Live Class Monitoring System (Face Emotion Recognition)

(Jasmin Babariya)
Data scientist,
STAPL, Pune

Abstract:

In this project, we have developed deep Convolutional neural networks (DCNN) or four more models for a facial expression recognition task. The goal is to classify each facial image into one of the seven facial emotion categories considered in this study. We trained CNN models with different depths using gray-scale images from the Kaggle website. We developed our models in Python and exploited Graphics Processing Unit (GPU) computation in order to expedite the training process. We applied cross-validation to determine the optimal hyperparameters and evaluated the performance of the developed models by looking at their training histories. We also present the visualization of different layers of a network to show what features of a face can be learned by DCNN models. Humans interact with each other mainly through speech, but also through body gestures, to emphasize certain parts of their speech and to display emotions. One of the important ways humans display emotions is through facial expressions which are a very important part of communication.

***Keywords: EDA, train-test split Classification,* ReLUs*, Xception, CNN,* Deep Face Model & Resnet50**

1. Problem Statement

Human emotions and intentions are expressed through facial expressions and deriving angles. the efficient and effective feature is the fundamental component of the 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.

2. Introduction

Human beings communicate with each other in the form of speech, gestures, and emotions. As such systems that can recognize the same are in great demand in many fields. With respect to artificial intelligence, a computer will be able to interact with humans much more naturally if they are capable of understanding humans it may also be emotion. It would also help during counseling and other healthcare related fields. In an E-Learning system, the presentation style may be varied depending on the student's state. However, in many cases, static emotion detection is not very useful. It is essential to know the user's feelings over a period of time in a live environment. Thus, the paper proposes a model that is aimed at real-time facial emotion recognition

3. Related Work

In recent years, researchers have made considerable progress in developing automatic expression classifiers. Some expression recognition systems classify the face into a set of prototypical emotions such as happiness, sadness, and anger. Others attempt to recognize the individual muscle movements that the face can produce in order to provide an objective description of the face. The expression typically results from the accumulation of several. Yu and Zhang used a five-layer ensemble CNN to achieve a 0.612

accuracy. They pre-trained their models on the FER-2013 dataset and then fine-tuned the model on the Static Facial Expressions in the Wild 2.0 (SFEW) dataset. Keauhou et al used a CNN-RNN architecture to train a model on individual frames of videos as well as static images. They made use of the Acted Facial Expressions in the Wild (AFEW) 5.0 dataset for the video clips and a combination of the FER-2013 and Toronto Face Database for the images. Instead of using long short-term memory (LSTM) units, they used IRNNs which are composed of rectified linear units (ReLUs). These IRNNs provided a simple mechanism for dealing with the vanishing and exploding gradient problem. proposed a network consisting of two convolution layers each followed by max-pooling and then four Inception layers. They used this network on seven different datasets including the FER-2013 dataset. They also compared the accuracies of their proposed network with an Alex Net network trained on the same datasets. They found that their architecture had better performance on the MMI and FER-2013 datasets with comparable performances on the remaining five datasets. The FER-2013 dataset in particular managed to reach an accuracy of 0.664.

Most other works in the same field attempted to solve the facial emotion recognition problem by the use of a combination of different datasets. In this paper, a single dataset, FER-2013 was chosen over such a combination of different datasets and then experiments were conducted with different models to find the highest accuracy that each model could reach

4. Theoretical Background

A Convolutional neural network is a neural network comprised of convolution layers that does the computational heavy lifting by performing convolution. Convolution is a mathematical operation on two functions to produce a third function. It is to be noted that the image is not represented as pixels, but as numbers representing the pixel value. In terms of what the computer sees, there will simply just be a matrix of numbers. The convolution operation takes place on these numbers.

We utilize both fully-connected layers as well as convolutional layers. In a fully-connected layer, every node is connected to every other neuron. They are the layers used in standard feed-forward neural networks. Unlike the fully connected layers, convolutional layers are not connected to every neuron. Connections are made across localized regions. A sliding" window" is moved across the image. The size of this window is known as the kernel or the filter. They help recognize patterns in the data. For each filter, there are two main properties to consider - padding and stride. Stride represents the step of the convolution operation, that is, the number of pixels the window moves across. Padding is the addition of null pixels to increase the size of an image. Null pixels here refer to pixels with a value of 0. If we have a 5x5 image and a window with a 3x3 filter, a stride of 1 and no padding, the output of the convolutional layer will be a 3x3 image. This condensation of a feature map is known as pooling. In this case," max pooling" is utilized. Here, the maximum value is taken from each sliding window and is placed in the output matrix. Convolution is very effective in image recognition and classification compared to a feed-forward neural network. This is because convolution allows reducing the number of parameters in

a network and taking advantage of spatial locality. Further, convolutional neural networks introduce the concept of pooling to reduce the number of parameters by down sampling. Applications of Convolutional

neural networks include image recognition, self-driving cars, and robotics. CNN is popularly used with videos, 2D images.

5. Experimental setup

This section details the data used for training and testing, how the data was preprocessed, the various models that were used, and an evaluation of each model.

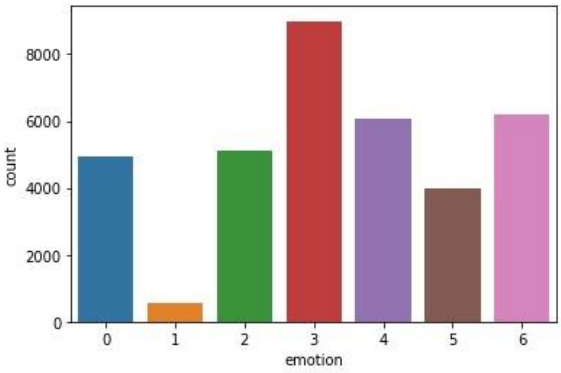
a)-Dataset:

In general, neural networks, especially deep neural networks, tend to perform better when larger amounts of training data set are present. With this in mind, the more popular

Instead, the Facial Expression dataset (FER2013) was chosen. The FER-2013 dataset was introduced in the ICML 2013 Challenges in Representation Learning [2]. It contains 35,887 images with the following basic expressions: angry, disgusted, fearful, happy, sad, surprised, and neutral. Figure 1 shows the distribution of each expression.

Each image is a frontal view of a subject, taken from the wild and annotated to one of the seven expressions. A sample of these expressions is shown in Figure 2. It is to be noted that the number of disgusted expressions (547) is much lower in comparison to the other expressions. There was also an obvious bias towards happy expressions due to the sheer number of sample data present for the expression. Emotion labels we have:

{0:'anger', 1:'disgust', 2:'fear', 3:'happiness', 4: 'sad ness', 5: 'surprise', 6: 'neutral'}

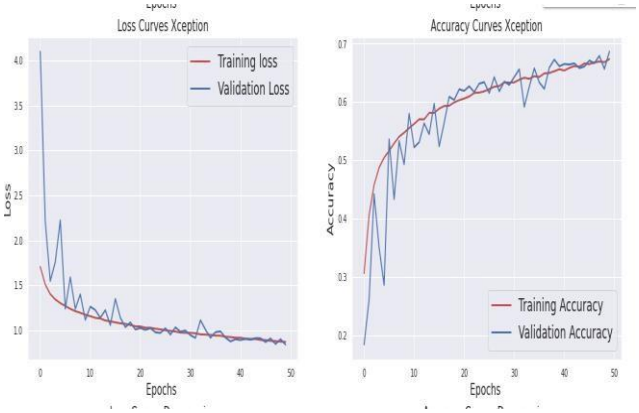


. b)-Preprocessing:

The dataset consisted of a number of images represented as strings of 2304 space-separated numbers which were then converted to a 48*48 matrix. Each number represented a pixel value. The original data of 35,887 images were split into a training set of 28,709 images and a testing set of 7,178 images - an 80:20 split. Generally, when it comes to deep learning, data is the biggest factor. The bigger the training set, the better the output. If there is less training data, there is a lot more variance in the final outputs due to a smaller set to train on. Bearing that in mind, having a testing set of 20% of the total images may be seen as excessive. However, to prevent overfitting it is necessary to have sizable testing set as well, Mollahosseiniet al.

Divided their 275k image dataset into 60% for training, 20% for testing, and 20% for validation. In this case, the validation set was foregone in favor of retraining the entire

model every time the hyperparameters were tuned. While this required more time and computational power, it provided a bigger training set in the end. A one-hot encoding scheme was used for the labels rather than classifying emotions with numbers from 0-6. During the live testing, Haar Cascades [15] were used to identify a face. This identified face was then taken as an image, converted to gray-scale and downscaled to a 48*48 image. Thus, the image was converted to a format identical to that which was used to train the model.



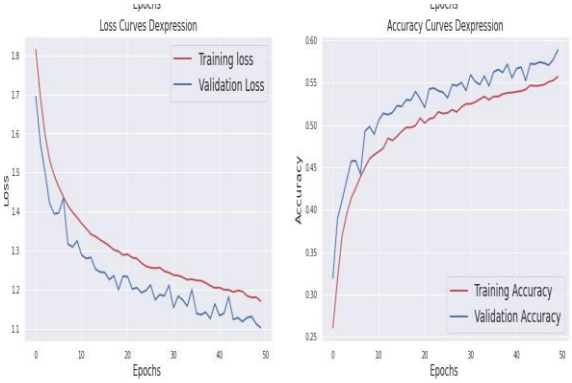
c)- Choosing a Model:

Different models were used in order to select the best foundation to work on the project. The neural networks were implemented using Keras with a Tensor Flow backend running in Python. All the implementations are written in Python 3.6 and are wholly reproducible with freely available software. Since there are 5 different models used, the individual training algorithm for each model is detailed with the model description itself.

1) Deep face model- Deep face is lightweight face recognition and facial attribute analysis (age, gender, emotion, and race) framework for python. We imported an image of Andrew Ng which looks sad our model gives us a "35 years old Asian sad Man" this result. To get better results we decided to train our own model**.**

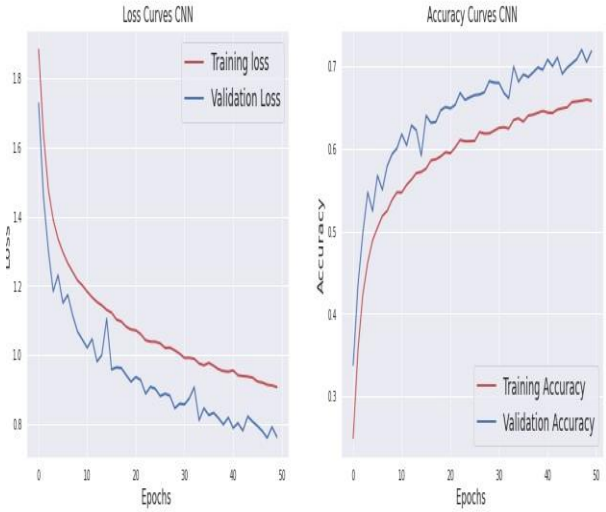
2) Xception model - Xception architecture is a linear stack of depth-wise separable convolution layers with residual connections. We used Adam as our optimizer after training for 50 epochs using Adam and a batch size of 785, we achieved 68 % accuracy on the test set.

3) DeXpression model- We propose a convolutional neural network (CNN) architecture for facial expression recognition. The proposed architecture is independent of any hand-crafted feature extraction and performs better than the earlier proposed convolutional neural network-based approaches. We visualize the automatically extracted features which have been learned by the network in order to provide a better understanding. we achieved
63 % accuracy on the test set

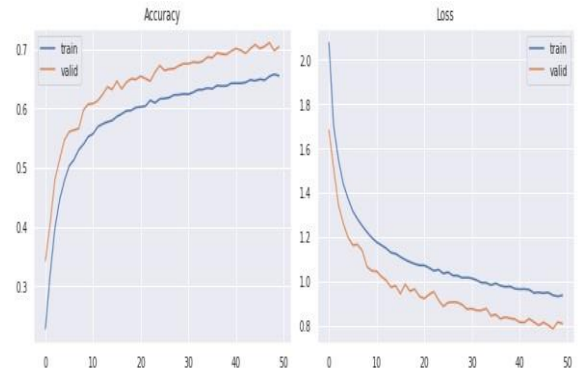


4) CNN model - A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that can take an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. We used RMSprop as our optimizer after training

for 50 epochs using RMSprop with a learning rate of 0.001 and a batch size of 785, we achieved 65 % accuracy on the test set.

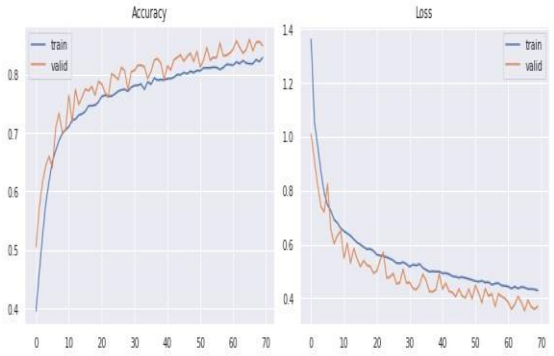


Resnet50 model (DCNN) - ResNet50 is a deep residual network with 50 layers. We used Adam as our optimizer after training for 50 epochs using Adam and a batch size of 785, we achieved 65.55% accuracy on the test set and 70% on the train set there is much less overfitting.



Deep Convolutional Neural Network-

The network consists of six two-dimensional convolutional layers, two max-pooling layers, and two fully connected layers. Max pooling uses the maximum value from each of a cluster of neurons at the prior layer. This reduces the dimensionality of the output array. The input to the network is a preprocessed face of 48 x 48 pixels. The model was developed based on the observation of the performance of the previous models. It was decided to go with a deeper network over a wide one. The advantage of using more layers is that it prevents memorization. A wide but shallow network memorizes well but does not generalize well. Multi-layer networks learn features at levels of abstractions allowing them to generalize well. The number of layers was selected so as to maintain a high level of accuracy while still being fast enough for real-time purposes. The proposed CNN differs from a simple CNN in that it uses 4 more convolutional layers and each of its convolutional layers differs in filter size. In addition, it utilized max pooling and dropout more effectively in order to minimize overfitting.



The second perspective is to improve the classifier algorithm to improve the prediction performance of the model and at the same time use relevant evaluation indicators to evaluate the prediction results. A fully connected layer with an L2 regularized penalty of 0.001 is then used along with an additional dropout of rate 0.5. Finally, a fully connected layer with a SoftMax activation function serves as the output layer. This model provided a

base accuracy of 0.79 on the testing set. The hyperparameters were then tuned, namely the batch size, the optimizer, and the number of epochs.

Each model was set to run for 50 epochs. However, in the interest of saving time and computational power, the network was allowed to stop training if there was no change in the accuracy over consecutive epochs. That is, the network would stop training if there was no change in the accuracy over 4 continuous epochs. This saved both time and computational power, especially in cases where there was no change in the accuracy within the earlier epochs themselves. The decision turned out to be a good one as none of the models exceeded 20 epochs.

d)-Testing:

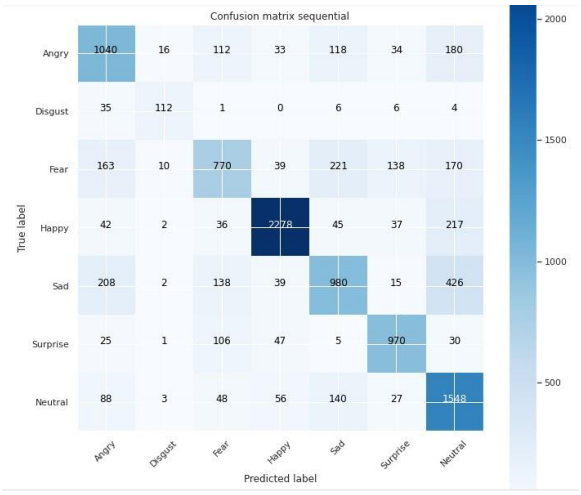
The dataset was initially split into an 80%-training set and a 20%-testing set. During the testing phase, each of the trained networks was loaded and fed the entire testing set one image at a time. This image was a new one that the model had never seen before. The image fed to the model was preprocessed in the same way as detailed in. Thus the model did not know already what the correct output was and had to accurately predict it based on its own training. It attempted to classify the emotion shown on the image simply based on what it had already learned along with the characteristics of the image itself. Thus, in the end, it gave a list of classified emotion probabilities for each image. The highest probability emotion for each image was then compared with the actual emotions associated with the images to count the number of accurate predictions.

The accuracy formula is detailed below. It simply counts the number of samples where the model correctly predicted the emotion and divides it by the total number of samples in the testing set. Here, the testing set consists of about 3,178 images. The accuracy formula is detailed below. It simply counts the number of samples where the model correctly predicted the emotion and divides it by the total number of samples in the testing set. Here, the testing set consists of about 7,178 images.

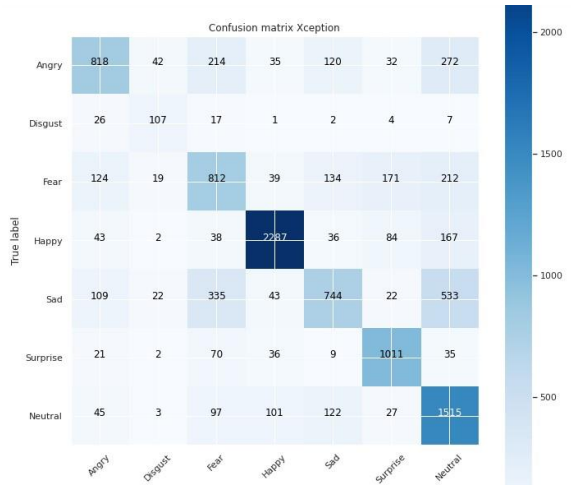
Accuracy = (Num.CorrectlyPredictedEmotions)/(TotalNum.Samples)

6. RESULTS

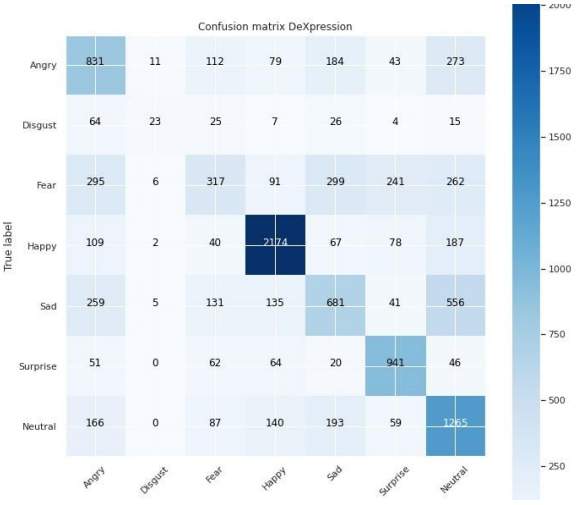
Upon turning the hyperparameters, the highest accuracy was achieved for each optimizer. We can observe the performance of the model by seeing the different model's confusion matrices more seeking the information regarding the data set of different emotions. we make step by step understanding of performance for all the models one by one. Confusion matrix for CNN model: -



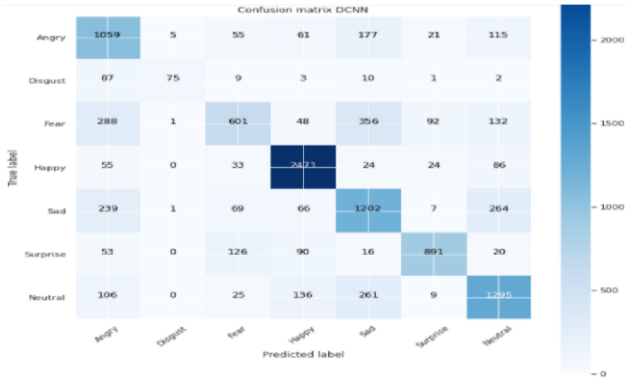
Confusion matrix for the Xception model: -



Confusion matrix for the DeXpression model: -



Confusion matrix for ResNet50(DCNN): -



.

6. Conclusion& Future work

In this project, the aim was to classify facial expressions into one of seven emotions by using various models on the FER2013 dataset. Models that were experimented with include, Deep face, Xception, DeXpression, CNN, and ResNet before arriving at the proposed model. The effects of different hyperparameters on the final model were then investigated. The final accuracy of 0.70 was achieved using the Adam optimizer with modified hyperparameters. It should also be noted that a nearly state-of-the-art accuracy was achieved with the use of a single dataset as opposed to a combination of many datasets. the model demonstrated has used significantly less data for training and a deep but simple architecture to attain near-state-of-the-art results. At the same time, it also has its shortcomings. While the model did attain near-state-of-the-art results, it also means that it did not achieve state-of the-art. Additionally, the relatively lower amount of data for emotions such as" disgust" makes the model have difficulty predicting it. This however does

illuminate a path for future work. If provided with more training data while still retaining the same network structure, the efficiency of the proposed system will be enhanced considerably. The ability of the model to make predictions in effectively real-time indicates that real-world uses of facial emotion recognition are barred only by the relative inaccuracies of the model itself. In the future, an in-depth analysis of the top2 predicted emotions may lead to a much more accurate and reliable system. Further training samples for the more difficult to predict emotion of disgust will definitely be required in order to perfect such a system.

7. REFERENCES

[1] P. Ekman, & W. V. Friesen. Constants across cultures in the face and emotion. Journal of Personality and Social Psychology, 17(2), 124-129. (1971)

[2] Raghuvanshi, A., & Choksi, V. (2016). Facial Expression Recognition with Convolutional Neural Networks. CS231n Course Projects.

[3] Ebrahimi Kahou, S., Michalski, V., Konda, K., Memisevic, R., and Pal, C. (2015). Recurrent neural networks for emotion recognition in video. Proceedings of the 2015 ACM on International Conference on Multimodal Interaction. 467-474. ACM.

[4] A. Mollahosseini, D. Chan and M. H. Mahoor. (2016). Going deeper in facial expression recognition using deep neural networks. 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, 1-10.

[5] Li M., Xu H., Huang X., Song Z., Liu X. and Li X. (2018). Facial Expression Recognition with Identity and Emotion Joint Learning. IEEE Transactions on Affective Computing. 1-1

[6] Tan L., Zhang K., Wang K., Zeng X., Peng X. and Qiao Y. (2017) Group emotion recognition with individual facial emotion CNNs and global image-based CNNs. Proceedings of the 19th ACM International Conference on Multimodal Interaction - ICMI 2017, 549-552. ACM.

[7] Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python, JMLR 12, 2825-2830.

[8] Open-Source Computer Vision. Face Detection using Haar Cascades: https://docs.opencv.org/3.4.1/d7/d8b/tutorial py face detection.html