Face Recognition And Emotion Detection Presentation

Indian Institute of Technology, Bombay

Team Info: Kiran Dapkar - 170050020 Mahesh Lomrar - 170050050 Sumit Jakhar- 170050111 Shreyas - 1700500..

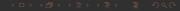
November 24, 2019

Face Recognition

- Face Detection in Python using the Viola-Jones algorithm on the CBCL Face Database published by MIT's Center for Biological and Computational Learning.
- Part One: The Basic Algorithm
- Part Two: The Additional Cascade

Code

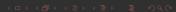
- viola_jones.py
 - An implementation of the Viola-Jones algorithm
 - Viola-Jones is an ensemble method which uses a series of weak classifiers to create a strong classifier.
- cascade.py
 - An implementation of the attentional cascade introduced by Paul Viola and Michael Jones.
- g face_detection.py
 - Methods to train and test a ViolaJones classifier on the training and test datasets.
 - Methods to train and test a CascadeClassifier on the training and test datasets.



Algorithm

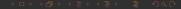
Data

- Get dataset from www.ai.mit.edu/courses/6.899/lectures/faces.tar.gz and compiled into pickle files.
- **Each** image is 19x19 and greyscale. There are Training set: 2,429 faces, 4,548 non-faces Test set: 472 faces, 23,573 non-faces.
- 3 training.pkl
 - An array of tuples. The first element of each tuple is a numpy array representing the image. The second element is its clasification (1 for face, 0 for non-face).
 - 2429 face images, 4548 non-face images.
- test.pkl
 - An array of tuples. The first element of each tuple is a numpy array representing the image. The second element is its clasification (1 for face, 0 for non-face).
 - 472 faces, 23573 non-face images.



Models

- 1 50.pkľ
 - A 50 feature Viola Jones classifier
- 200.pkl
 - A 200 feature Viola Jones classifier
- 3 cascade.pkl
 - An Attentional Cascade of classifiers looking at 1 feature, 5 features, 10 features, and 50 features.



Results

- The hyperparameter T for the ViolaJones class represents how many weak classifiers it uses.
- **2** For T=10, the model achieved 85.5% accuracy on the training set and 78% accuracy on the test set.
- 3 For T=50, the model achieved 93% accuracy on both the training and test set.

The End