Theoretical Part

Gabor Filter

We have learned applying Gabor filters to signals to take out the most important information and avoid noise. The same idea can also be applied to images to get rid of the redundancies and noises. We read the paper written by Miyuki Kamachi and Jiro Gyoba that each 256 by 256 pixel image will be convolved with gabor filters with a various wavelength and orientation. For cosine Gabor filter, we have:

\begin{equation}

G\_{\hat{k},+}(\hat{r}) = \frac{k^2}{\sigma^2}e^{\frac{-k^2||\hat{r}-\hat{r\_0}||^2}{2\sigma^2}}cos(\hat{k}(\hat{r}-\hat{r\_0})-e^{\frac{-\sigma^2}{2}}),

G\_{\hat{k},-}(\hat{r}) = \frac{k^2}{\sigma^2}e^{\frac{-k^2||\hat{r}-\hat{r\_0}||^2}{2\sigma^2}}sin(\hat{k}(\hat{r}-\hat{r\_0})

\end{equation}

where the + and - sign indicates the filter of even and odd phases and $\hat{k}$ is the filter wave-vector, which determines the special frequencies and orientations. [Ref] The results of the filtered image is represented by:

\begin{equation}

R\_{\hat{k},+/-} (\hat{r}) = \int G\_{\hat{k},+/-}(\hat{r}-\hat{r\_0}) I(\hat{r})d\hat{r}

\end{equation}

where I is the 256 by 256 pixel image, and

\begin{equation}

R\_{\hat{k}} = \sqrt{ R\_{\hat{k},+}^2 + R\_{\hat{k},-}^2 }

\end{equation}

[Ref] R is the amplitude of combined even and odd responses. Gabor filters can signify the facial features, so that the features are more obvious and are easier to be recognized and classified. Therefore, we are expecting that using Gabor filter on images will increase the classification accuracy rate. However, some of the filters may extract the most significant features on faces that are essential in recognizing facial expressions, while some may not helping by signify the noise as well. In order to find out the differences of effects of these Gabor filters on the images, we chose three different wavenumbers, which are $\pi/2$, $\pi/4$, and $\pi/8$. We also chose various orientations which goes from 0 to $\pi$ with interval of $\pi/6$. In total, we have 18 Gabor filters, and we would like to compare how their effects would influence the classification accuracy rate.

Implementation

First of all, we need to import the data. When we go through ever single image in the folder, we reshape it into a column vector and saved it in a big matrix IMG which contains all the image data. Since all the image data are in uint8 format, we need to change it to format double in order to do further classification. In order to get the unbiased classification rate, we first separated the entire data matrix into training set and testing set by using randperm to permute numbers. We separated them into 70 and 30. So we took out the first 70 percent of columns as our training set and the last 30 percent as testing set. Not only for the image matrix, we need to do the same for the label matrix as well.

We also need to o PCA to reduce the dimension of the matrix. To do that, we can use “svd” command to get U, $\Sigma$, and V matrices. By plotting the diagonal of $\Sigma$ matrix, we can see that not all of the components are equally important. First three are the most important. In order to investigate the different effects of energy used, we set threshold to 0.5, 0.7, and 0.9. To reconstruct the training set, we take out the first r columns of the U matrix and multiply with the original training data matrix. We did the same thing for testing set as well. Now, we have training and testing set with reduced dimension.

For SVM, we can just use the build in function “fitcecoc” to train our model by inputting training set data and training labels. After getting the trained model, we can use the “predict” command by inputting testing data to get predictions and then compare with the real labels for testing set. For LDA, the process is the same; we just changed the classification command to “fitcdiscr”. KNN,NAÏVE BAYES

For Gabor filter combines with neural network, the process is a little bit complicate than the other four methods. First of all, when we import the data, we need to save one more thing, which is the magnitude of the Gabor filter. This is the thing we need to use for the further classification. Also, for the labels, we need to do some pre-processing as well. Since the neural network will only take the numbers as label, but we have strings here, we need to convert strings to numbers. Therefore, we assigned each facial expression with a different number from 1 to 7.

MATLAB command

[mag,phase] = Imgaborfilt(image, w, o): this is a 2D Gabor filter which can be applied on images. For the inputs, image is the original image data in pixel; w is the wavelength, and o is the orientation. For outputs, mag is the magnitude of the Gabor filter, and phase is…

randperm(n): Randomly permute numbers from 1 to n.

mod = fitcecoc(x,label): This command will do multiclass classification using svm, where x is the data matrix and label is the corresponding labels of x.

mod = fitcdiscr(x,label): This command will give us the LDA model. The input x is the data matrix and label is the corresponding labels of x.

KNN

NAÏVE BAYES

label = predict(mod,x’): This command will predict the labels of the new input by using the trained model, where mod is the trained model and x’ is the new data. This command can be used for all trained classification models.