Facial Expression Recognition using Deep Learning

Ankita Nallana **Ritvika Reddy Nagula** **Ashok Koduru**

Machine Learning Fall 2018
Northeastern University
Boston, MA 02115

Abstract

In a world steering towards non verbal communication, understanding the facial expressions has widespread applications for the society. There is no denying in the fact that these sometimes subtle, yet complex signals in an expression contains a plethora of details hiding among the deep layers of a persons mind. This particular understanding of human emotions opens a path of opportunities in the field of human computer interaction. As the need to communicate with machines arise, facial expressions play a significant role in the ability to make smart decisions time and again.

1 Introduction

Communication between humans is always subjective and can be expressed in an umpteen number of ways. Facial expressions are one of the most intricate ones among them. They can often be characterized by the mood and context of the conversation and can be hard to decipher correctly even for humans. Hence, in this project we are trying to consider few universally common emotions and categorize them using image processing and try to build a machine learning model to do this. Computers understand images represented in the form of their pixel data. Thus, it becomes even more difficult for a computer to understand the emotion being portrayed in a given image.

The task of recognizing facial emotions can further be divided into two types i.e static image based methods where the features representation is the image itself and dynamic sequence based methods where features can also be derived from the temporal relationship between adjacent frames. In our project, we focus on facial emotion recognition from static images.

We built a deep learning model consisting of a 5 convoluted layers using ReLu (Rectified Linear Units) activation, maxpooling and dropout on the Kaggle FER 2013 competition [1] . The loss used is the categorical cross-entropy loss function. The last layer of the CNN uses a softmax activation function. We obtained an accuracy of 64.8% on the test set. The input for this CNN was simply the raw pixel data from the images.

2 Related Work

Recent years saw a significant spike in the research on Facial Emotion Recognition(FER). The State-of-the-Art FER system predicts emotions correctly with around 75% accuracy [1]. This system uses a CNN architecture for extraction of facial features thereby employing Data Augmentation on those features at both the test and training time. This resulted in the improvement of performance compared to the other models at the time. This particular paper in [2] uses special image correction methods along with the CNNs by normalizing the histogram of the facial image to reduce the noise and using an ensemble approach on them.

[1] Kaggle Dataset

Before this, we have another method of solving facial feature extraction by including geometric features such as shapes of the faces and facial landmarks [3], Haar features [4], Local binary patterns [5], Scale Invariant Feature Transformation(SIFT) [6], Histogram of Oriented Gradients(HOG) [7], pixel intensities [8]. The study in [9] shows that some advanced image features were also extracted by localization of facial landmarks by generating a 3D texture reference model generic to all kinds of faces to efficiently estimate the facial components out of the input picture.

Few particular studies also shows this topic in a different light of tracking the facial features by geometric modeling of the specific subsections of the face. This was used in two main approaches. Geometric based parameterization which is the old school way of tracking specific spots on the image sequences was explained in [10]. Few scholars have realized that instead of tracking the spatial points and moving parameters which vary with time, appearance based parameterizations consisting of the color and pixel information of facial regions can be combined with methods such as PCA, LDA and Adaboost gave a better understanding of the extraction which can easily be extended to non still based images such as moving frames and videos.

There are other methods too such as using a Domain Adaptation approach for emotion recognition. Domain adaptation is applied in the learning pipeline so as to develop more transferable representation of the data. This approach has been successfully applied in various computer vision problems such as object detection and recognition and similarly for emotions.

3 Methodology

3.1 Support Vector Machine - SVM

Support Vector Machines or SVMs are a subclass of Machine Learning models using mainly for classification and, more recently regression problems. Since our problem falls into the former category, we decided to use it as a baseline. The main idea behind SVM is to find a separating *hyperplane* that can capture the hidden representation of our target space i.e. the hyperplane, if correctly identified, can clearly discriminate between the categories that the dataset represents.

A dataset can have many hyperplanes which can achieve that separation. However, the best hyperplane is one that has the most distance from the nearest data point because this achieves maximum separation. If maximum separation is not achieved, then it implies that our hyperplane is closer to one set of data points which then might lead to a misclassification if a new data point is encountered.

Mathematically, this maximum margin can be expressed as:

$$\text{margin} = \text{arg min d(x)} = \text{arg min} \frac{|x \cdot w + b|}{\sqrt{\sum_{i=1}^{d} w_i^2}}$$

where w is the weight vector, x is the input vector and b is our bias. The expression $x \cdot w + b$ is essentially our linear classifier that we're trying to learn to minimize the loss.

In training the SVM Classifier, our objective function (or loss) is defined as:

$$H(\theta) = \sum_{l} \max(0, 1 - y \cdot f(x; \theta))$$

where the objective is that $H(\theta)$ for some hyperparameter θ is minimized.

3.2 Convolutional Neural Networks

A convolutional neural network is the same as a neural network except for the fact that it assumes the inputs to be images and consists of a stack of layers which transform the image volume into an output volume which is basically the probability of the image belonging to every target class [11]. A convolutional neural network consists of different types of layers namely,

- Convolutional Layer: This layer, often called the CONV layer, takes the local regions of the input into consideration and returns the output corresponding to that small local region.

- Activation Layer: In general, most of the layers in a neural network are associated with a specific activation function. The hidden layers in a CNN commonly use the ReLU activation function. The ReLU activation function is as follows:

$$F(x) = max(0, x)$$

- Pooling layers: Pooling layers downsample the images i.e the output from these layers is the volume of the input reduced along the height and width.

- Fully Connected Layers: These are the layers responsible for resulting in the final class scores. The output layer in all CNNs is a fully connected layer that has the same number of neurons as the number of target classes.

As we can see from Figure 1, the convolutional layers, along with the activation and pooling layers are involved in learning the features of the images present in the given data. The fully connected layers are used to perform the classification part. An activation function

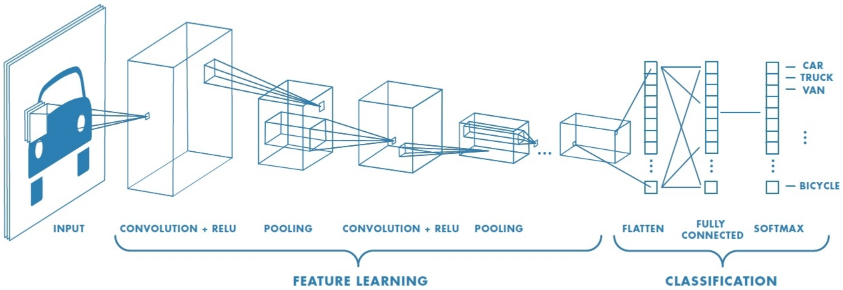

Figure 1: Example of a General CNN for classification

4 Experiment

4.1 Dataset

The Fer2013 Kaggle dataset contains a total of 35,887 images split into training, validation (public test) and test (private test) sets shown in Figure 2. Each image is assigned a numeric label pertaining to the emotion it is classified to be. There are seven key emotions that are targeted by the dataset. All of the emotions are almost equally distributed except disgust and happy. There are comparatively fewer images of the disgust emotion and more images of the happy emotion. Each image is a grey-scale image in a 48 x 48 pixel format. Examples are shown in Figure 3.

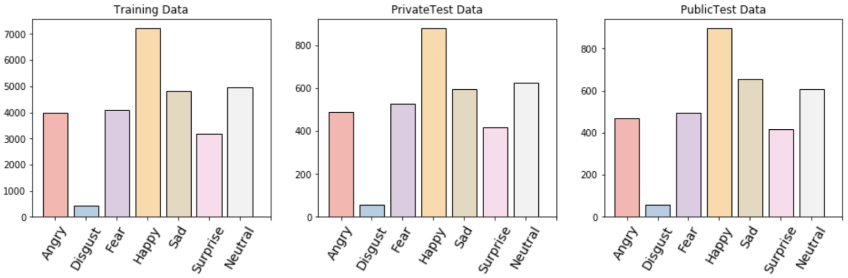

Figure 2: Distribution of Data

Type of Feature Input	Accuracy obtained
Raw Pixel Data	22%
Face Landmarks	47.1%
Histogram of Oriented Gradients (HOG)	15%

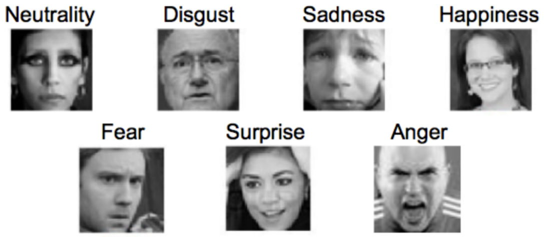

Figure 3: Image examples from the dataset

4.2 Baselines

4.2.1 Support Vector Machine (SVM)

The SVM run is our baseline model to compare against other models that we have worked with over the course of this project. We ran our SVM model over three kinds of input data:

Here is the confusion matrix displayed below from the raw pixel data SVM run. The predictions are all over the place and majorly incorrect thereby explaining the low accuracy rate:

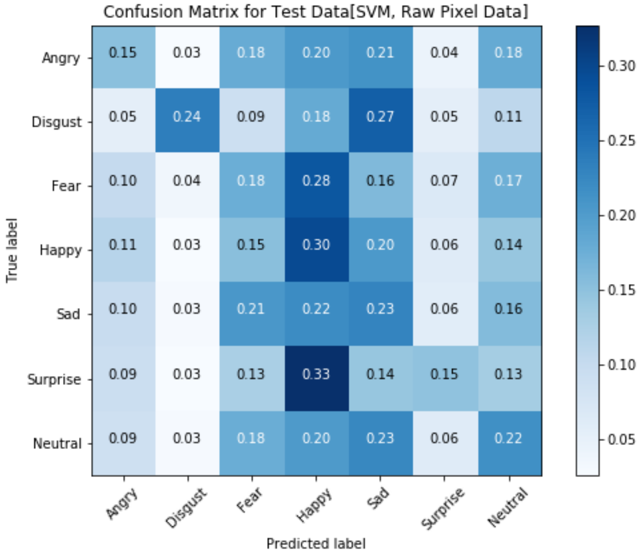

Figure 4: Confusion Matrix for SVM using raw pixel data

4.2.2 Convolutional Neural Network (CNN)

We started with a shallow CNN model built using 3 convolutional layers. Every convolutional layers uses a ReLU activation function and is followed by a batch normalization layer and a max-pooling layer. The architecture of this CNN is shown in Figure 5. A dropout layer follows the stack of convolutional, normalizing and pooling layers. This is in turn followed by a stack of 3 fully connected layers. The output layer uses a Softmax activation function as our aim is to predict the probabilities of the image belonging to all the seven emotions. This shallow network utilizes a a categorical cross-entropy loss and an ADAM optimizer. The emotion associated with the highest probability is then chosen as the predicted value. The input to the shallow CNN is simply the raw pixel data.

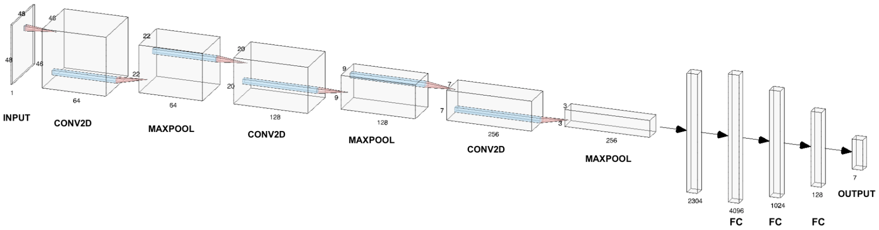

Figure 5: Shallow CNN Model Architecture

This model was trained for 100 epochs on a GPU on an ML-Engine provided by Google Cloud Platform. This baseline method gave us an accuracy of around 60%. Looking at the trends for the training and validation losses during training of the model in Figure 6, we can notice that after around 15 epochs, the validation loss keeps increasing and becomes way more than the training loss and thus we can conclude that this shallow model is an over-fitting model. The predictions using this model on the test data are shown in Figure 7.

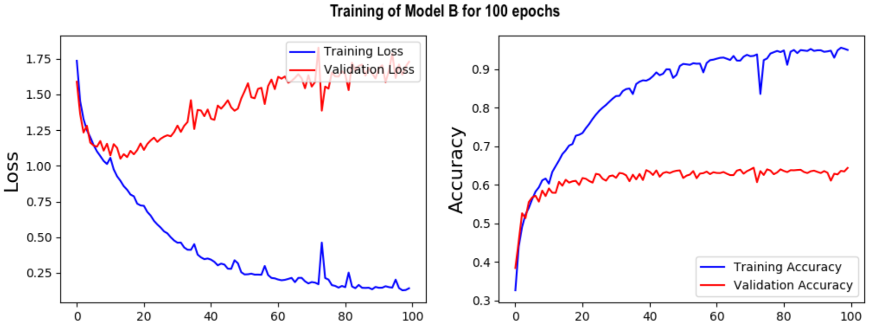

Figure 6: Training the Shallow CNN Model

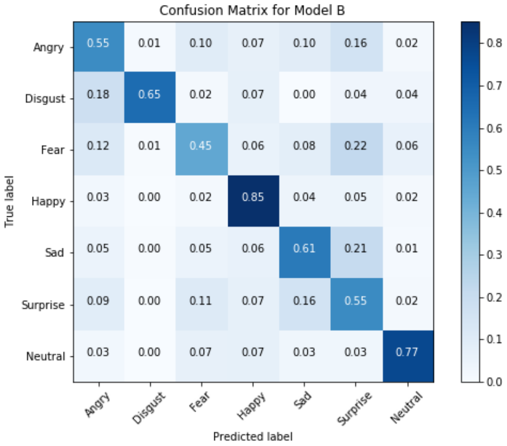

Figure 7: Confusion Matrix for the Shallow CNN Model

4.3 Results

4.3.1 Deep CNN

After looking at the previous shallow model, we increased the number of layers in the network and ended up using 5 convoluted layers. The model architecture is shown in Figure 8. The input is still the same raw pixel data in a 48x48 format.

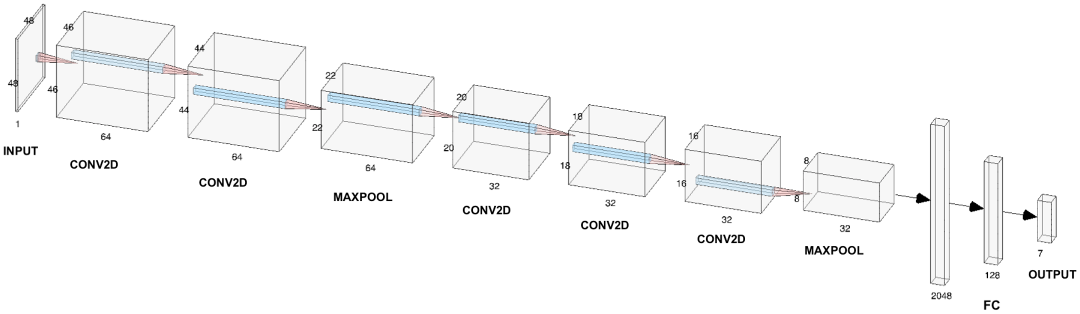

Figure 8: Deep CNN Model Architecture

This model was also trained for 100 epochs on a GPU on the Google Cloud Platform. We obtained an accuracy of 64.8%. We can observe from Figure 9 that the training converges around 40 epochs where the validation loss starts increasing slightly. The predicted results on the test set can be seen in Figure 10. It can be observed that the model classifies just 32% of the actual fear labels as fear and 24% as surprise. This is as expected because even in real life, it is difficult for a person to distinguish perfectly between fear and surprise. Our implementation can be found on Github [2].

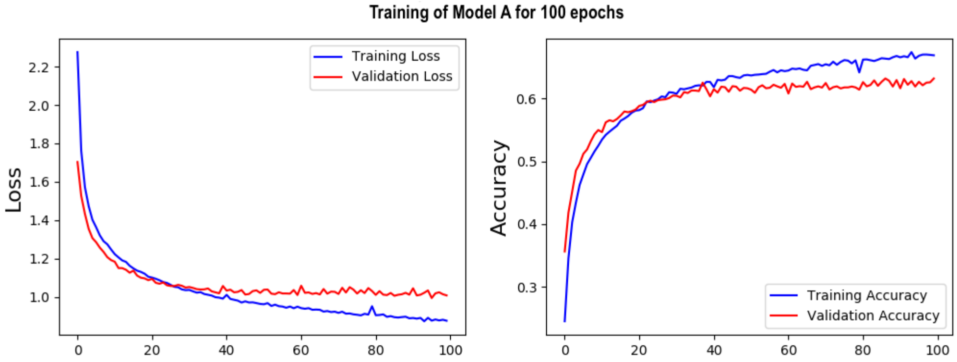

Figure 9: Training the Deep CNN Model

[2] Github Repo

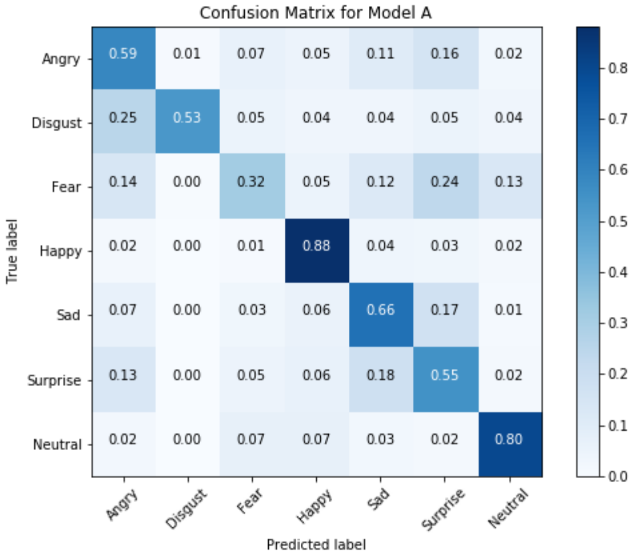

Figure 10: Confusion Matrix for the Deep CNN Model

5 Future Work

An extra addition in the models to improve the accuracy would be to take into account features derived from the faces in the images. Along with the CNN learning the features itself, there have been several implementations where extra facial features were computed by another method and fed into the fully connected layers of the CNN along with the features it learns as input. Some human emotions can be differentiated between each other via very minute differences in the facial patterns. For example, it is sometimes easy to mistake anger for disgust or vice-versa and hence such kind of subtleties are lost in the SVM model and it becomes imperative that we definitely need better defining features for the emotions. Upon further research we have come across Histogram of Oriented Gradients or HOG features to describe faces. The idea behind using HOG is that different emotions would have different and distinct gradients, particularly around the mouth and eye areas. This especially holds true when differentiating between emotions such as $happy$, sad, $surprised$. Although we used HOG features with our SVM run, we intend to use them with our CNN model and hope to improve on our accuracy with it.

References

[1] Christopher Pramerdorfer and Martin Kampel. Facial expression recognition using convolutional neural networks: state of the art. *arXiv preprint arXiv:1612.02903*, 2016.

[2] Yichuan Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013.

[3] Hiroshi Kobayashi and Fumio Hara. Facial interaction between animated 3d face robot and human beings. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, volume 4, pages 3732–3737. IEEE, 1997.

[4] Jacob Whitehill and Christian W Omlin. Haar features for facs au recognition. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 5–pp. IEEE, 2006.

[5] Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):915–928, 2007.

[6] Zisheng Li, Jun-ichi Imai, and Masahide Kaneko. Facial-component-based bag of words and phog descriptor for facial expression recognition. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 1353–1358. IEEE, 2009.

[7] S Mohammad Mavadati, Mohammad H Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F Cohn. Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013.

[8] Mohammad Reza Mohammadi, Emad Fatemizadeh, and Mohammad H Mahoor. Pca-based dictionary building for accurate facial expression recognition via sparse representation. *Journal of Visual Communication and Image Representation*, 25(5):1082–1092, 2014.

[9] Anbang Yao, Dongqi Cai, Ping Hu, Shandong Wang, Liang Sha, and Yurong Chen. Holonet: towards robust emotion recognition in the wild. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 472–478. ACM, 2016.

[10] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE, 2010.

[11] Karpathy. Cs231n convolutional neural networks for visual recognition. `http://cs231n.github.io/convolutional-networks/`.