

CSC320: Assignment #3

Due on Saturday, March 21, 2015

Senisa Soenardjo

March 23, 2015

Problem 1

Dataset description

This project uses a subset of the FaceScrub dataset, which is composed of publicly available images of celebrities with square bounding boxes of their faces in each image. The sample `get_data.py` script was used to batch download the images from the database and save them to the **uncropped** directory. The script was also modified to batch crop and desaturate each image as to make them usable for the sample PCA program. The modifications simply apply the given bounding boxes to the original image, desaturate and crop them to 32x32 pixel images that are then saved to the **cropped** directory. Examples of the original images and their resulting desaturated, cropped and resized counterparts can be found in Figures ??-??.



(a) Original image of actor.



(b) Resultant cropped image.

Figure 1: Original and resultant image of Diana Agron as generated by `get_data.y`



(a) Original image of actor.



(b) Resultant cropped image.

Figure 2: Original and resultant image of Aaron Eckhart as generated by `get_data.y`



(a) Original image of actor.



(b) Resultant cropped image.

Figure 3: Original and resultant image of Ashley Benson as generated by `get_data.y`

Problem 2

The sample data for each actor was split into three groups of unique images, the training set *train* (containing 100 images per actor), the validation set *valid* and the testing set *test* (both containing 10 images per actor). This was done through the use of a simple script in `p3_main.py`. In the script, the contents of the **cropped** directory is scoured for each actor and copied into newly created directories **train**, **valid** and **test**. The contents of **cropped** is retrieved using `os.listdir()` which does not retrieve the list in alphanumeric order. The first 100 images found of each actor in this list are sorted into the **train** folder, while the proceeding groups of 10 are sorted into **valid** and **test**. Then the sample code `pca_example.py` is used to calculate the average eigenface (Figure ??) as well as the first 25 eigenfaces. `pca_example.py` was modified to take a training directory containing training images as input, which was then referenced in the main program `p3.py`.



Figure 4: The average eigenface of the *train* set. Generated using `p3.py` and `pca_example.py`.



Figure 5: The first 25 eigenfaces of the training set. Generated using `pca_example.py`.

Problem 3

The function `actor_recog()` in the main program carries out face recognition for each image in the *valid* set, and after finding the best k value, which represents the number of eigenfaces to use in projection, carries out recognition on the *test* set. Values of 2, 5, 10, 20, 50, 80, 100, 150, and 200 were used among the k values. The correctness of each of these values are found in Figure ???. In these results, the k value with the most correctness was found to be 2. Thus, the first 2 eigenfaces were used for actor recognition on the *test* set. This resulted in a correctness percentage of **52.2%**, lower than the original correctness of the *valid* set with the same k value. Some examples of failed matches for the *test* set can be seen in Figures ???-??.

Number of Eigenfaces	2	5	10	20	50	80	100	150	200
Percentage of Correct Results	62.5%	51.3%	47.5%	51.3%	52.5%	55.0%	55.0%	56.3%	56.3%

Figure 6: Percentage of correct results as generated by `actor_recog()`. Notice that using 2 eigenfaces results in the highest correctness percentage.



(a) *Test* image of Diana Agron.



(b) Matched *training* image of Andrea Anders

Figure 7: Incorrect identificaion of Diana Agron as Andrea Anders.

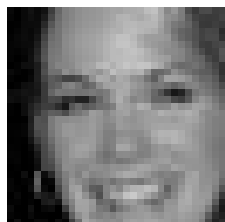


(a) *Test* image of Ashley Benson.

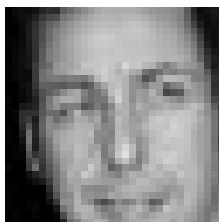


(b) Matched *training* image of Diana Agron

Figure 8: Incorrect identificaion of Ashley Benson as Diana Agron.

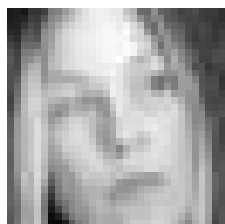


(a) *Test* image of Andrea Anders.

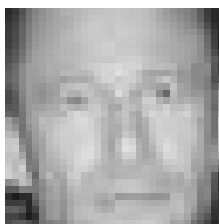


(b) Matched *training* image of Adam Sandler.

Figure 9: Incorrect identification of Andrea Anders as Adam Sandler.



(a) *Test* image of Christina Applegate.

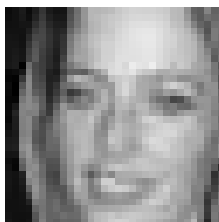


(b) Matched *training* image of Aaron Eckhart.

Figure 10: Incorrect identification of Christina Applegate as Aaron Eckhart.



(a) *Test* image of Adrien Brody.



(b) Matched *training* image of Gillian Anderson.

Figure 11: Incorrect identification of Adrien Brody as Gillian Anderson.

Problem 4

The function `masc_recog()` works similarly as `actor_recog()` from **Problem 3**, but instead of identifying the actor in the image, the relative masculinity/femininity of the face is identified. The original problem asks to identify the gender of the person in the image, though as gender is a personal identity and therefore separate from physical attributes, identifying whether a face is more masculine or feminine is more accurate. The program works by categorizing the masculine faced male actors from the original sample of actors and seeing whether a face had been matched with an actor in the masculine category. If the matched actor is within the masculine category, the image is identified as masculine. If the matched actor is not in the masculine category the image is identified as feminine. This function uses similar k values to represent the number of eigenfaces used as `actor_recog()`, 2, 5, 10, 20, 50, 80, 100, 150, and 200. The results of each of these values can be seen in Figure ???. As shown, using 200 eigenfaces produces the most correctness out of all the k values. As such, 200 eigenfaces were used in the masculinite recognition of the *test* set. This resulted in an **82.5%** correctness, which is once again lower than the findings of the *valid* set. Some examples of failed matches for the *test* set can be found in Figures ???-??.

Number of Eigenfaces	2	5	10	20	50	80	100	150	200
Percentage of Correct Results	91.3%	90.0%	90.0%	90.0%	88.8%	91.3%	91.3%	91.3%	92.5%

Figure 12: Percentage of correct results as generated by `gender_recog()`. Notice that using 200 eigenfaces results in the highest correctness percentage.



(a) *Test* set image of Adrian Brody.



(b) Matched *train* set image of Gillian Anderson

Figure 13: Incorrect identificaion of masculine featured Adrian Brody as feminine featured Gillian Anderson.



(a) *Test* set image of Aaron Eckhart.



(b) Matched *train* set image of Christina Applegate

Figure 14: Incorrect identificaion of masculine featured Aaron Eckhart as feminine featured Christina Applegate.

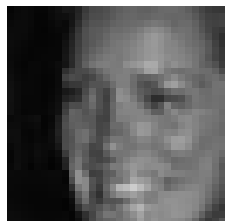
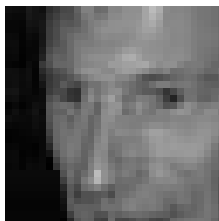
(a) *Testset* image of Andrea Anders.(b) Matched *train* set image of Aaron Eckhart.

Figure 15: Incorrect identificaion of feminine featured Andrea Anders as masculine featured Aaron Eckhart.

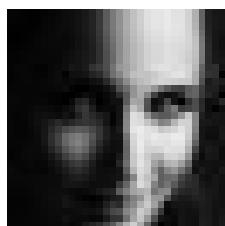
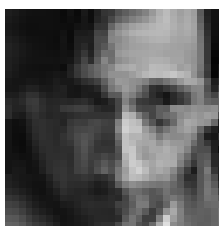
(a) *Test* set image of Christina Applegate.(b) Matched *train* set image of Aaron Eckhart.

Figure 16: Incorrect identificaion of feminine featured Christina Applegate as masculine featured Aaron Eckhart.

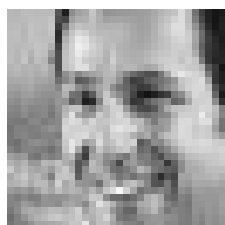
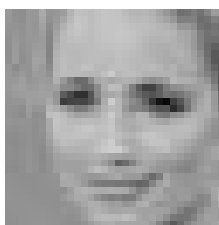
(a) *Test* set image of Adam Sandler.(b) Matched *train* set image of Diana Agron.

Figure 17: Incorrect identificaion of masculine featured Adam Sandler as feminine featured Diana Agron.