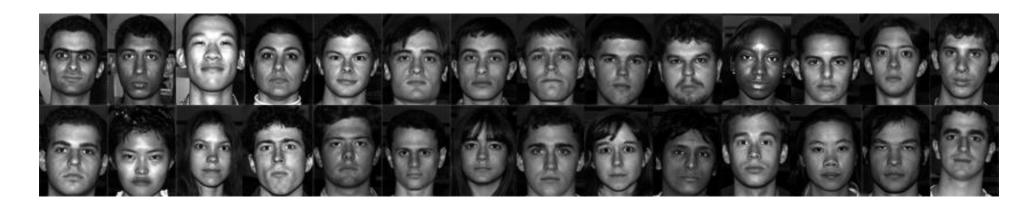
Machine Learning

Assignment #2

Face Recognition by Keras

Face Recognition by Keras

The Extended Yale Face Database



- All the images can be downloaded at:
 - Cropped Images (39 persons, 65 images each person)
 - http://vision.ucsd.edu/extyaleb/CroppedYaleBZip/CroppedYale.zip

Assignment #2

- 1. Setup Keras environment.
- 2. Modify the Keras sample code (or write your own) to define new model using VGG with only the first fully connected layer left (remove the rest).
- 3. Use Python to load input data (The Extended Yale Face Database)
- 4. Split the dataset by 35/30 in each class as training/testing dataset.
- 5. Run Keras to obtain the accuracy of test data for face recognition.
- 6. Improve your model using any technique from Lecture 8.

Ref:

- 1. kerascode.py
- 2. Introduction to Keras.pdf

Requirement for Assignment #1

- 1. Train your model with two initial settings:
 - Random weights
 - Pre-trained weights by PPMI.
- 2. Show the error curve/accuracy curve versus iterations.
- 3. Compare the overall accuracy (above settings & optimized one) to Nearest Neighbor (HW#1).
- 4. Submit two text files and your code/model to E-Course
 - Readme How to run your code
 - Report
 - Method description
 - Experimental results accuracy
 - Discussion of difficulty or problem encountered
- 5. Deadline: 05/07(Mon) 11:59p.m

Training Keras by using pre-trained weights

```
• def VGG16():
      input_layer...
       . . .
      output_layer...
      my_model = Model(intputs=input_layer,
 outputs=output_layer)
      my_model.complie(optimizer=SGD(1e-2), loss='mse')
```

- return my_model
- model = VGG16()
- # to save weight in model
- model.fit() # training...
- # after training...
- model.save_weights('file_path')

```
# to load weight into model...
model.load_weights('weight_path', by_name=True)
# by_name means load weights use layer name
model.fit() # training...
```