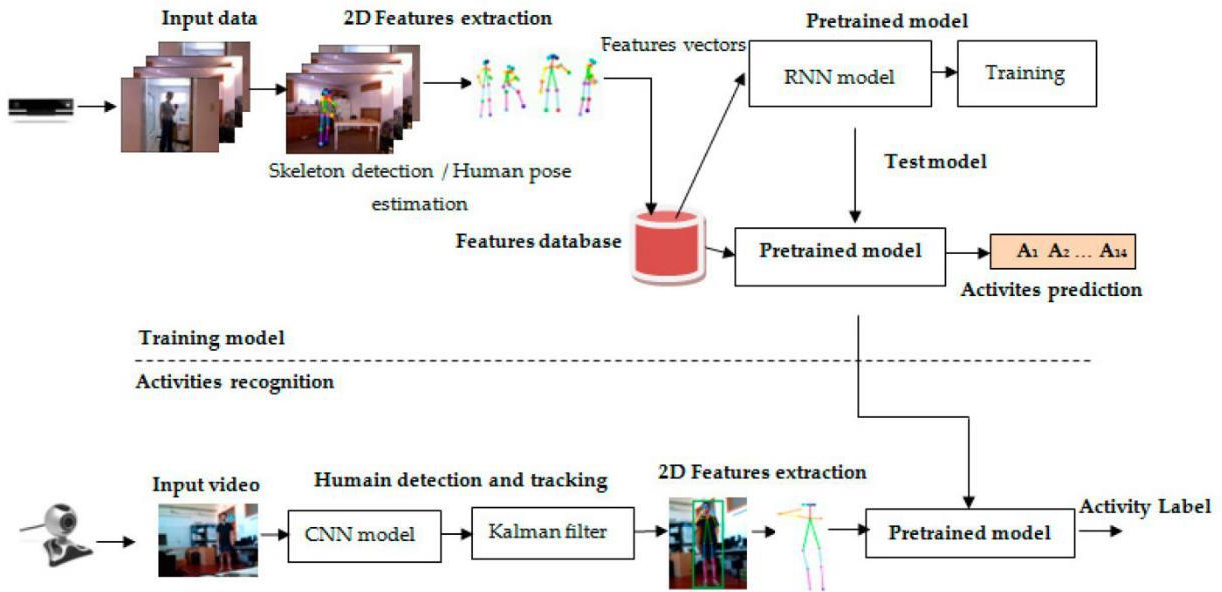
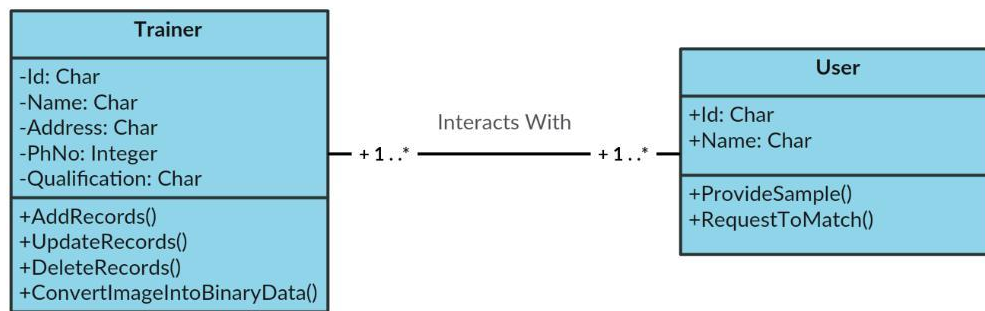


Figure 12: Project Flow Representation

## 5. DESIGN

### 5.1. CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.



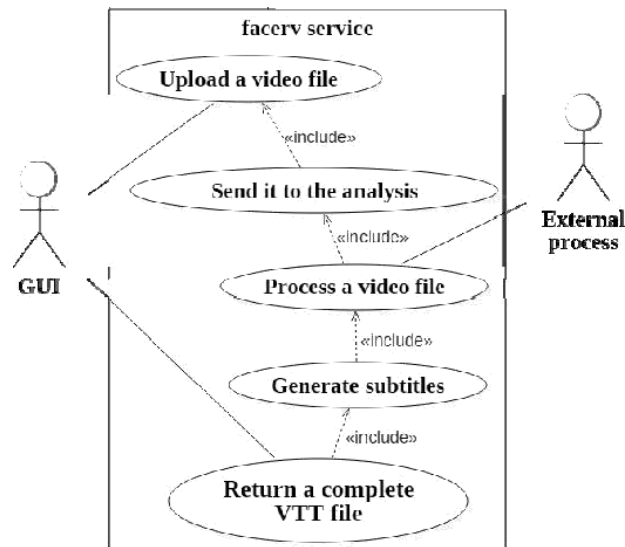
**Figure 13: Class diagram**

### 5.2. USE CASE DIAGRAM

A use case diagram is usually simple. It does not show the detail of the use cases:

- It only summarizes some of the relationships between use cases, actors, and systems.
- It does not show the order in which steps are performed to achieve the goals of each use case.

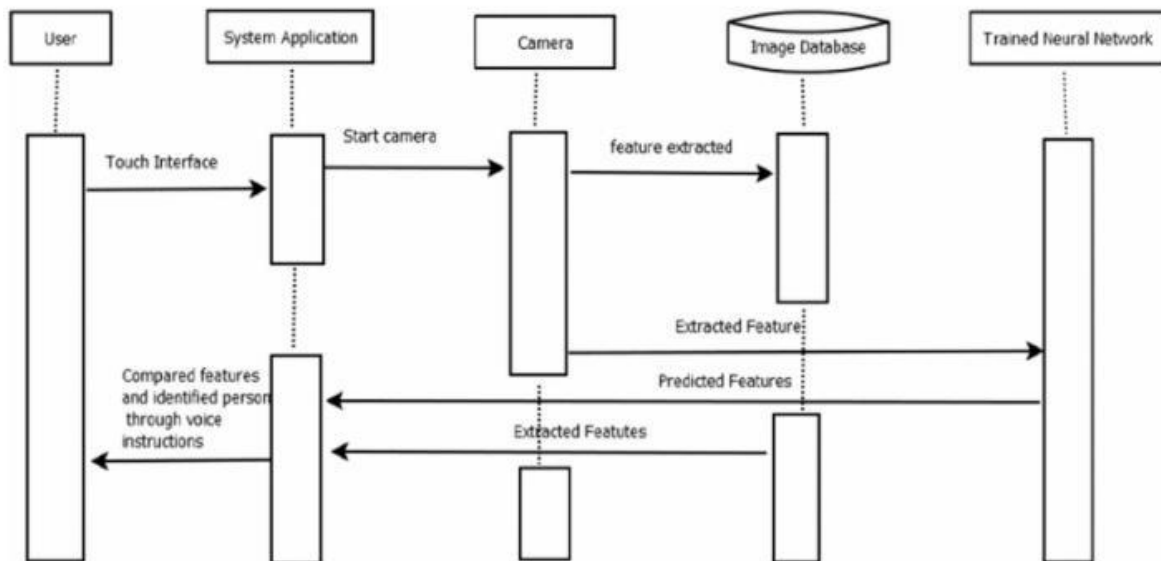




**Figure 14: Use Case Diagram**

### 5.3.SEQUENCE DIAGRAM

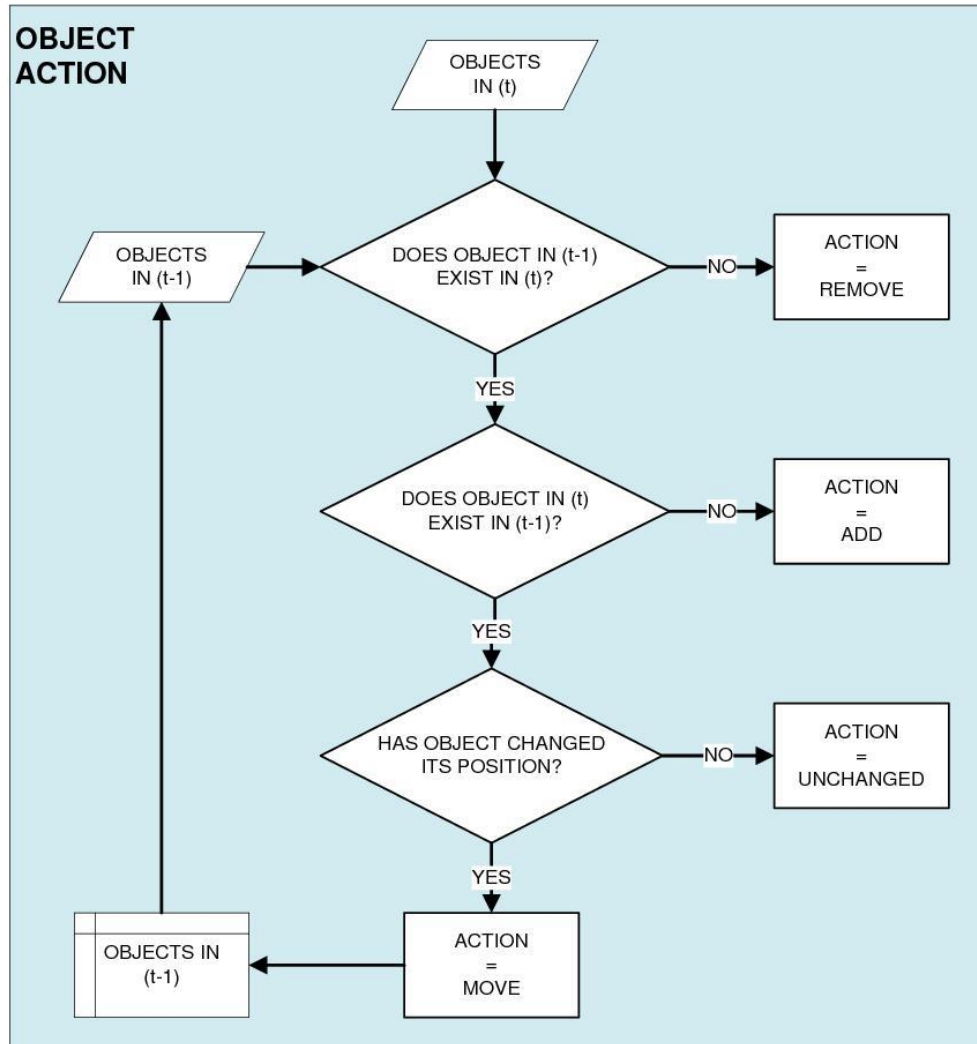
A **sequence diagram** or **system sequence diagram (SSD)** shows object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario.



**Figure 15: Sequence Diagram**

## 5.4. FLOWCHART

A flowchart is a picture of the separate steps of a process in sequential order.



**Figure 16: Flowchart**

## **6. CONCLUSION**

In UG Project Phase-1, we have worked on problem statement, literature survey and also done the experimental analyses which are required for the project to move forward. In experimental analysis we have discussed about the ResNet model and explained the algorithms to be used in the project. We also discussed about the flowcharts, use case diagrams, decision tree and sequence diagrams which are used in the project. Based on the experimental analysis we have designed the model for the project. Entire designing part is involved in UG Project Phase-1.

## **7. FUTURE SCOPE**

UG Project Phase-2 is the extension of UG Project Phase-1. UG Project Phase-2 involves all the coding and implementation of the design which we have retrieved from UG Project Phase-1. All the implementation is done and conclusions will be retrieved in the phase. We will also work on the applications, advantages, and disadvantages of the project in this phase. Future scope of the project will be also discussed in the UG Project Phase-2.

# **INTELLIGENT EMPLOYEE ACTIVITY DETECTOR IN RESTAURANT USING PRETRAINED RESNET DEEP LEARNING MODEL**

A UG Project Phase – II report submitted to  
**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD**

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

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

Submitted By

**MALYALA CHANDANAPRIYA**

**18UK1A0529**

**VIJAYAGIRI SAI HARSHA**

**18UK1A0560**

**SHAIK ABBAS**

**18UK1A0550**

**GUDURU SHIVA DHANUSH**

**18UK1A0575**

Under the guidance of

**Mr. P.ILANNA**

Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VAAGDEVI ENGINEERING COLLEGE**

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
VAAGDEVI ENGINEERING COLLEGE  
WARANGAL**



**CERTIFICATE  
UG PROJECT PHASE-II**

This is to certify that the UG Project Phase – II report entitled "**INTELLIGENT EMPLOYEE ACTIVITY DETECTOR IN RESTAURANT USING PRETRAINED RESNET DEEP LEARNING MODEL**" is being submitted by **MALYALA CHANDANAPRIYA (18UK1A0529), VIJAYAGIRI SAI HARSHA (18UK1A0560), SHAIK ABBAS (18UK1A0550), GUDURU SHIVA DHANUSH (18UK1A0575)** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2021- 2022.

**Project Guide**

**Mr. P.ILANNA**

**HOD**

**Dr. R. NAVEEN KUMAR**

**EXTERNAL**

## ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. Prasad Rao**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase - II in the institute.

We extend our heartfelt thanks to **Dr. R. Naveen Kumar**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase - II.

We express heartfelt thanks to the Major Project Coordinator, **Dr. G. Aruna Kranthi**, Assistant Professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase - II.

We express heartfelt thanks to the guide, **Mr. P. Ilanna**, Assistant Professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase - II.

Finally, we express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experiencing throughout thesis.

## **ABSTRACT**

Activity recognition has been an emerging field of research since the past few decades. Humans have the ability to recognize activities from a number of observations in their surroundings. These observations are used in several areas like video surveillance, health sectors, gesture detection, energy conservation, fall detection systems and many more. Intelligent employee activity detector in restaurant is used to understand and analyze the activities performed in an restaurant. A step-by-step procedure is followed in this paper to build an intelligent employee activity detector. A general architecture of the Resnet model is explained first along with a description of its workflow. Convolutional neural network which is capable of classifying different activities is trained using the kinetic dataset which includes more than 400 classes of activities. The videos last around tenth of a second. The Resnet-34 model is used for image classification of convolutional neural networks and it provides shortcut connections which resolves the problem of vanishing gradient. The model is trained and tested successfully giving a satisfactory result by recognizing over 400 human actions. Finally, some open problems are presented which should be addressed in future research.

***Keywords: Video Surveillance, Resnet, Convolutional Neural Network, Kinetic Dataset.***

# **TABLE OF CONTENTS**

## **LIST OF FIGURES**

## **LIST OF CHAPTERS**

## **PAGE NO**

<b>1. INTRODUCTION.....</b>	<b>1</b>
<b>2. CODE SNIPPETS .....</b>	<b>2</b>
<b>2.1. MODEL CODE.....</b>	<b>2</b>
<b>2.2. INPUT CODE.....</b>	<b>2</b>
<b>2.3. PYTHON CODE.....</b>	<b>2</b>
<b>2.4. TEXT FILE .....</b>	<b>4</b>
<b>3. CONCLUSION .....</b>	<b>11</b>
<b>4. APPLICATION.....</b>	<b>13</b>
<b>5. ADVANTAGES.....</b>	<b>13</b>
<b>6. DIS-ADVANTAGES.....</b>	<b>13</b>
<b>7. FUTURE SCOPE.....</b>	<b>14</b>
<b>8. BIBILIOGRAPHY .....</b>	<b>14</b>



## **LIST OF FIGURES**

## **PAGE NO:**

Figure 1: Importing libraries, pre trained model, and video file	2
Figure 2: loop over frames and adding frames into queue	3
Figure 3: Passing the blobs from the frames to obtain human activity recognition	3
Figure 4: 32 ACTIVITIES (1 st SET)	4
Figure 5: 32 ACTIVITIES (2 nd SET)	4
Figure 6: 32 ACTIVITIES (3 rd SET)	5
Figure 7: 32 ACTIVITIES (4 th SET)	5
Figure 8: 32 ACTIVITIES (5 th SET)	6
Figure 9: 32 ACTIVITIES (6 th SET)	6
Figure 10: 32 ACTIVITIES (7 th SET)	7
Figure 11: 32 ACTIVITIES (8 th SET)	7
Figure 12: 32 ACTIVITIES (9 th SET)	8
Figure 13: 32 ACTIVITIES (10 th SET)	8
Figure 14: 32 ACTIVITIES (11 th SET)	9
Figure 15: 32 ACTIVITIES (12 th SET)	9
Figure 16: 16 ACTIVITIES (13 th SET)	10
Figure 17: Pizza making	11
Figure 18: Washing hands	12
Figure 19: Tasting Beer	12

# 1. INTRODUCTION

Human Activity Recognition (HAR) is the problem of identifying a physical activity carried out by an individual dependent on a trace of movement within a certain environment. Activities such as walking, laying, sitting, standing, and climbing stairs are classified as regular physical movements and form our class of activity which is to be recognized. To record movement or change in movement, sensors such as triaxial accelerometer and gyroscopes, capture data while the activity is being performed. A triaxial accelerometer data detects acceleration or movement along the three axes and a gyroscope measures rotation along the three axes to determine direction. Data recorded is along three dimensions of the X, Y and Z axis at the specified frequency. For example, a frequency of 20Hz would indicate that 20 data points are recorded each second of the action. Various other physiological signals such as heartbeat, respiration, etc. and environmental signals such as temperature, time, humidity, etc. can further augment the recognition process. Activity recognition can be achieved by exploiting the information retrieved from these sensors.

In this architecture shows how existing state-of-the-art 2D architectures (such as ResNet, ResNeXt, DenseNet, etc.) can be extended to video classification via 3D kernels. These architectures have been successfully applied to image classification.

- The large-scale ImageNet dataset allowed such models to be trained to such high accuracy.
- The Kinetics dataset is also sufficiently large.

UG Project Phase-2 involves all the coding and implementation of the design which we have retrieved from UG Project Phase-1. All the implementation is done and conclusions are retrieved in this phase. We will also work on the applications, advantages, and disadvantages of the project in this phase. Future scope of the project will be also discussed in the UG Project Phase-2.

## 2. CODE SNIPPETS

### 6.1. PRE TRAINED MODEL CODE

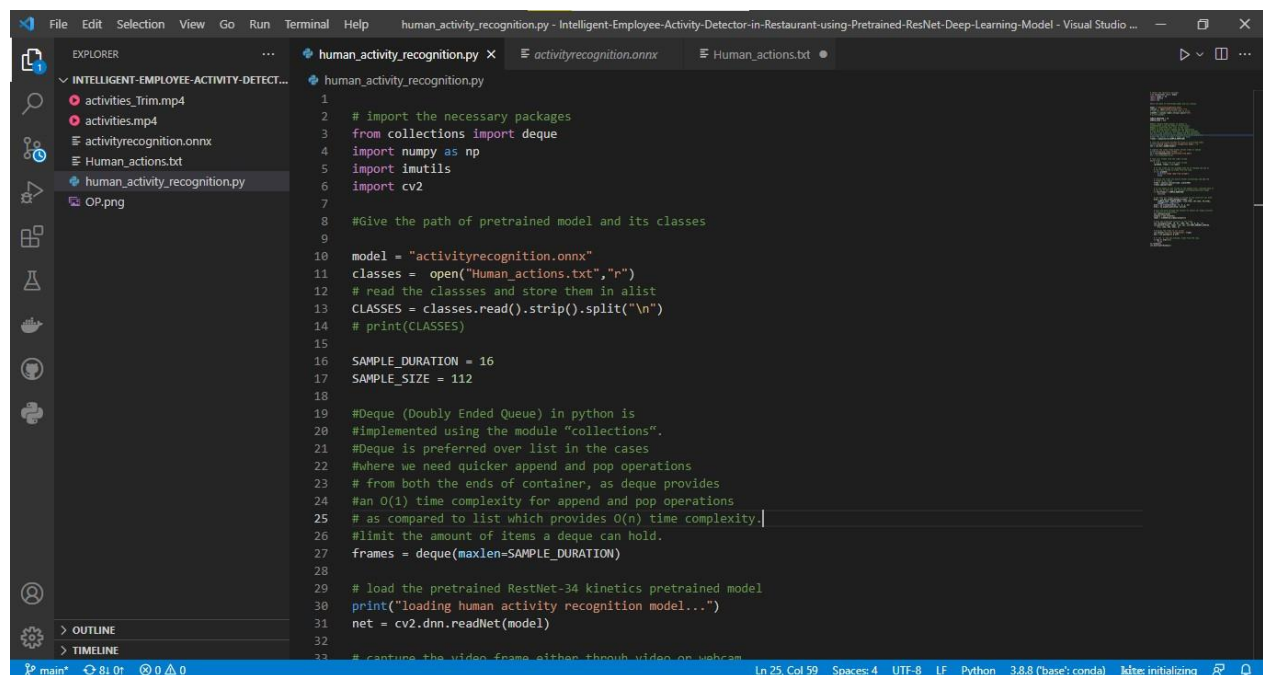
- **ACTIVITYRECOGNITION.ONNX**

### 6.2. INPUT VIDEO FILE:

- **ACTIVITIES\_TRIM.MP4**

### 6.3. PYTHON CODE:

- **HUMAN\_ACTIVITY\_RECOGNITION.PY**



```
1
2 # import the necessary packages
3 from collections import deque
4 import numpy as np
5 import imutils
6 import cv2
7
8 #Give the path of pretrained model and its classes
9
10 model = "activityrecognition.onnx"
11 classes = open("Human_actions.txt","r")
12 # read the classes and store them in alist
13 CLASSES = classes.read().strip().split("\n")
14 # print(CLASSES)
15
16 SAMPLE_DURATION = 16
17 SAMPLE_SIZE = 112
18
19 #Deque (Doubly Ended Queue) in python is
20 #implemented using the module "collections".
21 #Deque is preferred over list in the cases
22 #where we need quicker append and pop operations
23 # from both the ends of container, as deque provides
24 #an O(1) time complexity for append and pop operations
25 # as compared to list which provides O(n) time complexity.
26 #limit the amount of items a deque can hold.
27 frames = deque(maxlen=SAMPLE_DURATION)
28
29 # load the pretrained ResNet-34 kinetics pretrained model
30 print("loading human activity recognition model...")
31 net = cv2.dnn.readNet(model)
32
33 # capture the video frame either through video or webcam
```

Figure 1: Importing libraries, pre trained model, and video file

```

33 # capture the video frame either through video or webcam
34 print("accessing video stream...")
35 vs = cv2.VideoCapture(r"activities_Trim.mp4")
36 #vs = cv2.VideoCapture(0)
37
38 # loop over frames from the video stream
39 while True:
40     # read a frame from the video stream
41     (grabbed, frame) = vs.read()
42
43     # if the frame was not grabbed then we've reached the end of
44     # the video stream so break from the loop
45     if not grabbed:
46         print("no frame read from stream")
47         break
48
49     # resize the frame (to ensure faster processing) and add the
50     # frame to our queue
51     frame = imutils.resize(frame, width=800)
52     frames.append(frame)
53
54     # if our queue is not filled to the sample size, continue back to
55     # the top of the loop and continue polling/processing frames
56     if len(frames) < SAMPLE_DURATION:
57         continue
58
59     # now that our frames array is filled we can construct our blob
60     blob = cv2.dnn.blobFromImages(frames, 1.0,
61                                   (SAMPLE_SIZE, SAMPLE_SIZE), (114.7748, 107.7354, 99.4750),
62                                   swapRB=True, crop=True)
63     blob = np.transpose(blob, (1, 0, 2, 3))
64     blob = np.expand_dims(blob, axis=0)

```

**Figure 2: loop over frames and adding frames into queue**

```

54     # if our queue is not filled to the sample size, continue back to
55     # the top of the loop and continue polling/processing frames
56     if len(frames) < SAMPLE_DURATION:
57         continue
58
59     # now that our frames array is filled we can construct our blob
60     blob = cv2.dnn.blobFromImages(frames, 1.0,
61                                   (SAMPLE_SIZE, SAMPLE_SIZE), (114.7748, 107.7354, 99.4750),
62                                   swapRB=True, crop=True)
63     blob = np.transpose(blob, (1, 0, 2, 3))
64     blob = np.expand_dims(blob, axis=0)
65
66     # pass the blob through the network to obtain our human activity
67     # recognition predictions
68     net.setInput(blob)
69     outputs = net.forward()
70     label = CLASSES[np.argmax(outputs)]
71
72     # draw the predicted activity on the frame
73     cv2.rectangle(frame, (0, 0), (300, 40), (0, 0, 0), -1)
74     cv2.putText(frame, label, (10, 25), cv2.FONT_HERSHEY_SIMPLEX,
75               0.8, (255, 255, 255), 2)
76
77     # display the frame to our screen
78     cv2.imshow("Activity Recognition", frame)
79     key = cv2.waitKey(1) & 0xFF
80
81     # if the 'q' key was pressed, break from the loop
82     if key == ord("q"):
83         break
84     vs.release()
85     cv2.destroyAllWindows()

```

**Figure 3: Passing the blobs from the frames to obtain human activity recognition**

## 6.4. TEXT FILE WITH LIST OF ACTIVITIES:

- **HUMAN\_ACTIONS.TXT**

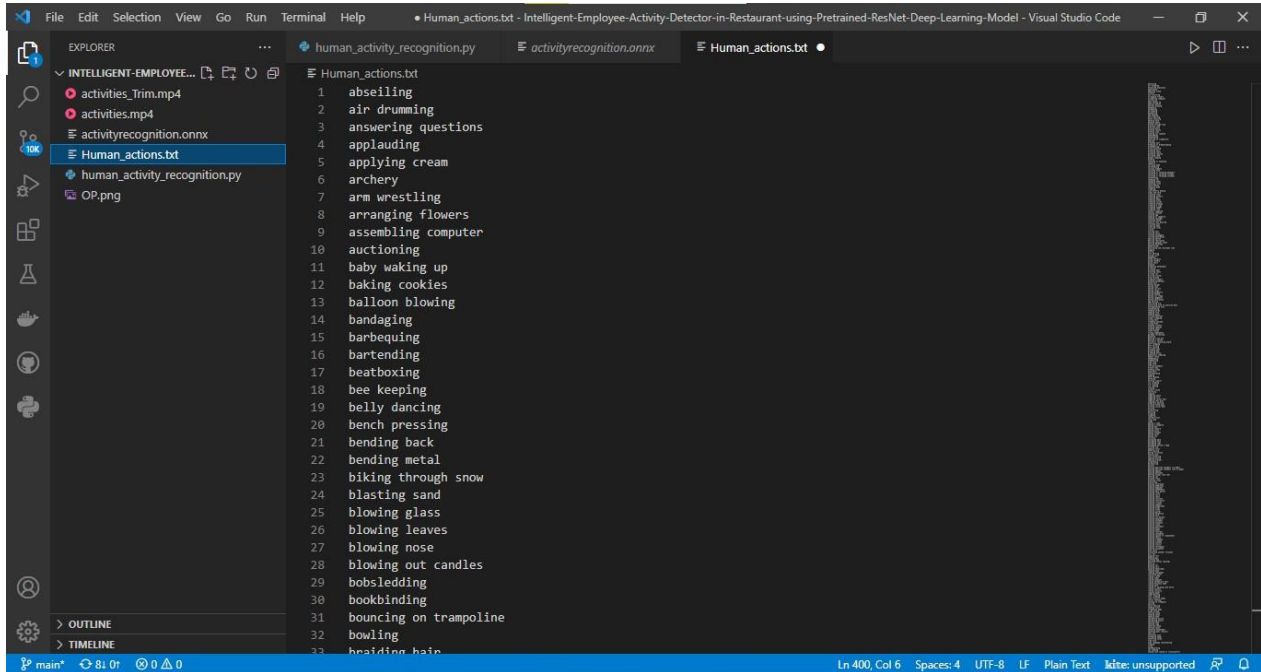


Figure 4: 32 ACTIVITIES (1 st SET)

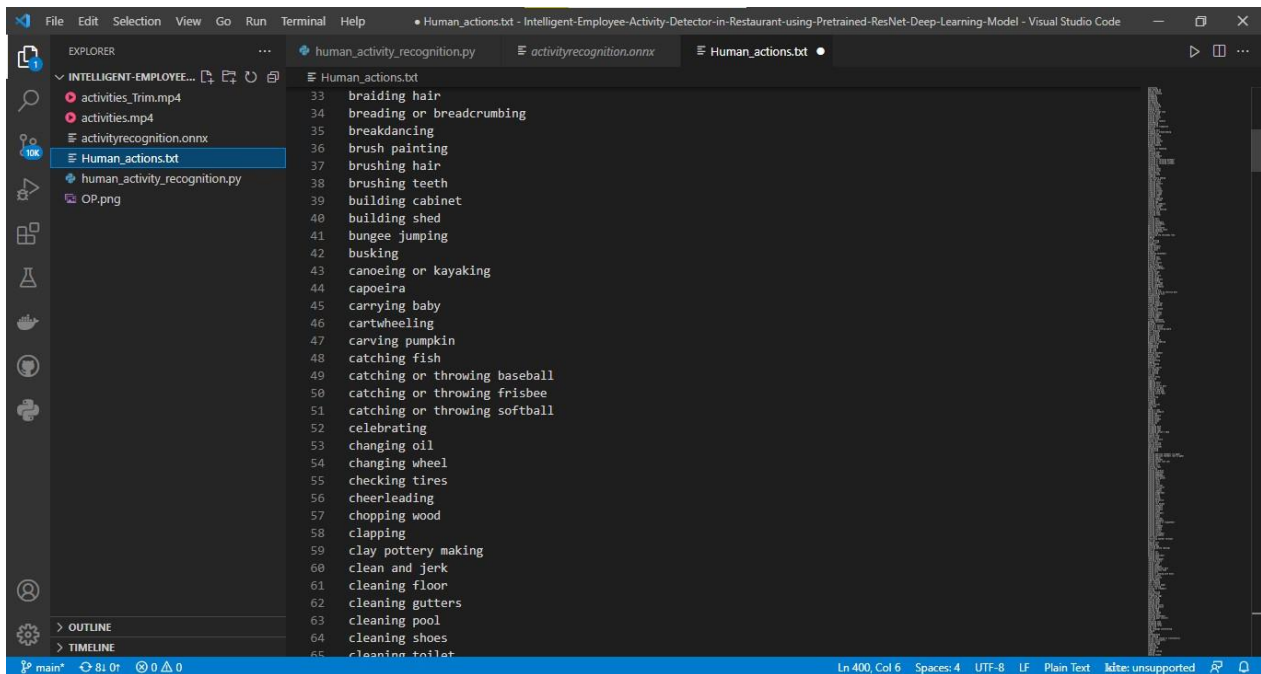


Figure 5: 32 ACTIVITIES (2 ND SET)

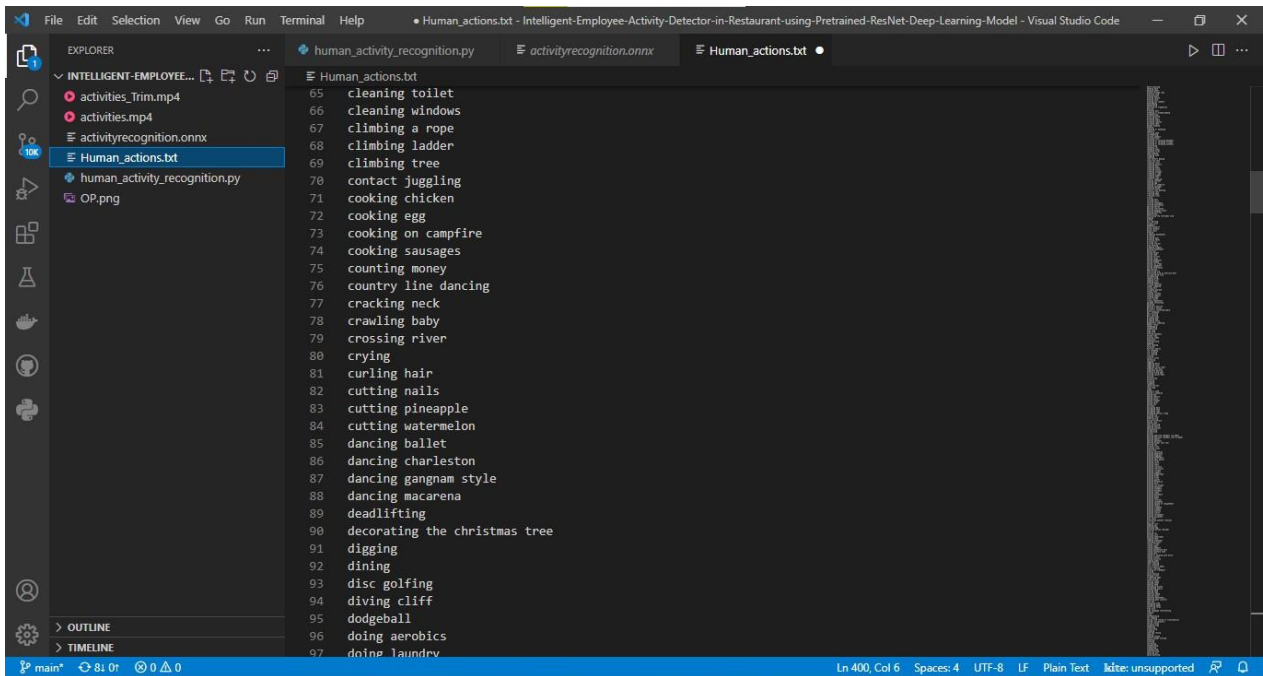


Figure 6: 32 ACTIVITIES (3 rd SET)

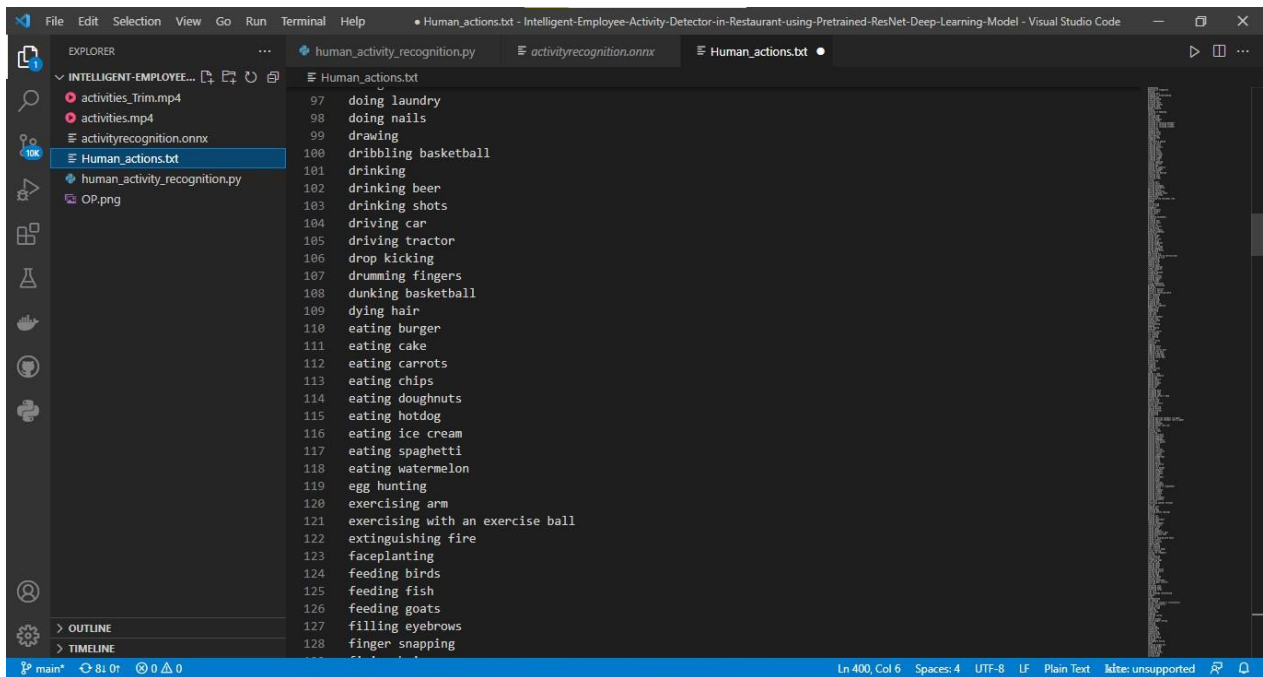
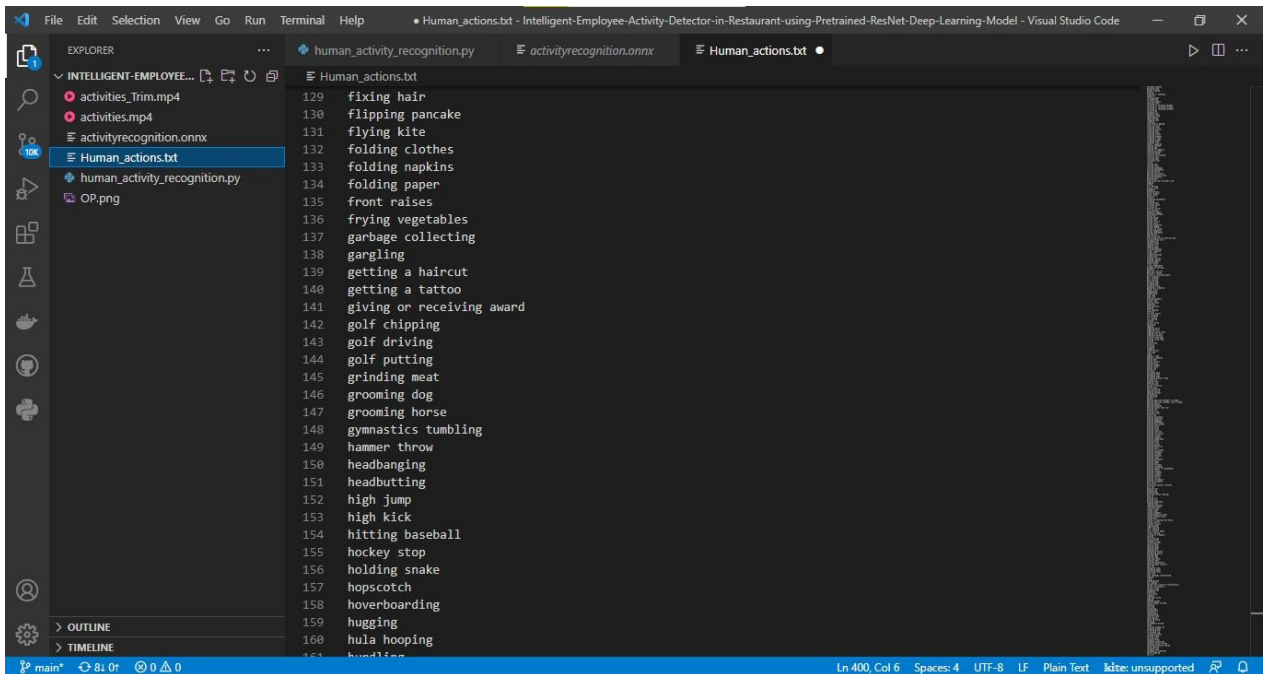
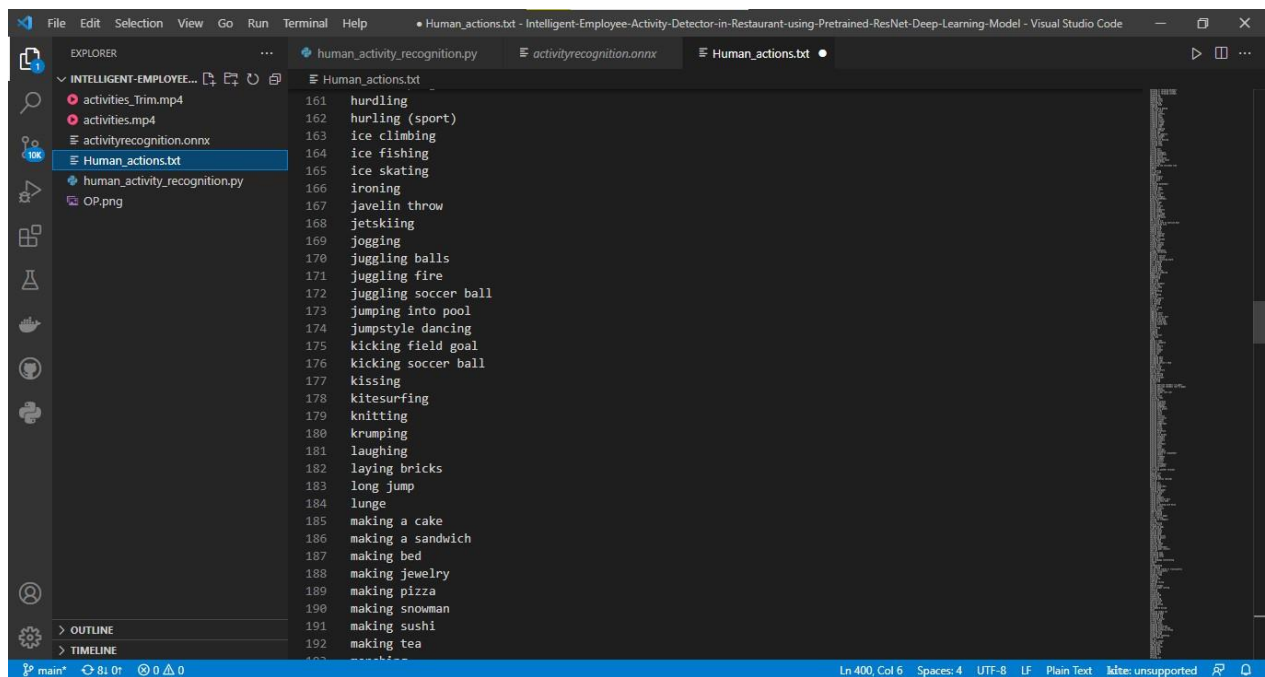


Figure 7: 32 ACTIVITIES (4 th SET)





**Figure 8: 32 ACTIVITIES (5 th SET)**



**Figure 9: 32 ACTIVITIES (6 th SET)**

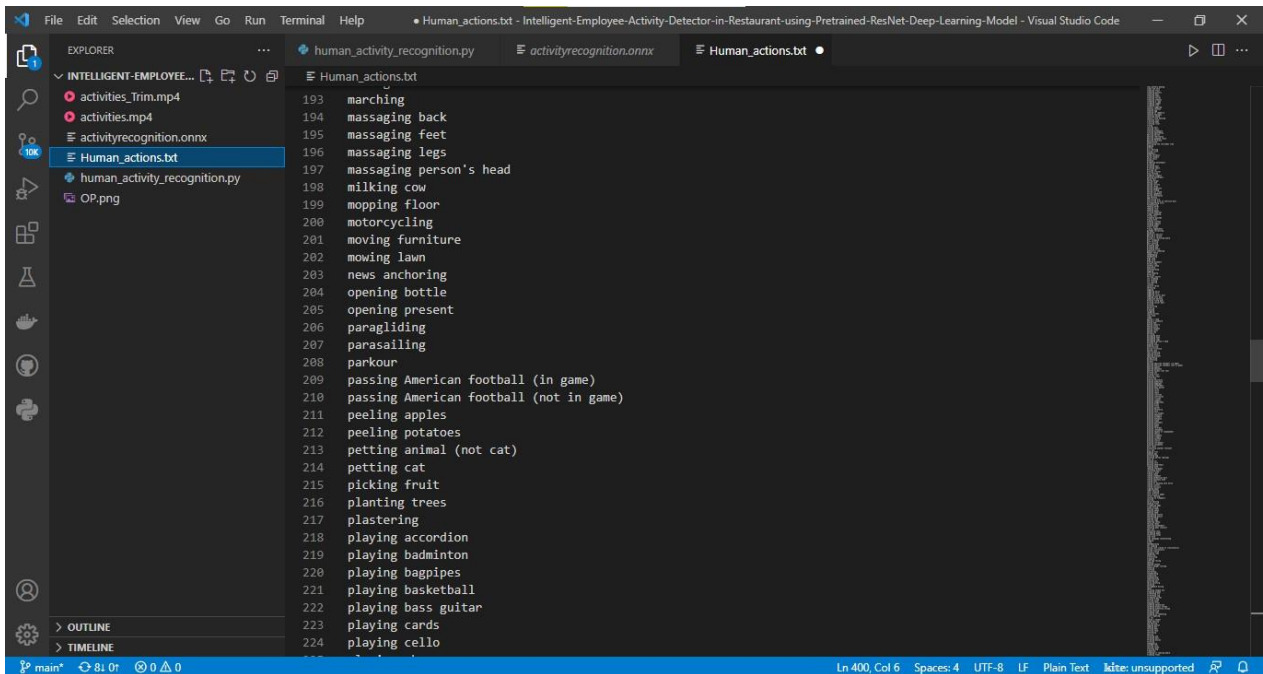


Figure 10: 32 ACTIVITIES (7 th SET)

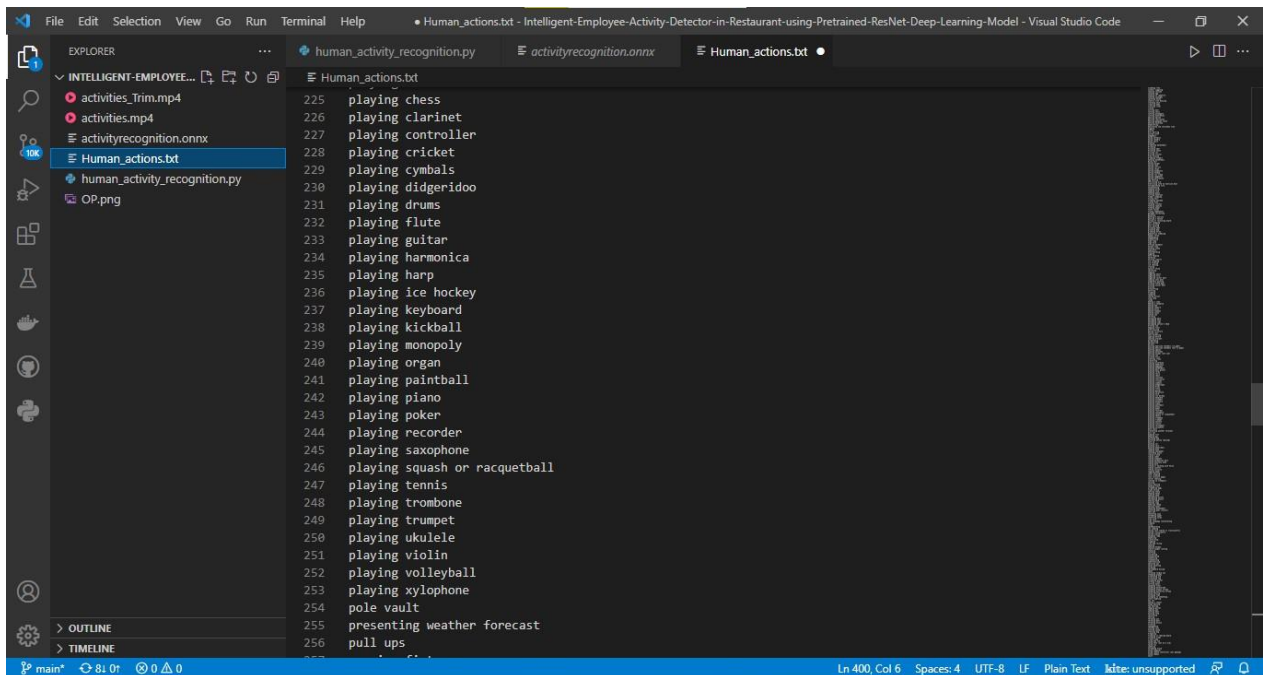
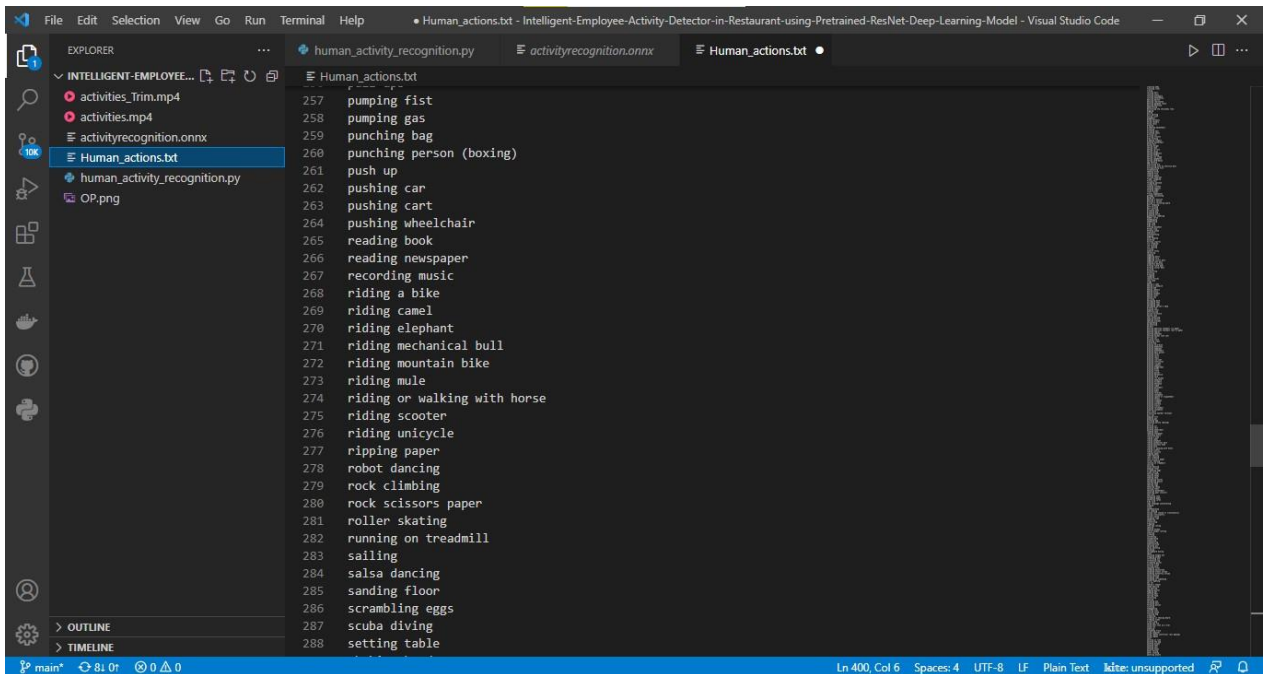
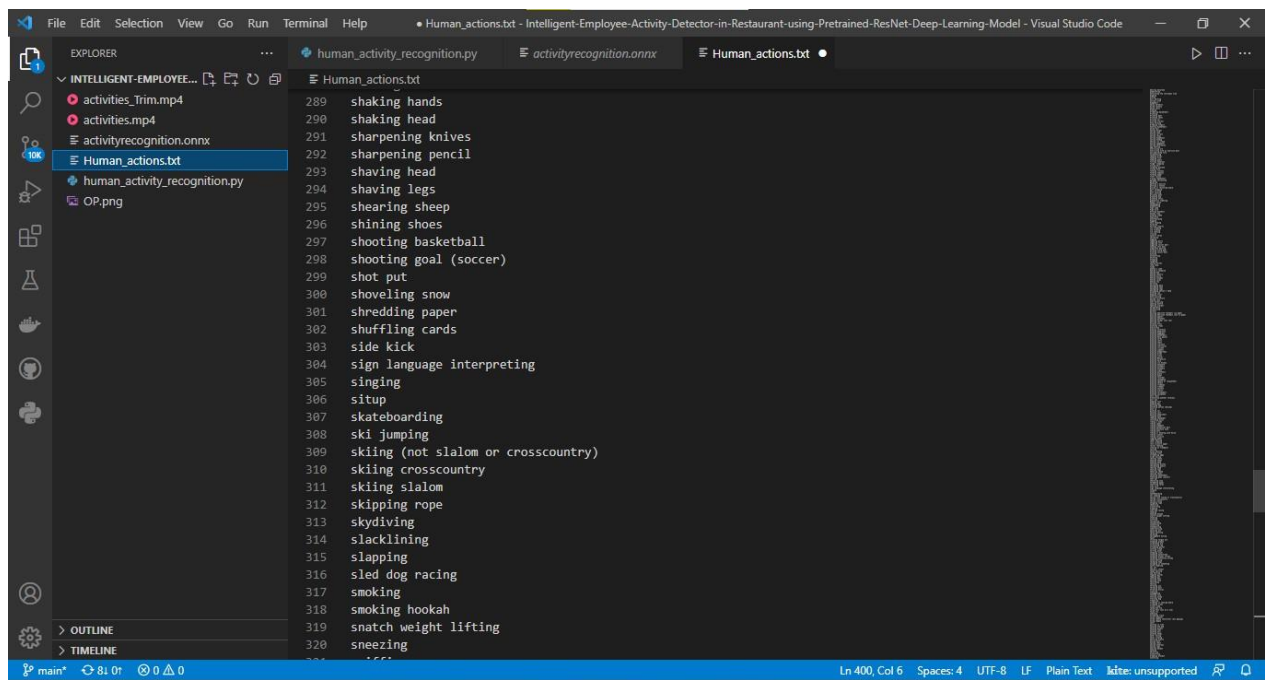


Figure 11: 32 ACTIVITIES (8 th SET)





**Figure 12: 32 ACTIVITIES (9 th SET)**



**Figure 13: 32 ACTIVITIES (10 th SET)**

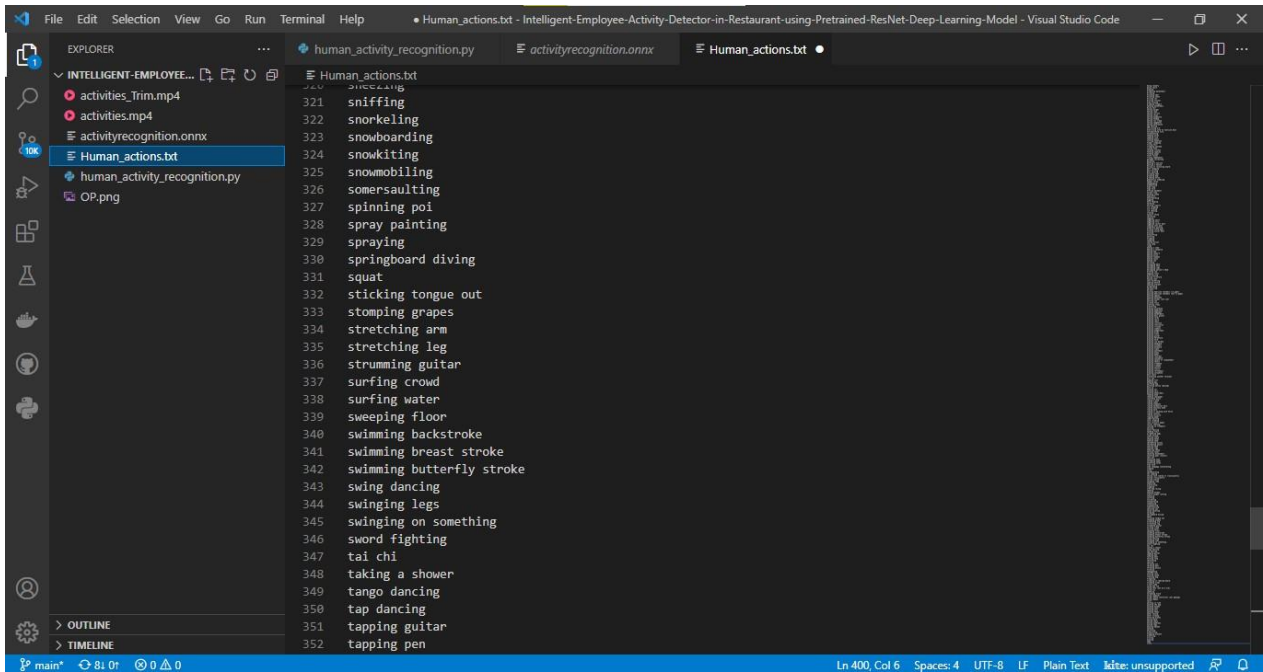


Figure 14: 32 ACTIVITIES (11 th SET)

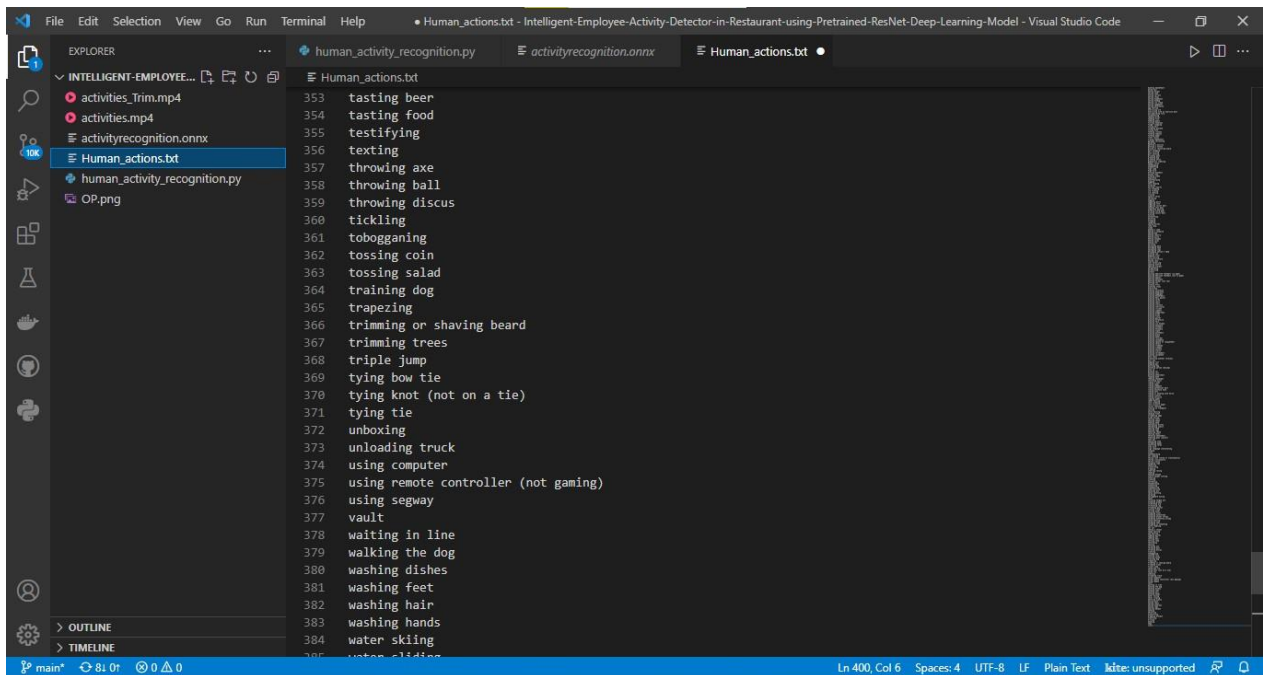
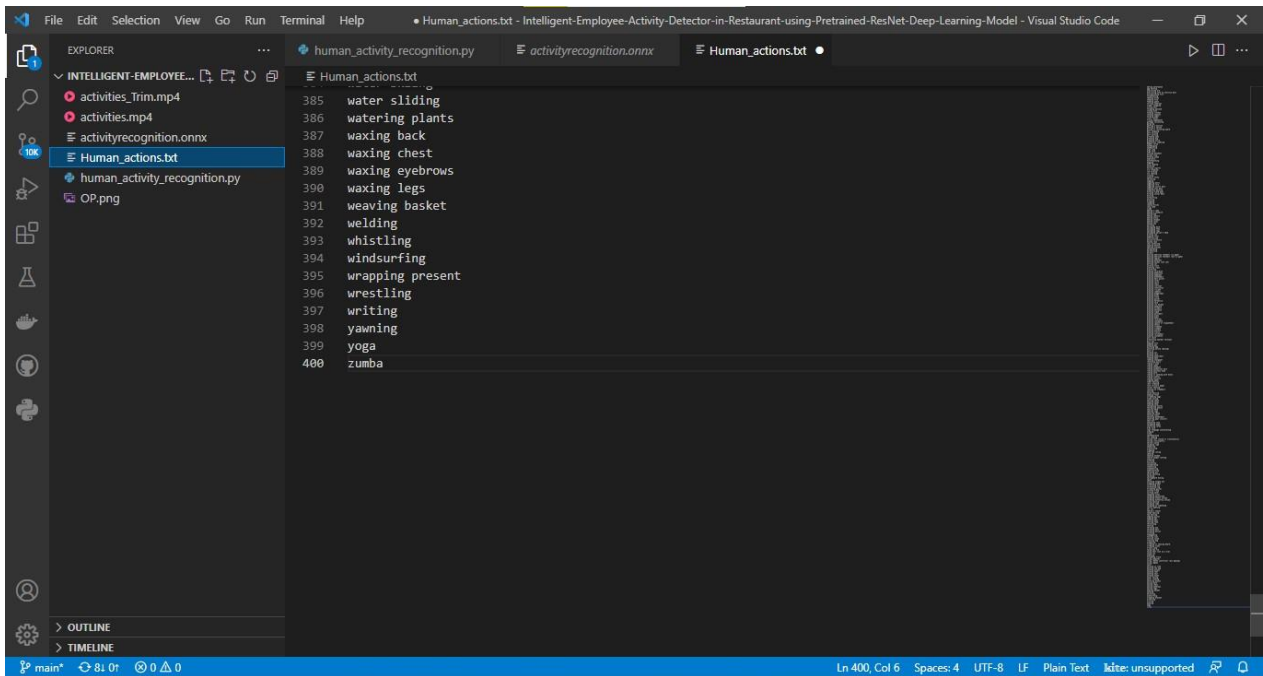


Figure 15: 32 ACTIVITIES (12 th SET)



**Figure 16: 16 ACTIVITIES (13 th SET)**

## 7. CONCLUSION

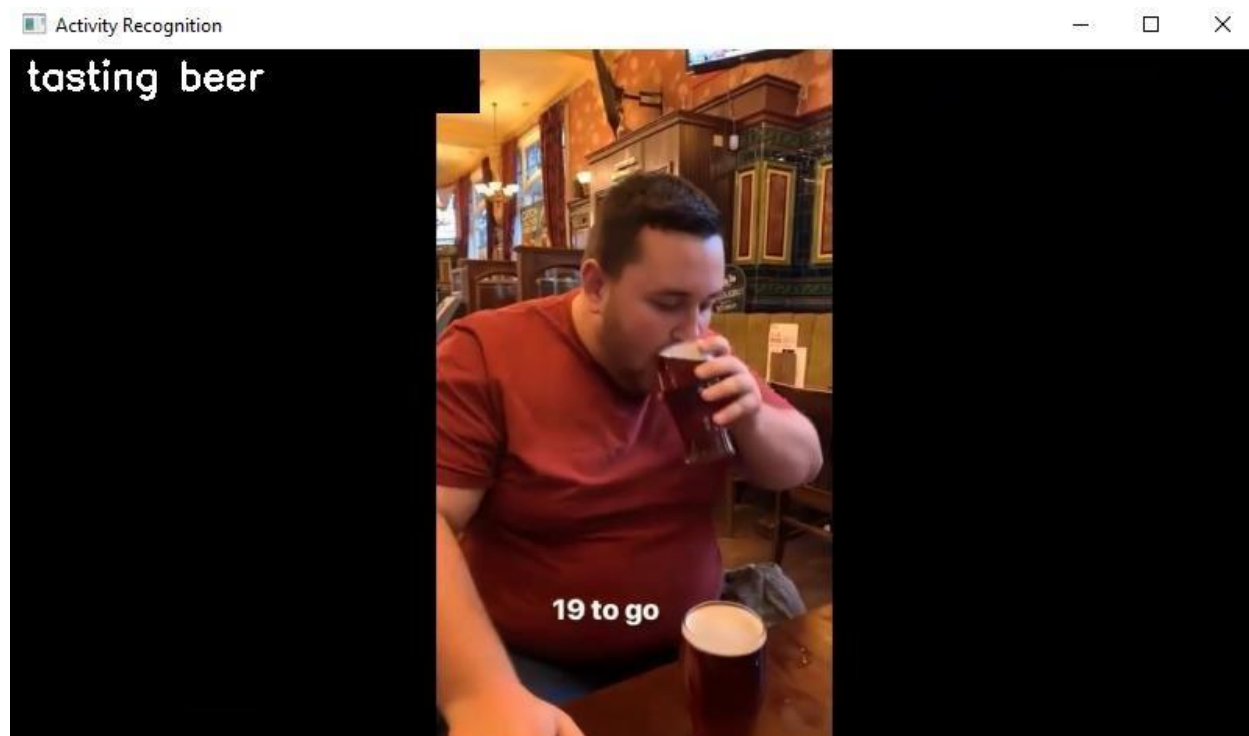
The following steps listed above are performed by our team, and herewith we attach snaps of our web page we achieved.



**Figure 17: Pizza making**



**Figure 18: Washing hands**



**Figure 19: Tasting Beer**

## **8. APPLICATION**

The following application could be used in a better understanding with

- Automatically classifying/categorizing a dataset of videos on disk.
- Monitoring a new employee to correctly perform a task.
- Proper steps, and procedures when making a pizza, including rolling out the dough, heating the oven, putting on the sauce, cheese, toppings, etc.
- Verifying that a food service worker has washed their hands after visiting the restroom or handling food that could cause cross-contamination (i.e. chicken and salmonella).
- Monitoring restaurant patrons and ensuring they are not over-served.

## **9. ADVANTAGES**

In this work the process of activity recognition is discussed and different methods of activity recognition are compared. Image recognition has become an important area of research for the advancement of computer vision. Actions could be anything like playing football, eating, dancing, etc. There has been a lot of progress in recognition of activities.

## **10. DIS-ADVANTAGES**

There are a lot of challenges like recognition of complex as well as simultaneous activities. Activities like walking while listening to music, singing while dancing are known as simultaneous activities. These activities become confusing and difficult to recognize.

## 11. FUTURE SCOPE

Many studies are still being done in order to fully overcome such problems. Sensor based technologies also face some challenges like installation of devices on different parts of human bodies to directly measure activities. It becomes a burden for users to wear sensors installed in their watches, clothes, bracelets, etc. External sensors are installed in the environment at different locations. GPS receivers are limited to the outdoor environment which makes the usage of sensors limited to particular regions. In a smart home, sensors need to be installed in every door and equipment of use. Installation and maintenance of such a huge network is quite cumbersome. These sensors can be replaced with the help of cameras.

There could be more than one action in a particular clip. If simultaneous activities are taking place like “texting” while “walking” or “eating” while “chatting”, then it will be labelled only under one of the classes and not both. Some activities require more emphasis on the object in order to differentiate, like playing different kinds of musical instruments. The proposed system could be used to monitor new staff to ensure they are working properly, keep a check in restaurants if the customers are served properly and automatically categorize a dataset of video on disk. Therefore, activity recognition systems are becoming a basic tool in many aspects of life. For further work, usage of a dataset with more than 400 activities could be made in order to increase the level of accuracy and make the system more flexible. It is observed that if there is a deep hierarchy of activities like yoga which has different positions, dance which has different forms, cooking in which there are different kinds of food and many other such activities, then it could significantly help to achieve better performance.

## 12. REFERENCES

- <https://pvimagesearch.com/2019/11/25/human-activity-recognition-with-opencv-and-deep-learning/>
- [https://e-archivo.uc3m.es/bitstream/handle/10016/26542/videobased\\_2015.pdf](https://e-archivo.uc3m.es/bitstream/handle/10016/26542/videobased_2015.pdf)
- <https://drive.google.com/drive/folders/1Dfl0PTIS9NRP-XhS-5djLnbSeSCf9saY>
- <https://www.frontiersin.org/articles/10.3389/frobt.2015.00028/full>