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. About 1.4% of global deaths in 2017 were from suicide. In some countries, this share increased to 5%. 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. Facial Emotion Recognition simply means identifying expressions that convey basic emotions such as fear, happiness, disgust, etc. with the advancement of computer vision techniques, high accurate emotion recognition model has been achieved.

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 0- angry, 1-Disgust, 2-Fear, 3-Happy, 4-Sad, 5-Surprise, 6-Neutral.

For the Testing of the model, 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 set of techniques to artificially increase the amount of data by generating new points from existing data. This includes making small changes to data or using deep learning models to generate new data points. This is widely useful to improve the performance and outcomes of machine learning models by forming new and different examples.

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. It also mean that the image will be distorted along an axis, mostly to create or rectify 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.

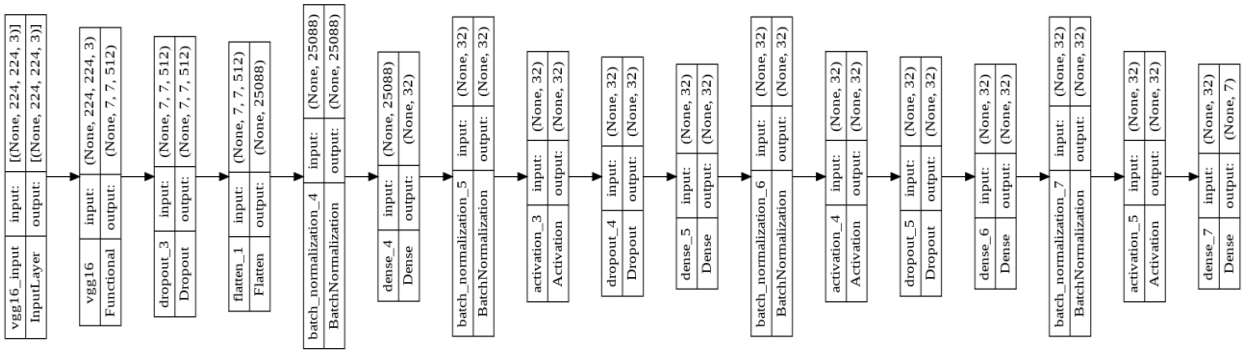
Model Training

A neural network with three or more layers is essentially what machine learning, which includes deep learning, is. These neural networks make an effort to mimic how the human brain functions, however they fall far short of being able to match it, enabling it to "learn" from vast volumes of data. Additional hidden layers can help to tune and refine for accuracy even if a neural network with only one layer can still make approximation predictions.

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 classify 1000 object categories. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224X224. 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.

Here we are going to use Adam optimizing algorithm. Adam optimizer is an extension to stochastic gradient descent. This method is widely reliable this days in deep learning application for computer vision and natural language processing.

The name Adam is derived from adaptive moment estimation. It computes individual adaptive learning rates for different parameters from estimated of first and second moments of the gradients.

Instead of adapting the parameter learning rates based on the average first moment as in RMSProp, Adam also makes use of the average of the second moments of the gradients.

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.

$$w_{t+1} = w_t - \alpha t.m_t$$

Where

$$m_t = \beta.m_{t-1} + (1+ \beta) [\partial L/ \partial w_t]$$

mt = aggregate of gradients at time t [current] (initially, mt = 0)

mt-1 = aggregate of gradients at time t-1 [previous]

Wt = weights at time t

Wt+1 = weights at time t+1

αt = learning rate at time t

∂L = derivative of Loss Function

∂Wt = derivative of weights at time t

β = Moving average parameter (const, 0.9)

RMSP method:

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

$$W_{t+1} = w_t - \alpha t / (v_t + \epsilon)^{1/2} * [\partial L / \partial w_t]$$

$$Where\ v_t = \beta . v_{t-1} + (1- \beta) * [\partial L / \partial w_t]^2$$

Wt = weights at time t

Wt+1 = weights at time t+1

αt = learning rate at time t

∂L = derivative of Loss Function

∂Wt = derivative of weights at time t

Vt = sum of square of past gradients. [i.e sum(∂L/∂Wt-1)] (initially, Vt = 0)

β = Moving average parameter (const, 0.9)

ϵ = A small positive constant (10-8)

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

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \left[\frac{\delta L}{\delta w_t}\right] v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left[\frac{\delta L}{\delta w_t}\right]^2$$

Parameters Used :

1. ϵ = a small +ve constant to avoid 'division by 0' error when (vt -> 0). (10-8)

2. β1 & β2 = decay rates of average of gradients in the above two methods. (β1 = 0.9 & β2 = 0.999)

3. α — Step size parameter/learning rate (0.001)

Cross Entropy Loss Function

The loss function is the function that determines how far the algorithm's current output is from what is desired. The purpose of this function, which derives from information theory, is to compare two averages of the distribution's number of bits. The difference between two probability distribution functions is calculated using the cross-entropy as the Log Loss function (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 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.

Furthermore, hardware implementation of the proposed approach is a research trend, which we are working on. Moreover, further machine learning techniques such as dictionary learning and semi-supervised learning can be performed to solve this issue.