Real-time Emotion Recognition From Facial Expressions

Minh-An Quinn, Grant Sivesind, Guilherme Reis

CS 229 - Stanford University

{minhan, gsivesin, greis}@stanford.edu

ABSTRACT

We built several models capable of recognizing seven basic emotions (happy, sad, angry, afraid, surprise, disgust, and neutral) from facial expressions. Using the FER-2013 dataset of labeled headshots, we achieve 45.95% test accuracy using an SVM and 66.67% using a CNN; on the CK+ dataset, we achieve 98.4% accuracy. We then transferred the skills learned on static images into a real-time emotion recognition system, which continuously detects faces from a video feed and classifies the predominant emotion of the individual. Though the real-time system can reliably classify some of the seven emotions, more work is necessary to build a robust real-time system that performs outside of laboratory conditions.

I. INTRODUCTION

We tackled the problem of recognizing the emotion of a person from an image of their facial expression. First, we built models capable of recognizing seven emotions (happy, sad, angry, afraid, surprise, disgust, and neutral). Given static, cropped headshots, our model would output a probability distribution over emotions of the pictured individual. Next, we transferred the skills learned on static datasets to implement a real-time emotion classifier. Using a webcam video feed, we built a system to continuously detect faces, extract, crop, and grayscale the face region, and classify the emotion of the person.

II. DATASET AND FEATURES

We used two main datasets to train our models: FER-2013 and CK+ (extended Cohn-Kanade).

The FER-2013 dataset consists of 28,000 labeled images in the training set, 3,500 labeled images in the development set, and 3,500 images in the test set. Each image in FER-2013 is labeled as one of seven emotions: happy, sad, angry, afraid, surprise, disgust, and neutral, with happy being the most prevalent emotion, providing a baseline for random guessing of 24.4%. The images in FER-2013 consist of both posed and unposed headshots, which are in grayscale and 48x48 pixels. The FER-2013 dataset was created by gathering the results of a Google image search of each emotion and synonyms of the emotions.

The CK+ dataset has a total of 5,876 labeled images of 123 individuals. Out of these images, we used 4,113 images for training, 881 for dev, and 881 for test. Each image is labeled with one of seven emotions: happy, sad, angry, afraid, surprise, disgust, and contempt. Images in the CK+ dataset are all posed with similar backgrounds, mostly grayscale, and 640x490 pixels.

Though the FER-2013 and CK+ database both have similarly labeled emotions, we found when developing our model that it was very easy to achieve extremely high accuracies on the CK+ dataset (as opposed to FER-2013). This is most likely because the CK+ dataset was posed, had fewer individuals and less diversity than FER-2013. Therefore, we focused more on improving our model's performance on the FER-2013 dataset, as the wider range of unposed images more closely reflected the images we would see when transferring the skills over to real-time.

III. RELATED WORK

A. STATIC IMAGES

There has been a decent amount of work done on emotional recognition in static images so we were able to draw on a few different papers for information on best practices and ways to improve our model. First, there were two projects from previous offerings of CS

229 that helped us start to formulate what our project would be and gave us a starting point for emotion recognition. [3][10] These two papers did a good job of laying the groundwork for our project by delving deeply into what advantages and disadvantages SVMs and CNNs bring to emotion recognition. Additionally, the main dataset we used, FER-2013 was created as a part of a larger project that included a competition for Kaggle.com. [2] We were able to look to that project to understand the dataset and what sort of results we could hope to achieve. In particular, the winner of the competition achieved a test accuracy of 71% by cleverly implementing a CNN with an output layer that fed into a linear SVM. [14]

We also explored previous work on the CK+ dataset to validate our high accuracy on the dataset. We found that the state of the art SVM model achieves an accuracy of 99.7% on CK+ and concluded that our accuracy of 98.4% was therefore reasonable. [8]

IV. Methods/Experiments

A. SVM

The first model that we tried was sklearn's one vs. one (OVO) SVM with an rbf kernel. [11] To establish a baseline, we first used raw, grayscale pixel values as the features for the SVM. With this combination, we achieved an accuracy that barely surpassed the method of choosing the most common emotion every time. We could only train on the first 5,000 training examples using this method as training the model took too long.

Next, we tried scaling the pixels so that each image had a mean pixel value of 0 and a variance of 1 and used the scaled pixel values as our new features. This improved our accuracy to close to 40%. However, we still had issues training the model within a reasonable time period, which stopped us from using more than the first 5,000 training examples.

We then used Principal Component Analysis (PCA) to attempt to isolate the most important components for our analysis. Reducing the dimensionality of the images allowed us to use the full training set. We experimented with different numbers of components, ranging from 10 to 250. We found that the best accuracy was with 25 components, where we were able to reach a test accuracy of 43%.

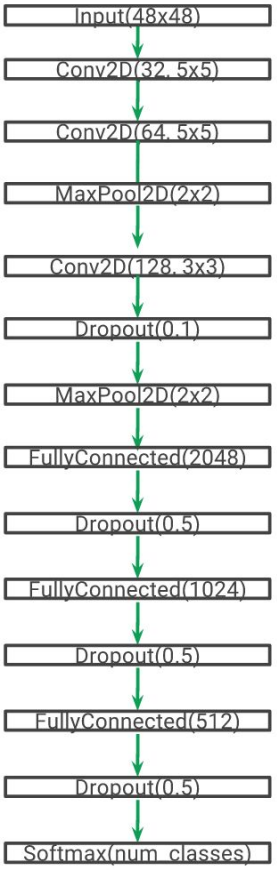
We then read a paper that argued for the use of one vs. all (OVA) SVM's in emotion recognition and decided to use sklearn's linear kernel SVM. [5] [11] We repeated the process of trying raw pixels and scaled pixels as features and PCA to limit the top components but were only able to achieve a maximum dev set accuracy of 34.77 %

We finally tried using Histogram of Oriented Gradients (HOG) to describe the distribution of gradients and edge directions in the images before processing them. The idea behind using HOG is that different emotions would have different and distinct gradients, particularly around the mouth and eye areas. While HOG did not significantly help with the accuracy of the OVO SVM, it bumped the accuracy of the OVA classifier up to our highest SVM accuracy of 45.95%

B. CNN

We evaluated both a variety of preprocessing techniques as well as several model architectures, ultimately developing a custom CNN model capable of attaining near-state-of-the-art accuracy of 66.67% on the FER-2013 test set.

For preprocessing, we experimented with centering (i.e., subtracting mean) and scaling data. We found it generally helpful to subtract the mean found in the train distribution from all sets before training/evaluating. While scaling was helpful with SVMs, it did not help in a CNN model. We also implemented data augmentation: we randomly rotate, shift, flip, crop, and sheer our training images. This yielded about a 10 p.p. increase in accuracies.



We implemented several CNN architectures from papers applying emotion recognition to these and other datasets. Ultimately, what yielded the best performance was our custom developed CNN architecture (left).

Analyzing error in neural networks is infamously difficult. We analyzed our error across different classes, as well as by visual inspection of images we classified

correctly and incorrectly. One early observation was that we fail much more at certain emotions (see confusion matrix), and that we were failing to classify images where it was necessary to rely on fine details in the images (e.g., small facial features or curves). Due to this, we increased the number of layers and decreased filter sizes to increase the number of parameters in our network, which had a clear effect in allowing us to fit the dataset better. This led to some overfitting, which we addressed by using dropout, early stopping around 100 epochs, and augmenting our training set. Given this, we only start learning training set noise after achieving approx. 60% dev set accuracy; this is clear from plotting accuracy during training. This leaves us with some suggestions for future work, which largely focus on enabling increased parameterization of the network.

V. Results

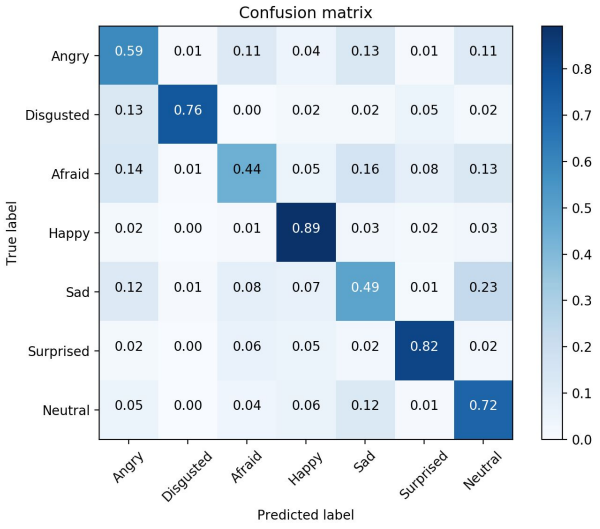
A. Results on FER-2013 Dataset

Model	Featurization/ Hyperparameters	Training Accuracy	Dev Accuracy	Test Accuracy [1]
SVM (OVO)[2]	Scaled pixels	43.36%	38.61%	40.17%
SVM (OVO)	Scaled pixels PCA - 25 comps	56.70%	43.74%	43.18%
Linear SVM (OVA)	HOG (4,4) pixels/cell	61.35%	44.99%	45.95%
CNN	Dachapally[4] [3]	53.88%	52.57%	52.38%
CNN	DeXpression [1] [4]	72.25%	63.86%	61.63%
CNN	Custom model	90.11%	67.68%	**66.67%**

[1] This was a blind, holdout set. Results were retroactively computed for this set for benchmarking purposes only, never for model tuning.
[2] Due to computational limits, only trained on a subset of training data.
[3] The original paper does test on our datasets. Not subtracting mean worsened results.
[4] No mean subtraction; augmentation; extra dense (2048, 1024) and dropout (0.5, 0.25) layers before final softmax. Mean subtraction produced inferior results. Training for 200 epochs with Adadelta. Batch normalization was replaced with reasonable values of dropout.. Denser layer added to allow model to fit more closely. Interestingly, adding denser layers only improved 3-4 p.p of accuracy.

Looking at our training accuracy vs. test accuracy, our relatively low variance shows that we avoid overfitting as long as we stop around 100 epochs. Training beyond that consistently results in overfitting. Despite that, our model suffers from high bias; increasing the parameters in our network did not lead to more overfitting, so long as it was counteracted by adding dropout.

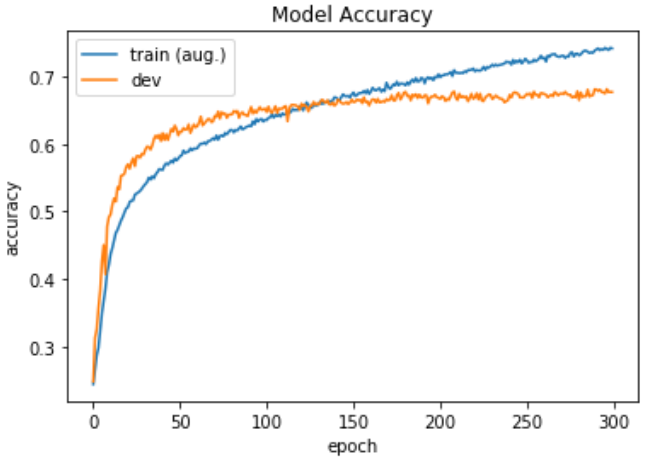
B. Confusion Matrix



*Confusion Matrix from FER-2013, Custom CNN
Generated with visualization code from sklearn [11]

Our precision is very high for Happy, Surprised, Neutral, and Angry. These are consistent with our qualitative observations using our real-time demo. Both precision and recall vary greatly by class. Precision is low for classes like Afraid; recall is high for classes like Happy. From visual inspection of the dataset, we notice that some emotion pairs -- Surprised and Afraid, or Disgusted and Surprised -- are quite similar, fooling even human subjects.

D. Model Accuracy Over Training Period



*Training on FER-2013, Custom CNN

E. Results on CK+ Dataset

We used a linear SVM with scaled pixel values to achieve a training accuracy of 99.59%, and a test accuracy of 98.47% on the CK+ dataset.

F. Real-Time Classification

We used OpenCV's Haar cascades to detect and extract a face region from a webcam video feed, then classified it using our CNN model. We found it best to neither subtract the training mean nor normalize the pixels in the detected face region before classifying it. Real-time classification better exposed our model's strengths: neutral, happy, surprised, and angry were generally well-detected. Illumination was a very important factor in the model's performance. This suggests that out training set may not truthfully represent the distribution of lighting conditions.

VII. Discussion

On the CK+ dataset, with a small number of models in posed, centered photos, and with good lighting, we achieve a test accuracy of 98.4%. We used a linear SVM to achieve this accuracy and while it is very high, it is in line with what other papers suggest achieve. [8] Despite this high accuracy, we decided to focus our efforts on the FER-2013 dataset as we believe it to more accurately reflect real-time conditions due to its automatically captured, non-posed photos.

On the FER-2013 dataset, we achieve a test accuracy of 66.67% with our best CNN. We are satisfied with this accuracy as the top score on the Kaggle competition using this dataset achieved an accuracy of 71%. Furthermore, human scores on the FER-2013 are accuracies of 65% +/- 5% further showing that our model is very accurate on static images. [2]

Due to the nature of real-time classification, it is hard to get a definitive metric of our real-time system's accuracy. Our system reliably classified some emotions (the most reliable classification being happy), but struggled when lighting conditions changed, or backgrounds were noisy. Though we tried training on FER-2013 to better reflect real-time data, conditions like lighting in real-time differed from static images, making it hard for our program to transfer over skills learned on static databases.

It is worth mentioning that when we displayed the top two most likely classifications during real time recognition, our performance increased significantly. This indicates that one challenge that real time emotion recognition faces is that the features of different emotions are very similar, and more analysis on ways to separate similar emotions may be warranted.

VIII. Future Work/Conclusion

For continued work on this project, we believe there are two major areas of focus that would improve our real-time emotion recognition system. First, we suggest fine tuning the architecture of the CNN used for the model to fit perfectly with the problem at hand. Some examples of this fine tuning include finding and removing redundant parameters, adding new parameters in more useful places in the CNN's structure, adjusting the learning rate decay schedule, adapting the location and probability of dropout and experimenting to find ideal stride sizes.

A second area of focus lies in adapting the datasets to more closely reflect real-time recognition conditions. For example, simulating low light conditions and "noisy" image backgrounds, could help the model become more accurate in real-time recognition. Additionally making sure that the distribution of models in the training dataset accurately reflects the distribution of subjects that the system will see when running in real-time. We noticed that our real-time recognition demo tended to perform better on caucasian males than it did on other races or genders. We believe this is in part due to a skew in the training data towards white males and a more balanced dataset might be able to correct this skew.

Overall, our models achieve state-of-the-art results on CK+, and are within 3 p.p. of doing so on FER-2013 with a much simpler CNN architecture. More work is necessary to make the real-time system robust outside laboratory conditions, and it is possible that a deeper, more finely-tuned CNN could improve results.

Code

https://github.com/gsivesind/229-project

Contributions

Minh-An Quinn - Research, Development and Experimentation, Project Milestone Write Up, Poster, Final Report Write Up

Guilherme Reis - Research, CNN Model development and Experimentation, Error Analysis, Milestone Write-up, Poster, Final Report Write-up

Grant Sivesind - Development and Experimentation, Error and Accuracy Analysis, Project Milestone Write Up, Poster, Final Report Write Up

References

[1]

Burkert, Peter, et al. "DeXpression: Deep Convolutional Neural Network for Expression Recognition." *ArXiv:1509.05371v2*, arxiv.org/pdf/1509.05371.pdf.

[2]

"Challenges in Representation Learning: A report on three machine learning contests." I Goodfellow, D Erhan, PL Carrier, A Courville, M Mirza, B Hamner, W Cukierski, Y Tang, DH Lee, Y Zhou, C Ramaiah.

[3]

Chen, Jason, et al. Recognizing Emotion from Static Images. Stanford University Department of Computer Science, cs229.stanford.edu/proj2016spr/report/026.pdf.

[4]

Dachapally, Prudhvi Raj. "Facial Emotion Detection Using Convolutional Neural Networks and Representational Autoencoder Units." *ArXiv:1706.01509*, arxiv.org/abs/1706.01509.

[5]

Dumas, Melanie. "Emotional Expression Recognition Using Support Vector Machines." Machine Perception Lab, Univeristy of California, 2001.

[6]

F Feng, R Li, X Wang, D Athanasakis, J Shawe-Taylor, M Milakov, J Park, R Ionescu, M Popescu, C Grozea, J Bergstra, J Xie, L Romaszko, B Xu, Z Chuang, and Y. Bengio. arXiv 2013.

[7]

Kanade, T., Cohn, J. F., & Tian, Y. (2000). Comprehensive database for facial expression analysis. Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), Grenoble, France, 46-53.

[8]

Kotsia, Irene, and Ioannis Pitas. "Facial Expression Recognition in Image Sequences Using Geometric Deformation Features and Support Vector Machines." IEEE Transactions on Image Processing, vol. 16, no. 1, 2007, pp. 172–187., doi:10.1109/tip.2006.884954.

[9]

Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression. Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010), San Francisco, USA, 94-101.

[10]

McLaughlin, Tom, et al. Emotion Recognition with Deep-Belief Networks. Stanford University Department of Computer Science, cs229.stanford.edu/proj2010/McLaughlinLeBayanbat-RecognizingEmotionsWithDeepBeliefNets.pdf.

[11]

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011

[13]

Siegman, Aron W, and Stanley Feldstein, editors. NONVERBAL BEHAVIOR AND COMMUNICATION. PSYCHOLOGY PRESS, 2017, www.paulekman.com/wp-content/uploads/2013/07/Facial-Expression.pdf.

[14]

Tang, Yichuan. Deep Learning using Linear Support Vector Machines. Department of Computer Science, University of Toronto, citeseerx.ist.psu.edu/viewdoc/download;jsessionid=BC50EE0E9E439263F6064A781B958745?doi=10.1.1.664.594&rep=rep1&type=pdf.

[15]

Bradski, Gary. OpenCV. N.p.: n.p., 2015. Computer software.