**System Requirements specification**

**Contents**

**1 Introduction**

1.1 Purpose

1.2 Project scope

**2.0 General description**

**3.0 Functional requirements**

3.1 Jupyter Notebook / Google Colab

3.2 Python Programming Language

3.3 Webpage

3.4 Database

**4.0 Nonfunctional requirements**

4.1 Usability

4.2 Availability

4.2 Reliability

4.3 Robustness

**5 System specifications**

5.1 Hardware specifications

5.2 Software specifications

**1.0 Introduction**

**1.1 Purpose**

The purpose of this document is to present a detailed description of the Facial emotion recognition using Convolutional Neural Networks. In this project a new approach is described which classifies the expressions with the help of image edge computing. Besides classifying facial emotions, this model requires lesser training time with more accuracy and robustness unlike traditional methods that uses feature extraction.

**1.2 Scope of the Project**

The model designed will be used for recognizing seven types of facial expressions that are considered as universal among all walks of cultures. They are anger, disgust, neutral, fear, happy, sad, and lastly surprise. Facial expression recognition systems run with an objective of understanding and analyzing the true perceptions of people with the help of their expressions. Another important aspect to be noted is that this model is more efficient and accurate when compared to the traditional methods because it is well trained and tested on FER-2013 dataset and uses maximum pooling to reduce the training time. This model can be used by any person who is interested in FER (Facial Expression Recognition) systems.

**2.0 General description**

The main theme of the project is to detect and classify the emotions of a person. Facial expression emotion recognition is an intuitive reflection of a person’s mental state, which contains rich emotional information, and is one of the most important forms of interpersonal communication. In this project a model is build to which facial images are given as input. The file name and its path of these images are taken by the source code to access the images in that address. This is done using a Webpage in which user upload the image. Once the execution is completed the bar charts are displayed as output that describes the classification of emotions.

**3.0 Functional requirements**

**3.1 Jupyter Notebook / Google Colab**

The source code for this model needs to be executed in either Jupyter Notebook or Google Colab. Jupyter Notebook provides a simple platform for executing python programs. It needs some of the disk memory to save python files. It also provides various tools that can help to import the python files from local disk and execute them whenever necessary.

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. The main advantage of Google Colab is that it provides free GPU (Graphics Processing Unit) and some other additional tools.

**3.2 Python Programming language**

We use python to design the model that works on Convolutional Neural Networks. This program consists of various python libraries like “Keras”, “matplotlib”, “tensorflow” etc,.

**3.3 Webpage**

This is where the images are uploaded and taken as input. The webpage consists of an “Upload” button through which the images are stored into the database. The uploaded image is analyzed and facial expressions present in it are classified and displayed as a bar chart.

**3.4 Database**

To store the image data some of the local disk memory is provided. The webpage is linked to this database. So the images uploaded in the webpage are directly sent to this database for storing.

**4.0 Nonfunctional requirements**

**4.1 Usability**

This model can be used by anyone through the designed webpage. The page is very simple to use for anyone who knows a little English vocabulary. The user does not need to understand any of the concepts in the source code. They just need to upload their facial images to know the weighted comparison of different expressions present in their faces.

**4.2 Availability**

This project can be used by anyone anytime without any constraints or restrictions. The webpage can be accessed by anybody who is willing to understand and classify various expressions present in the facial images.

**4.3 Reliability**

This model is quite efficient when compared to the traditional methods with less accuracy rates around 65%. The proposed model runs with an average accuracy rate of around 80%. Even training time is less when compared to other methods. The main advantage of this model lies in its performance and accuracy.

**4.4 Robustness**

This model is efficient even when the facial images are fuzzy. It provides accurate results and detects faces clearly in various types of backgrounds. To increase the robustness the model is trained with the scientific mixture of two datasets ie, Fer-2013 and LFW. And to know the standards of this model its reliability factor is compared to algorithms like FRR-CNN and R-CNN.

**5 System specifications**

**5.1 Hardware specifications**

* Network connectivity.
* Minimum 4GB RAM is required.
* i5/ Intel Processor
* Graphic Card is required.

**5.2 Software specifications**

* Operating System : Windows 10
* Server-side Script : Python 3.6
* IDE : PyCharm, Jupyter notebook
* Libraries Used : Numpy, Flask, keras, pandas,

FER(Face Emotion Recognizer)