INTERNSHIP REPORT

A Report Submitted to

Jawaharlal Nehru Technological University Kakinada, Kakinada

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

SAVARAM VENKATESWARA RAO(20KN1A05F1)

SHAIK JAANI(20KN1A05F6)

MANDALA USHA KIRANMAI (20KN1A05H6)

MENTOR:NAFIYA SIDDIQI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NRI INSTITUTE OF TECHNOLOGY

Autonomous

(Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada)

Accredited by NBA (CSE, ECE & EEE), Accredited by NAAC with 'A' Grade

ISO 9001: 2015 CertifiedInstitution

Pothavarappadu (V), (Via) Nunna, Agiripalli (M), Krishna Dist., PIN: 521212, A.P, India.

2022-2023

NRI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution, Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada)

Accredited by NBA (CSE, ECE & EEE), Accredited by NAAC with 'A' Grade
ISO 9001: 2015 Certified Institution

Pothavarappadu (V), (Via) Nunna, Agiripalli (M), Krishna Dist., PIN: 521212, A.P, India.

CERTIFICATE

This is to certify that the "**Internship report**" submitted by **USHA KIRANMAI MANDALA ,SHAIK JAANI , SAVARAM VENKATESWARA RAO (Regd. No.: 20KN1A05H6, 20KN1A05F6, 20KN1A05F1)** is work done by her and submitted during YEARS academic year, in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** at **BLACKBUCK ENGINEERS PVT LTD,Road No:36, Jubilee Hills, Hyderabad, Telangana.**

Kattyayani

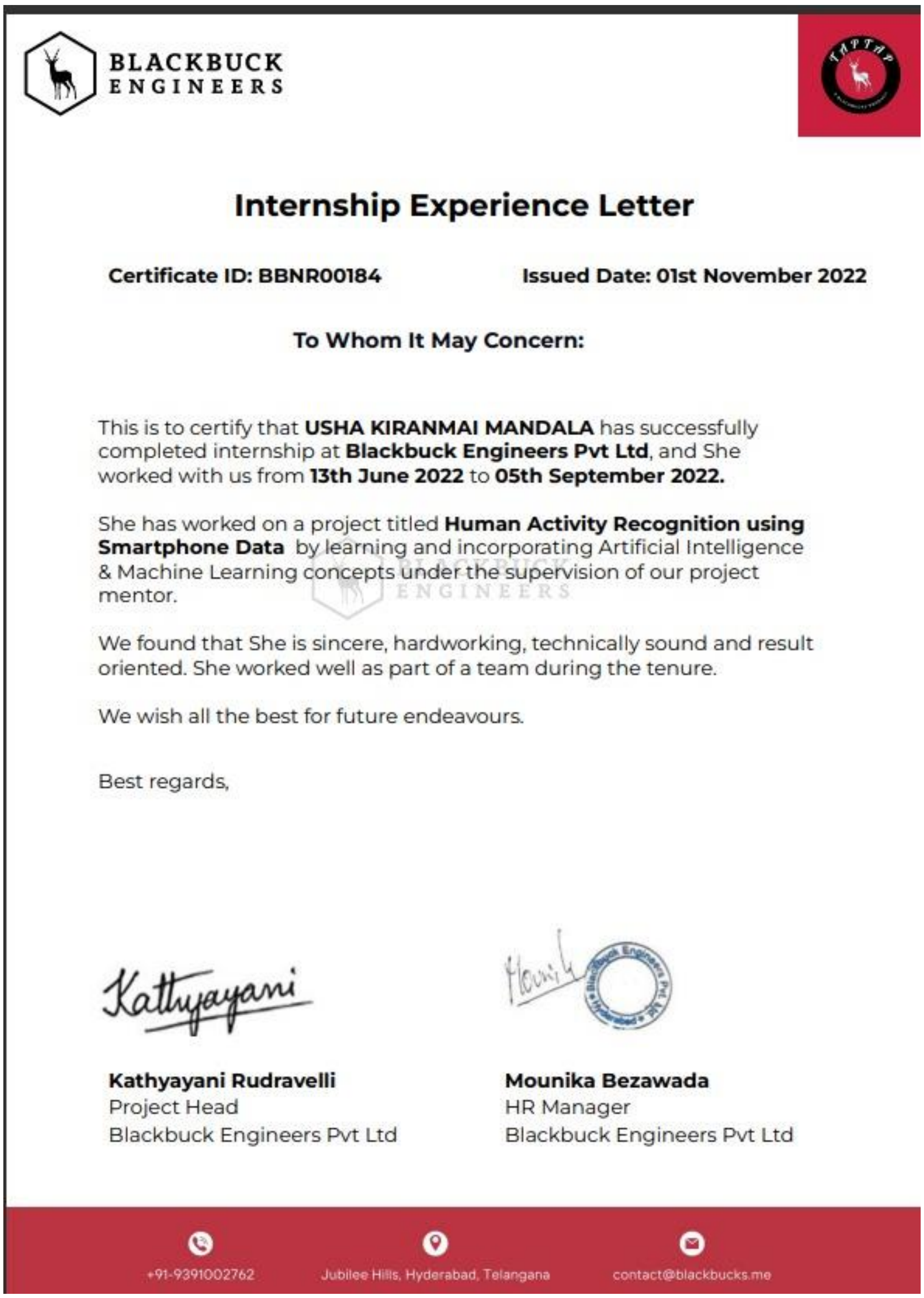
INTERNSHIP COORDIANTOR

Head of the Department

(Dr. D. SUNEETHA)

EXTERNAL EXAMINER

CERTIFICATE OF INTERNSHIP



BLACKBUCK ENGINEERS

Internship Experience Letter

Certificate ID: BBNR00184 **Issued Date: 01st November 2022**

To Whom It May Concern:

This is to certify that **USHA KIRANMAI MANDALA** has successfully completed internship at **Blackbuck Engineers Pvt Ltd**, and She worked with us from **13th June 2022** to **05th September 2022.**

She has worked on a project titled **Human Activity Recognition using Smartphone Data** by learning and incorporating Artificial Intelligence & Machine Learning concepts under the supervision of our project mentor.

We found that She is sincere, hardworking, technically sound and result oriented. She worked well as part of a team during the tenure.

We wish all the best for future endeavours.

Best regards,

Kathyayani

[illegible]

Blackbuck Engineers Pvt Ltd Hyderabad

Kathyayani Rudravelli
Project Head
Blackbuck Engineers Pvt Ltd

Mounika Bezawada
HR Manager
Blackbuck Engineers Pvt Ltd

+91-9391002762 Jubilee Hills, Hyderabad, Telangana contact@blackbucks.me

BLACKBUCK
ENGINEERS

Internship Experience Letter

Certificate ID: BBNR0160 **Issued Date: 01st November 2022**

To Whom It May Concern:

This is to certify that **SAVARAM VENKATESWARA RAO** has successfully completed internship at **Blackbuck Engineers Pvt Ltd**, and He worked with us from **13th June 2022** to **05th September 2022.**

He has worked on a project titled **Human Activity Recognition using Smartphone Data** by learning and incorporating Artificial Intelligence & Machine Learning concepts under the supervision of our project mentor.

We found that He is sincere, hardworking, technically sound and result oriented. He worked well as part of a team during the tenure.

We wish all the best for future endeavours.

Best regards,

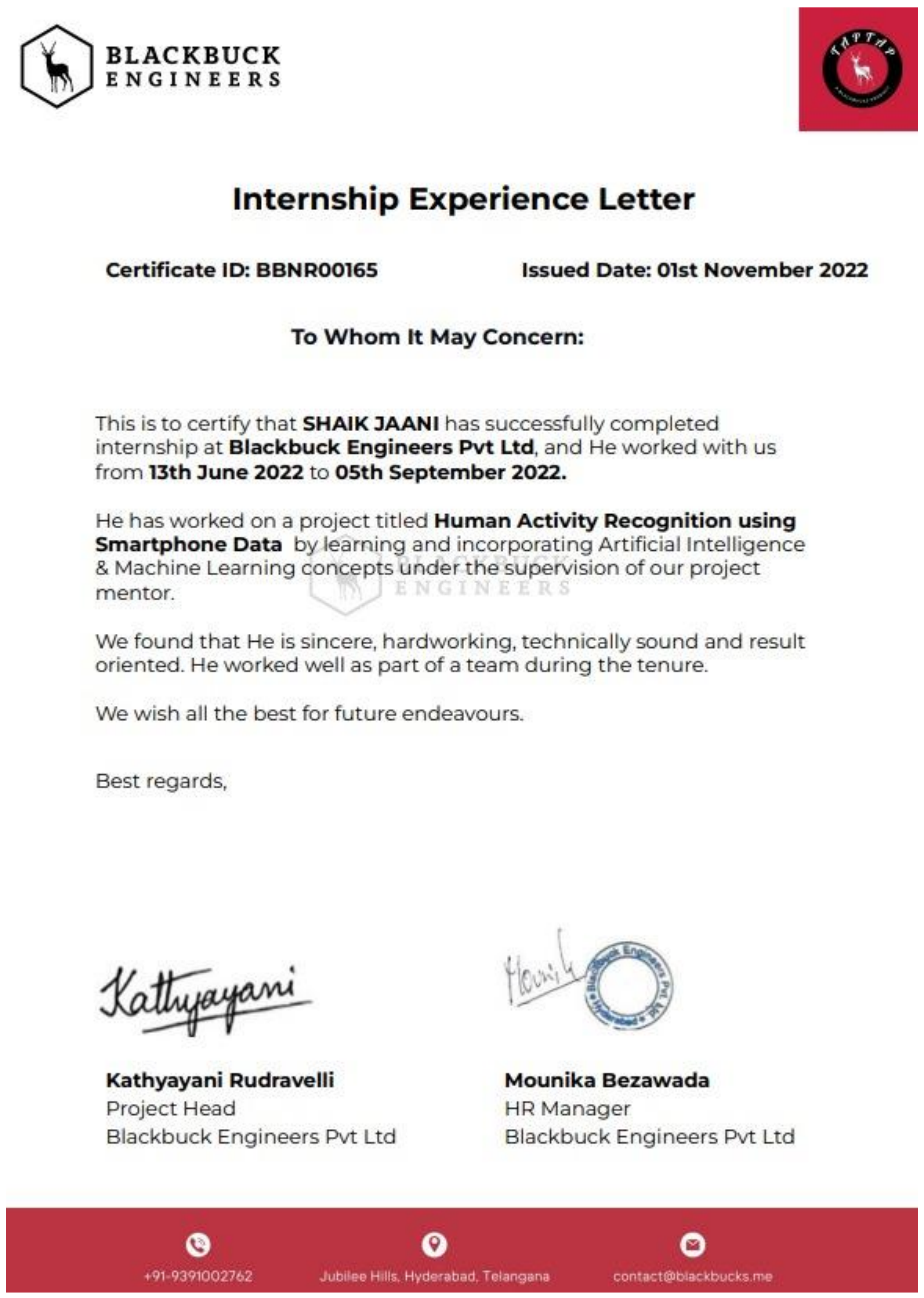
Kathyayani

[illegible]

Blackbuck Engineers Pvt Ltd Hyderabad

Kathyayani Rudravelli
Project Head
Blackbuck Engineers Pvt Ltd

Mounika Bezawada
HR Manager
Blackbuck Engineers Pvt Ltd



BLACKBUCK
ENGINEERS

TAPTAP

Internship Experience Letter

Certificate ID: BBNR00165 **Issued Date: 01st November 2022**

To Whom It May Concern:

This is to certify that **SHAIK JAANI** has successfully completed internship at **Blackbuck Engineers Pvt Ltd**, and He worked with us from **13th June 2022** to **05th September 2022.**

He has worked on a project titled **Human Activity Recognition using Smartphone Data** by learning and incorporating Artificial Intelligence & Machine Learning concepts under the supervision of our project mentor.

We found that He is sincere, hardworking, technically sound and result oriented. He worked well as part of a team during the tenure.

We wish all the best for future endeavours.

Best regards,

Kathyayani

Kathyayani Rudravelli
Project Head
Blackbuck Engineers Pvt Ltd

[illegible]

Blackbuck Engineers Pvt Ltd • Hyderabad •

Mounika Bezawada
HR Manager
Blackbuck Engineers Pvt Ltd

+91-9391002762 Jubilee Hills, Hyderabad, Telangana contact@blackbucks.me

ACKNOWLEDGEMENT

We take this opportunity to thank all who have rendered their full support to my work. The pleasure, the achievement, the glory, the satisfaction, the reward, the appreciation and the construction of my project cannot be expressed with a few words for their valuable suggestions.

We are expressing my heartfelt thanks to **Head of the Department, Dr. D. SUNEETHA** garu for her continuous guidance for completion of my Project work.

We are extending our sincere thanks to **Dean of the Department, Dr. K. V. SAMBASIVA RAO** for his continuous guidance and support to complete my project successfully.

We are thankful to the **Principal, Dr. C. NAGA BHASKAR** garu for his encouragement tocomplete the Project work.

We are extending my sincere and honest thanks to the **Chairman, Dr. R. VENKATA RAO garu & Secretary, Sri K. Sridhar** garu for their continuous support in completing the Project work.

Finally, I thank the Administrative Officer, Staff Members, Faculty of Department of CSE, NRI Institute of Technology and my friends, directly or indirectly helped us in the completion of this project.

NAME:USHAKIRANMAI MANDALA
SHAIK JAANI
SAVARAM VENKATESWARA RAO

REGDNO: 20KN1A05H6
20KN1A05F6
20KN1A05F1

ABSTRACT

Human activity recognition has wide applications in medical research and human survey system. In this project, we design a we identify human actions based on smart phone. The system uses a 3-dimentional smartphone accelerometer as the only sensor to collect time series signals, from which 31 features are generated in both time and frequency domain. Activity recognition aims to recognize the actions and goals of one or more agents from a series of observations on the agents' actions and the environmental conditions. Since the 1980s, this research field has captured the attention of several computer science communities due to its strength in providing personalized support for many different applications and its connection to many different fields of study such as medicine, human-computer interaction, or sociology.

Organization Information:

Blackbuck Engineers is started in 2013 with the aim of creating a great ecosystem of academia, research, industry, and individuals. Blackbucks is a premier partner to Govts International Institute of Digital Technologies, and IITs. Blackbuck delivers the TAPTAPAI Driven employability platform to transform the journey of students towards their dream goals while helping **HRs hire right students**.

Blackbuck has the largest chain of excellence in emerging tech across India.

Blackbucks runs post graduation programs in AI,ML and Data Science

www.theblackbucks.com

Programs and opportunities:

This ground-up approach helps us deliver not only the solution to our clients but also add value to at the core Blackbuck Engineers which operates in Five specific domains namely TapTap - AI Driven, Post Graduation Programs, Center of Excellence, Virtual Programming Labs and Happie Days - A social Networking site for the students. TapTap offer services in Campus Recruitment drives for the Employers as well as College authorities. Recruiters can Conduct Customized Online Assessments secured with Best-in-class Proctoring and Schedule the end-to-end hiring process. Under each division we further provide specific industry solutions on focused domains with cutting edge technologies. Blackbuck Engineers emphasize on building relationships with our clients by delivering projects on time and within budget.

INDEX

LearningObjectives/InternshipObjectives

- Internships are generally thought of to be reserved for college students looking to gainexperiencein aparticularfield.However,a widearray of peoplecan benefitfromTrainingInternshipsinordertoreceiverealworldexperienceanddeveloptheirskills.

- An objective for this position should emphasize the skills you already possess in the areaandyourinterestinlearningmore

- Internships are utilized in a number of different career fields, including architecture,engineering,healthcare,economics,advertisingandmanymore.

- Some internship is used to allow individuals to perform scientific research while othersarespecificallydesigned toallowpeople togainfirst-hand experienceworking.

- Utilizing internships is a great way to build your resume and develop skills that can beemphasizedinyourresumeforfuturejobs.WhenyouareapplyingforaTrainingInternship, make sure to highlight any special skills or talents that can make you standapart from the rest of the applicants so that you have an improved chance of landing theposition.

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

	DATE	DAY	NAMEOFTHETOPIC/MODULECOMPLETED
1stWEEK	13.06.2022	Monday	Introduce the Topic & the Problem Statement
	14.06.2022	Tuesday	Introduce the Topic & the Problem Statement
	15.06.2022	Wednesday	Introduce the Topic & the Problem Statement
	16.06.2022	Thursday	Introduce the Topic & the Problem Statement
	17.06.2022	Friday	Introduce the Topic & the Problem Statement

	DATE	DAY	NAMEOFTHETOPIC/MODULECOMPLETED
2ndWEEK	20.06.2022	Monday	Abstract Building
	21.06.2022	Tuesday	Abstract Building
	22.06.2022	Wednesday	Abstract Building
	23.06.2022	Thursday	Abstract Building
	24.06.2022	Friday	Abstract Submission

	DATE	DAY	NAMEOFTHETOPIC/MODULECOMPLETED
3rdWEEK	27.06.2022	Monday	Abstract Submission
	28.06.2022	Tuesday	Abstract Submission
	29.06.2022	Wednesday	Explain your Approach to Solving Problem
	30.06.2022	Thursday	Explain your Approach to Solving Problem
	01.07.2022	Friday	Explain your Approach to Solving Problem

4thWEEK	DATE	DAY	NAMEOFTHETOPIC/MODULECOMPLETED
	04.07.2022	Monday	Explain your Approach to Solving Problem
	05.07.2022	Tuesday	Explain Structure of Project
	06.07.2022	Wednesday	Explain Structure of Project
	07.07.2022	Thursday	Explain Structure of Project
	08.07.2022	Friday	Explain Structure of Project

5thWEEK	DATE	DAY	NAMEOFTHETOPIC/MODULECOMPLETED
	11.07.2022	Monday	Data Preprocessing
	12.07.2022	Tuesday	Data Preprocessing
	13.07.2022	Wednesday	Data Preprocessing
	14.07.2022	Thursday	Data Preprocessing
	15.07.2022	Friday	Data Preprocessing

6thWEEK	DATE	DAY	NAMEOFTHETOPIC/MODULECOMPLETED
	18.07.2022	Monday	Perform Analysis
	19.07.2022	Tuesday	Perform Analysis
	20.07.2022	Wednesday	Perform Analysis
	21.07.2022	Thursday	Perform Analysis
	22.07.2022	Friday	Perform Analysis

7thWEEK	DATE	DAY	NAMEOFTHETOPIC/MODULECOMPLETED
	25.07.2022	Monday	PPT Preparation
	26.07.2022	Tuesday	PPT Preparation
	27.07.2022	Wednesday	PPT Preparation
	28.07.2022	Thursday	PPT Preparation
	29.07.2022	Friday	PPT Preparation

8th WEEK	DATE	DAY	NAMEOFTHETOPIC/MODULECOMPLETED
	01.08.2022	Monday	PPT Submission
	02.08.2022	Tuesday	PPT Submission
	03.08.2022	Wednesday	Mid Review
	04.08.2022	Thursday	Mid Review
	05.08.2022	Friday	Mid Review

9th WEEK	DATE	DAY	NAMEOFTHETOPIC/MODULECOMPLETED
	08.08.2022	Monday	Mid Review
	10.08.2022	Tuesday	Mid Review
	11.08.2022	Wednesday	Building & Applying Algorithm
	12.08.2022	Thursday	Building & Applying Algorithm

10th WEEK	DATE	DAY	NAMEOFTHETOPIC/MODULECOMPLETED
	16.08.2022	Tuesday	Building & Applying Algorithm
	17.08.2022	Wednesday	Building & Applying Algorithm
	19.08.2022	Friday	Building & Applying Algorithm

11th WEEK	DATE	DAY	NAMEOFTHETOPIC/MODULECOMPLETED
	22.08.2022	Monday	Concluding Project
	23.08.2022	Tuesday	Concluding Project
	24.08.2022	Wednesday	Concluding Project
	25.08.2022	Thursday	Concluding Project
	26.08.2022	Friday	Concluding Project

12th WEEK	DATE	DAY	NAMEOFTHETOPIC/MODULECOMPLETED
	29.08.2022	Monday	Final Review
	30.08.2022	Tuesday	Final Review
	01.09.2022	Wednesday	Final Review
	02.09.2022	Thursday	Final Review
	05.09.2022	Friday	Final Review

CHAPTER 1

INTRODUCTION

1.INTRODUCTION

Human Activity Recognition (HAR) aims to identify the actions carried out by a person given a set of observations of him/herself and the surrounding environment. Recognition can be accomplished by exploiting the information retrieved from various sources such as environmental or body-worn sensors.

Human activity recognition plays a significant role in human-to-human interaction and interpersonal relations. Because it provides information about the identity of a person, their personality, and psychological state, it is difficult to extract. The human ability to recognize another person's activities is one of the main subjects of study of the scientific areas of computer vision and machine learning. As a result of this research, many applications, including video surveillance systems, human-computer interaction, and robotics for human behavior characterization, require a multiple activity recognition system.

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. Human activities, such as "walking" and "running," arise very naturally in daily life and are relatively easy to recognize. On the other hand, more complex activities, such as "peeling an apple," are more difficult to identify. Complex activities may be decomposed into other simpler activities, which are generally easier to recognize. Usually, the detection of objects in a scene may help to better understand human activities as it may provide useful information about the ongoing event

The goal of human activity recognition is to examine activities from video sequences or still images. Motivated by this fact, human activity recognition systems aim to correctly classify input data into its underlying activity category. Depending on their complexity, human activities are categorized into: (i) gestures; (ii) atomic actions; (iii) human-to-object or human-to-human interactions; (iv) group actions; (v) behaviors; and (vi) events. Figure 1 visualizes the decomposition of human activities according to their complexity.

Figure 1

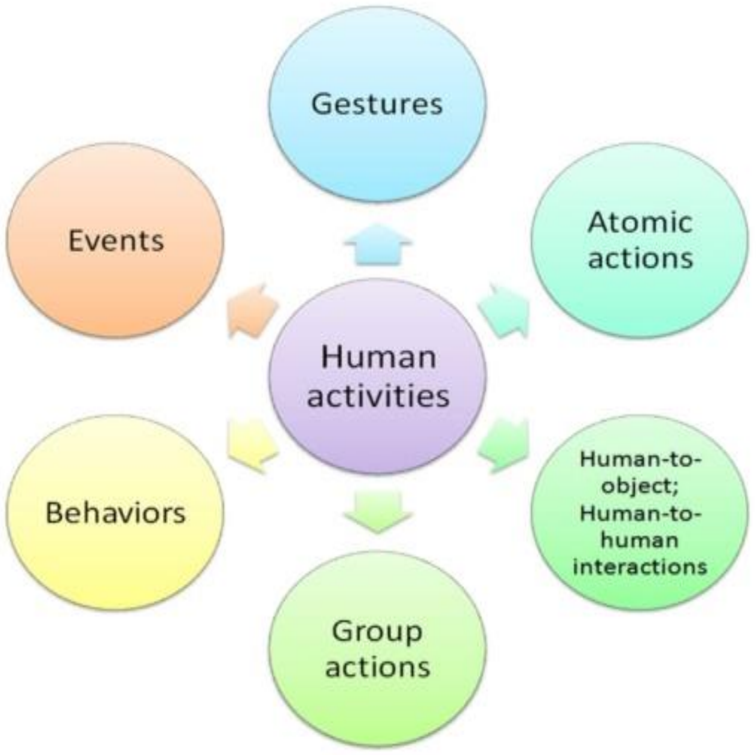


FIGURE 1. DECOMPOSITION OF HUMAN ACTIVITIES.

1.1ModuleDescription:

OS:

The python OS module provides functions used for interacting with the operating system and also get related information about it. The OS comes under Python's standard utility modules. This module offers a portable way of using operating system dependent functionality.

NumPy:

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

Dataset:

A Dataset is a set or collection of data. This set is normally presented in a tabular pattern. Every column describes a particular variable. And each row corresponds to a given member of the data set, as per the given question. This is a part of data management.

Validation:

A process associated with the collection and production of intelligence that confirms that an intelligence collection or production requirement is sufficiently important to justify the dedication of intelligence resources, does not duplicate an existing requirement, and has not been previously satisfied.

Sequential model:

A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor.

Tensor flow:

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

Matplotlib:

Matplotlib is **a comprehensive library for creating static, animated, and interactive visualizations in Python**. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update.

Math:

For straightforward mathematical calculations in Python, you can use the built-in mathematical operators, such as **addition (+), subtraction (-), division (/), and multiplication (*)**. But more advanced operations, such as exponential, logarithmic, trigonometric, or power functions, are not built in.

Capturing video using Open CV:

OpenCV has a function to read video, which is cv2. VideoCapture().
For ex;We can access our webcam using pass 0 in the function parameter. If you want to capture CCTV footage then we can pass RTSP url in the function parameter, which is really useful for video analysis.

MoviePy:

It is **a Python module for video editing**, which can be used for basic operations (like cuts, concatenations, title insertions), video compositing (a.k.a. non-linear editing), video processing, or to create advanced effects. It can read and write the most common video formats, including GIF.

CHAPTER 2
SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

Requirement Analysis

Existing System:

The existing system was manual where a person had to sit in front of a monitor to monitor and guide human activities, it was hectic, time consuming and costlier system and was prone to human errors and negligence. Further some systems started using sensor data to recognize human activities but they were needed to be worn by the user which limited the scope of activity recognition in open environment in general.

Proposed System:

Unlike the existing system, the proposed system takes input in the form of video and image to recognize the activity being performed in it. It is much faster and a cost effective solution. It uses deep learning to recognize the activities. This system can be used, incorporated or expanded further to cover a wide range of applications. Hence, it acts as a base system for various applications and tasks. Moreover, it can reduce the need of additional staff for entering data. Thereby, reducing the cost of the companies considerably.

CHAPTER-3
SOFTWARE REQUIREMENTS SPECIFICATIONS

3. SOFTWARE REQUIREMENTS SPECIFICATIONS

System configurations

The software requirement specification can produce at the culmination of the analysis task.The function and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, are presentation of system behavior, and indication of performance and design constrain , appropriate validatecriteria , and other information pertinent to requirements.

SoftwareRequirements:

• Operating system	:Windows7 Ultimate.
• CodingLanguage	:Python,CNN
• Front-End	:VisualStudio2012Professional.
• Data Base	:SQLServer2008.

Sensor-based activity recognition seeks the profound high-level knowledge about human activities from multitudes of low-level sensor readings. Conventional pattern recognition approaches have made tremendous progress in the past years. However, those methods often heavily rely on heuristic hand-crafted feature extraction, which could hinder their generalization performance. Recently, the recent advancement of deep learning makes it possible to perform automatic high-level feature extraction thus achieves promising performance in many areas

HardwareRequirement:

• System	:PentiumIV 2.4GHz.
• HardDisk	: 1TB.
• Ram	:4GB.

Here we use hard ware such as mobiles , CCTVs , computers , few chips etc
The most widely used hardware devises are watches and mobiles.

CHAPTER 4
TECHNOLOGY

4.TECHNOLOGY

Machine learning (ML):

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

PYTHON:

Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Using Neural networks:

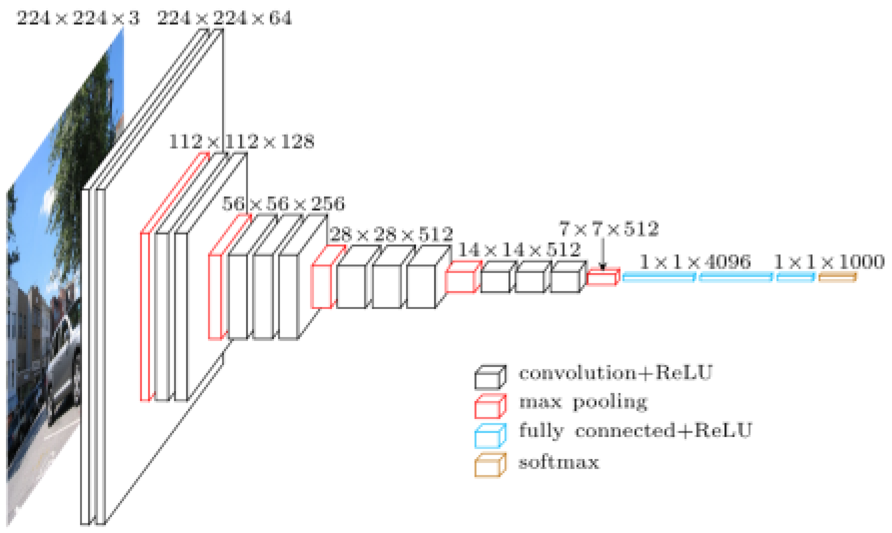
Tensor flow provides tf. keras a high – level API to build and train models in Tensor flow. Import the necessary packages Get the data from an external source or access the dataset. keras provide a wide range of datasets. Make sure that your image's data set correctly labelled. Divide dataset into Train images with labels, and test set also contain the images and labels. Do explore the data by looking at the shape, length of the data before training the model. So you can understand how no examples are there in the dataset. Preprocess the data, set the image size to 0 to 255 pixels according to the dataset. Display the 20 to 25 images with their label. using neural networks is the best way to classify images. identifying patterns and extracting features on images are what deep learning models can do, and they do it very well.

CONVOLUTION NEURAL NETWORK:

Image classification is not a hard topic anymore. Tensor flow has all the inbuilt Functionalities that of the complex mathematics for us. Without knowing the details of the neural network, we can use a neural network now. In today's project, I used a Convolutional Neural Network (CNN) which is an advanced version of the neural network. It condenses down a picture to some important features. CNN will figure out important portions of the images. A Convolution Neural Networks is a deep learning algorithm that takes images as input, learn objects/aspects from image and differentiate each image.

CHAPTER-5
Architecture

5. Architecture



CHAPTER-6
CODING

6.CODING

Make sure you have pafy, youtube-dl and moviepy packages installed

```
!pip install pafy youtube-dl moviepy
```

Import Required Libraries:

```
import os
import cv2
import math
import pafy
import random
import numpy as np
import datetime as dt
import tensorflow as tf
from moviepy.editor import *
from collections import deque
import matplotlib.pyplot as plt
%matplotlib inline

from sklearn.model_selection import train_test_split

from tensorflow.keras.layers import *
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.utils import plot_model
```

Set Numpy, Python and Tensorflow seeds to get consistent results

```
seed_constant = 23
np.random.seed(seed_constant)
random.seed(seed_constant)
tf.random.set_seed(seed_constant)
```

Download and Extract the Dataset

```
!wget -nc --no-check-certificate
https://www.crcv.ucf.edu/data/UCF50.rar
!unrar x UCF50.rar -inul –y
```

Visualize the Data with its Labels

```
# Create a Matplotlib figure
plt.figure(figsize = (30, 30))

# Get Names of all classes in UCF50
all_classes_names = os.listdir('UCF50')

# Generate a random sample of images each time the cell runs
random_range = random.sample(range(len(all_classes_names)), 20)

# Iterating through all the random samples
for counter, random_index in enumerate(random_range, 1):
```

```
# Getting Class Name using Random Index
selected_class_Name = all_classes_names[random_index]

# Getting a list of all the video files present in a Class Directory
video_files_names_list = os.listdir(f'UCF50/{selected_class_Name}')

# Randomly selecting a video file
selected_video_file_name = random.choice(video_files_names_list)

# Reading the Video File Using the Video Capture
video_reader = cv2.VideoCapture(f'UCF50/{selected_class_Name}/{selected_video_file_name}')

# Reading The First Frame of the Video File
_, bgr_frame = video_reader.read()

# Closing the VideoCapture object and releasing all resources.
video_reader.release()

# Converting the BGR Frame to RGB Frame
rgb_frame = cv2.cvtColor(bgr_frame, cv2.COLOR_BGR2RGB)

# Adding The Class Name Text on top of the Video Frame.
cv2.putText(rgb_frame, selected_class_Name, (10, 30), cv2.FONT_HERSHEY_SIMPLEX,1, (255, 0, 0), 2)

# Assigning the Frame to a specific position of a subplot
plt.subplot(5, 4, counter)
plt.imshow(rgb_frame)
plt.axis('off')
```

Read and Preprocess the Dataset

```
image_height, image_width = 64, 64
max_images_per_class = 8000

dataset_directory = "UCF50"
classes_list = ["WalkingWithDog", "TaiChi", "Swing",
"HorseRace"]

model_output_size = len(classes_list)
```

Extract, Resize and Normalize Frames

```
def frames_extraction(video_path):
    # Empty List declared to store video frames
    frames_list = []

    # Reading the Video File Using the VideoCapture
    video_reader = cv2.VideoCapture(video_path)

    # Iterating through Video Frames
    while True:

        # Reading a frame from the video file
        success, frame = video_reader.read()

        # If Video frame was not successfully read then break the loop
        if not success:
            break

        # Resize the Frame to fixed Dimensions
        resized_frame = cv2.resize(frame, (image_height, image_width))
    # Normalize the resized frame by dividing it with 255 so
that each pixel value then lies between 0 and 1
        normalized_frame = resized_frame / 255

        # Appending the normalized frame into the frames list
        frames_list.append(normalized_frame)

    # Closing the VideoCapture object and releasing all resources.
    video_reader.release()

    # returning the frames list
    return frames_list
```

Dataset Creation

```
def create_dataset():

    # Declaring Empty Lists to store the features and labels values.
    temp_features = []
    features = []
    labels = []

    # Iterating through all the classes mentioned in the classes list
    for class_index, class_name in enumerate(classes_list):
        print(f'Extracting Data of Class: {class_name}')

        # Getting the list of video files present in the specific class name directory
        files_list = os.listdir(os.path.join(dataset_directory, class_name))

        # Iterating through all the files present in the files list
        for file_name in files_list:

            # Construct the complete video path
            video_file_path = os.path.join(dataset_directory, class_name, file_name)

            # Calling the frame_extraction method for every video file path
            frames = frames_extraction(video_file_path)

            # Appending the frames to a temporary list.
            temp_features.extend(frames)

        # Adding randomly selected frames to the features list
        features.extend(random.sample(temp_features, max_images_per_class))

        # Adding Fixed number of labels to the labels list
        labels.extend([class_index] * max_images_per_class)

  # Emptying the temp_features list so it can be reused to store all frames of
the next class.
        temp_features.clear()

    # Converting the features and labels lists to numpy arrays
    features = np.asarray(features)
    labels = np.array(labels)

    return features, labels
```

Calling the create_dataset **method which returns features and labels**

```
features, labels =create_dataset()
```

Now we will convert class labels to one hot encoded vectors

```
# Using Keras's to_categorical method to convert labels into one-hot-encoded
vectors
one_hot_encoded_labels = to_categorical(labels)
```

Split the Data into Train and Test Sets

```
features_train, features_test, labels_train, labels_test =train_test_split(features,
one_hot_encoded_labels, test_size =0.2, shuffle =True, random_state =seed_constant)
```

Construct the Model

```
# Let's create a function that will construct our model
def create_model():

    # We will use a Sequential model for model construction
    model = Sequential()

    # Defining The Model Architecture
    model.add(Conv2D(filters = 64, kernel_size = (3, 3), activation = 'relu',
input_shape = (image_height, image_width, 3)))
    model.add(Conv2D(filters = 64, kernel_size = (3, 3), activation = 'relu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size = (2, 2)))
    model.add(GlobalAveragePooling2D())
    model.add(Dense(256, activation = 'relu'))
    model.add(BatchNormalization())
    model.add(Dense(model_output_size, activation = 'softmax'))

    # Printing the models summary
    model.summary()

    return model


# Calling the create_model method
model = create_model()

print("Model Created Successfully!")
```

Check Model's Structure:

```
plot_model(model, to_file ='model_structure_plot.png', show_shapes =True, show_layer_names
=True)
```

Compile and Train the Model

```
# Adding Early Stopping Callback
early_stopping_callback = EarlyStopping(monitor = 'val_loss', patience = 15,
mode = 'min', restore_best_weights = True)

# Adding loss, optimizer and metrics values to the model.
```

```
model.compile(loss = 'categorical_crossentropy', optimizer = 'Adam', metrics =
["accuracy"])

# Start Training
model_training_history = model.fit(x = features_train, y = labels_train,
epochs = 2, batch_size = 4 , shuffle = True, validation_split = 0.2, callbacks
= [early_stopping_callback])
```

Evaluating Your Trained Model

```
model_evaluation_history = model.evaluate(features_test, labels_test)
```

Save Your Model

```
# Creating a useful name for our model, incase you're saving multiple models
(OPTIONAL)
date_time_format = '%Y_%m_%d__%H_%M_%S'
current_date_time_dt = dt.datetime.now()
current_date_time_string = dt.datetime.strftime(current_date_time_dt,
date_time_format)
model_evaluation_loss, model_evaluation_accuracy = model_evaluation_history
model_name = f'Model___Date_Time_{current_date_time_string}___Loss_
{model_evaluation_loss}___Accuracy_{model_evaluation_accuracy}.h5'

# Saving your Model
model.save(model_name)
```

Function to Download YouTube Videos:

```
def download_youtube_videos(youtube_video_url, output_directory):
    # Creating a Video object which includes useful information regarding the
youtube video.
    video = pafy.new(youtube_video_url)

    # Getting the best available quality object for the youtube video.
    video_best = video.getbest()

    # Constructing the Output File Path
    output_file_path = f'{output_directory}/{video.title}.mp4'

    # Downloading the youtube video at the best available quality.
    video_best.download(filepath = output_file_path, quiet = True)

    # Returning Video Title

    Return video.title
```

Function To Predict on Live Videos Using Moving Average

```
def predict_on_live_video(video_file_path, output_file_path, window_size):
    # Initialize a Deque Object with a fixed size which will be used to
implement moving/rolling average functionality.
    predicted_labels_probabilities_deque = deque(maxlen = window_size)

    # Reading the Video File using the VideoCapture Object
    video_reader = cv2.VideoCapture(video_file_path)

    # Getting the width and height of the video
    original_video_width = int(video_reader.get(cv2.CAP_PROP_FRAME_WIDTH))
    original_video_height = int(video_reader.get(cv2.CAP_PROP_FRAME_HEIGHT))

    # Writing the Overlayed Video Files Using the VideoWriter Object
    video_writer = cv2.VideoWriter(output_file_path,
cv2.VideoWriter_fourcc('M', 'P', '4', 'V'), 24, (original_video_width,
original_video_height))

    while True:

        # Reading The Frame
        status, frame = video_reader.read()

        if not status:
            break

        # Resize the Frame to fixed Dimensions
        resized_frame = cv2.resize(frame, (image_height, image_width))

        # Normalize the resized frame by dividing it with 255 so that each
pixel value then lies between 0 and 1
        normalized_frame = resized_frame / 255

        # Passing the Image Normalized Frame to the model and receiving
Predicted Probabilities.
        predicted_labels_probabilities =
model.predict(np.expand_dims(normalized_frame, axis = 0))[0]

        # Appending predicted label probabilities to the deque object
        predicted_labels_probabilities_deque.append(predicted_labels_probabil
ities)

        # Assuring that the Deque is completely filled before starting the
averaging process
        if len(predicted_labels_probabilities_deque) == window_size:

            # Converting Predicted Labels Probabilities Deque into Numpy
array
            predicted_labels_probabilities_np =
np.array(predicted_labels_probabilities_deque)

            # Calculating Average of Predicted Labels Probabilities Column
Wise
            predicted_labels_probabilities_averaged =
predicted_labels_probabilities_np.mean(axis = 0)

            # Converting the predicted probabilities into labels by returning
the index of the maximum value.
            predicted_label =
np.argmax(predicted_labels_probabilities_averaged)
```

```
            # Accessing The Class Name using predicted label.
            predicted_class_name = classes_list[predicted_label]

            # Overlaying Class Name Text Ontop of the Frame
            cv2.putText(frame, predicted_class_name, (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)

        # Writing The Frame
        video_writer.write(frame)


        # cv2.imshow('Predicted Frames', frame)

        # key_pressed = cv2.waitKey(10)

        # if key_pressed == ord('q'):
        #     break

    # cv2.destroyAllWindows()


    # Closing the VideoCapture and VideoWriter objects and releasing all
resources held by them.
    video_reader.release()
    video_writer.release()
```

Download a Test Video:

```
# Creating The Output directories if it does not exist
output_directory = 'Youtube_Videos'
os.makedirs(output_directory, exist_ok = True)

# Downloading a YouTube Video
video_title =
download_youtube_videos('https://www.youtube.com/watch?v=8u0qjmHIOcE',
output_directory)

# Getting the YouTube Video's path you just downloaded
input_video_file_path = f'{output_directory}/{video_title}.mp4'
```

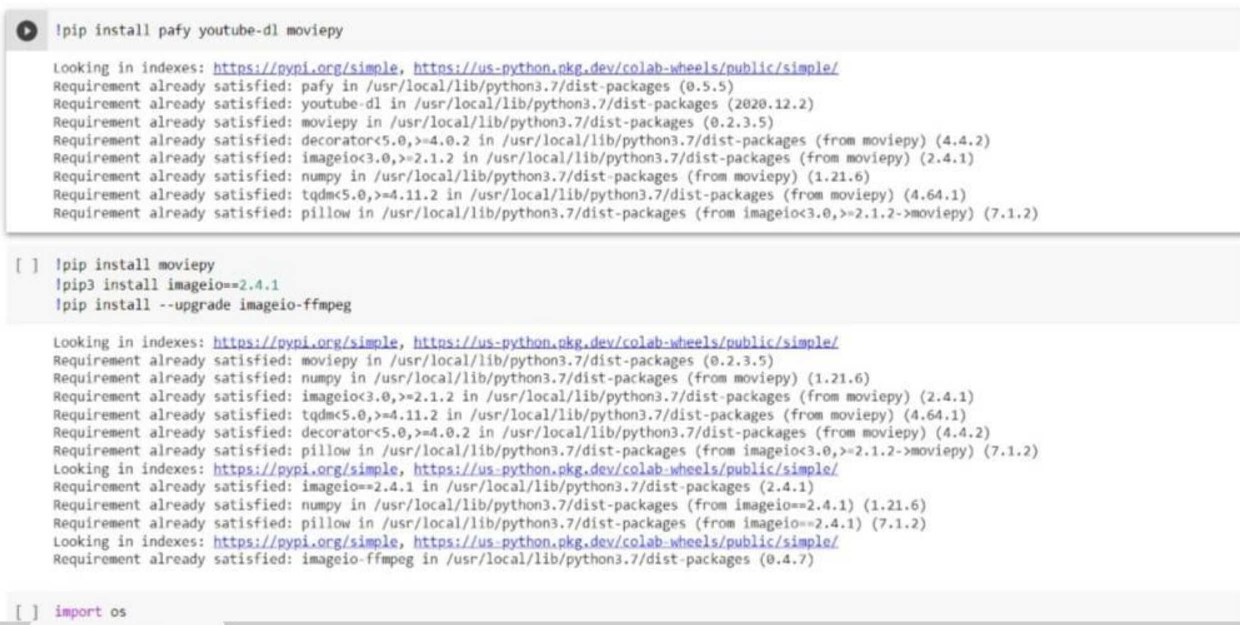
Results

```
# Setting sthe Window Size which will be used by the Rolling Average Proces
window_size = 1

# Constructing The Output YouTube Video Path
output_video_file_path = f'{output_directory}/{video_title} -Output-WSize
{window_size}.mp4'

# Calling the predict_on_live_video method to start the Prediction.
predict_on_live_video(input_video_file_path, output_video_file_path,
window_size)
```

```
# Play Video File in the Notebook
VideoFileClip(output_video_file_path).ipython_display(width = 700)
```

CHAPTER-7
SCREEN SHOTS

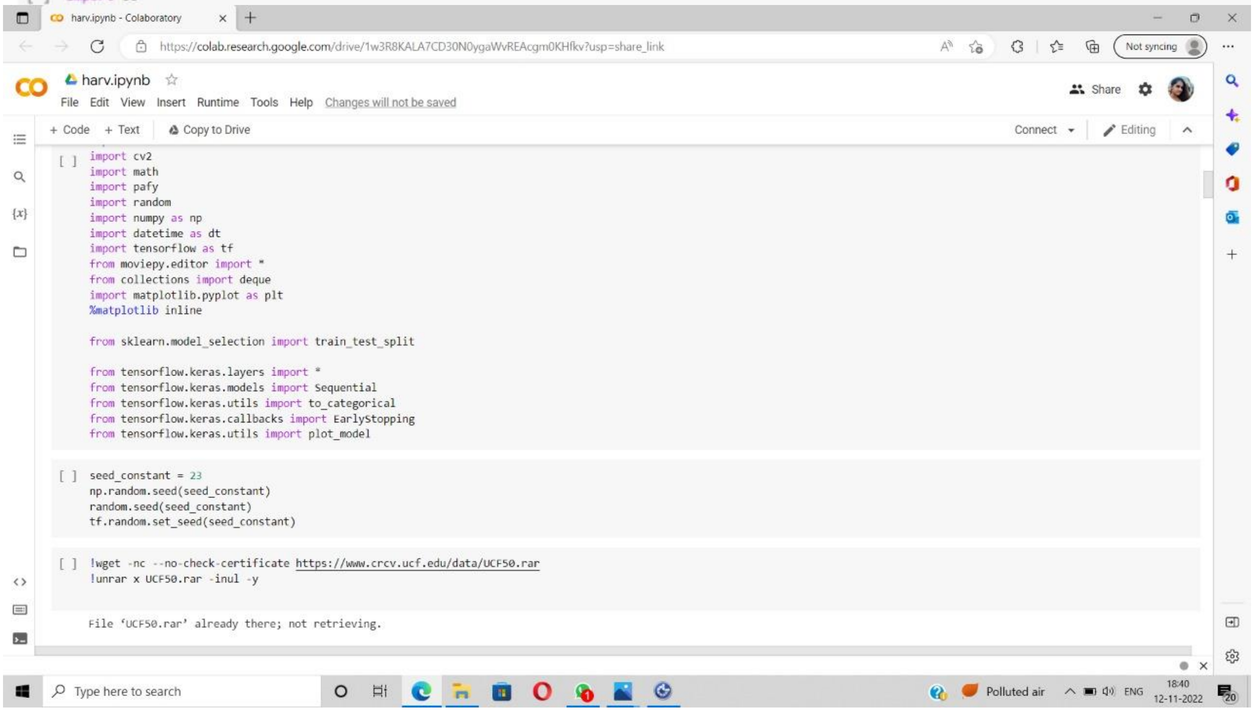
7. SCREEN SHOTS



```
!pip install pafy youtube-dl moviepy

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pafy in /usr/local/lib/python3.7/dist-packages (0.5.5)
Requirement already satisfied: youtube-dl in /usr/local/lib/python3.7/dist-packages (2020.12.2)
Requirement already satisfied: moviepy in /usr/local/lib/python3.7/dist-packages (0.2.3.5)
Requirement already satisfied: decorator<5.0,>=4.0.2 in /usr/local/lib/python3.7/dist-packages (from moviepy) (4.4.2)
Requirement already satisfied: imageio<3.0,>=2.1.2 in /usr/local/lib/python3.7/dist-packages (from moviepy) (2.4.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from moviepy) (1.21.6)
Requirement already satisfied: tqdm<5.0,>=4.11.2 in /usr/local/lib/python3.7/dist-packages (from moviepy) (4.64.1)
Requirement already satisfied: pillow in /usr/local/lib/python3.7/dist-packages (from imageio<3.0,>=2.1.2->moviepy) (7.1.2)
```

```
[ ] !pip install moviepy
    !pip3 install imageio==2.4.1
    !pip install --upgrade imageio-ffmpeg

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: moviepy in /usr/local/lib/python3.7/dist-packages (0.2.3.5)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from moviepy) (1.21.6)
Requirement already satisfied: imageio<3.0,>=2.1.2 in /usr/local/lib/python3.7/dist-packages (from moviepy) (2.4.1)
Requirement already satisfied: tqdm<5.0,>=4.11.2 in /usr/local/lib/python3.7/dist-packages (from moviepy) (4.64.1)
Requirement already satisfied: decorator<5.0,>=4.0.2 in /usr/local/lib/python3.7/dist-packages (from moviepy) (4.4.2)
Requirement already satisfied: pillow in /usr/local/lib/python3.7/dist-packages (from imageio<3.0,>=2.1.2->moviepy) (7.1.2)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: imageio==2.4.1 in /usr/local/lib/python3.7/dist-packages (2.4.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from imageio==2.4.1) (1.21.6)
Requirement already satisfied: pillow in /usr/local/lib/python3.7/dist-packages (from imageio==2.4.1) (7.1.2)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: imageio-ffmpeg in /usr/local/lib/python3.7/dist-packages (0.4.7)
```

```
[ ] import os
```

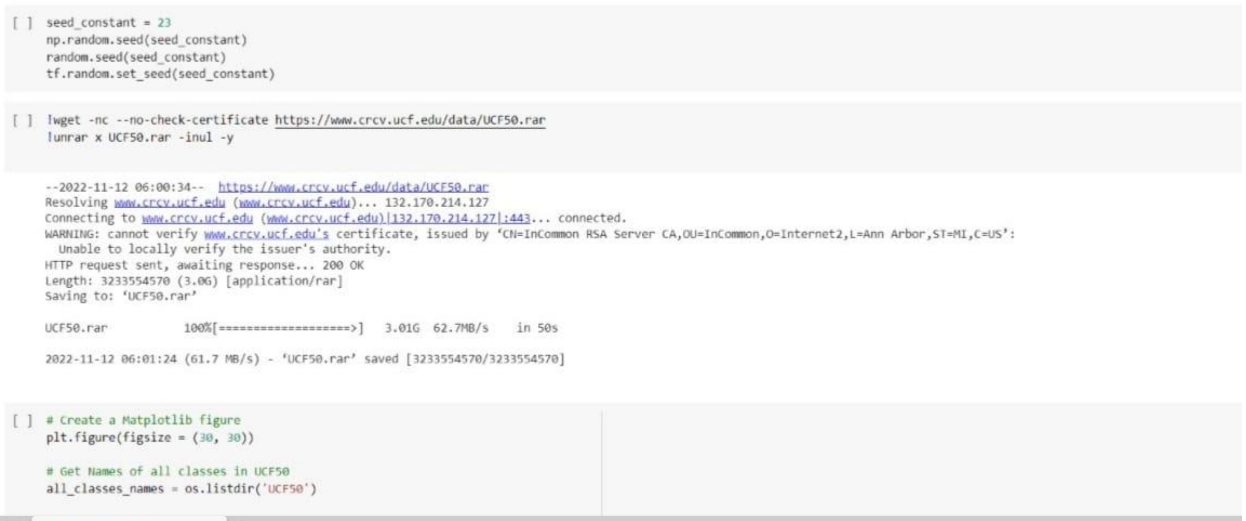


harv.ipynb - Colaboratory
https://colab.research.google.com/drive/1w3R8KALA7CD30N0ygaWvREAcgm0KHfkv?usp=share_link
Not syncing
harv.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
Share
+ Code + Text Copy to Drive
Connect Editing
Type here to search
Polluted air
ENG
18:40
12-11-2022

```
[ ] import cv2
    import math
    import pafy
    import random
    import numpy as np
    import datetime as dt
    import tensorflow as tf
    from moviepy.editor import *
    from collections import deque
    import matplotlib.pyplot as plt
    %matplotlib inline

    from sklearn.model_selection import train_test_split

    from tensorflow.keras.layers import *
    from tensorflow.keras.models import Sequential
    from tensorflow.keras.utils import to_categorical
    from tensorflow.keras.callbacks import EarlyStopping
    from tensorflow.keras.utils import plot_model
```

```
[ ] seed_constant = 23
    np.random.seed(seed_constant)
    random.seed(seed_constant)
    tf.random.set_seed(seed_constant)
```

```
[ ] !wget -nc --no-check-certificate https://www.crcv.ucf.edu/data/UCF50.rar
    !unrar x UCF50.rar -inul -y

    File 'UCF50.rar' already there; not retrieving.
```

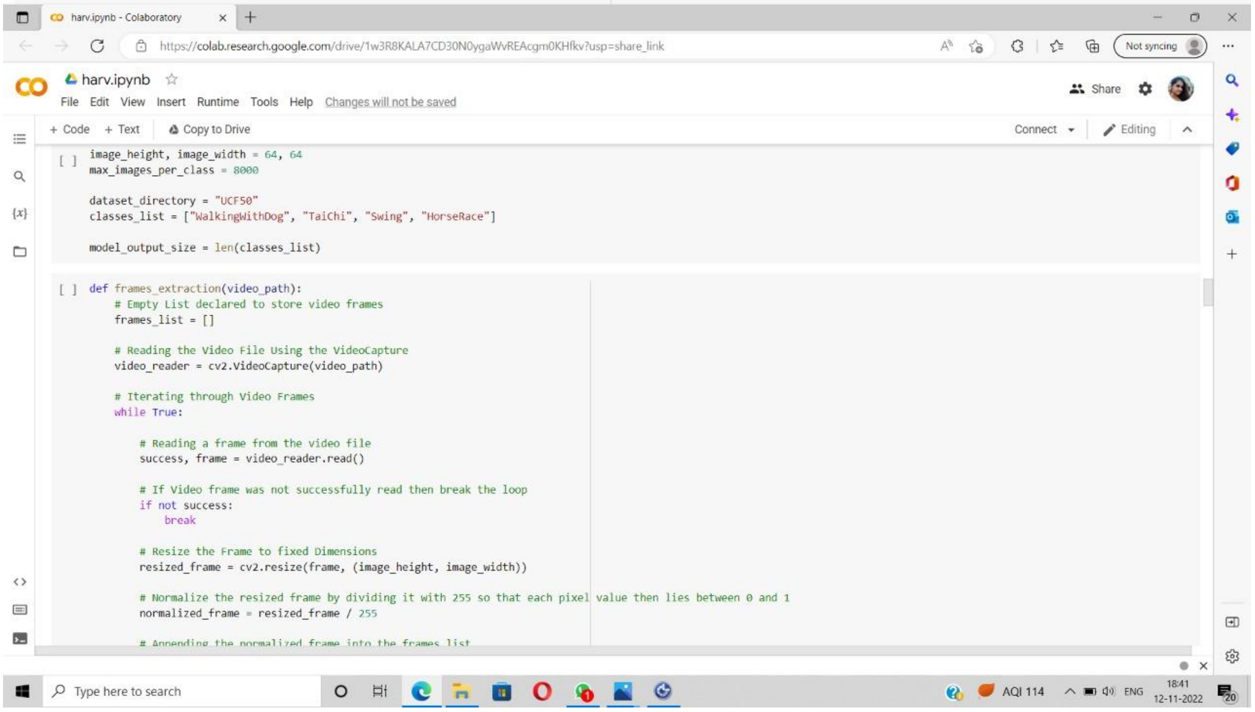


```
[ ] seed_constant = 23
    np.random.seed(seed_constant)
    random.seed(seed_constant)
    tf.random.set_seed(seed_constant)
```

```
[ ] !wget -nc --no-check-certificate https://www.crcv.ucf.edu/data/UCF50.rar
    !unrar x UCF50.rar -inul -y
```

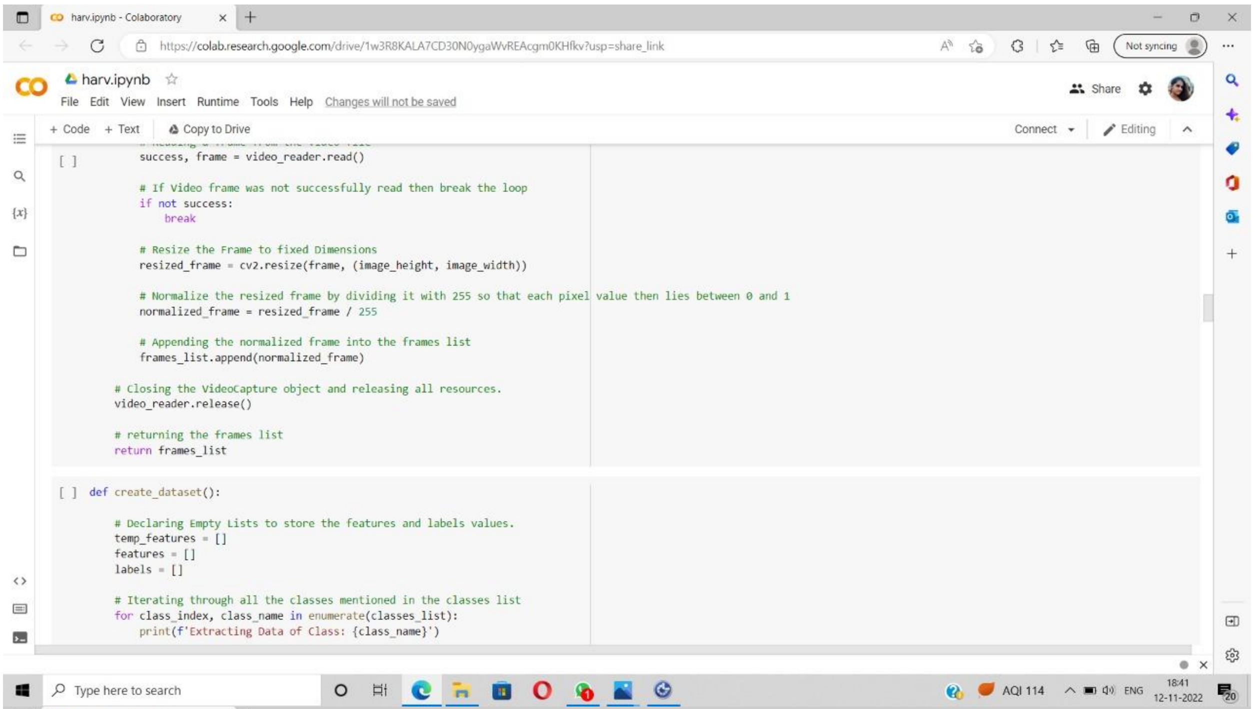
```
--2022-11-12 06:00:34--  https://www.crcv.ucf.edu/data/UCF50.rar
Resolving www.crcv.ucf.edu (www.crcv.ucf.edu)... 132.170.214.127
Connecting to www.crcv.ucf.edu (www.crcv.ucf.edu)|132.170.214.127|:443... connected.
WARNING: cannot verify www.crcv.ucf.edu's certificate, issued by ‘CN=InCommon RSA Server CA,OU=InCommon,O=Internet2,L=Ann Arbor,ST=MI,C=US’:
  Unable to locally verify the issuer's authority.
HTTP request sent, awaiting response... 200 OK
Length: 3233554570 (3.0G) [application/rar]
Saving to: ‘UCF50.rar’

UCF50.rar           100%[===================>]   3.01G  62.7MB/s    in 50s

2022-11-12 06:01:24 (61.7 MB/s) - ‘UCF50.rar’ saved [3233554570/3233554570]
```

```
[ ] # Create a Matplotlib figure
    plt.figure(figsize = (30, 30))

    # Get Names of all classes in UCF50
    all_classes_names = os.listdir('UCF50')
```


harv.ipynb - Colaboratory
https://colab.research.google.com/drive/1w3R8KALA7CD30N0ygaWvREAcgm0KHfkv?usp=share_link
Not syncing
harv.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
Share
+ Code + Text Copy to Drive
Connect Editing
Type here to search
AQI 114 ENG 18:41 12-11-2022

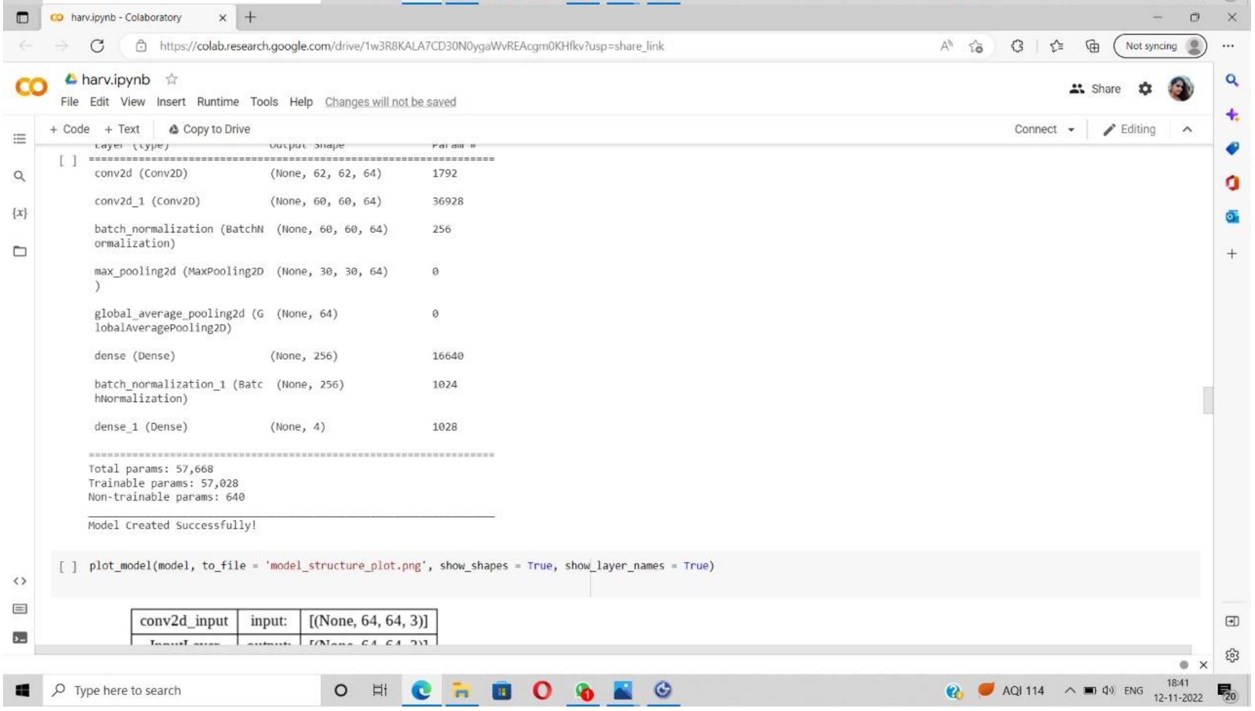
```
[ ] image_height, image_width = 64, 64
    max_images_per_class = 8000

    dataset_directory = "UCF50"
    classes_list = ["WalkingWithDog", "TaiChi", "Swing", "HorseRace"]

    model_output_size = len(classes_list)
```

```
[ ] def frames_extraction(video_path):
        # Empty List declared to store video frames
        frames_list = []

        # Reading the Video File Using the VideoCapture
        video_reader = cv2.VideoCapture(video_path)

        # Iterating through Video Frames
        while True:

            # Reading a frame from the video file
            success, frame = video_reader.read()

            # If Video frame was not successfully read then break the loop
            if not success:
                break

            # Resize the Frame to fixed Dimensions
            resized_frame = cv2.resize(frame, (image_height, image_width))

            # Normalize the resized frame by dividing it with 255 so that each pixel value then lies between 0 and 1
            normalized_frame = resized_frame / 255

            # Appending the normalized frame into the frames list
```



```
success, frame = video_reader.read()

            # If Video frame was not successfully read then break the loop
            if not success:
                break

            # Resize the Frame to fixed Dimensions
            resized_frame = cv2.resize(frame, (image_height, image_width))

            # Normalize the resized frame by dividing it with 255 so that each pixel value then lies between 0 and 1
            normalized_frame = resized_frame / 255

            # Appending the normalized frame into the frames list
            frames_list.append(normalized_frame)

        # Closing the VideoCapture object and releasing all resources.
        video_reader.release()

        # returning the frames list
        return frames_list
```

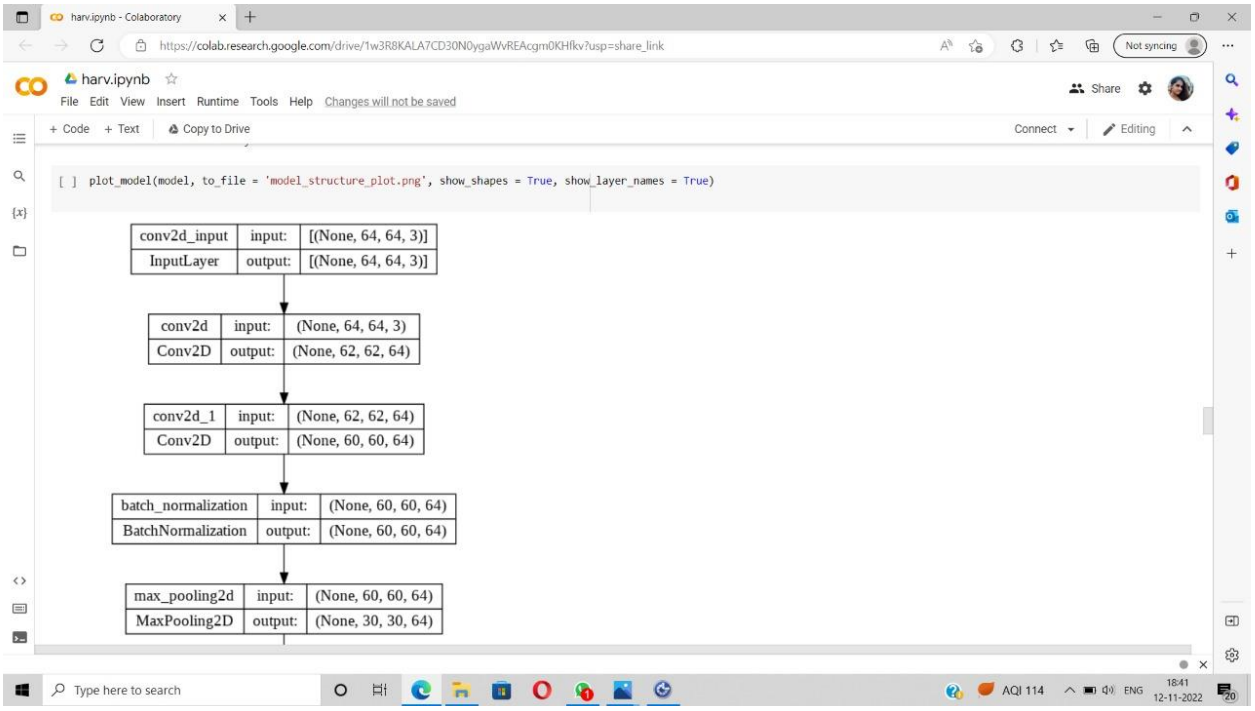
```
def create_dataset():

    # Declaring Empty Lists to store the features and labels values.
    temp_features = []
    features = []
    labels = []

    # Iterating through all the classes mentioned in the classes list
    for class_index, class_name in enumerate(classes_list):
        print(f'Extracting Data of Class: {class_name}')
```



```
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 64)        1792

 conv2d_1 (Conv2D)           (None, 60, 60, 64)        36928

 batch_normalization (BatchN  (None, 60, 60, 64)       256
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 30, 30, 64)       0
 )

 global_average_pooling2d (G  (None, 64)               0
 lobalAveragePooling2D)

 dense (Dense)               (None, 256)               16640

 batch_normalization_1 (Batc  (None, 256)              1024
 hNormalization)

 dense_1 (Dense)             (None, 4)                 1028

=================================================================
Total params: 57,668
Trainable params: 57,028
Non-trainable params: 640
_________________________________________________________________
Model Created Successfully!
```

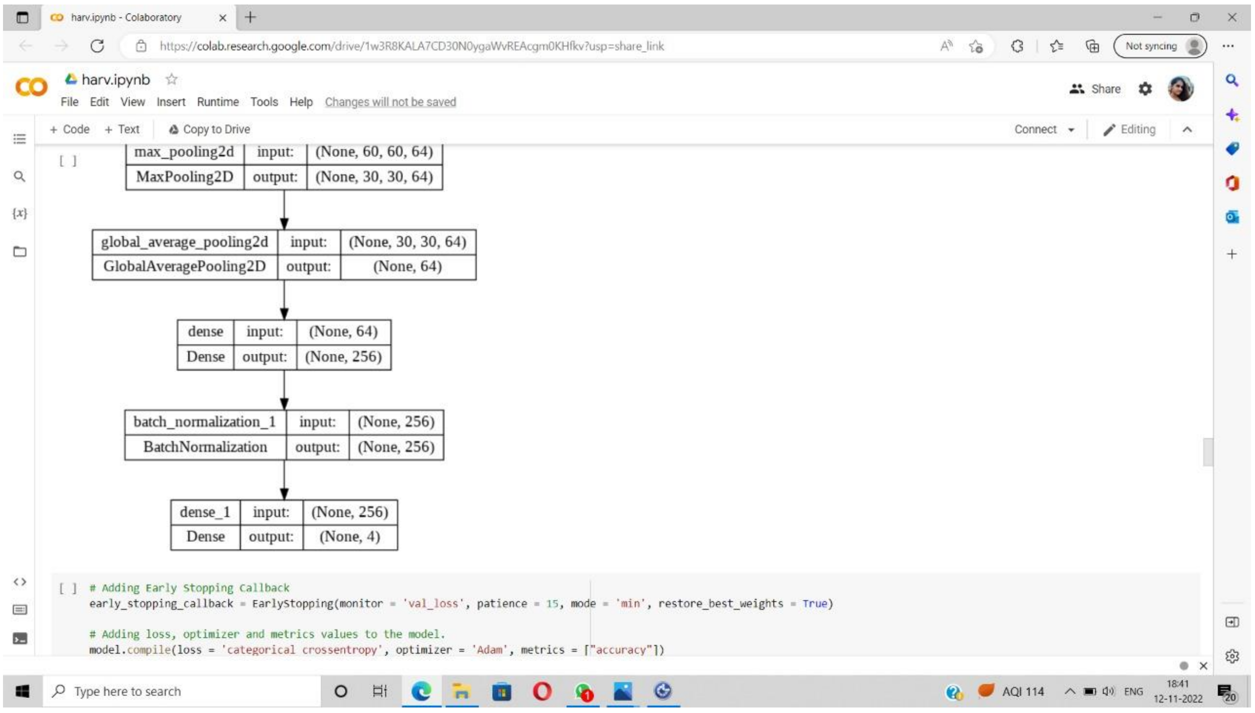
```
plot_model(model, to_file = 'model_structure_plot.png', show_shapes = True, show_layer_names = True)
```

conv2d_input	input:	[(None, 64, 64, 3)]


harv.ipynb - Colaboratory
https://colab.research.google.com/drive/1w3R8KALA7CD30N0ygaWvREAcgm0KHfkv?usp=share_link
harv.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing Share
Type here to search
AQI 114 ENG 18:41 12-11-2022

```
plot_model(model, to_file = 'model_structure_plot.png', show_shapes = True, show_layer_names = True)
```

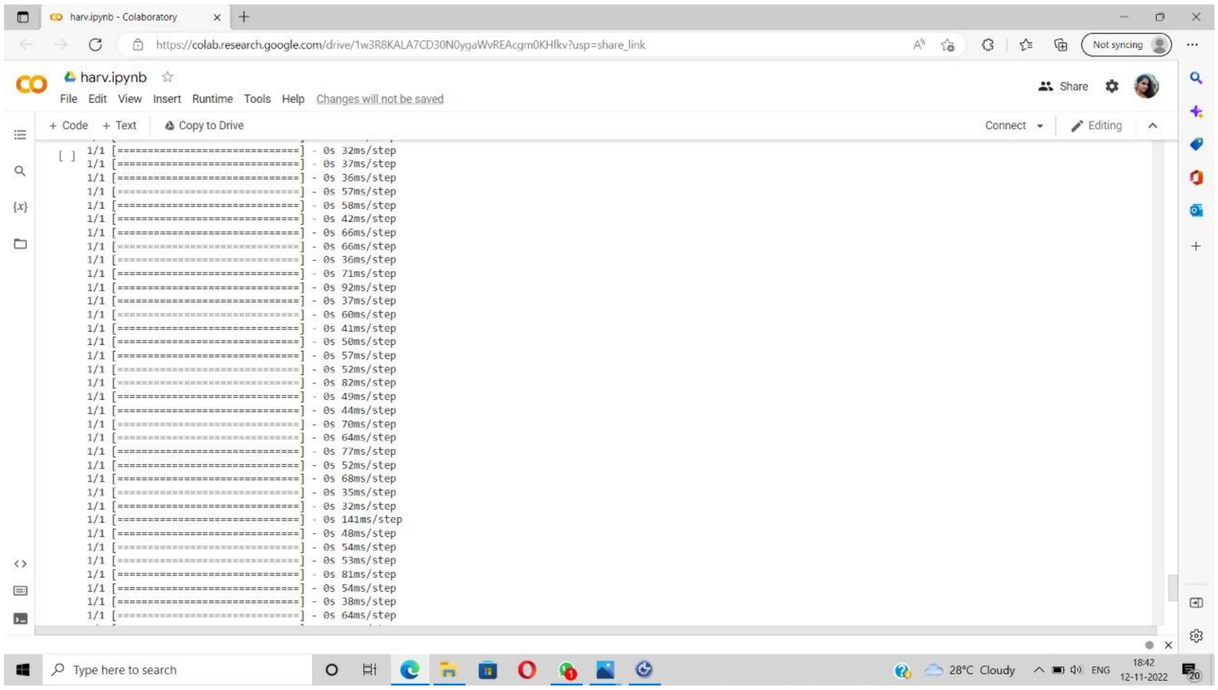
```
conv2d_input        | input:  | [(None, 64, 64, 3)]
InputLayer          | output: | [(None, 64, 64, 3)]

conv2d              | input:  | (None, 64, 64, 3)
Conv2D              | output: | (None, 62, 62, 64)

conv2d_1            | input:  | (None, 62, 62, 64)
Conv2D              | output: | (None, 60, 60, 64)

batch_normalization | input:  | (None, 60, 60, 64)
BatchNormalization  | output: | (None, 60, 60, 64)

max_pooling2d       | input:  | (None, 60, 60, 64)
MaxPooling2D        | output: | (None, 30, 30, 64)
```


harv.ipynb - Colaboratory
https://colab.research.google.com/drive/1w3R8KALA7CD30N0ygaWvREAcgm0KHfkv?usp=share_link
harv.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing Share
Type here to search
AQI 114 ENG 18:41 12-11-2022

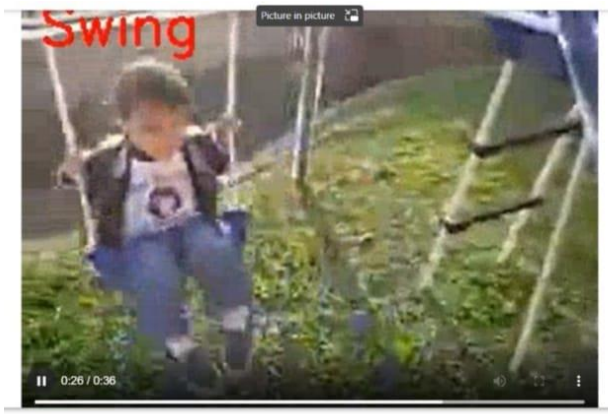
```
max_pooling2d            | input:  | (None, 60, 60, 64)
MaxPooling2D             | output: | (None, 30, 30, 64)

global_average_pooling2d | input:  | (None, 30, 30, 64)
GlobalAveragePooling2D   | output: | (None, 64)

dense                    | input:  | (None, 64)
Dense                    | output: | (None, 256)

batch_normalization_1    | input:  | (None, 256)
BatchNormalization       | output: | (None, 256)

dense_1                  | input:  | (None, 256)
Dense                    | output: | (None, 4)
```

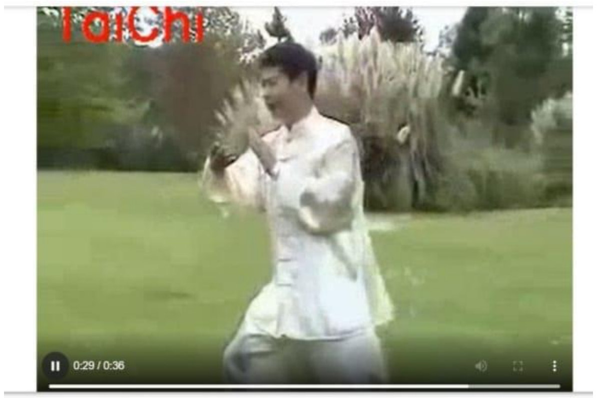
```
# Adding Early Stopping Callback
early_stopping_callback = EarlyStopping(monitor = 'val_loss', patience = 15, mode = 'min', restore_best_weights = True)

# Adding loss, optimizer and metrics values to the model.
model.compile(loss = 'categorical_crossentropy', optimizer = 'Adam', metrics = ["accuracy"])
```

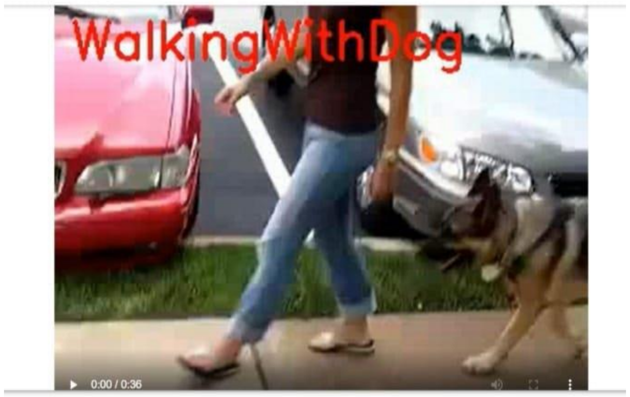
OUTPUTS


harv.ipynb - Colaboratory
https://colab.research.google.com/drive/1w3R8KALA7CD30N0ygaWvREAcgm0KHfkv?usp=share_link
harv.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Share
Connect
Editing
Type here to search
28°C Cloudy
ENG
18:42
12-11-2022

```
[ ] 1/1 [==============================] - 0s 32ms/step
    1/1 [==============================] - 0s 37ms/step
    1/1 [==============================] - 0s 36ms/step
    1/1 [==============================] - 0s 57ms/step
    1/1 [==============================] - 0s 58ms/step
    1/1 [==============================] - 0s 42ms/step
    1/1 [==============================] - 0s 66ms/step
    1/1 [==============================] - 0s 66ms/step
    1/1 [==============================] - 0s 36ms/step
    1/1 [==============================] - 0s 71ms/step
    1/1 [==============================] - 0s 92ms/step
    1/1 [==============================] - 0s 37ms/step
    1/1 [==============================] - 0s 60ms/step
    1/1 [==============================] - 0s 41ms/step
    1/1 [==============================] - 0s 50ms/step
    1/1 [==============================] - 0s 57ms/step
    1/1 [==============================] - 0s 52ms/step
    1/1 [==============================] - 0s 82ms/step
    1/1 [==============================] - 0s 49ms/step
    1/1 [==============================] - 0s 44ms/step
    1/1 [==============================] - 0s 70ms/step
    1/1 [==============================] - 0s 64ms/step
    1/1 [==============================] - 0s 77ms/step
    1/1 [==============================] - 0s 52ms/step
    1/1 [==============================] - 0s 68ms/step
    1/1 [==============================] - 0s 35ms/step
    1/1 [==============================] - 0s 32ms/step
    1/1 [==============================] - 0s 141ms/step
    1/1 [==============================] - 0s 48ms/step
    1/1 [==============================] - 0s 54ms/step
    1/1 [==============================] - 0s 53ms/step
    1/1 [==============================] - 0s 81ms/step
    1/1 [==============================] - 0s 54ms/step
    1/1 [==============================] - 0s 38ms/step
    1/1 [==============================] - 0s 64ms/step
```







CHAPTER-8
CONCLUSION

8.CONCLUSION

In this project, we designed a smart phone-based recognition system that recognizes many activities performed by humans such as walking,riding,swinginhg,talking,jumping,exercising etc.

Inorder to perform these actions CNN plays a major role . The main advantage of CNN compared to its predecessors is that **it automatically detects the important features without any human supervision.**

CHAPTER-9
BIBLOGRAPHY

9.BIBLOGRAPHY

The following web pages are referred during the analysis and execution phase of the project.

WEB LINKS:

- https://www.kaggle.com/code/manishlapasi/cnn-model
- https://pyimagesearch.com/2019/11/25/human-activity-recognition-with-opencv-and-deep-learning/
- https://learnopencv.com/introduction-to-video-classification-and-human-activity-recognition/
- https://youtu.be/TNBpYwgZcPo
- https://www.frontiersin.org/articles/10.3389/frobt.2015.00028/full
- https://machinelearningmastery.com/deep-learning-models-for-human-activity-recognition/
- https://towardsdatascience.com/understand-the-architecture-of-cnn-90a25e244c7