**Project Report**

**#Libraries used in this project**

**Tensorflow -->** TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

**keras -->** provides an approachable, highly-productive interface for solving machine learning (ML) problems, with a focus on modern deep learning. Keras covers every step of the machine learning workflow, from data processing to hyperparameter tuning to deployment.

**pandas -->** Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.

**Numpy -->** NumPy can be used to perform a wide variety of mathematical operations on arrays.

**tqdm -->** Tqdm is a popular Python library that provides a simple and convenient way to add progress bars to loops and iterable objects.

**opencv-contrib -->** These additional modules cover a wide range of computer vision tasks, including object detection, image processing, feature extraction, and more.

**#Algorithm**

we use CNN algorithm to built this project we used differet layers. CNN (Convolutional Neural Network) is a deep learning algorithm specialized in processing grid-like data, with layers like convolutional layers for feature extraction, pooling layers for downsampling, and fully connected layers for classification.

**#DataFrames in Pandas Example**

|  |  |
| --- | --- |
| **Col\_1(Location of images)** | **Col\_2** |
| **Images (main\_folder/subfolder/image\_name)** | **labels** |
| images/angry/faizan.png | angry |
| images/angry/Ali.png | angry |
| images/Sad/faizan2.png | sad |
| images/happy/faizan3.png | happy |

**#Preprocessing\_Images**

Then we process the each images by converting them into arrays by using method of which is in tensoflow which is load\_images and we load the images in **grayscale** (Grayscaling converts images from color (RGB) format to black and white format. Grayscaling of images involves converting them from color (RGB) to grayscale, where each pixel's intensity is represented by a single value ranging from 0 to 255, indicating the brightness of the pixel. This process simplifies image processing tasks by removing color information while retaining the image's structural details.)

**For converting images in grayscale:**

Load\_img(image,grayscale=True)

After gray scaling we stored the images to train the model.

**#Dependencies**

from keras.utils import to\_categorical

from keras.preprocessing.image import load\_img

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

import os

import pandas as pd

import numpy as np

**1. `keras.utils.to\_categorical`:** This function is used to convert class labels into one-hot encoded vectors. It's commonly used when dealing with categorical data in machine learning tasks, particularly in classification problems.

**2. `keras\_preprocessing.image.load\_img`:** This function is part of the Keras preprocessing module and is used to load an image file into a PIL (Python Imaging Library) object. It's often used in image processing tasks as a preliminary step before further manipulation or analysis.

**3. `keras.models.Sequential`:** This is a Keras class that allows you to create a sequential neural network model, where layers are added sequentially one after another. It's a convenient way to build deep learning models without explicitly defining the connections between layers.

**4. `keras.layers.Conv2D`, `keras.layers.MaxPooling2D`, `keras.layers.Flatten`, `keras.layers.Dense`, `keras.layers.Dropout`:** These are different types of layers commonly used in convolutional neural networks (CNNs). `Conv2D` represents a 2D convolutional layer, `MaxPooling2D` is a 2D max pooling layer, `Flatten` is used to flatten the input into a 1D array, `Dense` represents a fully connected layer, and `Dropout` is a regularization technique used to prevent overfitting by randomly dropping units during training.

**5. `os`:** This is a Python built-in module used for interacting with the operating system. It provides functions to perform tasks such as navigating file directories, creating and deleting directories, and executing system commands.

**6. `pandas`:** This is a popular Python library used for data manipulation and analysis. It provides data structures like DataFrame and Series, as well as functions for reading and writing data from various file formats, data cleaning, transformation, and analysis.

**7. `numpy`:** This is another widely used Python library for numerical computing. It provides support for large multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. numpy is often used in conjunction with other libraries like pandas and keras for data manipulation and model training.