

Emotion Recognition Neural Network

Final Project Paper for DATA622: Machine Learning & Big Data

Daniel Dittenhafer

December 13, 2016

Submit your final project, which includes the source code of your best model, plus a short (3-4 page) paper summarizing your work. Provide:

- an overview of the problem;
- why it's challenging;
- what techniques you used;
- comparison of performance;
- generalizable lessons learned/takeaways.

1 Overview

As part of the course requirements for Machine Learning & Big Data (DATA622), the problem of developing a machine learning model capable of classifying human face images into various emotions was presented. Specifically, the goal was to design and train a neural network to recognize the following 8 emotions and emit a probability of each of the emotion classes given the input 2 dimensional image.

1. anger
2. contempt
3. disgust
4. fear
5. happiness
6. neutral
7. sadness
8. surprise

Although this problem is not new to machine learning and solutions already exist (Microsoft Corporation, 2016), the complexities of the mathematics alone are challenging. This, combined with the application of a variety of neural network techniques, make for many unique solutions.

2 Techniques

The DATA622 class as a whole, under Professor Rowe's guidance, agreed to a general approach using the *Labeled Faces in the Wild* face database as a basis for training data (Huang, Ramesh, Berg, and Learned-Miller, 2007). We then labeled each of the included images using the Microsoft Emotion API. This was considered the "gold standard" for our training purposes and shared amongst the class as well as publicly in the [facial_expressions repository on GitHub](#) (Rowe, 2016).

In order to increase the size of our training and test data sets as well as apply regularization via data, each member of the class created ten image transformations using a variety of APIs and shared the transformations with the class as options for each of us when augmenting our data sets. The following code shows an example of a transformation applied during training (Joshi, 2015):

```
def cvBlurMotion1(img):

    size = 15
    kernel_motion_blur = np.zeros((size, size))
    kernel_motion_blur[int((size - 1) / 2), :] = np.ones(size)
    kernel_motion_blur = kernel_motion_blur / size

    img2 = cv2.filter2D(img, -1, kernel_motion_blur)
    return img2
```

The function, above, along with all code developed as part of this project can be found in the [emotional-faces repository on GitHub](#).

3 Performance

TBD

4 Conclusions

TBD

5 References

- Huang, G., M. Ramesh, T. Berg and E. Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Tech. rep. University of Massachusetts, Amherst, Oct. 2007. URL: <http://vis-www.cs.umass.edu/lfw/>.
- Joshi, P. OpenCV with Python By Example. Sep. 2015. URL: <https://www.packtpub.com/mapt/book/application-development/9781785283932/2/ch02lv11sec23/Embossing>.
- Microsoft Corporation. Emotion API. Dec. 2016. URL: <https://www.microsoft.com/cognitive-services/en-us/emotion-api>.
- Rowe, B. facial_expressions. Dec. 2016. URL: https://github.com/muxspace/facial_expressions.