Classifying Emotion in Real Time Using CNNs

CMPUT 399 Fall 2016 Team ReLU

Taylor Bos Kevin Gordon

bos ktgordon

Jun Ma Siqi Yan

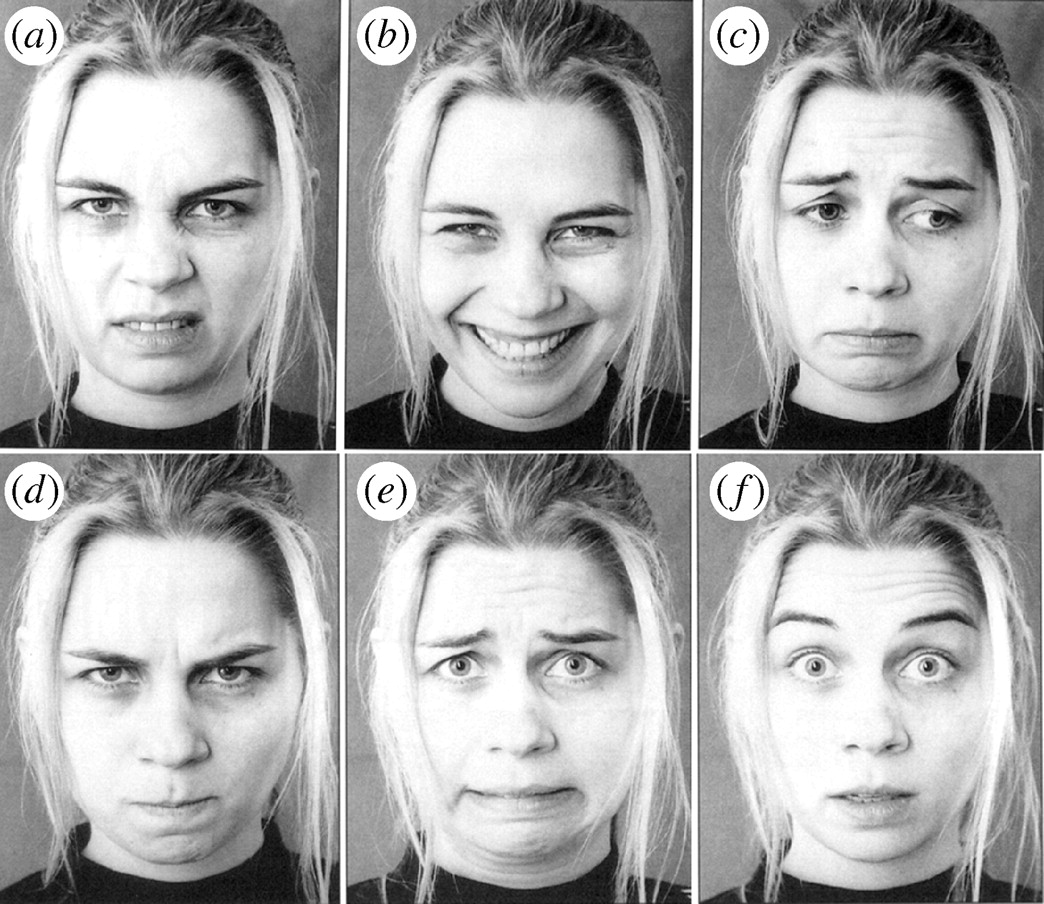
jun5 syan3

***Abstract*—We have created and trained two convolutional neural networks for classifying facial expressions in real time. These networks are very similar but using two different deep learning frameworks. The Darknet[12] five layer network displayed a training accuracy of 92.8% and test accuracy of 69.0% while the Caffe[4] three layer network was able to get to a test accuracy of 84%. We augmented the KDEF[2] and JAFFE datasets to train our networks. During implementation we take input from webcam, apply a face bounding box, and then the model displays which emotion it has decided is the most probable. This is done at about five frames per second.**

# Introduction

The topic of emotion has been highly researched across several fields of study including psychology, neuroscience, endocrinology, medicine, sociology, and more recently computer science. The classification of emotions has especially been a topic of heavy debate centering on two main viewpoints. Basic emotions focuses on the idea that emotion is discrete and measurable. Paul Ekman displayed in his paper “Constants Across Cultures in the Face and Emotion” that there are six universally recognizable emotions displayed through facial expressions: happiness, anger, sadness, disgust, surprise, and fear[5]. The other viewpoint centers around a multi dimensional analysis which has valence and arousal as the dimensions [7]. For simplicities sake we will be focusing on the emotion classification proposed by Ekman.

The study of relating computers and human emotion was named Affective Computing in 1995 in R. W. Picard’s book of the same name[6]. This area of study connects the fields of computer science and psychology and strives to develop systems that can recognize and stimulate human emotions. The primary application for affective computing is to teach a machine to recognize the emotions of its user and adapt in accordance to this. Other applications include perceptual information retrieval, biofeedback, machine intelligence, security and monitoring. Perhaps even microexpressions could eventually be classifying enabling recognition of smaller or more controlled emotions to be recognized resulting in the application of deception detection.

Figure 1: Ekman’s six basic emotions. <http://rstb.royalsocietypublishing.org/content/364/1535/3505>

Emotion recognition visually has its challenges as even though there are a fair number of datasets available most, if not all, of these datasets are posed expressions. This means that the expressions were prompted and therefore consist of what the model believes to match the emotion instead of the image being a result of an actual emotion. This of course creates a bias and the images tend to be of exaggerated expressions.

# Literature Survey

Since the early 90s affective computing has been a fairly active field with many different methods including speech, facial, body gesture, and physiological monitoring. Physiological monitoring primarily focuses on the autonomic nervous system as shown in the experiment “Using Neural Network to Recognize Human Emotions from Heart Rate Variability and Skin Resistance”[8] however preparing the dataset for an experiment like this is quite involved and time consuming. Also with how this physiological data is collected they found that recognizing real time emotional changes would be quite difficult through this method. An interesting study “Multimodal Deep Convolutional Neural Network for Audio-Visual Emotion Recognition” done recently combined the two approaches of facial expression and speech pattern recognition[9]. The study showed that combining approaches could be the key to higher accuracies. In the beginning of this field of study several approaches were explored such as using a recurrent neural network in 1993 by Kobayashi and Hara[10]. However convolutional networks were shown to outperform the accuracy of other models as shown in C. Nebauer’s paper “Evaluation of Convolutional Neural Networks for Visual Recognition” in 1998 [11]. Since then convolutional neural networks have been primarily used for the problem of facial emotion recognition.

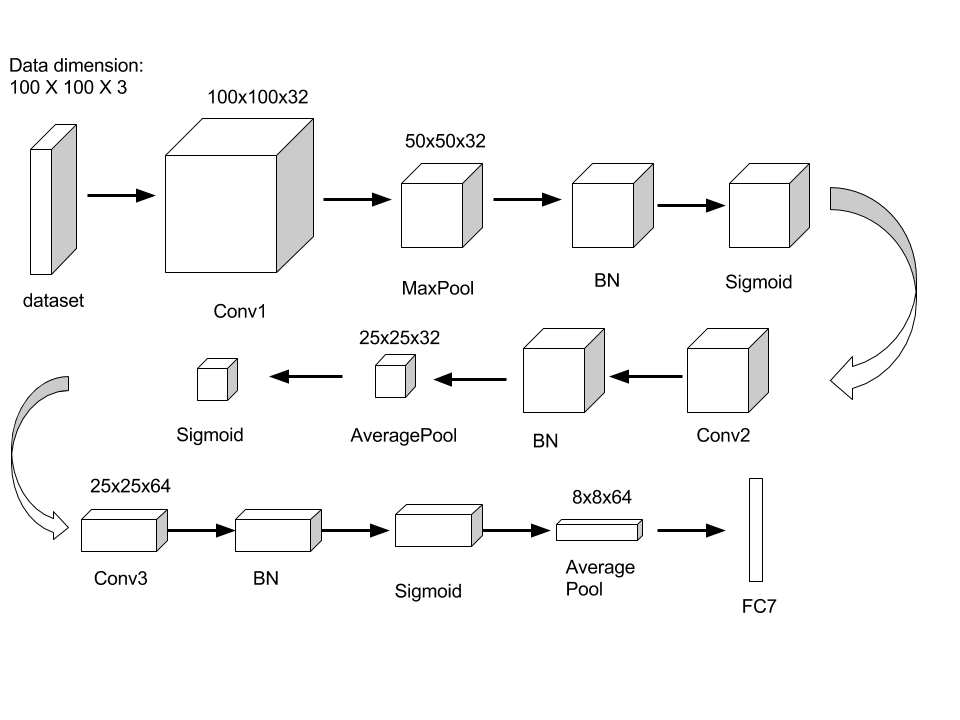
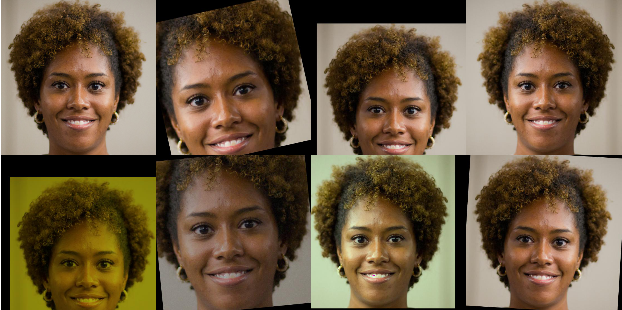
We were inspired by the study done by Dan Duncan, Gautam Shine, and Chris English in which they used a pretrained convolutional neural network to classify facial emotions in real time [1]. While their training accuracy was 90.9% their test accuracy was only 57.1% and they had a relatively slow frame rate of 2.5 frames/second. They suggested that a larger dataset with heavier preprocessing could improve the test accuracy in order to compensate for variability in conditions such as lighting. In order to compensate for the transition between facial expressions Duncan, Shine, and English theorized using running averages of the top classes however they were unable to implement this as a result of their slow frame rate.We have attempted to create a neural network that can work in real time with a higher test accuracy and a faster frame rate.

# Proposed Method

The dataset we used is the Karolinska Directed Emotional Faces (KDEF) which has 4900 images of 70 individuals with 7 emotional expressions (happiness, anger, fear, surprise, disgust, sadness, neutral) from five different angles [2]. We separated out only the images in which the individual is facing the front as the Haar Cascade filter can only detect frontal faces. This left us with 981 images. We then split the data into five parts: four for training and one for testing. Data augmentation was then applied to the images by making seven copies of the originals and randomly applying any of several augmentation methods, including: noise, crop, jitter, rotation (up to 20 degrees in either direction), horizontal flip, contrast, and colour. The final result was a dataset of 7848 images (including the 981 originals), resized to 100x100 pixels.

Our original plan was to use ResNet-50 pre-trained on a multi class dataset. We chose this as it is possible to only have to train the last few layers causing a fast training time. One issue with this model is that it uses far too much RAM to be efficient. Another issue is that the accuracy we were able to get it up to was only 18.6% which is just barely above random guessing and no where near where we wanted the accuracy to be at. After this we decided it might be best to create our own model.

As a proof of concept a two layer convolutional neural network was created using Caffe deep learning framework [4]. We were able to overfit our original data with this model. However, it must be noted that this was the KDEF data before augmentation so it was very uniform and therefore easily exploitable. The testing accuracy achieved after 2000 iterations was 79.125%, much better than the accuracy on ResNet-50. This however was just a simple model so we came up with quite a few ways to improve it.



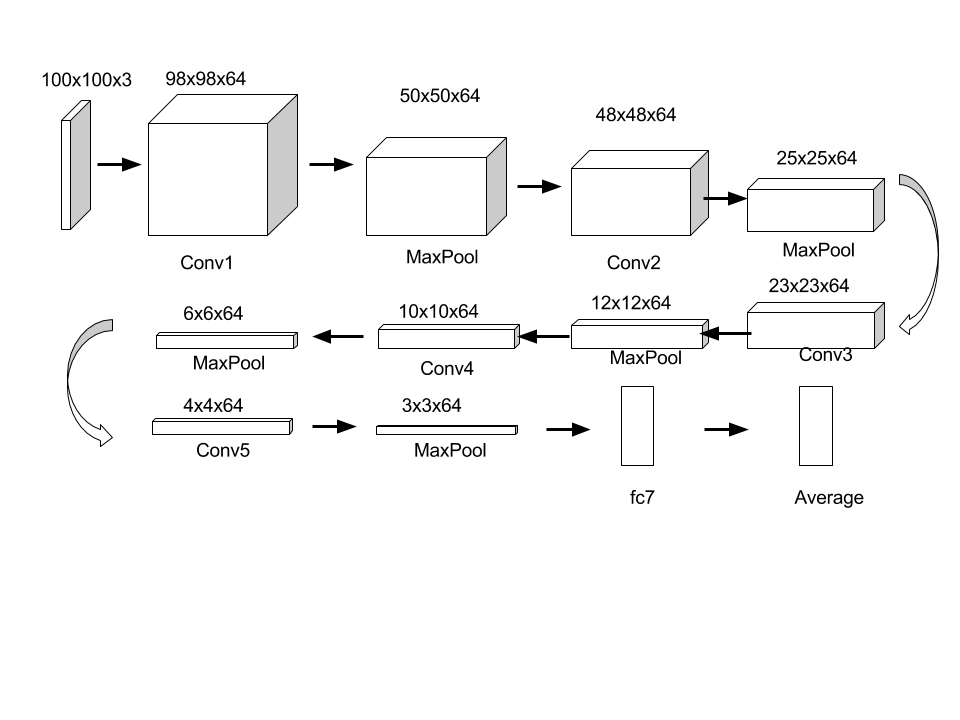


Figure 2(top): Original image (top left) and seven randomly augmented copies of the the original.

Figure 3(middle): The three layer convolutional neural network used in Caffe.

Figure 4(bottom): The 5 layer Darknet CNN

One of the major challenges we had in this project was running the models on our computers. We only had one computer which could run Caffe well so in order to test other techniques more efficient we decided to use another deep learning framework on a different computer. The framework chosen was Darknet [12].

We created a three and five layer convolutional neural network with batch normalization almost exactly the same in both frameworks. The major difference between our two parallel streams was the use of the dataset. When running in darknet the augmented data was used and when running in caffe the original data was used with cropping as the only modification. All models used the OpenCV Haar Cascade bounding box (haarcascade\_frontalface\_default.xml) for the input and achieved a test accuracy of around 70%. Going forward the darknet stream is using transfer learning on the pretrained model with KDEF and JAFFE (both augmented) and the results are promising. In the Caffe stream we are testing different models to try to increase testing and real time accuracy.

# Results and Discussions

The highest test accuracy we have achieved was with a three layer convolutional neural network with batch normalization using Caffe. When this model was trained on the original dataset we were able to obtain a test accuracy of 84% after 5000 iterations at a learning rate of 1e-3 however this model did not perform very well in real time. When the model was trained with the augmented data set we were not able to get the accuracy above 50% after 16800 iterations at a learning rate of 1e-3 to 1e-5 and a batch size of 100. While the test accuracy decreased, the real time webcam performance actually increased and was able to recognize happy, sad, and surprised quite well though it still confuses neutral, angry, and disgusted and it wouldn’t recognize fear. It also runs fairly fast at 5-7 fps on a laptop and 13-15 fps on a desktop. This model is the one visualized on the last page and its accuracies are displayed on the right.

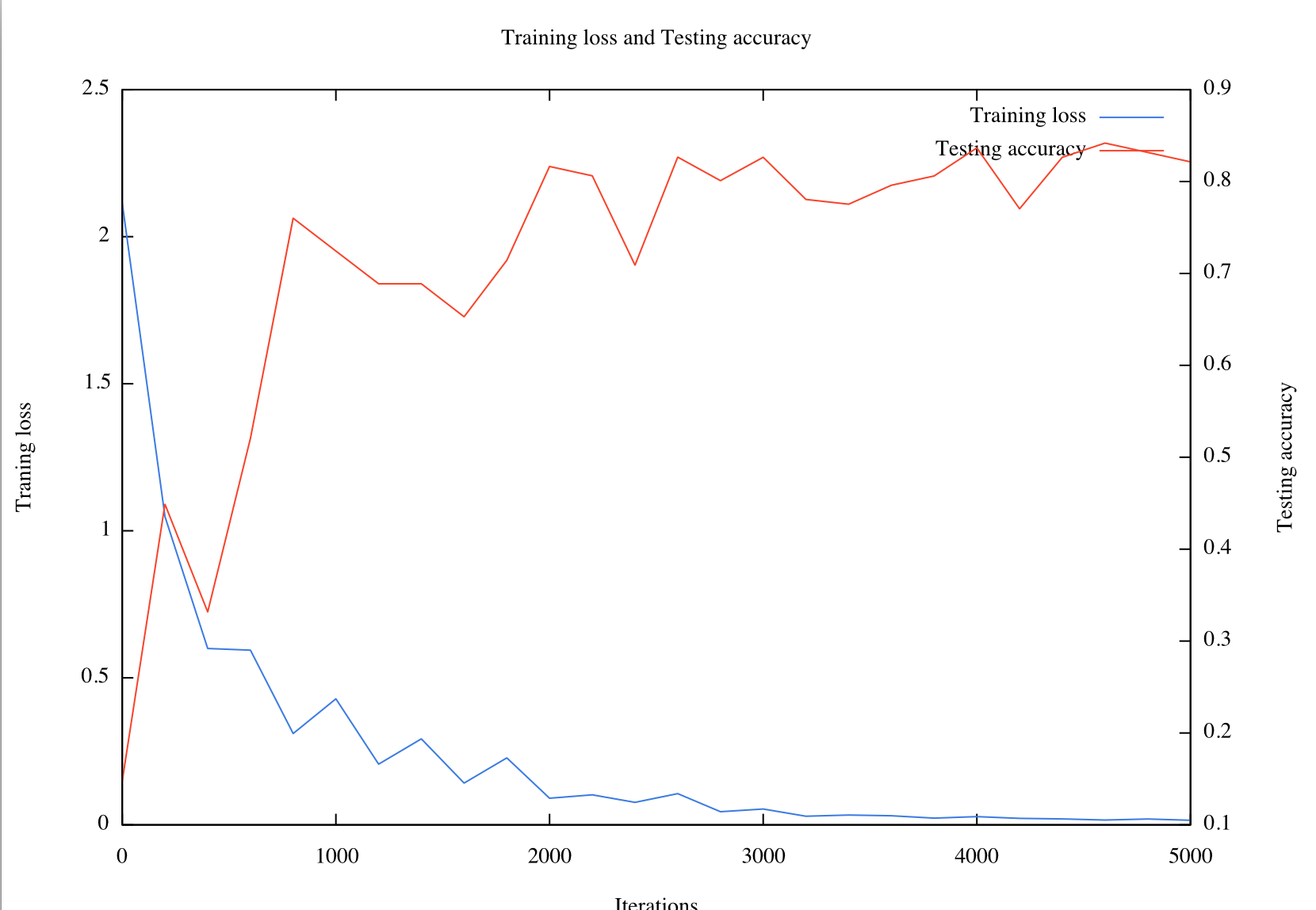
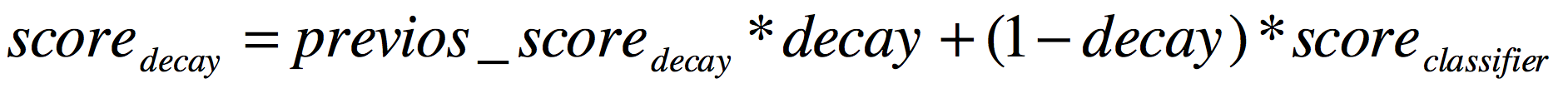
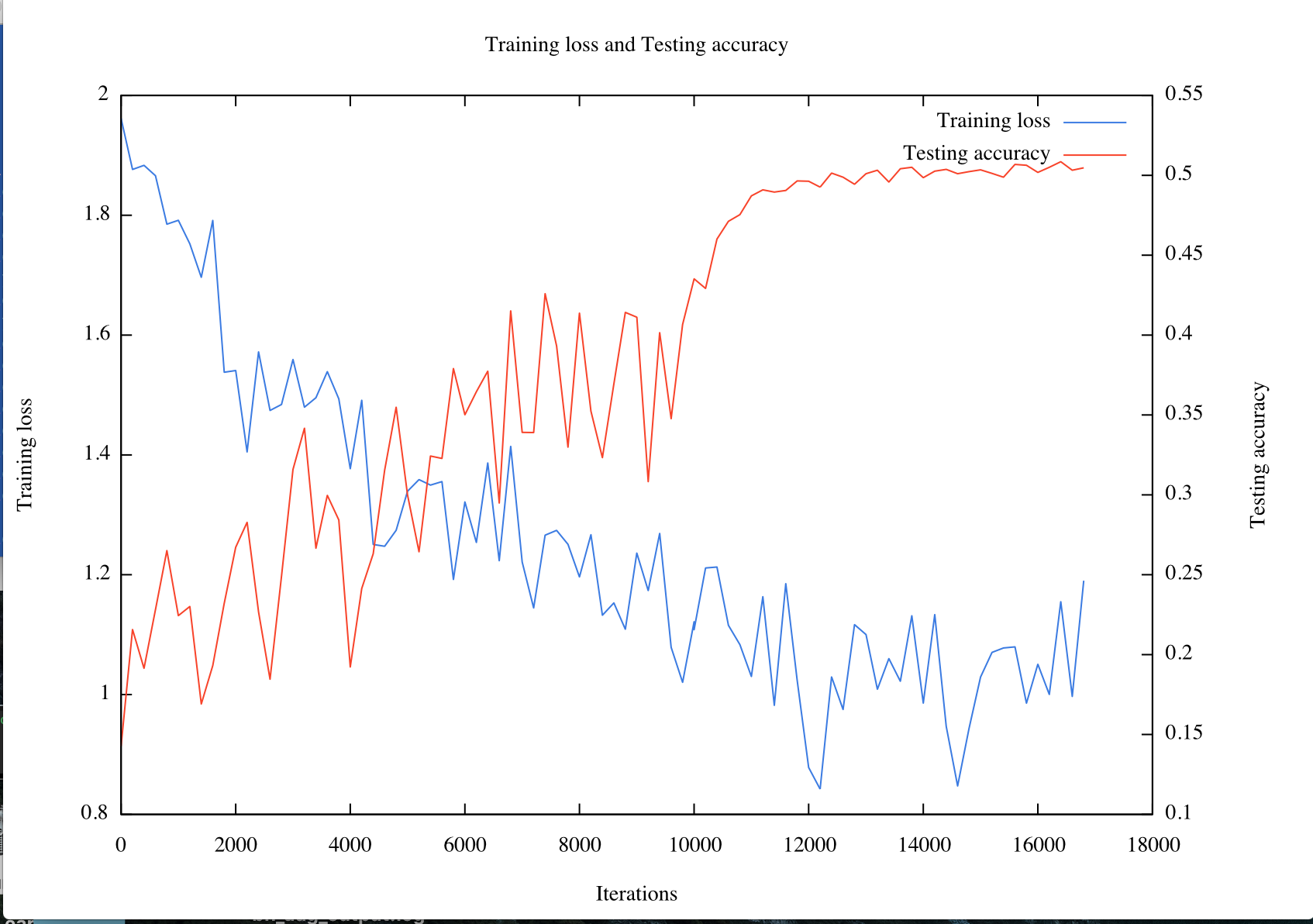
The best running model created on Darknet is a five layer convolutional neural network with batch normalization trained on the augmented KDEF data set. 1280 batches were ran during training with a batch size of 64 and a decaying learning rate of 1e-2. This model achieved a training accuracy of 92.8% and a test accuracy of 69.0% and it ran at around 5 frames per second. Transfer learning was then performed with this trained model with an augmented KDEF and JAFFE data set. This was done in 2000 batches of size 64 and a decaying learning rate of 1e-4. This model reduced the validation accuracy to 68.2% but saw the JAFFE training accuracy rise from 31.3% initially to 55.6%. This also qualitatively improved the real time webcam performance. A proposal by [1] to improve the real time performance was to calculate "running-average of the detected emotions from the input stream [to reduce] the effects of variation and noise". We tested a variation of the hypothesis with a running decay shown below, which resulted in the predictions being more stable and seemingly accurate, although no quantitative data was produced.  Figure 4: Caffe 3 layer model’s training loss and testing accuracy on the original (non-augmented) dataset. Images:981 Learning Rate:1e-3

Figure 5: Caffe 3 layer model’s training loss and testing accuracy on the augmented KDEF dataset. Images:7840 Figure 6: Caffe 3 layer model’s training loss and testing accuracy on the augmented KDEF and JAFFE datasets

We ran into quite a few challenges during the duration of this project. Caffe was the most major of these as there is very little documentation on how to use it so the learning curve is very steep. The easiest way to get Caffe to run was using a virtual machine however virtual machines are limited to the amount of memory they can use so we ran into RAM issues and the webcam was inaccessible when running Caffe this way. While we used multiple computers to run code only one computer could train models efficiently so of course having access to multiple fast computers would have been an asset.

We also ran into the most common problem in affective computing: a small dataset. A larger, more diverse dataset combined with a faster computer would have greatly improved our accuracies. We also noticed that while our test accuracies were fairly high (roughly 70-85%) the performance with the real time input using a webcam suffered. It is interesting to note that our several trained models classified the real time input quite differently. One model would tend towards neutral and another towards disgusted for example, even though these models were trained on the same data set and had similar test accuracies. One issue with real time classification is dealing with faces as they transition between facial expressions. Our models were trained on a dataset of still images that consisted of more exaggerated poses therefore the models performed quite well when classifying other exaggerated expressions however the accuracy suffers when the models are met with transitioning or softer expressions.

##### References

1. **stanford** D. Duncan and G. Shine and C. English, “Facial Emotion Recognition in Real Time,” CS231n Stanford University, Mar. 23, 2016.
2. **kdef** Lundqvist, D., Flykt, A., & Öhman, A. (1998). The Karolinska Directed Emotional Faces - KDEF, CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, ISBN 91-630-7164-9.
3. **nimstim** "Development of the MacBrain Face Stimulus Set was overseen by Nim Tottenham and supported by the John D. and Catherine T. MacArthur Foundation Research Network on Early Experience and Brain Development. Please contact Nim Tottenham at tott0006@tc.umn.edu for more information concerning the stimulus set.”
4. **caffe** Jia, Yangqing and Shelhamer, Evan and Donahue, Jeff and Karayev, Sergey and Long, Jonathan and Girshick, Ross and Guadarrama, Sergio and Darrell, Trevor, “Caffe: Convolutional Architecture for Fast Feature Embedding,” arXiv preprint arXiv:1408.5093, 2014.
5. **ekman** P. Ekman and W. V. Frieden, “Universals and cultural differences in the judgements of facial expressions of emotion,” Journal of Personality and Social Psychology, vol. 53, 1987.
6. **affective computing** R. W. Picard, “Affective Computing,” 1995.
7. M. M. Bradley and M. K. Greenwald and M. C. Petry and P. J. Lang, “Remembering Pictures: Pleasure and Arousal in Memory,” Journal of Experimental Psychology: Learning, Memory, & Cognition. 18:379-390, 1992.
8. **physiological** Lee, Yoo, Park, Kim, Jeong, Lee, “Using Neural Network to Recognize Human Emotions From Heart Rate Variability and Skin Resistance,” Engineering in Medicine and Biology Society, 2005.
9. **face and speech** Zhang, Shiqing, et al. "Multimodal Deep Convolutional Neural Network for Audio-Visual Emotion Recognition." *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*. ACM, 2016.
10. **rnn** Kobayashi, Hiroshi, and Fumio Hara. "Dynamic recognition of basic facial expressions by discrete-time recurrent neural network." *Neural Networks, 1993. IJCNN'93-Nagoya. Proceedings of 1993 International Joint Conference on*. Vol. 1. IEEE, 1993.
11. **cnn vs fc** Nebauer, Claus. "Evaluation of convolutional neural networks for visual recognition." *IEEE Transactions on Neural Networks* 9.4 (1998): 685-696.
12. **darknet** Joseph Redmon, “Darknet: Open Source Neural Networks in C,” <http://pjreddie.com/darknet/>. 2013-2016

**Hours:**

“The page also mentions every group member's contribution in hours for writing the entire report.”

I’m (Taylor) assuming when you say writing the entire report you mean the coding and computation work as well. The four of us put in five days in the library working together for around five to eight hours each. Each group member also put in a considerable amount of individual hours. Work was delegated as working computers were limited. Siqi and Jun focused on the Caffe model, Kevin on the darknet model, and Taylor wrote the report with input from the others. Of course as we worked together we helped each other with the tasks we took upon ourselves to complete. All in all, every group member put in more than 30 hours.

**Signatures:**

