Face Emotion Detection

Sarthak Kapaliya

### **Abstract**

**Emotion is something that makes humans different from machines. The one thing a human can’t stow away, despite attempting to their fullest is emotions. Emotion fundamentally triggers the sensory system resulted due to specific action. Emotions are best seen on the face of a person. Facial Expression is one of the most natural communication modes among human beings. There are various facial emotions which are neutral, happy, sad, surprise, fear, disgust, and anger. So Image-based analysis of facial emotion can help in many ways to improve human-computer interaction. Human-computer interaction has outstanding demand as many tasks are being automated and computer operated. Most of the time Face emotion detection is used for surveillance, crowd analytics, and security. The main application of technology can be used for student Counselling sessions. According to Our World in Data, 800000 people die from suicide every year. The suicide rate around the world in 2017 was roughly 1.4%. This percentage rose to 5% in some nations. This can be decreased if we use emotion detection while counseling sessions or some measure to know how an individual is feeling. This technology can also be used with virtual assistants like Siri and Alexa as they can respond to particular emotions. Another Example where this tech is used is in Crime and Law enforcement during a Criminal interrogation. The media or Entertainment sector can have a successful result from this technology as it can them feedback about their content or movies. We have used Public dataset for emotion detection. FER2013 dataset and CK+48 data were used for model training purpose. We used the deep neural networks model VGG16 and achieved an accuracy of 89%. The model has a precision score of 0.81.**

## Data preprocessing system

Emotions are the most important part of a human being. Humans can recognize and differentiate between faces. This is believed to be achieved by computers nowadays. Recognizing facial expressions that communicate fundamental emotions like fear, happiness, disgust, etc. is known as facial emotion recognition. A highly accurate emotion identification model has been developed thanks to the development of computer vision techniques.

### Data Collection

We have collected different types of facial emotion data from many online dataset repositories. We have used the Facial Emotion Recognition 2013 dataset for training purposes. It contains approximately 30000 facial images in RGB form of differential expression with a size restricted to 48x48 pixels. It contains mainly 7 types of emotion labels. They are angry(0), Disgust(1), Fear(2), Happy(3), Sad(4), Surprise(5), Neutral(6).

For model testing, we used different data. CK+48 is a small dataset. It contains 7 classes fear, sadness, anger, disgust, happiness, contempt, and surprise. Images are 48 x48 in size with a grey-scaled color palette. There is a good variation and feature distribution that can be used in testing to obtain a good results. It has a frontal view with a clear images of faces.

### Data Augmentation

Data augmentation is a group of methods for creating additional data points from previously collected data in order to artificially increase the amount of data. This includes making minor adjustments to the data or creating new data points using deep learning models. By creating new and varied examples, this has a wide range of applications for enhancing the performance and results of machine learning models.

We are using the Tensorflow framework for the deep learning model. Tensorflow provides an image preprocessing technique for data augmentation by generating batches of tensor image.

We have done the following data augmentation operation:

1. Rotation: in this, we just rotate the image by a certain specified degree. If the rotation degree is set to 40 then the new image will be 40 degrees and rotate to the original one.
2. Shearing: It is also used to transform the orientation of the image. Additionally, it implies that the image will be warped along a particular axis, typically to alter or modify the perception angles.
3. Zooming: It allow us to either zoom in or zoom out. Specified zoom-in range allow us to get different image which can be helpful for training the ML model.
4. Flipping: It allow us to flip the orientation of the image. We can use horizontal or vertical flip. This operation can be misleading for model. If the image is flipped, along wrong axis then it can make no sense during the training of the deep learning model. So in face detection we don’t need vertical flip.
5. Rescale: We rescale the image pixel in the range 0 to 255.
6. Shifting: We shift the image by a certain length making it different form the real image. It has height and width shift for example.

Here are the demonstration how Data Augmentation change the Face image. It creates different types of images similar to original ones. This is applied to whole dataset to increase the dataset:

### C:\Users\LENOVO\Downloads\download.pngModel Training

A neural network with three or more layers is essentially what machine learning, which includes deep learning, is. Although these neural networks try to emulate the way the human brain works, they are unable to match it, allowing the computer to "learn" from enormous amounts of data. Even if a neural network with only one layer can still provide approximate predictions, more hidden layers can help to tune and improve for accuracy.

Here we are going to use VGG16 as a deep learning model.

VGG16 is a convolutional neural network of 16 layer deep. It is a pre-trained model that has been trained on ImageNet database. The pre-trained model can categorise 1000 different types of objects. The network has therefore acquired rich feature representations for a variety of images. The network can accept images up to 224x224 in size. This model has achieved 92% in the ImageNet Challenge for 14 million images belonging to 1000 classes.

It has fixed input size of 224x224 and have RGB channels which result to (224, 224, 3) tensor. Here it calculate probabilities of different classes. After every prediction we get probabilities associated to different classes based on similarity. The classification vector has to make sure that these probabilities add to 1 and to check it we use Softmax function.

The 16 in VGG16 refer to 16 layers that have weights. In VGgg16 there are thirteen convolutional layers, five max pooling layers and three dense layers i.e. learnable parameters layer. It contain 3x3 filter with stride 1 and same padding and maxpool layer of 2x2 filter of stride 2. The convolutional and maxpool layer are consistently arranged through out the whole structure.

The first Conv-1 layer has 64 filters, Conv-2 has 128 and conv-3 has 256 and Conv-4 and Conv-5 have 512 filters.

We have added extra dense layer to the existing layers of the VGG16 and also applied batch normalization and dropout to it.

We have added three times dense layer with 32 filters having batch normalization and dropout with activation function ReLU.

A good optimizing algorithm can help a deep learning model in training by getting difference in result in minutes, hours and days.

In this case, we'll apply the Adam optimization method. An addition to stochastic gradient descent is the Adam optimizer. These days, deep learning applications for computer vision and natural language processing are commonly regarded as dependable.

Adam is a shortened version of adaptive moment estimation. Using estimated values for the first and second moments of the gradients, it calculates individual adaptive learning rates for various parameters.

### Adam Optimation

It uses both momentum and sealing term for gradient cost function

Algorithm:

Initialize Vdw=0,Sdw=0,Vdb=0,Sdb=0

On iteration t:

Computer dw , db using current mini batch gradient descent

Vdw = β1 Vdw + ( 1- β1)dw, Vdw = β1 Vdb + ( 1- β1)db

(momentum β1 exponentially weighted average)

Sdw = β2 Sdb + ( 1- β2)dw2 , Sdb = β 2 Sdb + ( 1- β2)db

( RMSProp β2 )

Vdwcorr = Vdw/ ( 1- β1t), Vdbcorr = Vdb/ ( 1- β1t)

Sdwcorr = Sdw/ ( 1- β2t) , Sdbcorr = Sdb/ ( 1- β2t)

W = W – α Vdwcorr / √ (Sdwcorr + € ) ,b = b – α Vdbcorr / √ (Sdbcorr + €)

Hyperparameter choices:

α : Learning rate. It needs to be tune.

β1 : beta 1 it is moving average weight of dw. Default value 0.9

β2 : beta 2 moving average weight of dw2 and db2. Default value 0.999

€ : 10-8

Adam additionally uses the average of the second moments of the gradients, as opposed to adjusting the parameter learning rates based on the average first moment as in RMSProp .

Basically it combines the idea of moment optimization with RMSProp and exponential decay

#### How Adam works

**Moment’s method:**

Generally, the main aim is to accelerate the gradient descent algorithm with exponentially weighted average of the gradient. To converge faster toward minima we use averages.

**RMSP method:**

An improved version of AdaGrad is Root mean square prop. Here we take exponential moving average.

Adam Optimizer inherits the strength or the positive attributes of the above two methods and build upon them to give optimized results.

After taking account final equation are

#### Cross Entropy Loss Function

The loss function measures how far the algorithm's current output deviates from the desired output. The goal of this information theory-derived function is to compare two averages of the distribution's bit count. The cross-entropy is used as the Log Loss function to compare two probability distribution functions (not the same, but they measure the same thing).

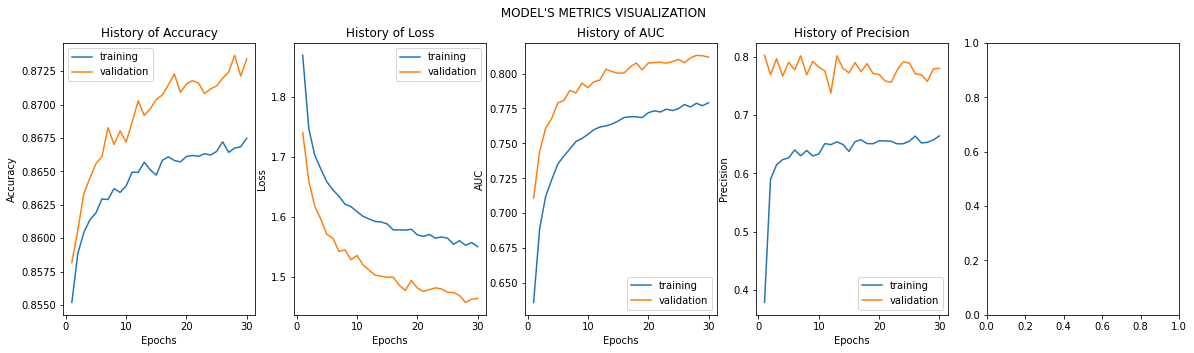
We've employed for binary and multiclass problems, categorical cross-entropy is utilized; the label must be encoded as a categorical, one-hot encoding representation (for three classes: [0, 1, 0], [1, 0, 0]).

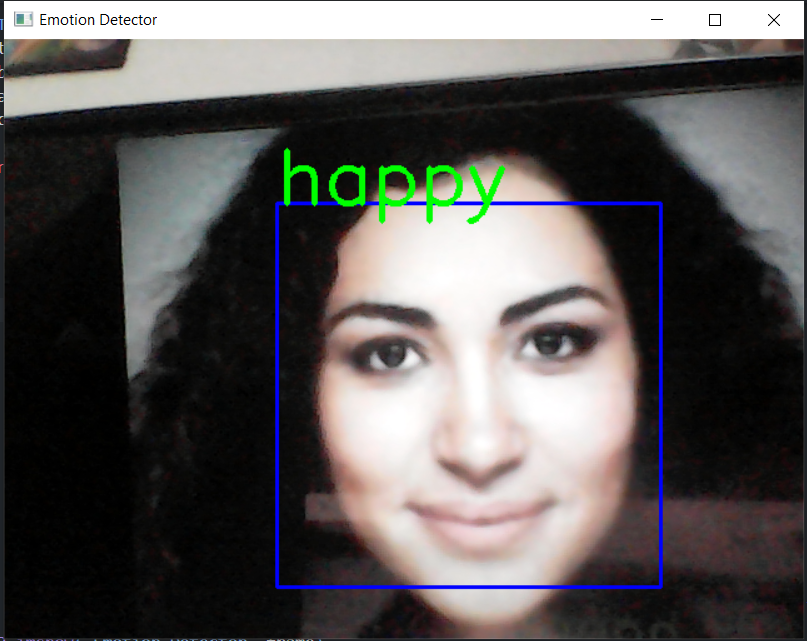
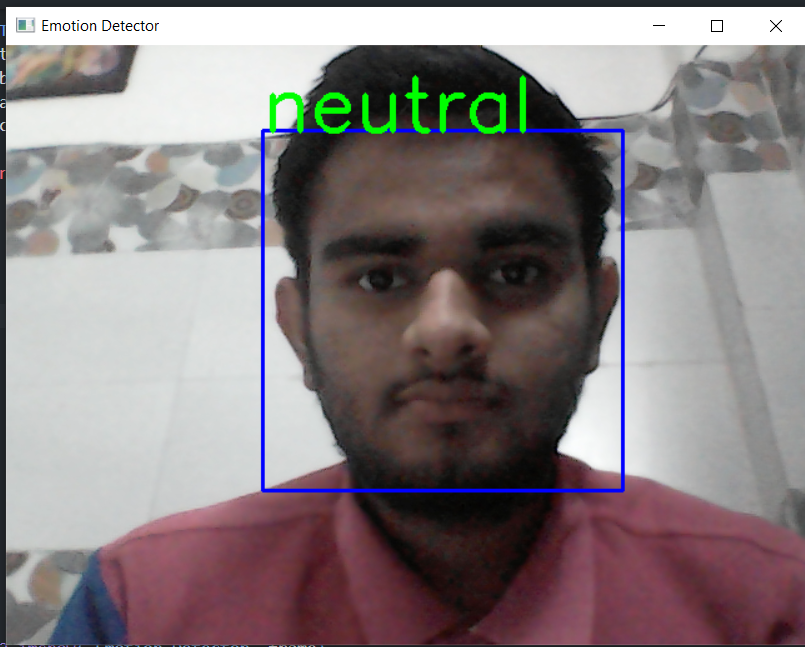
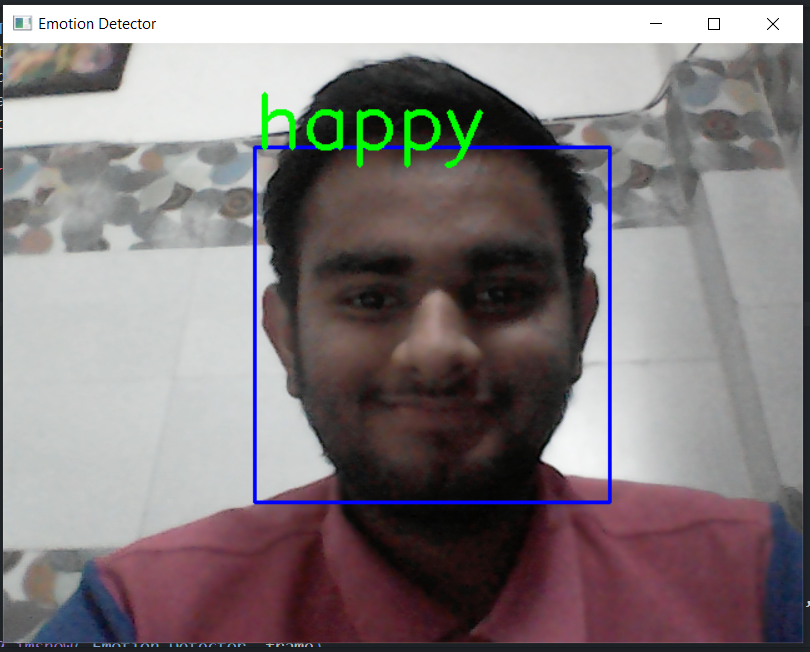
### Model Evaluation

Model evaluation is the most important step and it help us to evaluate and improve our model. The main criteria we used for evaluation was Validation and testing techniques. The model was train on 28000 images of 7 different classes/emotion. The model was validated with 8000 images while training to improve. For training and validation we used FER 2013 dataset only. For Testing we used CK+48 dataset images that contain different image from FER2013 data. CK+48 data contain 981 images.

We used different evaluation metrics. We used 5 classification metrics: Accuracy, Precision, Recall, AUC and F1 Score.

## Results:

In this paper we have proposed an Emotion detection model. Deep Neural networks are used for the precise prediction of emotion from the face images. Feature are extracted using deep learning methods. The effectiveness of the Deep learning Pre trained model is evaluated by classification metrics like Precision Recall F1 Score and Accuracy. We have used two different type of Dataset available publicly. The FER2013 dataset contains more than 30000 images from which we have used 28000 for training of the VGG16 Pre Trained Model. The Rest of the Images were used to validate model while training which uses transfer learning methods. For the testing of the model we used CK+48 dataset. The model is performing well for detecting all & emotion provided in the dataset.



**The proposed model has an accuracy of 89% while having a precision of 81 percent for classification. We have achieved an F! Score of 0.42 and AUC of 0.734.**

Additionally, we are working on a research trend where the proposed approach is being implemented on hardware. Additionally, this problem can be resolved using additional machine learning techniques including dictionary learning and semi-supervised learning.