Face Emotion Recognition

By

Ashish Gupta

Data Science Trainee, Almabetter, Bangalore

Contents: -

Acknowledgement

First of all, I would like to thank Almabetter for giving me opportunity to learn amazing things of deep learning. I would like to thank my all teachers and student success manager for helping me to solve my doubts.

Beside from my lecturer, I like to thank my friends for helping to understand the project related questions more clearly. I thank them for their effort.

Abstract: These Human facial expressions convey a lot of information visually rather than articulately. Facial expression recognition plays a crucial role in the area of human-machine interaction. Automatic facial expression recognition system has many applications including, but not limited to, human behavior understanding, detection of mental disorders, and synthetic human expressions. Recognition of facial expression by computer with high recognition rate is still a challenging task. Two popular methods utilized mostly in the literature for the automatic FER systems are based on geometry and appearance. Facial Expression Recognition usually performed in four-stages consisting of pre-processing, face detection, feature extraction, and expression classification. Facial recognition is a very useful tool and has been researched extensively in recent years. The applications for facial recognition vary from use in security cameras to emotion detection. Emotion detection in particular is a facet of facial recognition that has great potential in a wide range of fields. In order to tailor the software for emotion detection, a series of steps must be taken. In this project I applied deep learning method (convolutional neural network) to identify the key seven human emotions: anger, disgust, fear, happiness, sadness, surprise and neutrality.

Keywords: Emotions, fer2013, recognition, deep learning, neural network etc.

1. PROBLEM STATEMENT:

The Indian education landscape has been undergoing rapid changes for the past 10 years owing to the advancement of web-based learning services, specifically, eLearning platforms.

Global E-learning is estimated to witness an 8X over the next 5 years to reach USD 2B in 2021. India is expected to grow with a CAGR of 44% crossing the 10M users mark in 2021. Although the market is growing on a rapid scale, there are major challenges associated with digital learning when compared with brick-and-mortar classrooms. One of many challenges is how to ensure quality learning for students. Digital platforms might overpower physical classrooms in terms of content quality but when it comes to understanding whether students are able to grasp the content in a live class scenario is yet an open-end challenge.

In a physical classroom during a lecturing teacher can see the faces and assess the emotion of the class and tune their lecture accordingly, whether he is going fast or slow. He can identify students who need special attention. Digital classrooms are conducted via video telephony software program (ex-Zoom) where it’s not possible for medium scale class (25-50) to see all students and access the mood. Because of this drawback, students are not focusing on content due to lack of surveillance. While digital platforms have limitations in terms of physical surveillance but it comes with the power of data and machines which can work for you. It provides data in the form of video, audio, and texts which can be analyzed using deep learning algorithms. Deep learning backed system not only solves the surveillance issue, but it also removes the human bias from the system, and all information is no longer in the teacher’s brain rather translated in numbers that can be analyzed and tracked.

We will solve the above-mentioned challenge by applying deep learning algorithms to live video data. The solution to this problem is by recognizing facial emotions.

Attribute Information:

This dataset contains information on emotion, pixels & Usage. recognizing your emotions just looking at your facial expressions.

1. Emotion: emotion that contains numeric code from 0-6 (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).
2. Pixels: pixels values in the range of 0-255.
3. Usage: training, public test and private test.

2. INTRODUCTION

Facial emotions are very important and play significant role in communication among humans. Emotions convey the motive of the persons. Facial emotional recognition (FER) is an interdisciplinary domain standing at the crossing of behavioral science, psychology, neurology and artificial intelligence. In this new competitive world security are very much concerned in public and private sectors. Emotional face recognition has become one of the most active research areas at present and many organizations are showing interest to invest. Due to this reason emotional face recognition is one of the hottest research areas in computer vision and machine learning. Very few different emotional face recognition algorithms have been designed. Face is the most commonly used for identification, recognition and verification. Emotional face recognition is so important in day today's life. Passwords and cards are the most extensive used in biometric system; because of non-contact process face recognition is the most advantageous and effective systems. It is the most complex object recognition biometric system. Because human faces are the most similar structure. Due to this reason many automatic face recognition algorithms have been designed but still the recognition rate is not up to the mark. Still there is no fixed benchmark database for testing existing algorithms. The most drawback of face recognition are inclusion of facial expression, illumination, aging, hair style, pose, scaling, frontal, present or absent of spectacle and occlusion. Lighting condition, facial expression and deep learning are the most talking in the recent past. Bright lighting causes image saturation. Emotions often mediate and facilitate interactions among human beings. Thus, understanding emotion often brings context to seemingly bizarre and/or complex social communication. Emotion can be recognized through a variety of means such as voice intonation, body language, and more complex methods; such electroencephalography (EEG) [1]. However, the easier, more practical method is to examine facial expressions. There are seven types of human emotions shown to be universally recognizable across different cultures [2]: anger, disgust, fear, happiness, sadness, surprise, contempt. Interestingly, even for complex expressions where a mixture of emotions could be used as descriptors, cross-cultural agreement is still observed [3]. Therefore, a utility that detects emotion from facial expressions would be widely applicable. Such advancement could bring applications in medicine, marketing and entertainment [4]. The task of emotion recognition is particularly difficult for two reasons: 1) There does not exist a large database of training images and 2) classifying emotion can be difficult depending on whether the input image is static or a transition frame into a facial expression. With the advent of modern technology our desires went high and it binds no bounds. In the present era a huge research work is going on in the field of digital image and image processing. The way of progression has been exponential and it is ever increasing. Image Processing is a vast area of research in present day world and its applications are very widespread. Image processing is the field of signal processing where both the input and output signals are images. One of the most important applications of Image processing is Facial expression recognition. Our emotion is revealed by the expressions in our face. Facial Expressions plays an important role in interpersonal communication. Facial expression is a nonverbal scientific gesture which gets expressed in our face as per our emotions. Automatic recognition of facial expression plays an important role in artificial intelligence and robotics and thus it is a need of the generation. Some application related to this include Personal identification and Access control, Videophone and Teleconferencing, Forensic application, Human-Computer Interaction, Automated Surveillance, Cosmetology and so on. The objective of this project is to develop Automatic Facial Expression Recognition System which can take human facial images containing some expression as input and recognize and classify it into seven different expression class.

3. Problem Definition:

Human facial expressions can be easily classified into 7 basic emotions: happy, sad, surprise, fear, anger, disgust, and neutral. Our facial emotions are expressed through activation of specific sets of facial muscles. These sometimes subtle, yet complex, signals in an expression often contain an abundant amount of information about our state of mind. Through facial emotion recognition, we are able to measure the effects that content and services have on the audience/users through an easy and low-cost procedure. For example, retailers may use these metrics to evaluate customer interest. Healthcare providers can provide better service by using additional information about patients' emotional state during treatment. Entertainment producers can monitor audience engagement in events to consistently create desired content. Humans are well-trained in reading the emotions of others, in fact, at just 14 months old, babies can already tell the difference between happy and sad. But can computers do a better job than us in accessing emotional states? To answer the question, we designed a deep learning neural network that gives machines the ability to make inferences about our emotional states. In other words, we give them eyes to see what we can see.

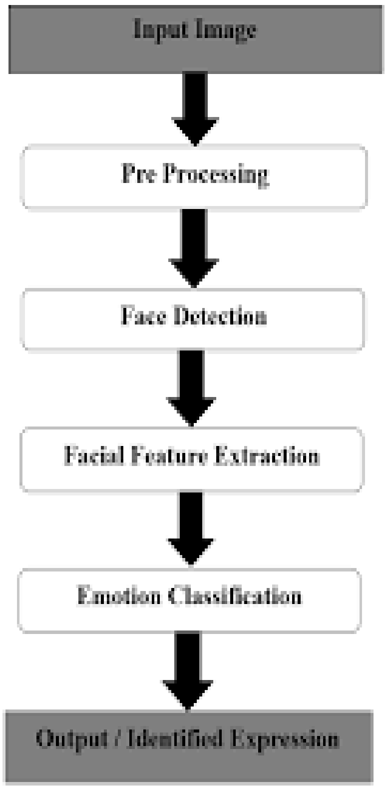


Figure 1: problem definition

Facial expression recognition is a process performed by humans or computers, which consists of: 1. Locating faces in the scene (e.g., in an image; this step is also referred to as face detection), 2. Extracting facial features from the detected face region (e.g., detecting the shape of facial components or describing the texture of the skin in a facial area; this step is referred to as facial feature extraction), 3. Analyzing the motion of facial features and/or the changes in the appearance of facial features and classifying this information into some facial-expression interpretative categories such as facial muscle activations like smile or frown, emotion (affect)categories like happiness or anger, attitude categories like (dis)liking or ambivalence, etc.(this step is also referred to as facial expression interpretation).

4. Four basic steps:

As per various literature surveys it is found that for implementing this project four basic steps are required to be performed.

i. Preprocessing
ii. ii. Face registration
iii. iii. Facial feature extraction
iv. iv. Emotion classification

Description about all these processes is given below-

- Preprocessing:

Preprocessing is a common name for operations with images at the lowest level of abstraction both input and output are intensity images. Most preprocessing steps that are implemented are –

a. Reduce the noise

b. Convert the Image to Binary/Grayscale.

c. Pixel Brightness Transformation.

d. Geometric Transformation.

- Face Registration:

Face Registration is a computer technology being used in a variety of applications that identifies human faces in digital images. In this face registration step, faces are first located in the image using some set of landmark points called "face localization" or "face detection". These detected faces are then geometrically normalized to match some template image in a process called "face registration".

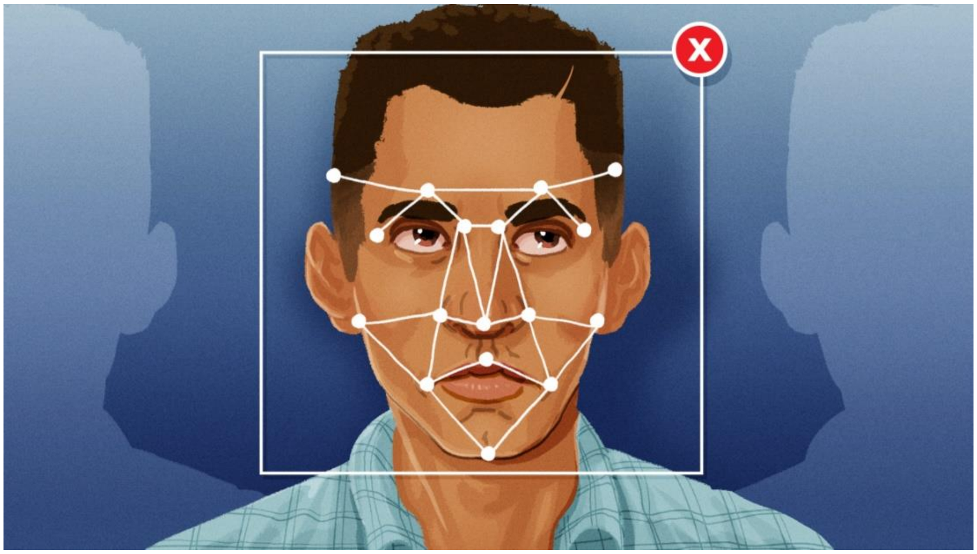


Figure 2: Face Registration

- Facial Feature Extraction:

Facial Features extraction is an important step in face recognition and is defined as the process of locating specific regions, points, landmarks, or curves/contours in a given 2-D image or a 3D range image. In this feature extraction step, a numerical feature vector is generated from the resulting registered image. Common features that can be extracted area.

a. Lips

b. Eyes

c. Eyebrows

d. Nose tip

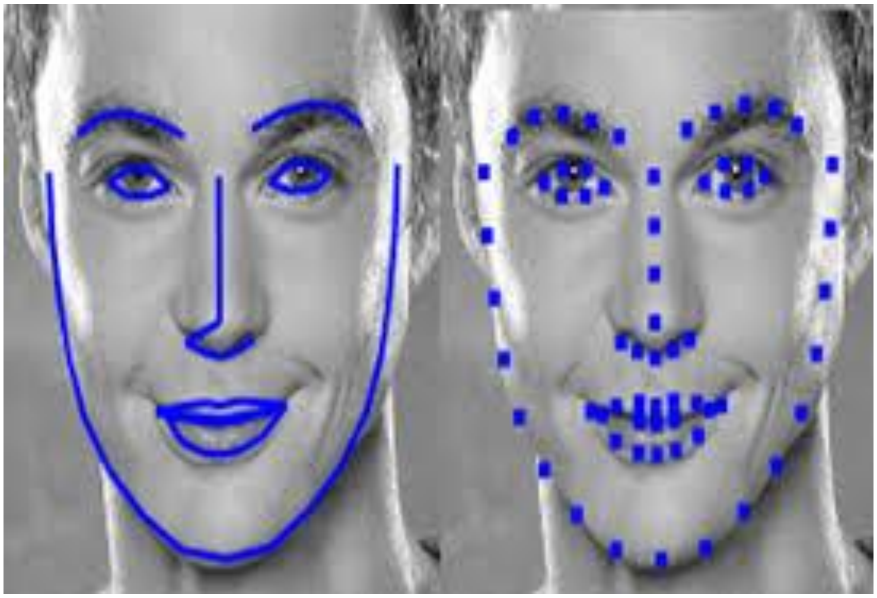


Figure 3: Feature Extraction

- ❖ Emotion Classification:

In the third step, of classification, the algorithm attempts to classify the given faces portraying one of the seven basic emotions.

Figure 4: Face Emotion Classification

- Neural Network Approach:

The neural network contained a hidden layer with neurons. The approach is based on the assumption that a neutral face image corresponding to each image is available to the system. Each neural network is trained independently with the use of on-line back propagation. Neural Network will be discussed later.

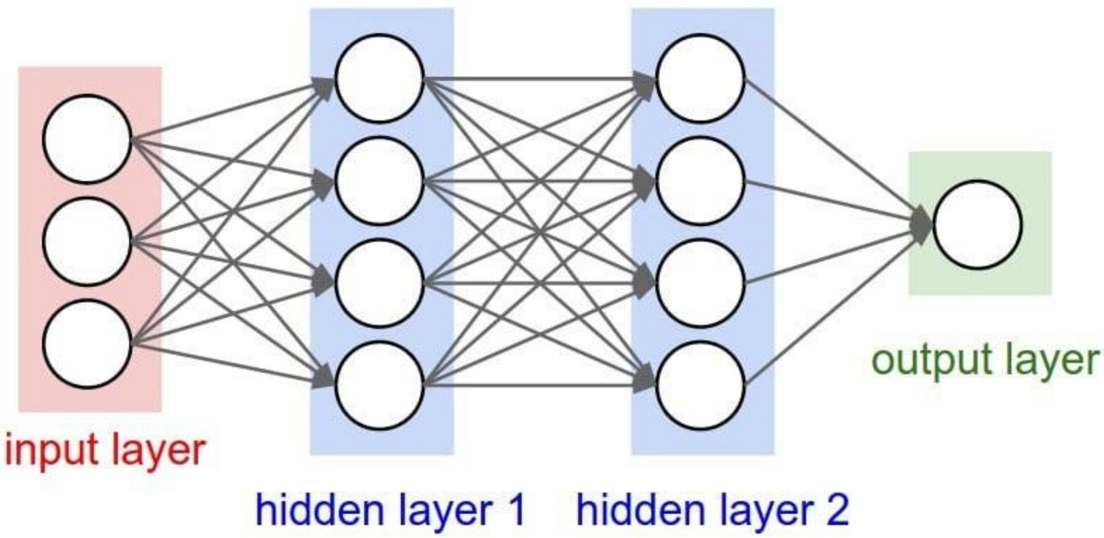


Figure 5: Artificial Neural Network

5. EDA

Exploratory Data Analysis (EDA) is an approach to analyzing datasets to summarize their main characteristics, often with visual methods. EDA is used for seeing what the data can tell us before the modeling task. EDA is the process of investigating the dataset to discover patterns, and anomalies (outliers), and form hypotheses based on our understanding of the dataset. EDA involves generating summary statistics for numerical data in the dataset and creating various graphical representations to understand the data better. Our dataset contains 35887 rows and 3 columns.

	emotion	pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training

Figure 5.1: EDA

	emotion	pixels	Usage
35882	6	50 36 17 22 23 29 33 39 34 37 37 37 39 43 48 5...	PrivateTest
35883	3	178 174 172 173 181 188 191 194 196 199 200 20...	PrivateTest
35884	0	17 17 16 23 28 22 19 17 25 26 20 24 31 19 27 9...	PrivateTest
35885	3	30 28 28 29 31 30 42 68 79 81 77 67 67 71 63 6...	PrivateTest
35886	2	19 13 14 12 13 16 21 33 50 57 71 84 97 108 122...	PrivateTest

Figure 5.2: EDA

Above image shows us head and tail of the data.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35887 entries, 0 to 35886
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   emotion  35887 non-null  int64 
 1   pixels   35887 non-null  object
 2   Usage    35887 non-null  object
dtypes: int64(1), object(2)
memory usage: 841.2+ KB
```

Figure 5.3: data info.

Above image talks about data info. In emotion column all emotions categorize in 7 codes i.e., 0: 'anger', 1: 'disgust', 2: 'fear', 3: 'happy', 4: 'sad',5: 'surprise', 6: 'neutral' . in pixels column image is present in array.

Below shown images present in our dataset.

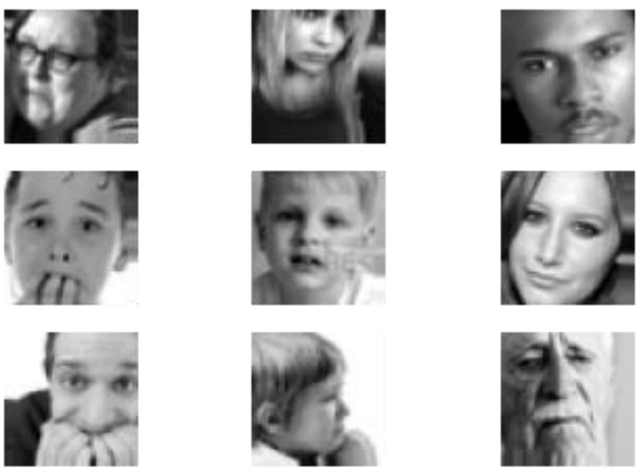


Figure 5.4: image data

6. Data Preparation:

We will divide our data into train, test split. After splitting we reshape our dataset into (48,48,1). Whenever we preprocess with image, we generate images from given dataset to get train on more different types of data. To generate data, we will use image data generator. In this function we have different types of parameters through we can generate different types of data. Parameter like horizontal_flip=True means changing angle of a photo, rescale, rotation range, zoom_range, width_shift, height_shift.

In machine learning, early stopping is a form of regularization used to avoid overfitting when training a learner with an iterative method, such as gradient descent. Such methods update the learner so as to make it better fit the training data with each iteration.

Early stopping is a method that allows you to specify an arbitrary large number of training epochs and stop training once the model performance stops improving on a holdout validation dataset.

When training deep learning models, the checkpoint is the weights of the model. These weights can be used to make predictions as is, or used as the basis for ongoing training. The API allows you to specify which metric to monitor, such as loss or accuracy on the training or validation dataset.

Specifically, the learning rate is a configurable hyperparameter used in the training of neural networks that has a small positive value, often in the range between 0.0 and 1.0. The learning rate controls how quickly the model is adapted to the problem.

```
# having early stopping and model check point

from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau

# early stopping
es = EarlyStopping(monitor = 'val_accuracy', min_delta=0.01, patience=15, verbose = 1, mode ='auto')

# model checkpoint
mc = ModelCheckpoint(filepath="best_model.hdf5", monitor = "val_accuracy", verbose = 1, save_best_only=True, mode = "auto")

# Reduce learning rate when a metric has stopped improving
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=3, verbose=1, mode='auto', min_delta=0.0001, min_lr=0.001)

# putting callback in a list
call_back = [es,mc,reduce_lr]
```

Figure 6: keras callbacks

7. Algorithm

Step 1: - Collection of a data set of images. 35887 pre-cropped, 48-by-48-pixel grayscale images of faces each labeled with one of the 7 emotion classes: anger, disgust, fear, happiness, sadness, surprise, and neutral.

Step 2: - Pre-processing of images.

Step 3: - Detection of a face from each image.

Step 4: - The cropped face is converted into grayscale images.

Step 5: - The pipeline ensures every image can be fed into the input layer as a (48, 48, 1) NumPy array.

Step 6: - Convolution generates feature maps.

Step 7: - Batch normalization is a layer that allows every layer of the network to do learning more independently. It is used to normalize the output of the previous layers. Using batch normalization learning becomes efficient also it can be used as regularization to avoid overfitting of the model.

Step 8: - ReLU helps to prevent the exponential growth in the computation required to operate the neural network.

Step 9: - Pooling method called MaxPooling2D that uses (2, 2) windows across the feature map only keeping the maximum pixel value.

<u>Step 10: -</u> Dropout is a technique used to prevent a model from overfitting. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase.

<u>Step 11: -</u> During training, Neural network Forward propagation and Backward propagation performed on the pixel values.

<u>Step 12: -</u> The SoftMax function presents itself as a probability for each emotion class. The model is able to show the detail probability composition of the emotions in the face.

```
def model_(input_shape=(48,48,1)):
  # first input model
    visible = tf.keras.Input(shape=input_shape)

    # Conv Block 1
    conv1 = tfl.Conv2D(64, (3,3), padding='same', input_shape=(48,48,1))(visible)
    bn1 = tfl.BatchNormalization()(conv1)
    rl1 = tfl.ReLU()(bn1)
    mp1 = tfl.MaxPooling2D(pool_size=(2,2))(rl1)
    dp1 = tfl.Dropout(0.25)(mp1)

    # Conv Block 2
    conv2 = tfl.Conv2D(128,(5,5), padding='same')(dp1)
    bn2 = tfl.BatchNormalization()(conv2)
    rl2 = tfl.ReLU()(bn2)
    mp2 = tfl.MaxPooling2D(pool_size=(2,2))(rl2)
    dp2 = tfl.Dropout(0.25)(mp2)

    # Conv Block 3
    conv3 = tfl.Conv2D(512,(3,3), padding='same')(dp2)
    bn3 = tfl.BatchNormalization()(conv3)
    rl3 = tfl.ReLU()(bn3)
    mp3 = tfl.MaxPooling2D(pool_size=(2,2))(rl3)
    dp3 = tfl.Dropout(0.25)(mp3)

    # Conv Block 4
    conv4 = tfl.Conv2D(512,(3,3), padding='same')(dp3)
    bn4 = tfl.BatchNormalization()(conv4)
    rl4 = tfl.ReLU()(bn4)
    mp4 = tfl.MaxPooling2D(pool_size=(2,2))(rl4)
    dp4 = tfl.Dropout(0.25)(mp4)

    #Flatten and output

    flatten = tfl.Flatten(name = 'flatten')(dp4)
    #dn = tfl.Dense(256, activation='relu')(flatten)
    #bn = tfl.BatchNormalization()(dn)
    #dp = tfl.Dropout(0.25)(bn)

    ouput = tfl.Dense(7, activation='softmax')(flatten)

    # create model
    model = tf.keras.Model(inputs =visible, outputs = ouput)
    # summary layers
    print(model.summary())
```

Figure 7.1: CNN model

8. Evaluation

- g Loss function used is categorical cross entropy.
- Loss function simply measures the absolute difference between our prediction and the actual value.
- Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. So, predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high loss value.
- Adam optimizer combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems.

```
model = model_()
for layer in model.layers:
    layer.trainable = True
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
```

Figure 8.1: compile model

```
============================
Total params: 3,192,839
Trainable params: 3,190,407
Non-trainable params: 2,432
____________________________
```

Figure 8.2: parameters

9. Model Training & Validation Accuracy

- To get model output finally we fit the model.
- Performance As it turns out, the final CNN had a validation and training accuracy as 65% and 73% respectively. This actually makes a lot of sense. Because our expressions usually consist a combination of emotions, and only using one label to represent an expression can be hard.

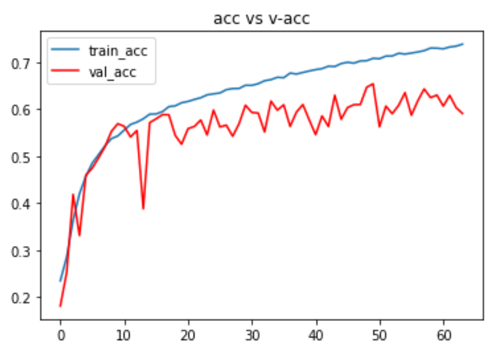
```
batch_size = 128
epochs = 100

history = model.fit(train_flow,
                    steps_per_epoch=len(X_train) / batch_size,
                    epochs=epochs,
                    validation_data = test_flow,
                    validation_steps = len(X_test) / batch_size,
                    callbacks = [es,mc,reduce_lr],
                    verbose = 1)
```

Figure 9: fitting model

10. Visualization

1. Train accuracy vs validation accuracy 2. Train loss vs validation loss



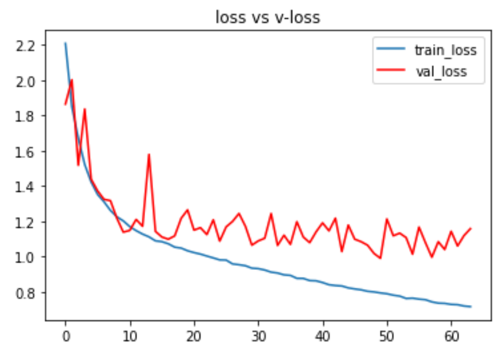


Figure 10.1: training & validation accracy Figure 10.2: training & validation loss

11. Conclusion

1) when the model predicts incorrectly, the correct label is often the second most likely emotion.

2) The facial expression recognition system presented in this research work contributes a resilient face recognition model based on the mapping of behavioral characteristics with the physiological biometric characteristics. The physiological characteristics of the human face with relevance to various expressions such as happiness, sadness, fear, anger, surprise and disgust are associated with geometrical structures which restored as base matching template for the recognition system.

3) The behavioral aspect of this system relates the attitude behind different expressions as property base.

4) CNN model training accuracy 73% & validation accuracy 65%.

5) Training loss 0.7175 , validation loss is 0.9910 & learning rate is 0.001 .

❖ **FUTURE SCOPE**

It is important to note that there is no specific formula to build a neural network that would guarantee to work well. Different problems would require different network architecture and a lot of trail and errors to produce desirable validation accuracy. This is the reason why neural nets are often perceived as "black box algorithms.".

In this project we got an accuracy of almost 70% which is not bad at all comparing all the previous models. But we need to improve in specific areas like-

- ❖ number and configuration of convolutional layers
- ❖ number and configuration of dense layers

- dropout percentage in dense layers

But due to lack of highly configured system we could not go deeper into dense neural network as the system gets very slow and we will try to improve in these areas in future. We would also like to train more databases into the system to make the model more and more accurate but again resources become a hindrance in the path and we also need to improve in several areas in future to resolve the errors and improve the accuracy.

12. Output

Getting following output from my trained model:

Figure 12.1: surprise

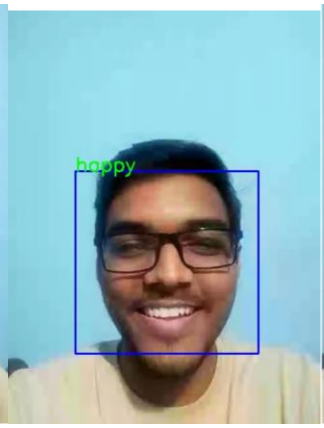


Figure 12.2: happy

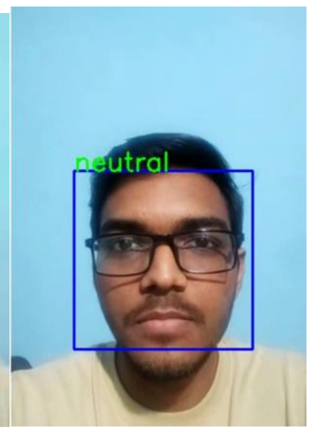


Figure 12.3: neutral

Figure 12.4: sad

Figure 12.5: anger

Figure 12.6: fear

13. References

- ❖ Deploy Machine Learning Models Using StreamLit Library- Data Science

  [https://www.youtube.com/watch?v=5XnHlluw-Eo&t=601s]

- ❖ Convolutional Neural Networks (CNN) With TensorFlow by Sourav from Edureka
- ❖ [https://www.youtube.com/watch?v=umGJ30-15_A]
- ❖ Emotion Detection using OpenCV & Python | Real time Emotion Detection | Deep Learning | Edureka [https://www.youtube.com/watch?v=G1Uhs6NVi-M]
- ❖ Emotion Detection using CNN | Emotion Detection Deep Learning project |Machine Learning | Data Magic [https://www.youtube.com/watch?v=UHdrxHPRBng]
- ❖ Emotion Detection using Convolutional Neural Networks and OpenCV | Keras | Realtime [https://www.youtube.com/watch?v=Bb4Wvl57LIk]
- ❖ Report: IMAGE PROCESSING FACIAL EXPRESSION RECOGNITION [https://www.rcciit.org/students_projects/projects/it/2018/GR8.pdf]
- ❖ Face emotion recognition deep learning approach https://www.slideshare.net/AshwinRachha/facial-emotion-recognition-a-deep-learning-approach
- ❖ https://1library.net/document/q01lrpxz-human-emotion-recognition-facial-expression-approach-features-classifier.html

14. The Library & Packages:

- OpenCV:

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. The library is used extensively in companies, research groups and by governmental bodies. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDAand OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

OpenCV's application areas include:

♣ 2D and 3D feature toolkits

♣ Egomotion estimation

♣ Facial recognition system

♣ Gesture recognition

♣ Human–computer interaction (HCI)

♣ Mobile robotics

♣ Motion understanding

♣ Object identification

♣ Segmentation and recognition

♣ Stereopsis stereo vision: depth perception from 2 cameras

♣ Structure from motion (SFM)

♣ Motion tracking

♣ Augmented reality

To support some of the above areas, OpenCV includes a statistical machine learning library that contains:

♣ Boosting

♣ Decision tree learning

♣ Gradient boosting trees

♣ Expectation-maximization algorithm

♣ k-nearest neighbor algorithm

♣ Naive Bayes classifier

♣ Artificial neural networks

♣ Random forest

♣ Random forest

♣ Support vector machine (SVM)

♣ Deep neural networks (DNN)

- NumPy:

NumPy is an acronym for "Numeric Python" or "Numerical Python". It is an open-source extension module for Python, which provides fast precompiled functions for mathematical and numerical routines. Furthermore, NumPy enriches the programming language Python with powerful data structures for efficient computation of multi-dimensional arrays and matrices. The implementation is even aiming at huge matrices and arrays. Besides that, the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

♣ A powerful N-dimensional array object

♣ Sophisticated (broadcasting) functions

♣ Tools for integrating C/C++ and Fortran code

♣ Useful linear algebra, Fourier Transform, and random number capabilities.

- NumPy Array:

A NumPy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.

- SciPy:

SciPy (Scientific Python) is often mentioned in the same breath with NumPy. SciPy extends the capabilities of NumPy with further useful functions for minimization, regression, Fourier-transformation and many others. NumPy is based on two earlier Python modules dealing with arrays. One of these is Numeric. Numeric is like NumPy a Python module for high-performance, numeric computing, but it is obsolete nowadays. Another predecessor of NumPy is Numarray, which is a complete rewrite of Numeric but is deprecated as well. NumPy is a merger of those two, i.e., it is built on the code of Numeric and the features of Numarray.

- Keras:

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Keras contains numerous implementations of commonly used neural network building blocks suchas layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel.

Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine. It also allows use of distributed training of deep learning models on clusters of Graphics Processing Units (GPU).

- TensorFlow:

TensorFlow is 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.

- SoftMax Function:

♣ Softmax function calculates the probabilities distribution of the event over „n" different events. In general way of saying, this function will calculate the probabilities of each target class over all possible target classes. Later the calculated probabilities will be helpful for determining the target class for the given inputs.

♣ The main advantage of using Softmax is the output probabilities range. The range will 0 to 1, and the sum of all the probabilities will be equal to one. If the softmax function used for multi-classification model it returns the probabilities of each class and the target class will have the high probability.

♣ The formula computes the exponential (e-power) of the given input value and the sum of exponential values of all the values in the inputs. Then the ratio of the exponential of the input value and the sum of exponential values is the output of the softmax function.

✓ Properties Of Softmax Function:
✓ ♣ The calculated probabilities will be in the range of 0 to 1.
✓ ♣ The sum of all probabilities is equal to 0.