Convolution for Image Recognition

Jae-Ho Lee, Smruthi Iyengar, Kyle Listermann

I. Abstract

We put together a model that predicts emotions in faces, some emotions better than others. Performance could be improved with more layers but with what we did we achieved about a 60% accuracy. We fixed some overfitting issues to achieve this accuracy.

II. Introduction

In the project, our goal is to utilize modeling and algorithms to identify the facial expressions of people, so as to classify and organize them. Facial expression recognition is a very important and necessary link in the development of AI. Social security systems, mobile payment facial recognition systems, car autopilot character recognition systems and robotic service systems, among many other aspects of people's lives, will use facial expression recognition knowledge and technology. Furthermore, recognizing facial expressions requires humans to have very complex logical thinking and a certain degree of familiarity. The cost of these identifications is considerable. If AI can identify facial expressions more accurately and quickly with specific frameworks and sophisticated algorithms, then computers will be able to record, store and analyze facial images on a large scale at low cost, greatly accelerating social development.

III. Related Works

The related works explore multiple different steps into thinking about how well neural networks work and what different concepts they can be helpful in. One of those fields consist of micro expressions of and how well neural networks can detect emotion from very little facial expression at all. They used a Dual Temporal Scale Convolutional Neural Network. Others aimed to test same datasets using different number of epochs to find the test errors with each. There was also an attempt to use very specific datasets of which would more likely give better results given the fact that you're talking about similar items in a dataset. Such as all-female or all male or a group with similar patterns with expressions. Another aimed to get best method from one dataset and used on different datasets to see if the method was good for more data than just within the one dataset which is what would make it useful. In another experiment author's used multiple different methods of data augmentation for classifying dogs and cats such as GANS and neural loss. This can also apply to how the computer will see some facial expressions as visually similar. These are all different ways to approach future work in the field to help design

IV. Data

The dataset was obtained from a facial expression recognition challenge in kaggle. It consists of two variables: "emotion" and "pixels". Each "pixels" value was converted to a 48x48x1 numpy array in order to predict "emotion" as an input to a convolutional neural network.

Variables:

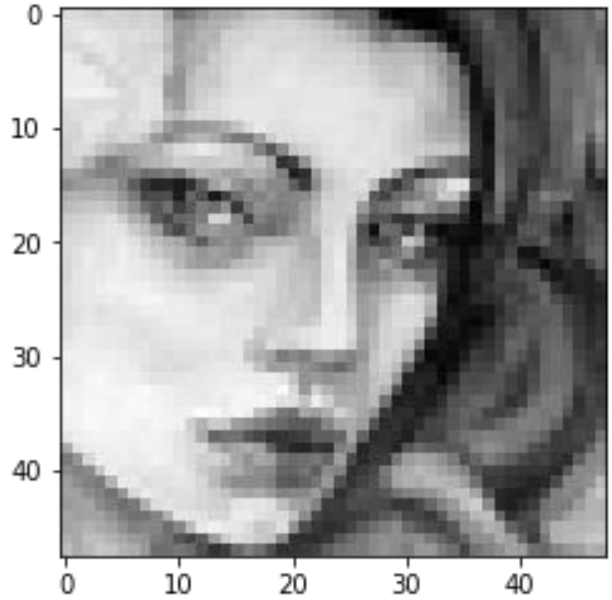
- "emotion" (#s in 0-6)
- "pixels" (48x48x1 3D arrays)

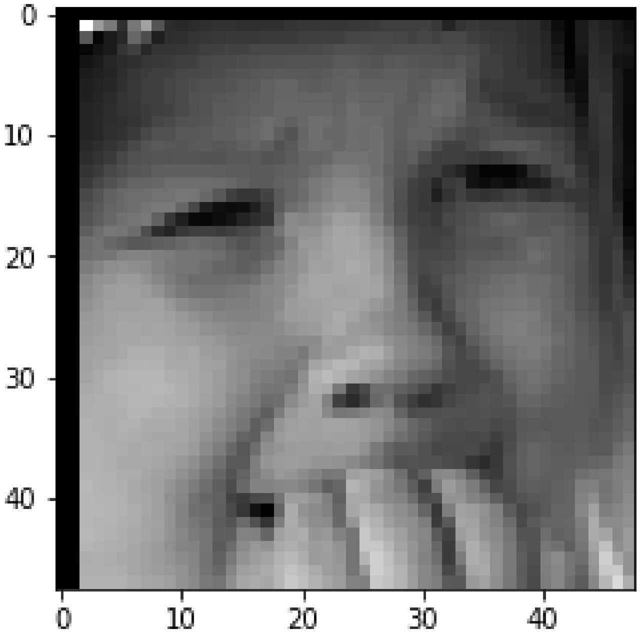
Data was also normalized and augmented to boost model performance. All data was normalized by dividing pixel values by 255. Augmentation techniques -- flipping, rotating, padding, and re-cropping the images -- were applied to 200 random samples of the training data.

Final dataset:

- training (28,909)
- validation (3,589)
- testing (3,589)

	Emotion	Emotion
3	8989	0.250481
6	6198	0.172709
4	6077	0.169337
2	5121	0.142698
0	4953	0.138017
5	4002	0.111517
1	547	0.015242





V. Methods

The convolutional neural network of our choice is defined below, where:

- Conv2: N-M (N = height and width of filter, M = number of neurons, ReLU activation)
- Maxpool: P (P = height and width of pooling filter)
- Dropout: D (D= proportion of dropped neurons)
- FC: F (F = number of neurons, ReLU activation)

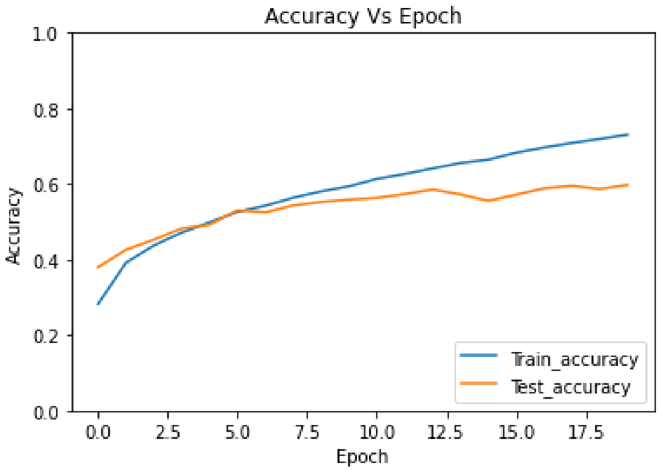
Input 48 x 48 grayscale image
Conv2: 3-64
Maxpool: 2
Dropout: 0.2
Conv2: 3-128 Conv2: 3-128
Maxpool: 2
Dropout: 0.2
Conv2: 3-256 Conv2: 3-256
Maxpool: 2
Dropout: 0.2
FC: 1000
Dropout: 0.3
Soft-max

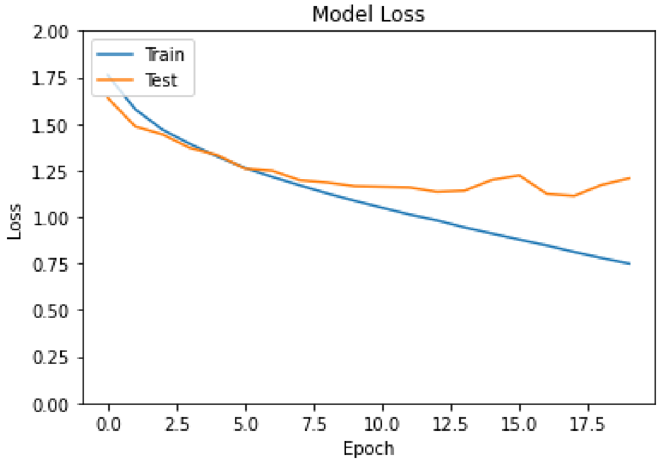
Number of parameters: 10,368,263

Paddings were added in order to maintain the same output dimensions as the input dimensions for all convolutional layers. Dropout layers were included in order to regularize the model and prevent overfitting. A categorical cross-entropy loss function was optimized with the RMSprop algorithm with a learning rate of 5e-5. Our model was initially based on the of VGG19 architecture, following the choice of using ReLU for all activation. However, we decided to reduce the size of the model and include dropout layers as the accuracy and loss graphs our previously tested models showed signs of overfitting.

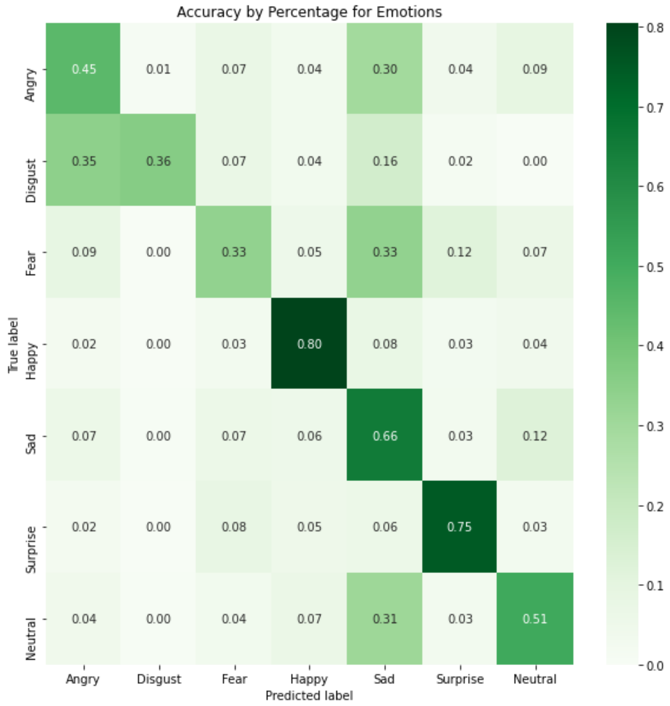
VI. Results

A. Training Results





B. Summary Tables



VII. Discussion

After running the model for 20 epoches, the train accuracy is at about 73%. The Test accuracy is at approximately 60%. There is no evidence of over or underfitting in the model. The model is working correctly.

Both the Test and the Train data have a loss of about 1.75 at the start. But, by the end of 20 epochs, the train has a loss of .72 while the test's loss plateaus at about 1.15. The cost function decreases at a higher rate for the train data after epoch 5.

The Overall accuracy when using the validation data was 59.77%. As we can see from the confusion matrix below there were some emotions that were misclassified at higher rates than others. The True positive rate for the emotions is 80% happy,75% surprise,66% sad,45% angry,51% neutral, 36% disgust, and 33% for fear. Angry, neutral, fear, and disgust had lower accuracies than the model did on average. It is interesting that angry was misclassified as disgust for only 1% of its observations but disgust was misclassified as angry for 35% of its observations. Angry, Fear, and Neutral were all most commonly misclassified as sad. Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

VIII. Conclusion

Data augmentation and adding droup out layers to the model was needed in order to fix the model's problem of overfitting.

The overall accuracy of the model is about 60%. The model could be up or down sampled to adjust for the unbalance in the output data. This would help the model perform better. The model predicts some emotions better than others. So by balancing the data this should solve that problem. We could also see if performance would improve if we added more layers to the model so it could be more like a VGG-19 model.

IX. References

[1] A. De Souza, and T. Oliveira-Santos, “Facial Expression Recognition with Convolutional Neural Networks: Coping with Few Data and the Training Sample Order,” Pattern Recognition.

[2] M. Kumbhar, “Facial Expression Based on Image Feature” ResearchGate.

[3] M. Peng, “Dual Temporal Scale Convolutional Neural Network for Micro-Expression Recognition” Frontiers.

[4] J. Wong and L. Perez, “The Effectiveness of Data Augmentation in Image Classification using Deep Learning,” Stanford University.

[5] D. Ciresan, “Flexible, High Performance Convolutional Neural Networks for Image Classification” IDSIA, USI and SUPSI.

[6] P. Carrier and A. Courville, “Facial Expression Recognition,” https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data Kaggle.

[7] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks For Large-Scale Image Recognition” University of Oxford.