Overview of Code Modules

Austin Stone, as46569
University of Texas
Datamining

May 2, 2014

sortFaces.sh

This is a shell script I wrote to sort all the images into folders based on the direction that they are looking. I am including this shell script in my submission, although it needs to be modified based on your where you put your directory. The modifications should be fairly obvious.

load_image_db.m

This checks to see if the image database has already been loaded and saved as a .mat file, and if not loads all the images into a structure with fields for the data for each angle increment that the faces look.

get_img_mat.m

This function loads a user specified number of images from a specified folder. It is used by load_image_db.m

load_img.m

This was used by get_img_mat.m to load in images of various color types and convert them to greyscale matrices.

center_relative_to_mean.m

This function takes as input the mean image and a new, uncentered image and centers the new image relative to the mean image by shifting it up and down and left and right until its cosine similarity with the mean image is maximized.

center_database.m

This function iterates through all the fields of the image database structure and centers all the images using center_relative_to_mean.m and saves the database of centered images.

pca_by_svd.m

This function does pca via singular value decomposition. It takes as input the data matrix and the number of principal components to calculate and it returns the principal components, the values of the data matrix projected onto the principal components, and a vector which encodes the variance explained by each principal component.

get_classification_info.m

This function takes the image database and calculates for each field (angle orientation) the principal components, the variance each principal component explains, the mean, and each image's projection onto the principal components. Outputs a structure containing these values for each field.

display_images.m

Simply displaces a user specified number of images from the database. Used only for generating figures.

get_classification_stats.m

This takes as input a structure with of images of new faces unseen before, a structure of new images of faces seen before previously, and a matrix of images that are not faces. It returns a structure for each orientation field containing the average min distance for each image class, the average distance for each image class, the average max distance for each image class, and the standard deviation of these values.

get_smaller_mat.m

This function returns a subset of the images from a larger database. It was convenient to load all the centered images into one large structure, and then get test and validation subsets using this function.

graph_stats.m

Just generates a bar graph of the classification statistics. This was only used to generate figures for my paper.

normalize_test_images.m

This can be used to normalize all the image matrices for all fields (orientations). I used it to normalize the images used to compute classification statistics and the test images I used to determine the quality of my classification algorithm.

classify_new_img.m

Determines the classification of a new image.

odds_of_face.m

This is called by the classify_new_img.m function to get the odds of a new image being a face.

reconstruct_plot.m

This was only used to build the eigenface reconstruction plot in my paper. It takes an image and producing a tiling showing its reconstruction from its eigenface weighting.

savefigs.m

Saves all open figures.

test_prediction_accuracy.m

Takes the same input as get_classification_info.m. Takes unknown images of each class and evaluates my classification algorithm. Returns the true positive, false positive, true negative, and false negative rate of my classification algorithm.