**Hackathon – III**

Table of Contents

**OBJECTIVE: ........................................................................................................................................1**

**BELOW IS THE LIST OF FILES PROVIDED: ..............................................................................................2** EXP\_RECOGNITION.PY: ................................................................................................................................... 2 FACE\_RECOGNITION.PY: ................................................................................................................................. 2 EXP\_RECOGNITION\_MODEL.PY, FACE\_RECOGNITION\_MODEL.PY: ......................................................................... 3 HAARCASCADE\_FRONTALFACE\_DEFAULT.XML, LBPCASCADE\_FRONTALFACE.XML: .................................................... 3

**STEPS TO UPLOAD YOUR CODE TO THE SERVER TO ACCESS IT FROM THE MOBILE APP: .......................3**

**MOBILE APP URL: ...............................................................................................................................3**

**TASKS: ...............................................................................................................................................4**

This hackathon has been designed to help you practice, reinforce and apply various concepts learned in Module - 3.

**Objective:**

Upon successful completion of this Hackathon, you will integrate a system accessible through a mobile app, which can recognize expressions and identify the person in it.

**Datasets:**

• **For Face Recognition:** IMFDB Data is provided. (Data is provided in the shared folder) For more details, click on the link: http://cvit.iiit.ac.in/projects/IMFDB/

• **For Expression Recognition:** IMFDB Data Segregated by expressions is provided. *Note: The data is not uniformly spread across all the classes.*

The Expressions available are as follows:

▪ ANGER

▪ DISGUST

▪ FEAR

▪ HAPPINESS

▪ NEUTRAL

▪ SADNESS

▪ SURPRISE

**Below is the list of files provided:**

1. exp\_recognition\_model.py 2. exp\_recognition.py 3. face\_recognition.py 4. face\_recognition\_model.py 5. haarcascade\_frontalface\_default.xml 6. lbpcascade\_frontalface.xml

exp\_recognition.py:

• **detected\_face:** (To detect faces in the image, upon which expression recognition model)

o It uses Viola-jones face detector o It returns only one image which has maximum area out of all the detected faces

in the image o If no face is detected, it returns zero (0)

• **get\_expression:** (To return the detected Expression from the app)

o Captured image from mobile is passed as parameter in base64 encoding in the

API call. o The code to decode the image in base64 encoding is provided within the

function. o Load the trained model and use it for expression recognition o This function should return the Expression in string form ex: "Anger" o **Caution: Don't change the definition or function name; for loading the model use the *current\_path* variable*(It gives the path of the directory where the python file is getting executed from)*. Example is provided in comments in the file**

face\_recognition.py:

• **get\_similarity:** (To return the similarity between two faces from the app)

o Captured image from mobile is passed as parameter in base64 encoding in the

API call. o The code to decode the image in base64 encoding is provided within the

function. o Load the trained Siamese model o Get the features for both the faces from the model and return the relevant

similarity measure such as euclidean, cosine etc. o **Caution: Don't change the definition or function name; for loading the model use the *current\_path* variable*(It gives the path of the directory where the python file is getting executed from)* . Example is provided in comments in the file.**

• g**et\_face\_class:** (To return the face class from the app)

o Captured image from mobile is passed as parameter in base64 encoding in the

API call.

o The code to decode the image in base64 encoding is provided within the

function. o Load the trained Siamese model o This should return the Face Class in string form ex: "AnilKapoor" o Along with the Siamese, you need the classifier as well, which is to be fine-tuned

with the classes that you want to recognize o **Caution: Don't change the definition or function name; for referring to any**

**path use the *current\_path(It gives the path of the directory where the python file is getting executed from)* variable. Example is provided in comments.**

exp\_recognition\_model.py, face\_recognition\_model.py:

o Define your models, transformation and all necessary helper functions here

respectively for Expression Recognition and face Recognition model o You can ‘import’ these into the files **exp\_recognition.py, face\_recognition.py** o **READ ALL CODE COMMENTS CAREFULLY**

Haarcascade\_frontalface\_default.xml, lbpcascade\_frontalface.xml:

• A Cascade is basically a classifier which is used to detect particular objects from the source. The haarcascade\_frontalface\_default.xml and lbpcascade\_frontalface.xml are cascades designed by OpenCV to detect the frontal face

• Place these files in the same directory as capture\_face\_imagesv1.py, capture\_expression\_imagesv1.py, exp\_recognition.py, face\_recognition.py

**Steps to upload your code to the server to access it from the Mobile app:**

• Upload your files to the given ftp server and test your model on the mobile app

• Steps to upload the updated code files to the server for the mobile app is present in the document “FileZilla installation”

**Mobile App URL:**

<Will be made available here a few days before Hackathon>

• Open this link in your Mobile phone and join as a tester by clicking on the button “Become a tester”

• In the page redirected, click on “download it on google play”

• Install the app in your android phone

• For App usage documentation refer to “Mobile\_APP\_Documentation”

**Tasks:**

For tasks refer the starter code at the following link https://drive.google.com/file/d/17QCNvUNHvNirQWo06uTre7JCtx88kB6U/view?pli=1