Facial Expression Classification into Emotions

David Orozco, Christopher Lee, Yevgeniy Arabadzhi, Deval Gupta

Abstract

The purpose of this project was to classify images of people's facial expressions into 8 distinct emotion classes. The Cohn-Kanade AU-Coded Facial Expression Database (CK) and the Japanese Female Facial Expression (JAFFE) Databases were compiled and preprocessed. These data were input into pretrained networks (AlexNet, VGG, and ResNet); weights were fine tuned using transfer learning. We achieved an average 90% accuracy across all classes in testing.

Problem & Motivation

Emotion classification introduces a new sets of challenges, since the model needs to be able to differentiate intra-facial features. We use Deep Learning for, both, feature selection and expression classification. **AlexNet** provides a good start for **Transfer Learning**. Because of the many classes the net was trained on, there is an expectation that universal features had been learned by the network.

Datasets

Two datasets were used for fine-tuning the network. Each underwent its own pre-selection process, which resulted in a smaller, but cleaner, dataset. CK[1] , for example, initially consisted of video expressions, changing from neutral to one of seven emotion classes. The frames of each video were then split into classes; a margin of frames around the split were removed.

Table 1: Dataset Size Reduction, after Preselection

	Initial Size	Post Size	Initial # Classes	Post # Classes
CK[1]	5877	4468	7	8
JAFFE[2]	213	213	7	7

It is worth noting that two other datasets were processed, but not used. These were the SoF[3] and 10k-US[4] datasets.

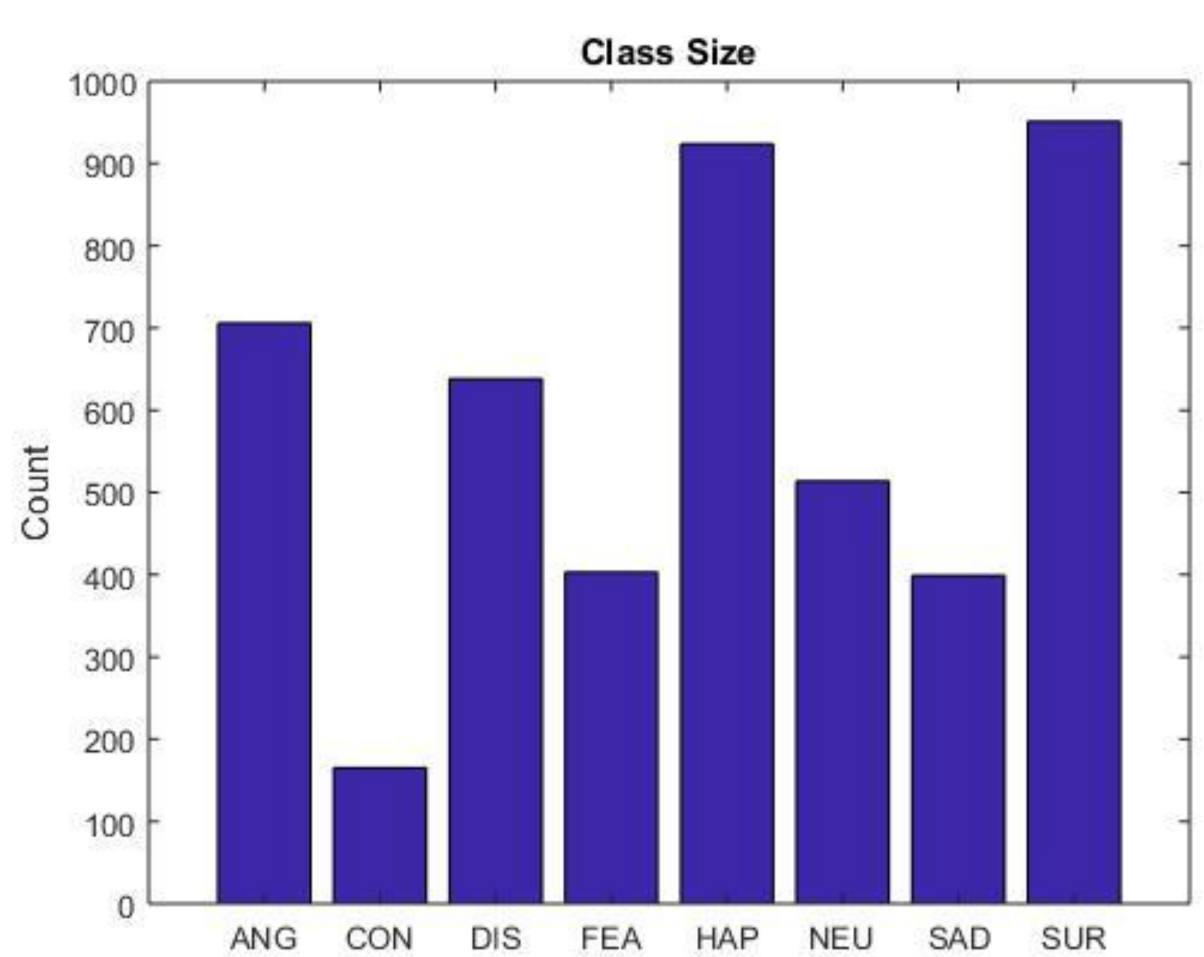


Figure 1: Final Class Distribution

Methods

Three pretrained networks (VGG, AlexNet, and ResNet) were pretained on the ImageNet dataset. The network architectures were adapted for the specific task of emotion classification as shown below, and the initialized weights were then fine tuned on our datasets. This process is known as transfer learning.

Preprocessing

In order to create a uniform robust dataset, each image was preprocessed by using a Haar Cascade Face Detection algorithm. Using OpenCV and a pretrained classifier to find faces, each the image was cropped for the found face. The blue rectangle in Figure 2 shows an example of face finding before crop.

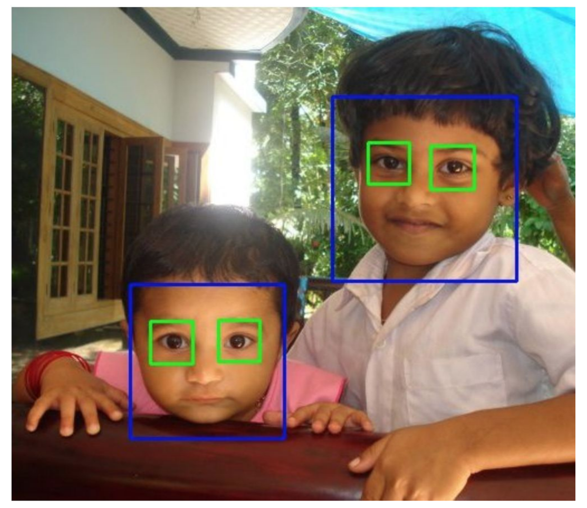


Figure 2: Haar Cascade Face Detection Example [5]

Transfer Learning

Since Imagenet was never trained on faces or facial expressions, the last 3 layers needed to be altered and fine tuned to meet our specifications. However, as we can see in Figure 3 the convolutional layers most likely are also specialized to ImageNet, thus fine tuning was done on entire network to boost performance to our specific task.

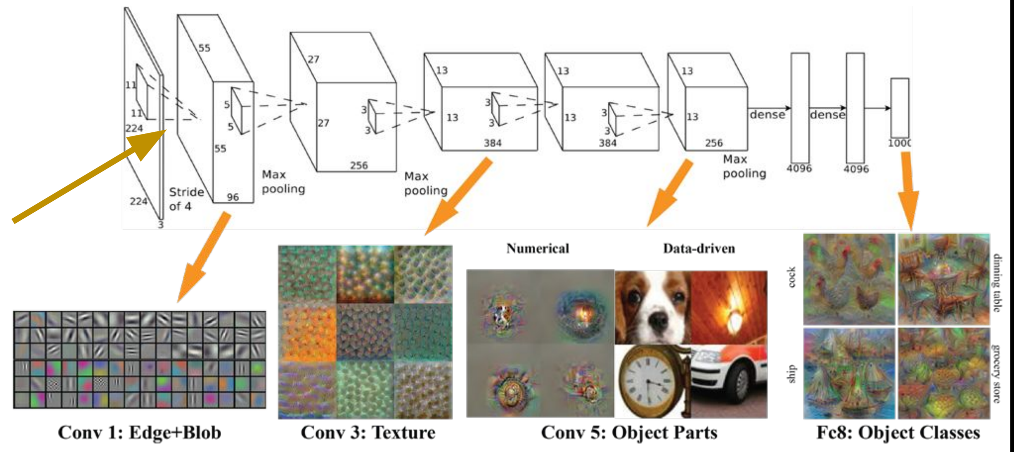
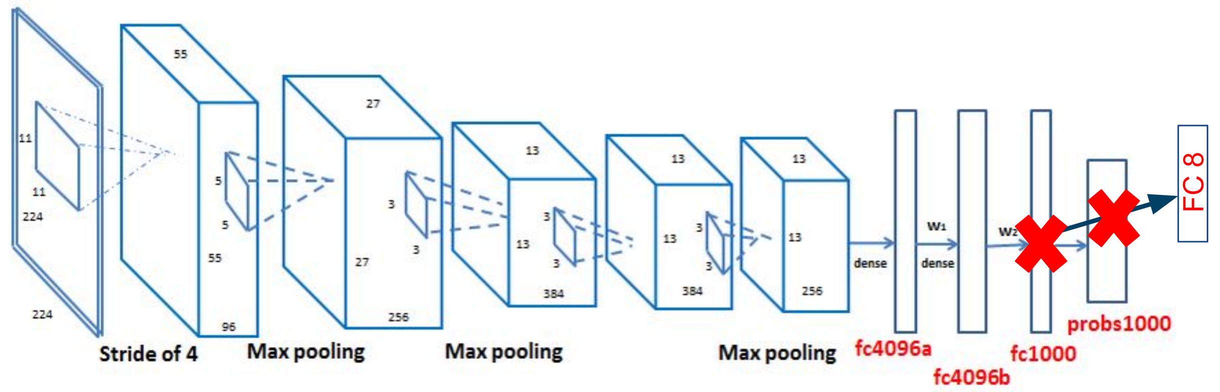


Figure 3: Feature Visualizations

AlexNet, ResNet, VGG



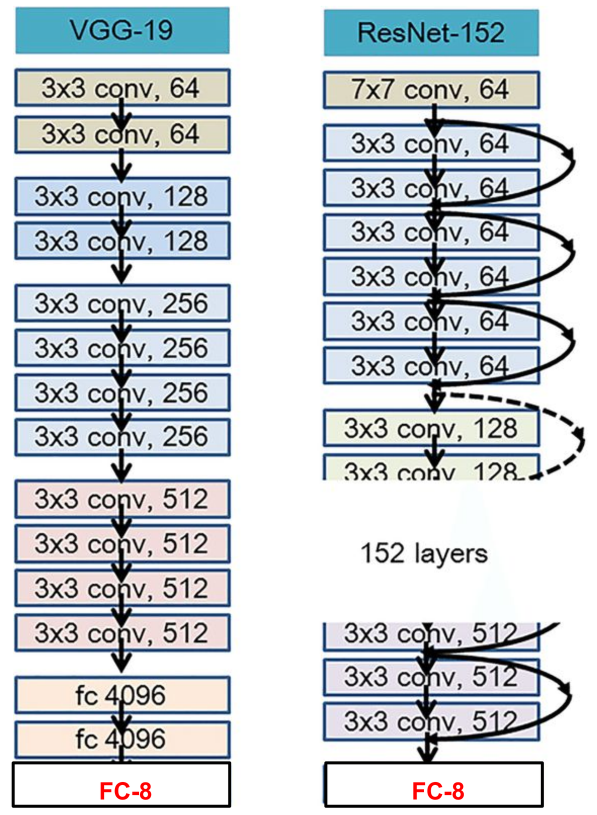


Figure 4 AlexNet Architecture Changes (above):
Figure 5 VGG and ResNet Architecture Changes (right):

Training & Testing

Training data was augmented to ensure robust training. The methods employed included: 240 pixel resize, random 224 pixel crop, random horizontal flip, 10% random grayscale, +/- 1% angle deviation, and grayscale pictures to RGB. A early stopping method using 10% of the data was also implemented. Another 10% of the data was used solely for testing purposes. Finally, images of one team member's faces were also tested.

Results

Figure 6: Accuracy for each class

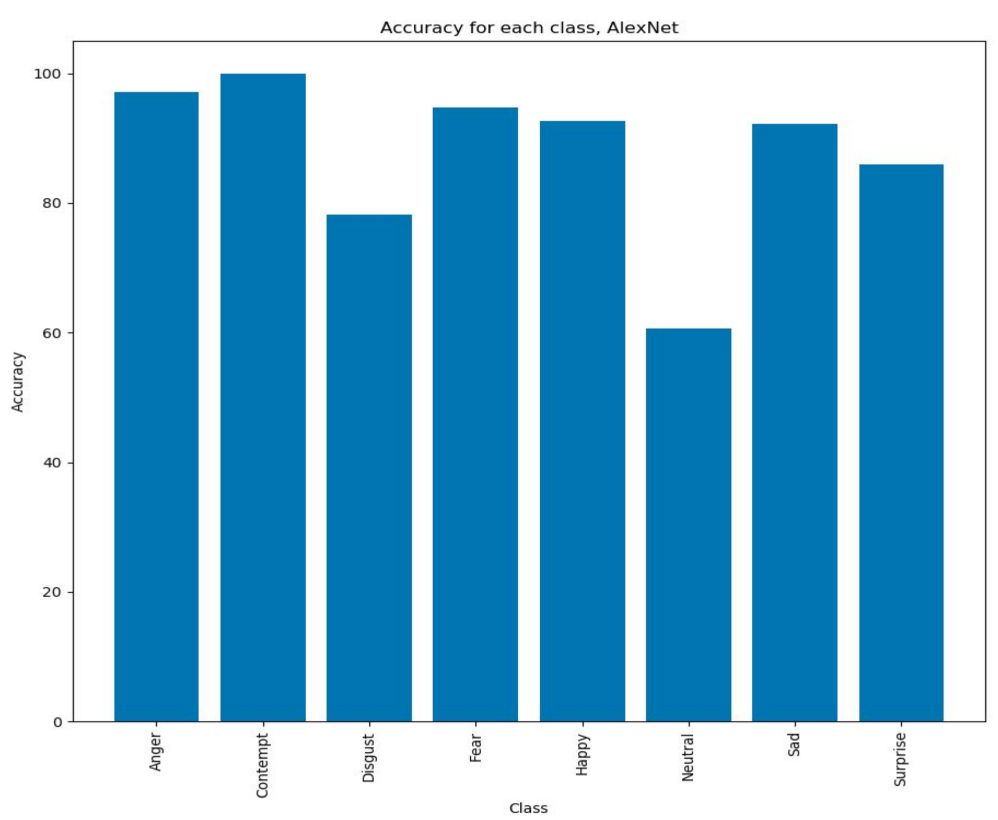
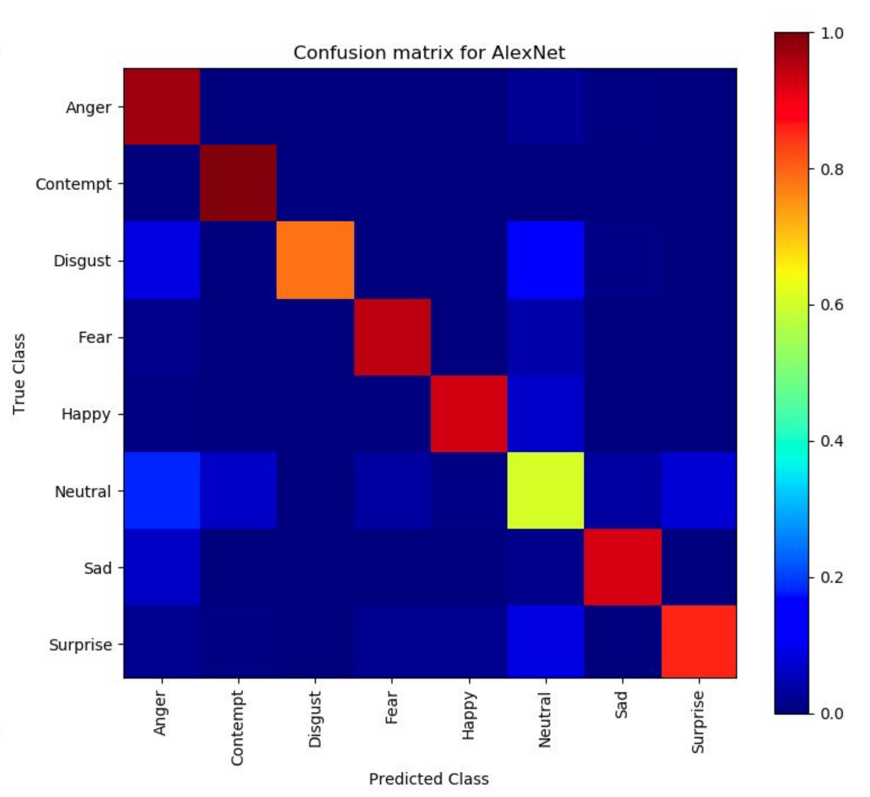
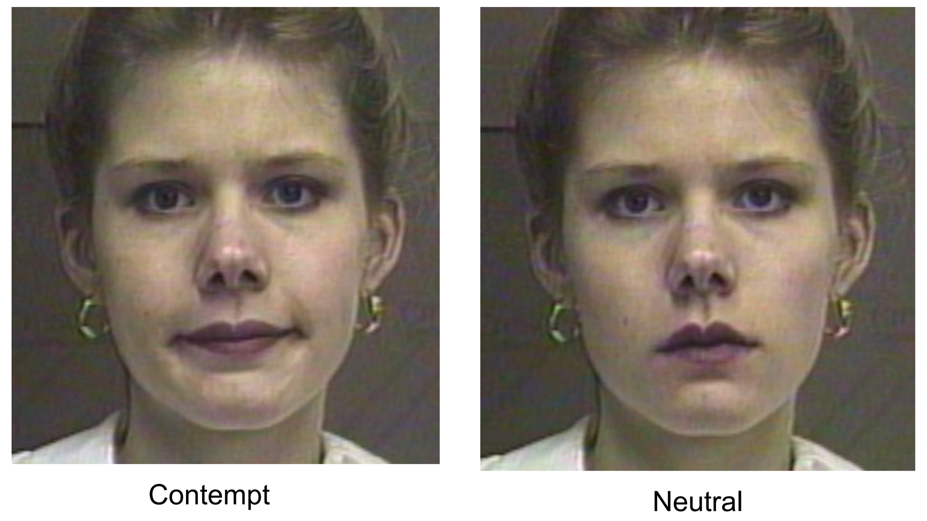


Figure 7: Confusion matrix



In general, our network had an average of 90% accuracy across all classes, with the highest accuracy in contempt and the lowest in neutral. From the confusion matrix: the test set faces from the neutral class seemed to have false-positives as well as some false negatives. This might be due to some of our data being inconsistent (i.e. some images from the CK dataset may lean toward neutral due to the range of expressions). This is similarity in expression is shown in Table 2.

Table 2: neutral vs. contempt examples



Conclusions

We achieved an average 90% accuracy across all classes. One possible way to improve the network's performance would be to clean up more of the training data. We plan to include testing performance on our faces in the final submission.

References

[1]: Cohn-Kanade AU-Coded Facial Expression, http://www.consortium.ri.cmu.edu/data/ck/
[2]: The Japanese Female Facial Expression, http://www.kasrl.org/jaffe.html
[3]: Specs on Faces Dataset, https://sites.google.com/view/sof-dataset
[4]: 10k US Adult Faces Database, http://www.wilmabainbridge.com/facememorability2.html
[5]: https://docs.opencv.org/3.4/d7/d8b/tutorial_py_face_detection.html