Facial Emotion Recognition in Real Time

Dan Duncan
duncand@stanford.edu

Gautam Shine
gshine@stanford.edu

Chris English
chriseng@stanford.edu

Abstract

We have developed a convolutional neural network for classifying human emotions from dynamic facial expressions in real time. We use transfer learning on the fully-connected layers of an existing convolutional neural network which was pretrained for human emotion classification. A variety of datasets, as well as our own unique image dataset, is used to train the model. An overall training accuracy of 90.7% and test accuracy of 57.1% is achieved. Finally, a live video stream connected to a face detector feeds images to the neural network. The network subsequently classifies an arbitrary number of faces per image simultaneously in real time, wherein appropriate emojis are superimposed over the subjects' faces (https://youtu.be/MDHtzOdnSgA). The results demonstrate the feasibility of implementing neural networks in real time to detect human emotion.

1. Introduction

Emotions often mediate and facilitate interactions among human beings. Thus, understanding emotion often brings context to seemingly bizarre and/or complex social communication. Emotion can be recognized through a variety of means such as voice intonation, body language, and more complex methods such electroencephalography (EEG) [1]. However, the easier, more practical method is to examine facial expressions. There are seven types of human emotions shown to be universally recognizable across different cultures [2]: anger, disgust, fear, happiness, sadness, surprise, contempt. Interestingly, even for complex expressions where a mixture of emotions could be used as descriptors, cross-cultural agreement is still observed [3]. Therefore a utility that detects emotion from facial expressions would be widely applicable. Such an advancement could bring applications in medicine, marketing and entertainment [4].

The task of emotion recognition is particularly difficult for two reasons: 1) There does not exist a large database of training images and 2) classifying emotion can be difficult depending on whether the input image is static or a transition frame into a facial expression. The latter issue is particularly difficult for real-time detection where facial expressions vary dynamically.

Figure 1. a) Input image to the convolutional neural network b) Output of the application, with emojis superimposed over the subjects' faces to indicate the detected emotion

Most applications of emotion recognition examine static images of facial expressions. We investigate the application of convolutional neural networks (CNNs) to emotion recognition in real time with a video input stream. Given the computational requirements and complexity of a CNN, optimizing a network for efficient computation for frame-by-frame classification is necessary. In addition, accounting for variations in lighting and subject position in a non-laboratory environment is challenging. We have developed a system for detecting human emotions in different scenes, angles, and lighting conditions in real-time. The result is a novel application where an emotion-indicating emoji is superimposed over the subjects' faces, as shown in Fig. 1.

2. Background/Related Work

Over the last two decades, researchers have significantly advanced human facial emotion recognition with computer vision techniques. Historically, there have been many approaches to this problem, including using pyramid histograms of gradients (PHOG) [5], AU aware facial features [6], boosted LBP descriptors [7], and RNNs [8]. However,

recent top submissions [9], [10] to the 2015 Emotions in the Wild (EmotiW 2015) contest for static images all used deep convolutional neural networks (CNNs), generating up to 62% test accuracy.

A recent development by G. Levi et. al [11] showed significant improvement in facial emotion recognition using a CNN. The authors addressed two salient problems: 1) a small amount of data available for training deep CNNs and 2) appearance variation usually caused by variations in illumination. They used Local Binary Patterns (LBP) to transform the images to an illumination invariant, 3D space that could serve as an input to a CNN. This special data preprocessing was applied to various publicly available models such as VGG_S [12]. The model was then re-trained on the large CASIA WebFace data-set [13] and transfer-learned on the Static Facial Expressions in the Wild (SFEW) dataset, which is a smaller database of labeled facial emotions released for the EmotiW 2015 challenge [14]. Final results showed a test accuracy up to 54.56%, an improvement of 15% over baseline scores. Figure 2 visualizes the first convolutional layer of VGG_S, revealing the different kernels optimized for feature detection. Since this modified VGG_S neural network is pre-trained for facial recognition and freely available, we chose to use VGG_S as a starting point in developing our own model.

A notable implementation of a CNN to real-time detection of emotions from facial expressions is by S. Oullet [15]. The author implemented a game, where a CNN was applied to an input video stream to capture the subject's facial expressions, acting as a control for the game. This work demonstrated the feasibility of implementing a CNN in real-time by using a running-average of the detected emotions from the input stream, reducing the effects of variation and noise.

3. Approach

To develop a working model, we use two different freely-available datasets: the extended Cohn-Kanade dataset (CK+) [16, 17] and 2) the Japanese Female Facial Expression (JAFFE) database [18]. The CK+ dataset, although small, provides well-defined facial expressions in a controlled laboratory environment. The JAFFE database provides additional images with more subtle facial expressions with laboratory conditions. We also uniquely developed our own new (home-brewed) database that consists of images from five individuals. Many images were recorded for each of the seven primary emotions (anger, disgust, fear, happy, neutral, sad, surprise) from each subject. We subsequently applied jitter to these images to account for variations in lighting and subject position in the final implementation.

Initially we directly implemented the VGG_S network from ref. [11] for image classification. We were unable to obtain similar results and at best could obtain a test accuracy of 24% on the CK+ dataset, and 14% on the JAFFE dataset. There were incorrect classifications for images that appeared to clearly convey a particular emotion, such as that in Figure 3a. We pre-processed the data by subtracting the mean image (Figure 3b), but the results showed no improvement. The resulting confusion matrices for the JAFFE and CK+ datasets are shown in Figures 4 and 5, respectively. Many different facial expressions were incorrectly classified as 'fear' by VGG_S for both datasets.

It is not clear why the accuracy was lower from that reported in ref [11]. One issue is that the facial expressions in the JAFFE dataset are quite subtle, exacerbating the ability to differentiate emotions. Another issue is that there are few images labeled with 'fear' and 'disgust' in both the JAFFE

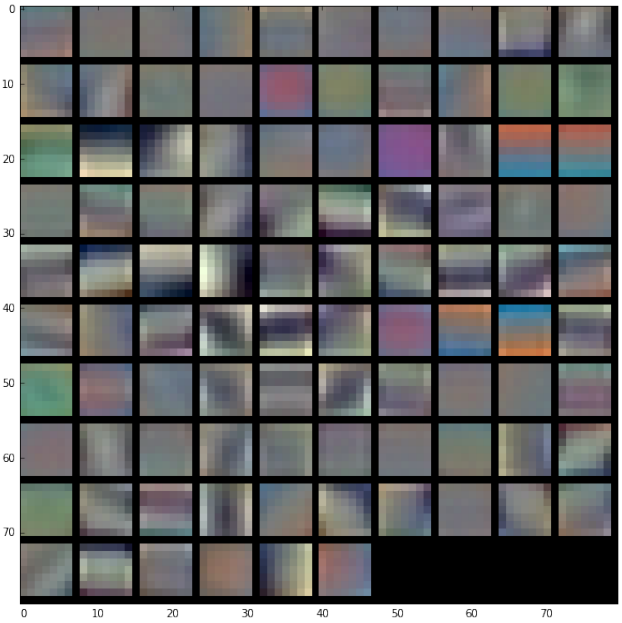

Figure 2. Visualization of the filters used in the first convolutional layer of VGG_S net. See section 2 for discussion.

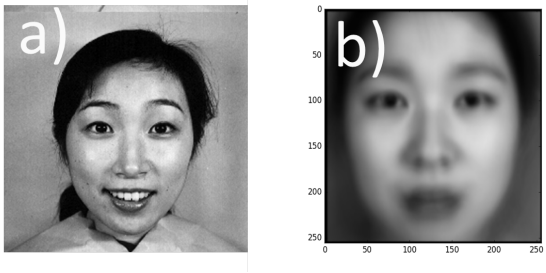

Figure 3. a) Example image from the JAFFE dataset that was tested on VGG_S. The image was labeled as 'happy' and predicted as 'angry'. b) Mean image of the JAFFE dataset. See section 3 for discussion

and CK+ datasets, making it difficult to train the network to recognize these two emotions correctly. Also, the authors in ref. [11] may have included some data pre-processing they neglected to describe in their paper.

To improve the classification accuracy of VGG_S, we applied transfer learning to the network on the JAFFE and CK+ datasets, as well as our own dataset. For our own 'home-brewed' dataset, we recorded images from 5 different individuals. Multiple images were recorded for each of the seven primary emotions (anger, disgust, fear, happy, neutral, sad, surprise) from each subject, resulting in a total of 2118 labeled images. We omitted the emotions 'disgust' and 'contempt', since they are very difficult to classify. In any implementation of a CNN in a real environment, effects that are usually omitted in a laboratory must be accounted for. These include variations in lighting, distance from the camera, incorrect face cropping, and variations in orientation of the subjects face. To account for some of these issues, we implemented randomized jitter in the home-brewed dataset. The jitter was applied by randomly changing both cropping (10% variation) and brightness (20% variation) in the input images, as demonstrated in Figure 7. By re-examining the same images with different variations in cropping and lighting, VGG_S could learn to account for these effects.

Due to time constraints, we only trained the last few fully-connected layers of the network (fc6, fc7, fc8). The architecture (see Figure 6) of VGG_S consists of five convolutional layers, three fully-connected layers, followed by a softmax classifier, for a total of approximately $7x10^6$ convolutional parameters and $7x10^7$ fully-connected parameters. We also applied a Haar-Cascade filter (see Figure 6) provided by OpenCV to crop the input image faces, which significantly improved test and training accuracy. The loss versus training iteration is shown in Figure 8. A small batch size was required for training due to lengthy computation times and is the cause of the 'jagged' profile of the curve. Convergence to low loss appears to occur after 100 iterations.

4. Experiment

Once a newly trained version of VGG_S was obtained, we connected a video stream to the network using a standard webcam. The run-time for image cropping using the face-detector was 150 ms and that for a forward pass in VGG_S was 200 ms. These operations limited the frame-rate of our emotion-recognition algorithm to 2.5 frames/second, sufficient for a real-time demonstration. Note that for any number N of subjects in the camera's view, the run-time for a single frame would be increased to 150 ms + N x 200 ms to account for all persons in the frame. Our network was developed on a laptop that had insufficient GPU capabilities (for VGG_S) to utilize CUDNN, a GPU-accelerated library provided by Nvidia for deep neural networks. Thus, other machines with optimized graphics capabilities, such as the Jetson TX1, could implement our solution with substantially reduced run-time. We tried to implement VGG_S on the Jetson TK1, but found that the

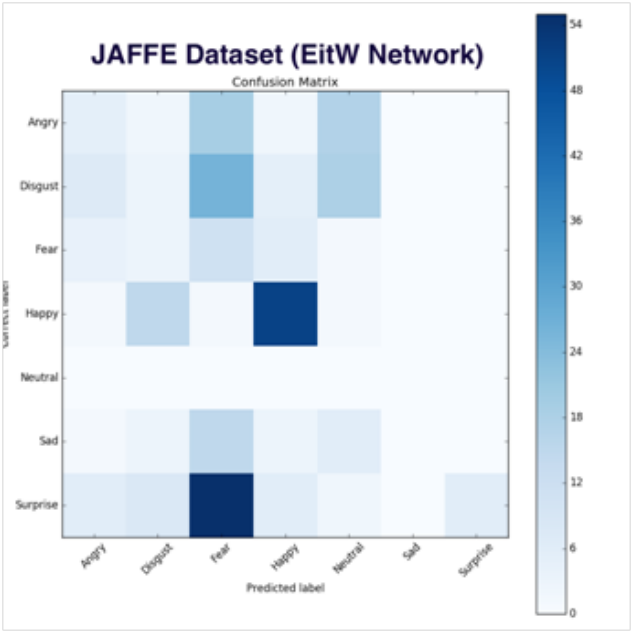

Figure 4. Confusion matrix for the unaltered VGG_S net on the JAFFE dataset, with a 14% accuracy.

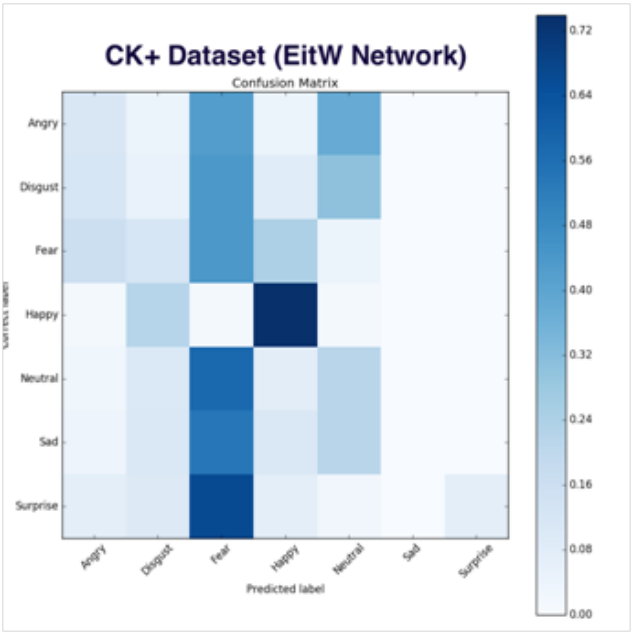

Figure 5. Confusion matrix for the unaltered VGG_S net on the CK+ dataset, with a 24% accuracy.

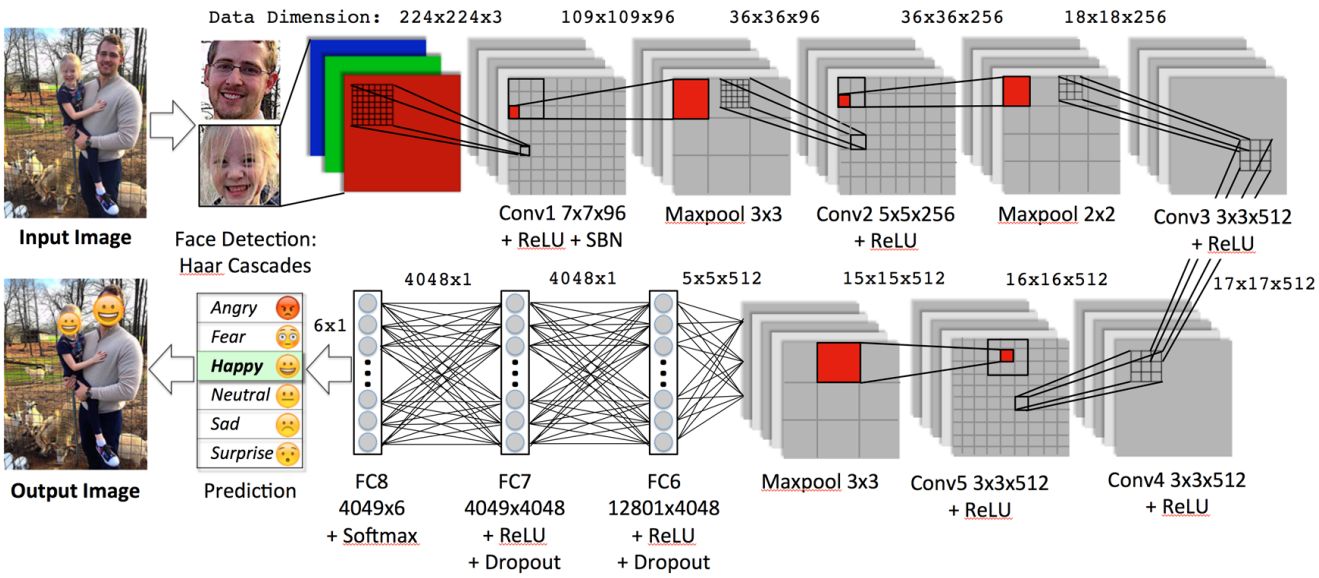

Figure 6. Architecture of the convolutional neural network used in this project. The input image and the cropped faces using a Haar-Cascade detector are also shown. The resulting classification from the network chooses a particular emoji to apply over the subjects' faces.

run-time memory requirements (> 2 GB) were too large for the board to process without crashing.

It is interesting to examine the second fully-connected layer's neurons' responses to different facial expressions. We use the cosine similarity, defined as $s = A \cdot B / |A||B|$, to measure the similarity between two outputs A and B from the second layer (fc7). Figure 9 shows the output of the second fully-connected layer for two different input images displaying the emotion 'happy'. The cosine similarity between each of the layers in the output is quite high, revealing an average similarity of $s = 0.94$. However, if one examines the cosine similarity for the same subjects but different

Figure 7. Effect of jitter applied to the input image. Each image results from changing the offset in the face cropping and the brightness.

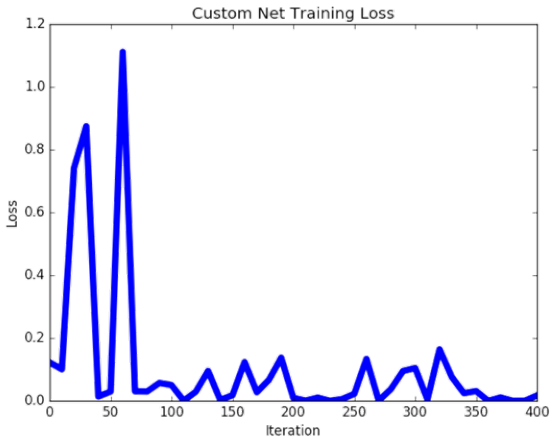

Figure 8. Training loss versus iteration for the VGG_S convolutional neural network. Only the fc6-fc8 layers were adjusted in training.

different emotions (see Figure 10), the resulting similarity is only $s = 0.04$. This large difference in similarity demonstrates that the layers in the network have "learned" to differentiate emotions, regardless of the individual in the input image. They accomplish this by highlighting the salient features of the corresponding emotion necessary for classification.

Figures 11-13 show the confusion matrices for our custom VGG_S network trained on the JAFFE, CK+, and home-brewed datasets, respectively. The worst performance is obtained on the JAFFE dataset, with a training

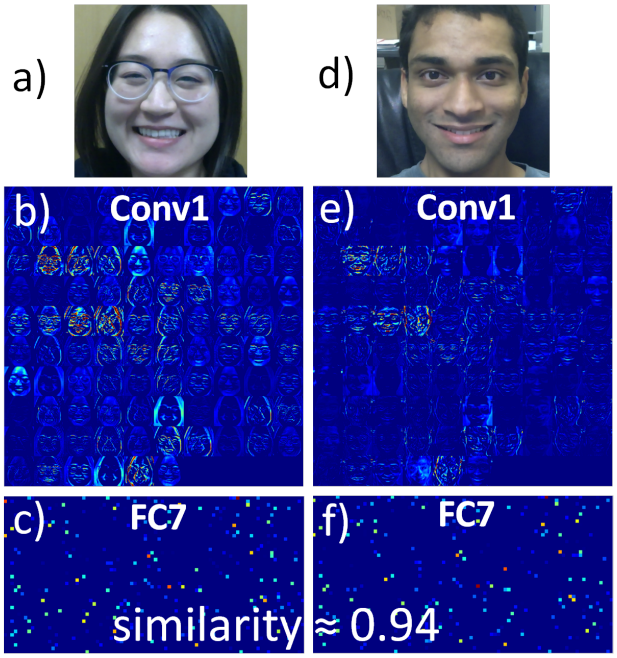

Figure 9. (a-c): input image, output of second fully-connected layer, and cosine similarity for 'happy' emotion. (d-f) same sequence of figures for a different subject showing the same emotion.

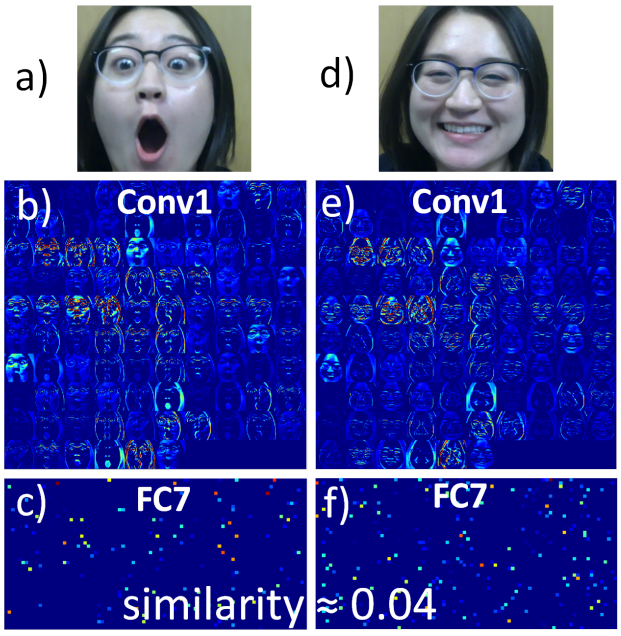

Figure 10. (a-c): input image, output of second fully-connected layer, and cosine similarity for 'surprise' emotion. (d-f) same sequence of figures for the same subject showing the 'happy' emotion

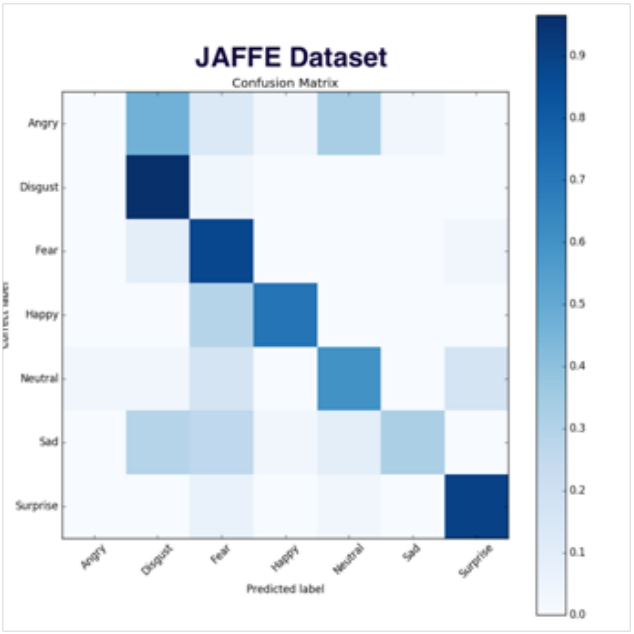

Figure 11. Confusion matrix during training for the custom VGG_S network on the JAFFE dataset.

accuracy of 62%. Here, the differences between different emotions among the subjects is very subtle, particularly between the emotions 'disgust' and 'fear'. Also, these images are in grey-scale format, whereas our VGG_S network is optimized for RGB images. The lack of RGB format can sometimes exacerbate the ability of the network to distinguish between important features and background elements. The CK+ dataset shows much improved performance (90.7% training accuracy), with some incorrect classification for angry, disgust, neutral and sad emotions due to the low number of available labeled images. Figure 13 shows the confusion matrix for our home-brewed dataset. Here, the overall training accuracy is as high as 90.9%. VGG_S excelled with classifying the home-brewed dataset, which consisted primarily of exaggerated facial expressions. This dataset was effective for the implementation of our demonstration, which typically involved exaggerated expressions from subjects. Note that, with the pre-trained 'off-the-shelf' VGG_S network from ref. [11], we could only obtain training accuracies of 14% and 24% on the JAFFE and CK+ datasets, respectively. Overall, we obtained 57% test accuracy on the remaining images in our own dataset.

5. Conclusions

The goal of this project was to implement real-time facial emotion recognition. Using our custom trained VGG_S network with a face-detector provided by OpenCV, we successfully implemented an application wherein an emoji indicat-

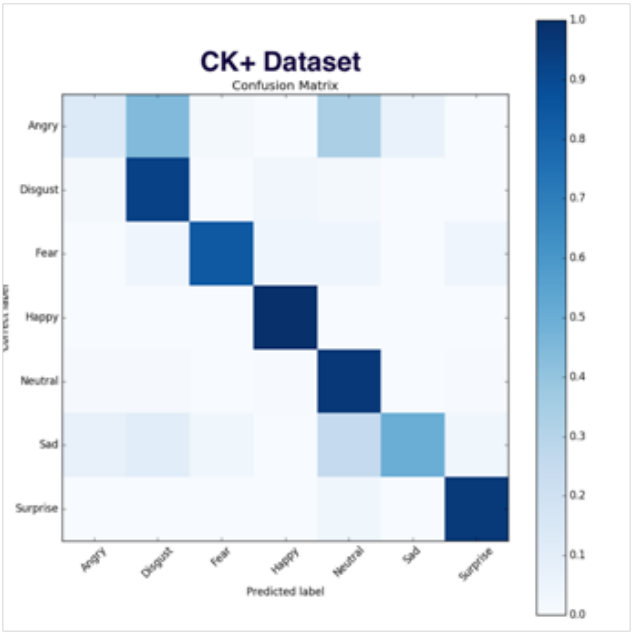

Figure 12. Confusion matrix during training for the custom VGG_S network on the CK+ dataset.

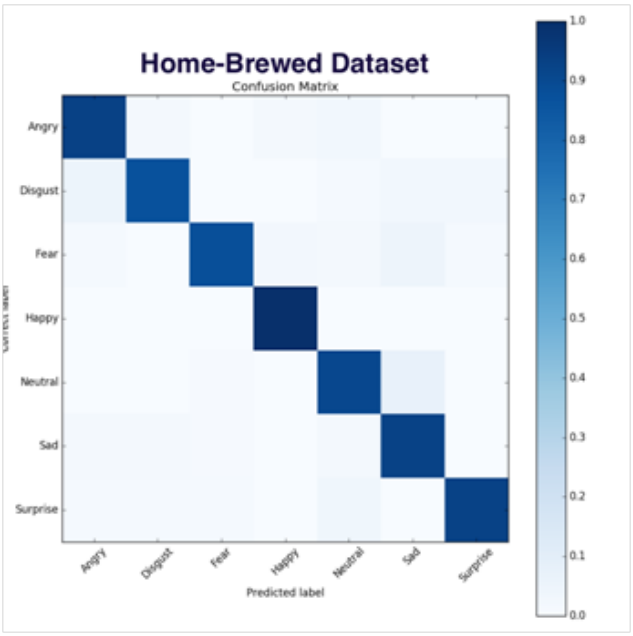

Figure 13. Confusion matrix during training for the custom VGG_S network on out homebrew dataset.

ing one of six expressions (anger, fear, neutral, happy, sad, surpirse) is superimposed over a user's face in real time (see https://youtu.be/MDHtzOdnSqA for a demonstration). Some of the results are shown in Figure 14.

While we achieved a successful implementation, significant improvements can be made by addressing several key

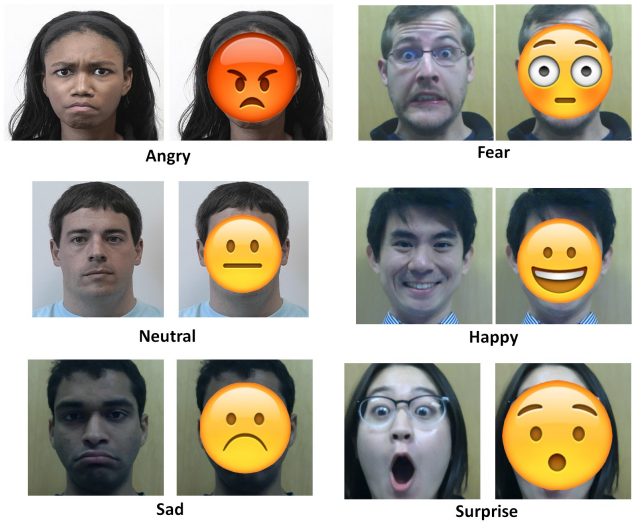

Figure 14. Samples of the video demonstration. Each pair of images indicates a pair of images. The emotion detected by the CNN is indicated by the type of emoji superimposed over the subject's face.

issues. First, a much larger dataset should be designed to improve the model's generality. While we achieved $> 90\%$ accuracy in laboratory conditions (perfect lighting, camera at eye level, subject facing camera with an exaggerated expression), any deviation from that caused the accuracy to fall significantly. In particular, any shadow on a subject's face would cause an incorrect classification of 'angry'. In addition, the webcam needs to be level to the subjects' faces for accurate classification. Augmenting our home-brewed dataset to include off-center faces could have addressed this problem. Heavier pre-processing of the data would have certainly improved test-time accuracy. For example, adjusting the brightness to the same level on all the images might have removed the requirement for providing jittered input images. Also, fully training a network other than VGG_S might yield substantial improvements in computation speed, since VGG_S is relatively slow.

As mentioned previously, a particularly difficult aspect of real-time recognition is deciding how to classify transition frames from neutral to fully formed expressions of emotion. One viable solution is to use a running average of the top classes reported by each frame which would ameliorate the problem of noise/errors caused by dynamic expressions [15]. Unfortunately, the relatively slow frame-rate of our demonstration made this solution untenable. Regardless, our implementation appeared to classify a subject's emotion reliably. Future implementations that run at higher frame-rates would require a running average.

Our project was implemented under Python 2.7 and the source code can be downloaded from a github repository [19].

References

[1] P. Abhang, S. Rao, B. W. Gawali, and P. Rokade, "Article: Emotion recognition using speech and eeg signal a review," *International Journal of Computer Applications*, vol. 15, pp. 37–40, February 2011. Full text available.

[2] P. Ekman, *Universals and cultural differences in facial expressions of emotion*. Nebraska, USA: Lincoln University of Nebraska Press, 1971.

[3] P. Ekman and W. V. Friesen, "Universals and cultural differences in the judgements of facial expressions of emotion," *Journal of Personality and Social Psychology*, vol. 53, 1987.

[4] A. Kołakowska, A. Landowska, M. Szwoch, W. Szwoch, and M. R. Wróbel, *Human-Computer Systems Interaction: Backgrounds and Applications 3*, ch. Emotion Recognition and Its Applications, pp. 51–62. Cham: Springer International Publishing, 2014.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *NIPS*, vol. 1, p. 4, 2012.

[6] A. Yao, J. Shao, N. Ma, and Y. Chen, "Capturing au-aware facial features and their latent relations for emotion recognition in the wild," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, (New York, NY, USA), pp. 451–458, ACM, 2015.

[7] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," *Image and Vision Computing*, vol. 27, no. 6, pp. 803 – 816, 2009.

[8] S. Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal, "Recurrent neural networks for emotion recognition in video," *ICMI*, pp. 467–474, 2015.

[9] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, (New York, NY, USA), pp. 435–442, ACM, 2015.

[10] B. Kim, J. Roh, S. Dong, and S. Lee, "Hierarchical committee of deep convolutional neural networks for robust facial expression recognition," *Journal on Multimodal User Interfaces*, pp. 1–17, 2016.

[11] G. Levi and T. Hassner, "Emotion recognition in the wild via convolutional neural networks and mapped binary patterns," in *Proc. ACM International Conference on Multimodal Interaction (ICMI)*, November 2015.

[12] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014.

[13] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *CoRR*, vol. abs/1411.7923, 2014.

[14] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 2106–2112, Nov 2011.

[15] S. Ouellet, "Real-time emotion recognition for gaming using deep convolutional network features," *CoRR*, vol. abs/1408.3750, 2014.

[16] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. 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*, pp. 94–101, June 2010.

[17] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 46–53, 2000.

[18] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 200–205, Apr 1998.

[19] https://github.com/GautamShine/emotion-conv net