Convolutional Neural Networks for Facial Expression Recognition

Shima Alizadeh, Azar Fazel

October 17, 2024

1 Introduction

Facial expressions are a crucial aspect of human communication, conveying emotions and non-verbal cues that play a significant role in interpersonal relations. Recognizing these expressions is essential for developing natural human-machine interfaces and has applications in behavioral science and clinical practice. However, reliable expression recognition by machines was a challenge at the time (2015). The study explores the use of Convolutional Neural Networks (CNNs) to classify facial images into seven emotional categories: anger, happiness, fear, sadness, disgust, surprise, and neutral.

Human facial expressions are one of the most important aspects of nonverbal communication, in which expressions can range from subtle signaling to serving major cues. Being able to interpret these with accuracy is critical for the design of more intuitive human-computer interfaces and is very important in behavioral research and clinical psychology. By 2015, developing machines that could reliably recognize facial expressions remained a challenge. The study explores the use of Convolutional Neural Networks (CNNs) to classify facial images into seven emotional categories: anger, happiness, fear, sadness, disgust, surprise, and neutral.

2 Related Work

In years prior to the release of the research paper, other 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 best known psychological framework for describing nearly the entirety of facial movements is the Facial Action Coding System (FACS). FACS is a system to classify human facial movements by their appearance on the face using Action Units (AU). An AU is one of 46 atomic elements of visible facial movement or its associated deformation; an expression typically results from the accumulation of several AUs.

3 Methods

The paper uses CNNs with variable depths for the task of facial expression recognition and uses the following neural network architecture:

$$[Conv-(SBN)-ReLU-(Dropout)-(Max-pool)]M-[Affine-(BN)-ReLU-(Dropout)]N-Affine-Softmax.$$

- Conv - Convolutional neural network layer
- SBN - Spatial batch normalization layer
- Relu - Rectifier activation function layer
- Dropout - Dropout layer
- Max-pool - Max pooling layer

- M - stands for repeating the entire CNN sequence M times (depth of CNN part)
- Affine - Fully connected layer
- BN - batch normalization
- N - stands for repeating the entire Fully connected sequence for N times

The first part of the network refers to M convolutional layers that can possess spatial batch normalization (SBN), dropout, and max-pooling in addition to the convolution layer and ReLU nonlinearity. After M convolution layers, the network is led to N fully connected layers that always have Affine operation and ReLU nonlinearity, and can include batch normalization (BN) and dropout. Finally, the network is followed by the affine layer that computes the scores and softmax loss function.

4 Dataset, Features, Preprocessing

The paper uses a dataset provided by Kaggle website, which consists of about 37,000 well-structured 48×48 pixel gray-scale images of faces which were divided into 29,000 training images, 4,000 validation images and 4,000 test images. The images are processed in such a way that the faces are almost centered and each face occupies about the same amount of space in each image. Each image has to be categorized into one of the seven classes that express different facial emotions. These facial emotions have been categorized as: 0 = Angry, 1 = Disgust, 2 = Fear, 3 = Happy, 4 = Sad, 5 = Surprise, and 6 = Neutral. The authors extended the dataset by flipping each image in the dataset horizontally. For preprocessing, the mean of the training dataset pixels was subtracted from the entire dataset. The authors also tried to add manual features to the network by adding HOG features after the M conv layers.

5 Analysis

5.1 Experiments

First, the paper authors build 2 CNN models: a shallow and a deep one (to observe the effect of adding more CNN layers to the network).

- **Shallow CNN model** had 2 convolution layers and one FC layer. In the first convolution layer, they had 32×3 filters, with the stride of size 1, along with batch normalization and dropout, but without max-pooling. In the second convolution layer, they had 64 3×3 filters, with the stride of size 1, along with batch normalization and dropout and also max-pooling with a filter size 2×2. In the FC layer, they had a hidden layer with 512 neurons and Softmax as the loss function. All layers, has ReLu as the activation function.
- **Deep CNN model** had 4 convolution layers, and 2 FC layers. The first cnn layer has 64×3 filters. The second has 128×5 filters. Third and fourth layers have 512×3 filters. All layers have the same stride, batch normialization, max-polling and dropout. And for the FC layers - the first has with 256 neurons and the second has 512 neurons. They used the Softmax as the loss function.

Since HOG features are very sensitive to edges - they are used for facial expression recognition. That's the reason why the authors wanted to explore if there is any way to apply HOG features along with raw pixels to the model and observe the performance of the model when it has a combination of two different features.

So, for this, they built a new learning model containing two neural networks: the first one contained cnn layers, and the second one had only fully connected layers. The features developed by the first network are concatenated with the HOG features and the resultant hybrid features were fed into the second network. In order to evaluate the performance of the network with hybrid features, **they again, trained two networks - a shallow and a deeper one (with the same architecture as the shallow and deep networks that they trained for the first experiment)**.

Hyper-Parameters:

- Shallow model (Learning Rate: 0.001, Regularization: 1e-6, epochs: 35, batch size: 128)
- Deep model (Learning Rate: 0.01, Regularization: 1e-7, epochs: 35, batch size: 128)

5.2 Paper Results

- First, the authors compared the performance of the shallow and the deep models (without HOG features) using accuracy plots, and confusion matrix, which is a tool for evaluating the performance of a classification model by summarizing the correct and incorrect predictions across different classes. As seen in Figure 2, the deep model enabled us to increase the validation accuracy. Furthermore, one can observe that the deep network has reduced the overfitting behavior of the learning model by adding more non-linearity and hierarchical usage of anti-overfitting techniques such as dropout and batch normalization in addition to L2 regularization. Figure 3 and 4 shows the confusion matrix of the models. As demonstrated in these figures, the deep network results in higher true predictions for most of the labels.
- Figures 5 and 6 illustrate how the combination of HOG features impacts the performance of each model (shallow and deep). As seen in these figures, the accuracy of the model is very close to the accuracy we got from the model that has no HOG features. This means that CNN is strong enough to extract sufficient information including those coming from HOG features by using only raw pixel data.

Expression	Shallow Model	Deep Model
Angry	41%	53%
Disgust	32%	70%
Fear	54%	46%
Happy	75%	80.5%
Sad	32%	63%
Surprise	67.5%	62.5%
Neutral	39.9%	51.5%

Figure 1: The accuracy of each expression in the shallow and deep models

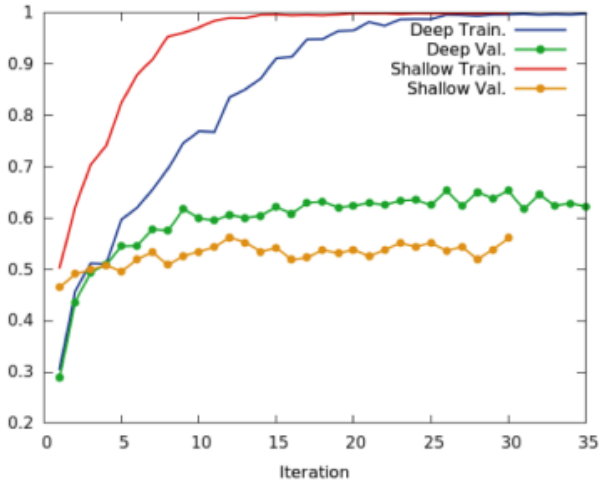

Figure 2: The accuracy of the shallow and deep models for different numbers of iterations

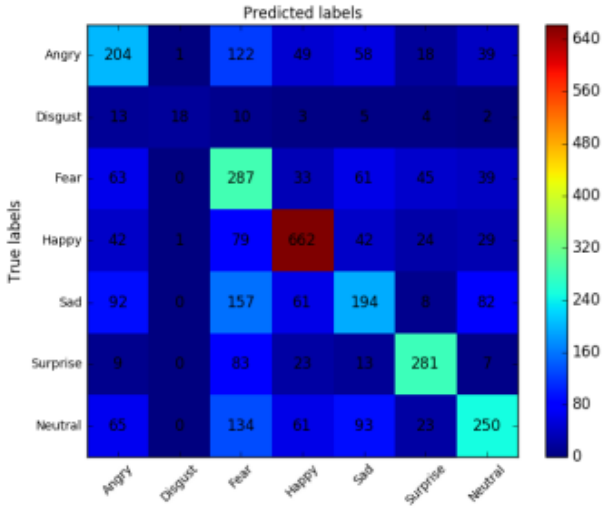

Figure 3: The confusion matrix for the shallow model

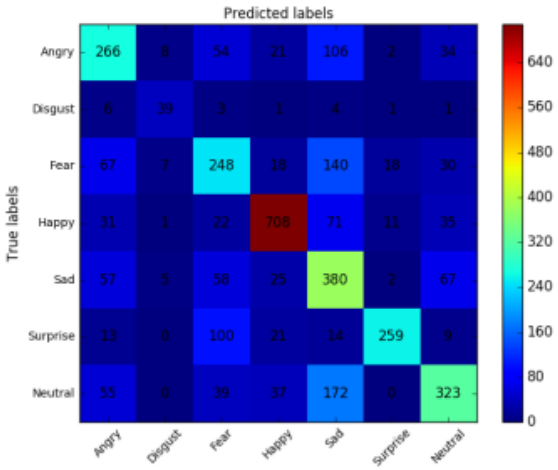

Figure 4: The confusion matrix for the deep model

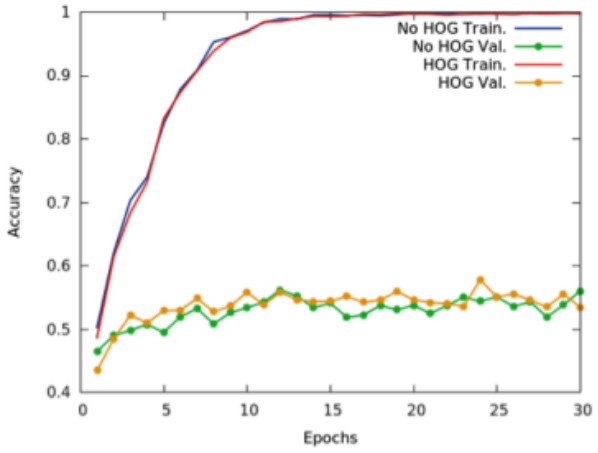

Figure 5: The accuracy of the shallow model with hybrid features for different numbers of iterations

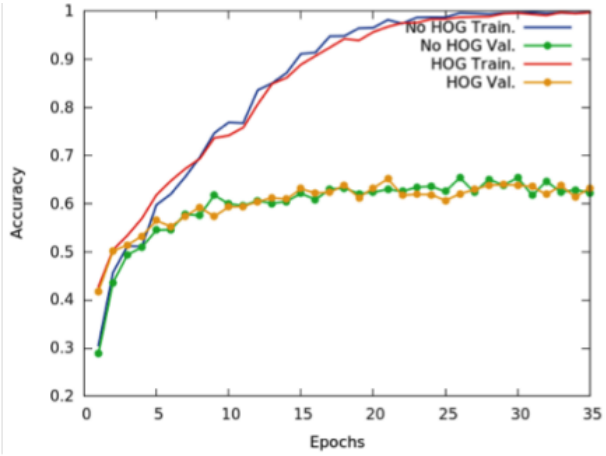

Figure 6: The accuracy of the deep model with hybrid features for different numbers of iterations

5.3 Our Results

Expression	Shallow Model	Deep Model
Angry	33.47%	46.07%
Disgust	44.64%	60.71%
Fear	36.06%	49.00%
Happy	77.39%	79.57%
Sad	53.92%	53.09%
Surprise	69.00%	77.15%
Neutral	32.94%	61.26%

Table 1: The accuracy of each expression in the shallow and deep models

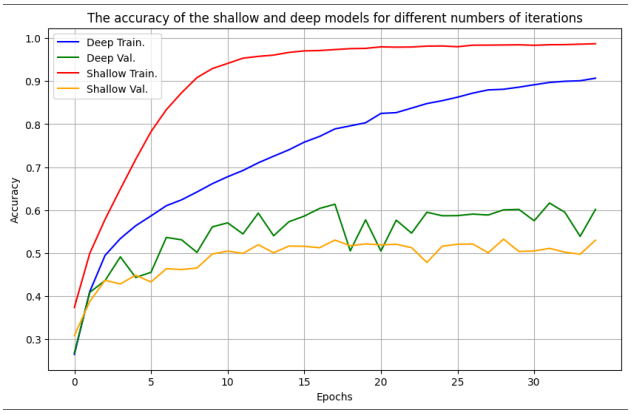

Figure 7: The accuracy of the shallow and deep models for different numbers of iterations

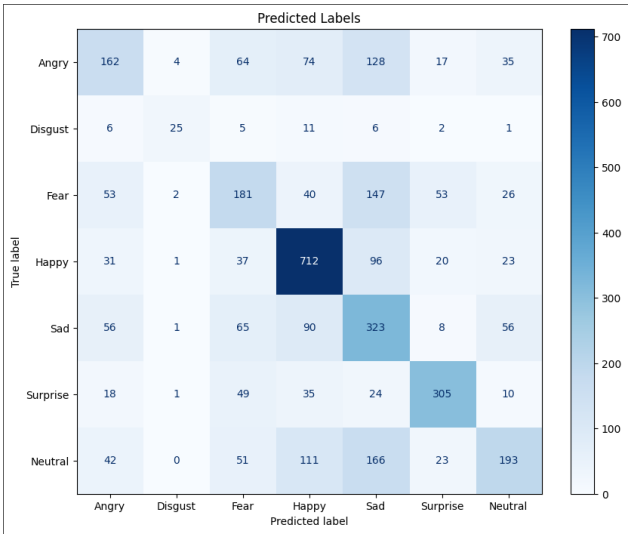

Figure 8: The confusion matrix for the shallow model

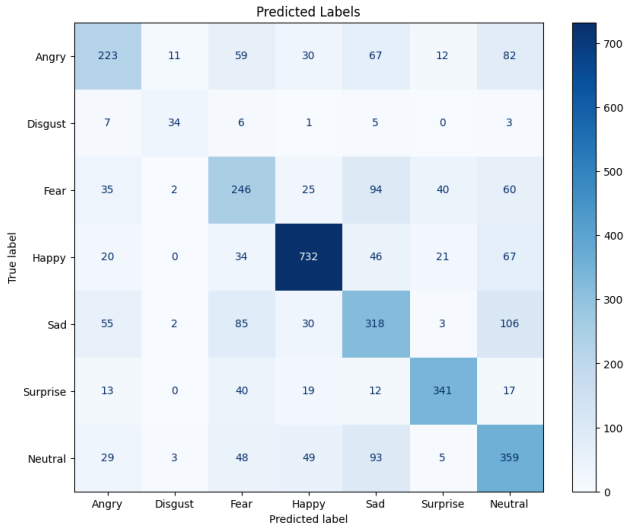

Figure 9: The confusion matrix for the deep model

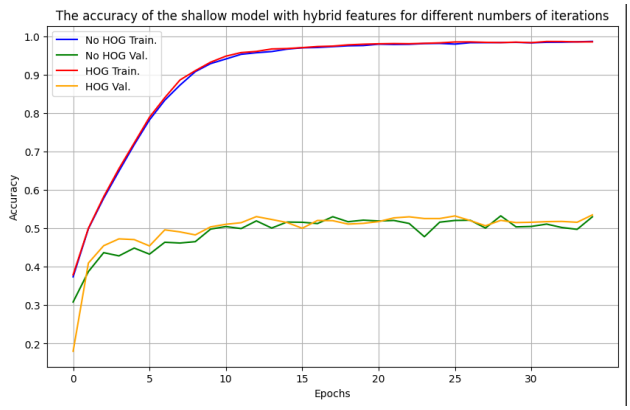

Figure 10: The accuracy of the shallow model with hybrid features for different numbers of iterations

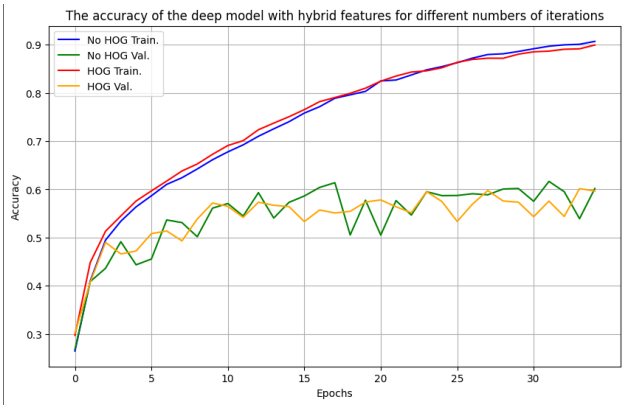

Figure 11: The accuracy of the deep model with hybrid features for different numbers of iterations

5.4 Our results vs Paper results:

- **Accuracy of the shallow and deep models** - According to the paper **their shallow model** accuracy was 55% on validation set and 54% on the test set, and **our shallow model** accuracy was 53.02% on validation set, and 52.9% accuracy on test set. **Their deep model** accuracy was 65% on the validation set and 64% on the test set, and our deep model accuracy was on 60.18% validation set, and 62.77% on test set. **So, the results are very close.** (Our test results can be found in our .ipynb notebook on our GitHub)
- **Accuracy With the HOG features** - For the shallow model: figure 5 (the paper) is almost the same as figure 10, and for the deep model, figure 6 (the paper) is almost the same as figure 11.
- **Confusion matrix are also very similar**

6 Conclusion

We developed various CNNs for a facial expression recognition problem and evaluated their performances using visualization techniques. The results demonstrated that deep CNNs are capable of learning facial characteristics and improving facial emotion detection. Also, the HOG didn't help in improving the model accuracy, which means that the CNN networks can intrinsically learn the key facial features by using only raw pixel data. For this project, we trained all the models from scratch using CNN packages Tensorflow.

7 Challenges

The authors' paper does not provide an official GitHub implementation, so we had to implement models from scratch. Furthermore, they utilized HOG features, which were unfamiliar to us, requiring us to explore this topic in detail.