***FACE EMOTION AND RECOGNITION***

***RAHUL SINGH WALDIA***

***Data science trainees,***

***AlmaBetter, Bangalore***

***Abstract***

*A Facial expression is the visible manifestation of the affective state, cognitive activity,*

*intention, personality and psychopathology of a person and plays a communicative role in*

*interpersonal relations. Automatic recognition of facial expressions can be an important component of natural human-machine interfaces; it may also be used in behavioral science and in clinical practice. An automatic Facial Expression Recognition system needs to perform detection and location of faces in a cluttered scene, facial feature extraction, and facial expression classification. Facial expression recognition system is implemented using Convolution Neural Network (CNN). CNN model of the project is based on LeNet Architecture. Kaggle facial expression dataset with seven facial expression labels as happy, sad, surprise, fear, anger, disgust, and neutral is used in this project. The system achieved 56.77 % accuracy and 0.57 precision on testing dataset.*

***Keywords****: Facial Expression Recognition, Convolutional Neural Network, Deep Learning,Theano*

***1. Overview***

***1.1 Background***

*A Facial expression is the visible manifestation of the affective state, cognitive activity,*

*intention, personality and psychopathology of a person and plays a communicative role in*

*interpersonal relations. 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. Automatic recognition of facial expressions can be an important component of natural human-machine interfaces; it may also be used in behavioral science and in clinical practice. It have been studied for a long period of time*

*and obtaining the progress recent decades. Though much progress has been made, recognizing facial expression with a high accuracy remains to be difficult due to the complexity and varieties of facial expressions .On a day to day basics humans commonly recognize emotions by characteristic features,displayed as a part of a facial expression. For instance happiness is undeniably associated with a smile or an upward movement of the corners of the lips. Similarly other emotions are characterized by other deformations typical to a particular expression. Research into automatic recognition of facial expressions addresses the problems surrounding the representation and categorization of static or dynamic characteristics of these deformations of face pigmentation .In machine learning, a convolutional neural network (CNN, or ConvNet) is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. Individual cortical neurons respond to*

*stimuli in a restricted region of space known as the receptive field. The receptive fields of*

*different neurons partially overlap such that they tile the visual field. The response of an*

*individual neuron to stimuli within its receptive field can be approximated mathematically by*

*a convolution operation. Convolutional networks were inspired by biological processes*

*and are variations of multilayer perceptron designed to use minimal amounts of*

*preprocessing.They have wide applications in image and video recognition, recommender systems and natural language processing. The convolutional neural network is also known as shift invariant or space invariant artificial neural network (SIANN), which is named based on its shared weights architecture and translation invariance characteristics.*

*LeNet is one of the very first convolutional neural networks which helped propel the field of*

*Deep Learning. This pioneering work by Yann LeCun was named LeNet5 was used mainly for character recognition tasks such as reading zip codes, digits, etc. The basic architecture of LeNet*

***1. Convolution:***

*The primary purpose of Convolution in case of a CNN is to extract features from the input*

*image. Convolution preserves the spatial relationship between pixels by learning image*

*features using small squares of input data. The convolution layer’s parameters consist of a set of learnable filters. Every filter is small spatially (along width and height), but extends through the full depth of the input volume. For example, a typical filter on a first layer of a CNN might have size 3x5x5 (i.e. images have depth 3 i.e. the color channels, 5 pixels width and height). During the forward pass, each filter is convolved across the width and height of the input volume and compute dot products between the entries of the filter and the input at any position.As the filter convolve over the width and height of the input volume it produces a 2-dimensional activation map that gives the responses of that filter at every spatial position. Intuitively, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some color on the first layer, or eventually entire honeycomb or wheel-like patterns on higher layers of the network. Now, there will be an entire set of filters in each convolution layer (e.g. 20 filters), and each of them will produce a separate 2-dimensional activation map. A filter convolves with the input image to produce a feature map. The convolution of anotherfilter over the same image gives a different feature map. Convolution operation captures the local dependencies in the original image. A CNN learns the values of these filters on its own during the training process (although parameters such as number of filters, filter size, architecture of the network etc. still needed to specify before the training process). The more number of filters, the more image features get extracted and the better network becomes at recognizing patterns in unseen images. The size of the Feature Map (Convolved Feature) is controlled by three parameters • Depth: Depth corresponds to the number of filters we use for the convolution operation.• Stride: Stride is the size of the filter, if the size of the filter is 5x5 then stride is 5.*

*• Zero-padding: Sometimes, it is convenient to pad the input matrix with zeros around*

*the border, so that filter can be applied to bordering elements of input image matrix.*

*Using zero padding size of the feature map can be controlled.*

***2. Rectified Linear Unit:***

*An additional operation called ReLU has been used after every Convolution operation. A*

*Rectified Linear Unit (ReLU) is a cell of a neural network which uses the following activation*

*function to calculate its output given x:*

*R(x) = Max(0,x) (2.2)*

*Using these cells is more efficient than sigmoid and still forwards more information compared to binary units. When initializing the weights uniformly, half of the weights are negative. This helps creating a sparse feature representation. Another positive aspect is the relatively cheap computation. No exponential function has to be calculated. This function also prevents the vanishing gradient error, since the gradients are linear functions or zero but in no case non-linear functions.*

***3. Pooling (sub-sampling)***

*Spatial Pooling (also called subsampling or downsampling) reduces the dimensionality of each feature map but retains the most important information. Spatial Pooling can be of different types: Max, Average, Sum etc. In case of Max Pooling, a spatial neighborhood (for example, a2×2 window) is defined and the largest element is taken from the rectified feature map within that window. In case of average pooling the average or sum of all elements in that window is taken. In practice, Max Pooling has been shown to work better.*

*Max Pooling reduces the input by applying the maximum function over the input xi. Let m be*

*the size of the filter*

*Figure 1.2 :* ***Max Pooling***

*The function of Pooling is to progressively reduce the spatial size of the input representation.*

*In particular, pooling .Makes the input representations (feature dimension) smaller and more manageable • Reduces the number of parameters and computations in the network, therefore, controlling over-fitting .Makes the network invariant to small transformations, distortions and translations in the input image (a small distortion in input will not change the output of Pooling. • Helps us arrive at an almost scale invariant representation. This is very powerful since objects can be detected in an image no matter where they are located.*

***4. Classification (Multilayer Perceptron):***

*The Fully Connected layer is a traditional Multi-Layer Perceptron that uses a softmax*

*activation function in the output layer. The term “Fully Connected” implies that every neuron in the previous layer is connected to every neuron on the next layer. The output from the convolutional and pooling layers represent high-level features of the input image. The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset.Softmax is used for activation function. It treats the outputs as scores for each class. In the Softmax, the function mapping stayed unchanged and these scores are interpreted as the un-normalized log probabilities for each class. where j is index for image and K is number of total facial expression class.*

*Apart from classification, adding a fully-connected layer is also a (usually) cheap way of*

*learning non-linear combinations of these features. Most of the features from convolutional*

*and pooling layers may be good for the classification task, but combinations of those features*

*might be even better. The sum of output probabilities from the Fully Connected Layer is 1. This is ensured by using the as the activation function in the output layer of the Fully Connected Layer. The Softmax function takes a vector of arbitrary real-valued scores and squashes it to a vector of values between zero and one that sum to one.*

***1.2 Problem definition***

*Human emotions and intentions are expressed through facial expressions and deriving an efficient and effective feature is the fundamental component of facial expression system. Facial expressions convey non-verbal cues, which play an important role in interpersonal relations. Automatic recognition of facial expressions can be an important component of natural human-machine interfaces; it may also be used in behavioral science and in clinical practice. An automatic Facial Expression Recognition system needs to solve the following problems: detection and location of faces in a cluttered scene, facial feature extraction, and facial expression classification.*

***1.2 Objective***

*The objective of the project is:*

*1. To implement Convolutional Neural Networks for classification of facial expressions.*

***1.3 Scope of the Project***

*In this project facial expression recognition system is implemented using convolution neural*

*network. Facial images are classified into seven facial expression categories namely Anger,*

*Disgust, Fear, Happy, Sad, Surprise and 'Neutral. Kaggle dataset is used to train and test the*

*classifier.*

*7*

***2. Literature Review***

*Two different approaches are used for facial expression recognition, both of which include two different methodologies, exist [6]. Dividing the face into separate action units or keeping it as a whole for further processing appears to be the first and the primary distinction between the main approaches. In both of these approaches, two different methodologies, namely the ‘Geometric based’ and the ‘Appearance-based’ parameterizations, can be used.*

*Making use of the whole frontal face image and processing it in order to end up with the*

*classifications of 6 universal facial expression prototypes: disgust, fear, joy, surprise, sad ness and anger; outlines the first approach. Here, it is assumed that each of the above mentioned emotions have characteristic expressions on face and that’s why recognition of them is necessary and sufficient. Instead of using the face images as a whole, dividing them into some sub-sections for further processing forms up the main idea of the second approach for facial expression analysis. As expression is more related with subtle changes of some discrete features such as eyes, eyebrows and lip corners; these fine-grained changes are used for analyzing automated recognition.*

*There are two main methods that are used in both of the above explained approaches. Geometric Based Parameterization is an old way which consists of tracking and processing the motionsof some spots on image sequences, firstly presented by Suwa et al to recognize facial expressions . Cohn and Kanade later on tried geometrical modeling and tracking of facial features by claiming that each AU is presented with a specific set of facial muscles .*

*The disadvantages of this method are the contours of these features and components to be*

*adjusted manually in this frame, the problems of robustness and difficulties come out in cases*

*of pose and illumination changes while the tracking is applied on images, as actions & expressions tend to change both in morphological and in dynamical senses, it becomes hard to estimate general parameters for movement and displacement. Therefore, ending up with robust decisions for facial actions under these varying conditions becomes to be difficult. Rather than tracking spatial points and using positioning and movement parameters that vary within time, color (pixel) information of related regions of face are processed in Appearance Based Parameterizations; in order to obtain the parameters that are going to form the feature vectors. Different features such as Gabor, Haar wavelet coefficients, together with feature extraction and selection methods such as PCA, LDA, and Adaboost are used within this framework.For classification problem, algorithms like Machine learning, Neural Network, Support Vector Machine, Deep learning, Naive Bayes are used.*

*have built a Facial expression recognition system upon recent research to classify images of human faces into discrete emotion categories using convolutional neural networks [9]. Alizadeh, Shima, and Azar Fazel have developed Facial Expression Recognition*

*system using Convolutional Neural Networks based on Torch model [10].*

***3. Methodology***

*The facial expression recognition system is implemented using convolutional neural network.*

*During training, the system received a training data comprising grayscale images of faces with their respective expression label and learns a set of weights for the network. The training step took as input an image with a face. Thereafter, an intensity normalization is applied to the image. The normalized images are used to train the Convolutional Network. To ensure that the training performance is not affected by the order of presentation of the examples, validation dataset is used to choose the final best set of weights out of a set of trainings performed with samples presented in different orders. The output of the training step is a set of weights that achieve the best result with the training data. During test, the system received a grayscale image of a face from test dataset, and output the predicted expression by using the final network weights learned during training. Its output is a single number that represents one of the seven basic expressions.*

*Raw Image*

*Normalization*

*CNN Train*

*CNN*

*Weights*

*Raw Image*

*Normalization*

*CNN*

*Facial*

*Expression*

*CNN*

*Weights*

***3.1 Dataset***

*The dataset from a Kaggle Facial Expression Recognition Challenge (FER2013) is used for the training and testing. It comprises 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. Dataset has training set of 35887 facial images with facial expression labels.. The dataset has class imbalance issue, since some classes have large number of examples while some has few. The dataset is balanced using oversampling, by increasing numbers in minority classes. The balanced dataset contains 40263 images, from which 29263 images are used for training, 6000 images are used for testing, and 5000 images are used for validation.*

***3.2 Architecture of CNN***

*A typical architecture of a convolutional neural network contains an input layer, some convolutional layers, some fully-connected layers, and an output layer. CNN is designed with some modification on LeNet Architecture [10]. It has 6 layers without considering input and output.*

*1. Input Layer:*

*The input layer has pre-determined, fixed dimensions, so the image must be pre-processed*

*before it can be fed into the layer. Normalized gray scale images of size 48 X 48 pixels from*

*Kaggle dataset are used for training, validation and testing. For testing propose laptop webcam images are also used, in which face is detected and cropped using OpenCV Haar Cascade Classifier and normalized.*

*2. Convolution and Pooling (ConvPool) Layers:*

*Convolution and pooling is done based on batch processing. Each batch has N images and CNN filter weights are updated on those batches. Each convolution layer takes image batch input of four dimension N x Color-Channel x width x height. Feature map or filter for convolution are also four dimensional (Number of feature maps in, number of feature maps out, filter width, filter height). In each convolution layer, four dimensional convolution is calculated between image batch and feature maps. After convolution only parameter that change is image width and height.*

*New image width = old image width – filter width + 1*

*New image height = old image height – filter height + 1*

*After each convolution layer downsampling / subsampling is done for dimensionality*

*reduction. This process is called Pooling. Max pooling and Average Pooling are two famous*

*pooling method. In this project max pooling is done after convolution. Pool size of (2x2) is*

*12 taken, which splits the image into grid of blocks each of size 2x2 and takes maximum of 4*

*pixels. After pooling only height and width are affected. Two convolution layer and pooling layer are used in the architecture. At first convolution layer size of input image batch is Nx1x48x48. Here, size of image batch is N, number of colorchannel is 1 and both image height and width are 48 pixel. Convolution with feature map of 1x20x5x5 results image batch is of size Nx20x44x44. After convolution pooling is done with pool size of 2x2, which results image batch of size Nx20x22x22. This is followed by second convolution layer with feature map of 20x20x5x5, which results image batch of size Nx20x18x18. This is followed by pooling layer with pool size 2x2, which results image batch of size Nx20x9x9.*

*3. Fully Connected Layer*

*This layer is inspired by the way neurons transmit signals through the brain. It takes a large*

*number of input features and transform features through layers connected with trainable*

*weights. Two hidden layers of size 500 and 300 unit are used in fully-connected layer. The*

*weights of these layers are trained by forward propagation of training data then backward*

*propagation of its errors. Back propagation starts from evaluating the difference between*

*prediction and true value, and back calculates the weight adjustment needed to every layer*

*before. We can control the training speed and the complexity of the architecture by tuning the*

*hyper-parameters, such as learning rate and network density. Hyper-parameters for this layer*

*include learning rate, momentum, regularization parameter, and decay.*

*The output from the second pooling layer is of size Nx20x9x9 and input of first hidden layer*

*of fully-connected layer is of size Nx500. So, output of pooling layer is flattened to Nx1620*

*size and fed to first hidden layer. Output from first hidden layer is fed to second hidden layer.*

*Second hidden layer is of size Nx300 and its output is fed to output layer of size equal to*

*number of facial expression classes.*

*4. Output Layer*

*Output from the second hidden layer is connected to output layer having seven distinct classes.*

*Using Softmax activation function, output is obtained using the probabilities for each of the*

*seven class. The class with the highest probability is the predicted class.*

*13*

*4. Results and Analysis*

*CNN architecture for facial expression recognition as mentioned above was implemented in*

*Python. Along with Python programming language, Numpy, Theano and CUDA libraries were*

*used.*

*Training image batch size was taken as 30, while filter map is of size 20x5x5 for both*

*convolution layer. Validation set was used to validate the training process. In last batch of every*

*epoch in validation cost, validation error, training cost, training error are calculated. Input*

*parameters for training are image set and corresponding output labels. The training process*

*updated the weights of feature maps and hidden layers based on hyper-parameters such as*

*learning rate, momentum, regularization and decay. In this system batch-wise learning rate was*

*used as 10e-5, momentum as 0.99, regularization as 10e-7 and decay as 0.99999.*

*The testing of the model is carried out using 6000 images. The classifier provided 56.77 %*

*accuracy. The confusion matrix for seven facial expression classes is shown below:*

*Precision, Recall and F1-Score*

*Precision*

*Recall*

*F1-score*

*Anger*

*0.39*

*0.42*

*0.41*

*Disgust*

*0.95*

*0.99*

*0.97*

*Fear*

*0.45*

*0.38*

*0.39*

*Happy*

*0.68*

*0.69*

*0.69*

*Sad*

*0.44*

*0.38*

*0.41*

*Surprise*

*0.69*

*0.65*

*0.67*

*Neutral*

*0.45*

*0.49*

*0.47*

*Average*

*0.57*

*0.57*

*0.57*

*The overall precision and recall are 0.57 and 0.57 respectively. The model performs really well on classifying positive emotions resulting in relatively high precision scores for happy and surprised. Disgust has highest precision and recall as 0.95 and 0.99 as images in this class were oversampled to address class imbalance. Happy has a precision of 0.68 and recall of 0.69 which could be explained by having the most examples (6500) in the training set. Interestingly, 16 surprise has a precision of 0.69 and recall of 0.65 having the least examples in the training set.There must be very strong signals in the surprise expressions.*

*Model performance seems weaker across negative emotions on average. In particularly, the*

*emotion sad has a low precision of only 0.44 and recall 0.38. The model frequently*

*misclassified angry, fear and neutral as sad. In addition, it is most confused when predicting*

*sad and neutral faces because these two emotions are probably the least expressive (excluding crying faces).The overall F1-score is also 0.57 . F1-score is highest for disgust due to oversampling of images. Happy and surprise have higher F1-score as 0.69 and 0.67 respectively. Fear has least F1-score as 0.39 and sad, anger and neutral also have low F1-score. CNN Classifier is then used to classify image taken from webcam in Laptop. Face is detected in webcam frames using Haar cascade classifier from OpenCV. Then detected face is cropped and normalized and fed to CNN Classifier. Some classification results using webcam are listed in Appendix A.*

***5. Summary***

***5.1 Conclusion***

*In this project, a LeNet architecture based six layer convolution neural network is implemented to classify human facial expressions i.e. happy, sad, surprise, fear, anger, disgust, and neutral. The system has been evaluated using Accuracy, Precision, Recall and F1-score. The classifier achieved accuracy of 56.77 % , precision of 0.57, recall 0.57 and F1-score 0.57.*