FACIAL EMOTION RECOGNITION

1. DATASET

JAFFE (Japanese Female Facial Expressions):

The JAFFE dataset contains 213 images of seven facial expressions (neutral, sadness, surprise, happiness, fear, anger, and disgust) in poses of ten female Japanese models collected by the Psychology Department at Kyushu University.

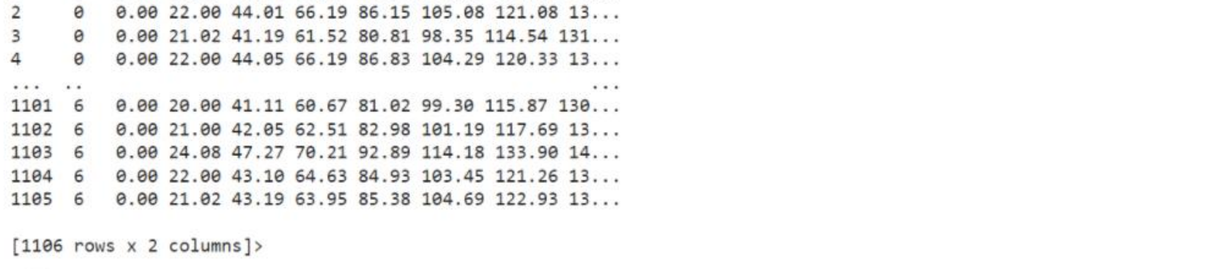
CK+ (Extended Cohn-Kanade Dataset):

The CK+ comprises a total of 593 sequences((327 sequences having discrete emotion labels-neutral, sadness, surprise, happiness, fear, anger,contempt and disgust) across 123 subjects.

The above mentioned datasets were collected from Kaggle and 1106 images uniformly distributed across the emotions-neutral, sadness, surprise, happiness, fear, anger, and disgust were selected.

The images are resized and converted to grayscale. The faces are detected using dlib frontal face detector . From the faces detected ,the coordinates of 68 key facial landmarks that map to facial structures(eyes,nose,mouth,jaw,etc) on the face are estimated using dlib's pre-trained facial landmark detector.

The Euclidean distance between all 68 landmarks is calculated and stored as 68x68 array.



```
<bound method NDFrame.head of       0  0.00 22.09 43.29 64.63 84.72 102.20 119.40 132.41 144.27 156.54 165.61 171.
0     0   0.00 23.09 46.39 68.25 88.06 107.56 124.34 13...
1     0   0.00 23.19 44.55 67.47 88.32 107.70 124.42 14...
2     0   0.00 22.00 44.01 66.19 86.15 105.08 121.08 13...
3     0   0.00 21.02 41.19 61.52 80.81 98.35 114.54 131...
4     0   0.00 22.00 44.05 66.19 86.83 104.29 120.33 13...
...  ..                                               ...
1101  6   0.00 20.00 41.11 60.67 81.02 99.30 115.87 130...
1102  6   0.00 21.00 42.05 62.51 82.98 101.19 117.69 13...
1103  6   0.00 24.08 47.27 70.21 92.89 114.18 133.90 14...
1104  6   0.00 22.00 43.10 64.63 84.93 103.45 121.26 13...
1105  6   0.00 21.02 43.19 63.95 85.38 104.69 122.93 13...

[1106 rows x 2 columns]>
```

4.2.2 SOFTWARE REQUIREMENTS AND LIBRARIES REQUIRED

Google Colaboratory is used to build the machine learning model for facial emotion recognition.Colab is a free Jupyter notebook environment that runs entirely in the cloud.It is used to write and execute code in Python and supports several popular machine learning libraries.

The libraries required to develop the model for facial emotion recognition are:

Numpy – adding support for large, multi- dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Pandas- built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

Scikit-learn -free software machine learning library for the Python programming language.

Mathplotlib - plotting library for the Python programming language and its numerical mathematics extension NumPy.

Tensorflow - a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

Keras - Open Source Neural Network library written in Python that runs on top of Theano or Tensorflow. It is designed to be modular, fast and easy to use.

Dlib - Dlib is principally a C++ library, however, you can use a number of its tools from python applications.Dlib implements a variety of machine learning algorithms, including classification, regression, clustering, data transformation,image processing and structured prediction.

OpenCV-OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

3. DATA PREPROCESSING

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model in order to improve efficiency. It is the first and crucial step while creating a machine learning model.

The JAFFE_CK.csv dataset(1106 rowsx2 columns) is loaded.The 68x68 Euclidean distances between facial landmarks (independent variables or features) and the emotion value (dependent variable) are extracted from the dataset into.Then,the emotion list is converted into a numpy array with integers(0-6) that represent seven emotions and encoded into categorical data using the to_categorical method in Keras.This method converts the given input numpy array into a numpy array with 7 columns containing binary values.

The dataset is then split into training and testing sets using the train_test_split function in Scikit-learn.Here,66 % of the data is used for training and 33 percent is used for testing.

Finally, feature scaling is done on the Euclidean distances between facial landmarks in order to normalize them. This is done using n object of MinMaxScaler class in Scikit-learn which scales and translates each feature individually such that it is in the range 0 to 1.Then, the fit_transform method is called to fit and transform the training data.

4. BUILDING NEURAL NETWORK

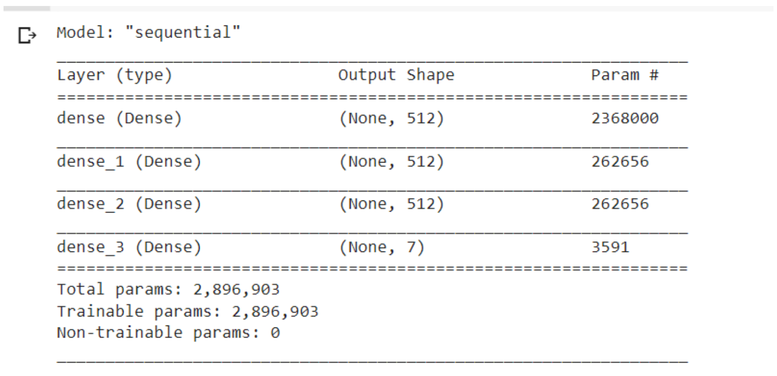
In the dataset used, input has 4624 values(68*68) and output has 7 emotion values(happy,sad,angry,fear,neutral,surprise,disgust).

A sequential model that allows linear stacking of layers is created and fully connected layers defined using Dense class are added one at a time.The input layer has 4624(68x68) Euclidean distances which is specified when creating the first hidden layer. The model has an input layer with 4624 features.The three hidden layers have 512 nodes each and use the ReLU activation function. The output layer has 7 nodes(one for each emotion) and is activated using Softmax activation function as we have multiple classes of emotions. Activation functions are mathematical equations that determine the output of a neural network for a given input set.The function is attached to each neuron in the network, and determines whether it should be activated (“fired”) or not, based on whether each neuron’s input is relevant for the model’s prediction.

The rectified linear activation function or ReLU activation function has been used as it is better suited for Multi class and overcomes the vanishing gradient problem which detrimentally affects the accuracy of the model. ReLU is also very quick to evaluate. The softmax activation function is is a more

generalized logistic activation function that is used for multiclass classification. It is generally used in the output layer, where it takes value and transforms it into probability distribution.

Later, the model is compiled.The loss function used is categorical cross entropy which is a multi-class classification loss function.Optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses.Here,Adam optimize is used.Metrics is used to specify the way in which the neural network's performance is judged.Here, metrics used is accuracy.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 512)               2368000
_________________________________________________________________
dense_1 (Dense)              (None, 512)               262656
_________________________________________________________________
dense_2 (Dense)              (None, 512)               262656
_________________________________________________________________
dense_3 (Dense)              (None, 7)                 3591
=================================================================
Total params: 2,896,903
Trainable params: 2,896,903
Non-trainable params: 0
_________________________________________________________________
```

Fig c.a :building neural network

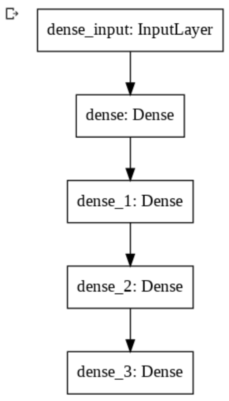
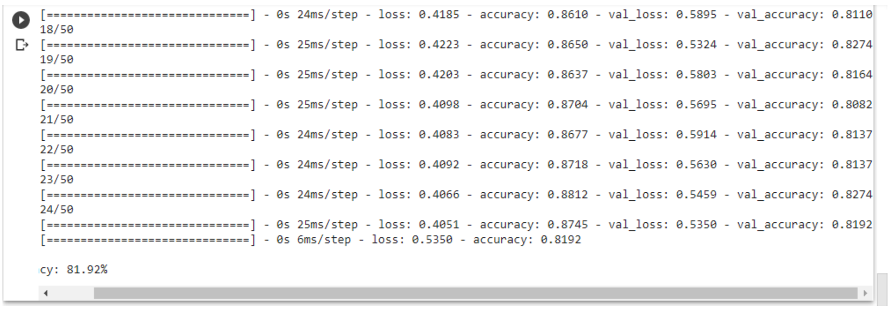


Fig c.b : building neural network 1

4.2.5 TRAINING THE MODEL

The model is trained using the fit function over 50 epochs with batch size of 64 and the testing set is used fit validation.Furthermore,callbacks such as EarlyStopping and ReduceLROnPlateau are used. A callback is a set of functions to be applied at given stages of the training procedure. Callbacks are used to get a view on internal states and statistics of the model during training and improve the model. The EarlyStopping function is used to avoid overfitting and has various metrics/arguments that can be modified to set up when the training process should stop.   ReduceLRonPlataeu is a class which reduces learning rate when a metric shows no improvement is seen in patience number of epochs.



```
[==============================] - 0s 24ms/step - loss: 0.4185 - accuracy: 0.8610 - val_loss: 0.5895 - val_accuracy: 0.8110
18/50
[==============================] - 0s 25ms/step - loss: 0.4223 - accuracy: 0.8650 - val_loss: 0.5324 - val_accuracy: 0.8274
19/50
[==============================] - 0s 25ms/step - loss: 0.4203 - accuracy: 0.8637 - val_loss: 0.5803 - val_accuracy: 0.8164
20/50
[==============================] - 0s 25ms/step - loss: 0.4098 - accuracy: 0.8704 - val_loss: 0.5695 - val_accuracy: 0.8082
21/50
[==============================] - 0s 24ms/step - loss: 0.4083 - accuracy: 0.8677 - val_loss: 0.5914 - val_accuracy: 0.8137
22/50
[==============================] - 0s 24ms/step - loss: 0.4092 - accuracy: 0.8718 - val_loss: 0.5630 - val_accuracy: 0.8137
23/50
[==============================] - 0s 24ms/step - loss: 0.4066 - accuracy: 0.8812 - val_loss: 0.5459 - val_accuracy: 0.8274
24/50
[==============================] - 0s 25ms/step - loss: 0.4051 - accuracy: 0.8745 - val_loss: 0.5350 - val_accuracy: 0.8192
[==============================] - 0s 6ms/step - loss: 0.5350 - accuracy: 0.8192

cy: 81.92%
```

Fig d: training the model

4.2.6 TESTING THE MODEL ON REAL TIME IMAGES

As seen earlier, the sequential ANN model was built,trained,evaluated and found to have an accuracy of 81.92%

Real-time images are captured via the device's web camera or obtained from the drive.The image is read and converted to grayscale to increase robustness with the help of OpenCV functions.The face in the image is detected using the pre-trained Haarcascade frontal face classifier and a rectangle is drawn around it.

Later, using dlib's frontal face detector and shape predictor, 68 facial landmarks are located and the Euclidean distances between the landmarks are calculated.These distances are stored in a numpy array and scaled and normalised into the 0 to 1 range.The predict function takes these distance values as input and the model predicts the probabilities of the seven emotions.The emotion with the highest probability is written onto the image and displayed using OpenCV functions