1. Problem Statement

This project implements the following classifiers for face recognition.

- ML estimation with Gaussian assumption followed by Bayes rule
- Nearest neighbor (NN) rule
- PCA followed by Bayes classifier and NN rule
- LDA followed by Bayes classifier and NN rule

Two different scenarios are performed using the Bayes classifier and the NN classifier:

- Test the impact of facial expressions, illumination variations
- Test the impact of training data size

2. Data Sets

Three datasets are used in this project:

- **DATA.MAT:** Cropped images of 200 subjects, 3 images each, each image of size 24x21. The file data.mat has a variable "face" of size (24x21x600). The images corresponding to the person labeled n, n = {1, . . . , 200} can be indexed in Matlab as face(:,:,3*n-2), face(:,:,3*n-1) and face(:,:,3*n). The first image is a neutral face, the second image is a face with facial expression, and the third image has illumination variations.
- **POSE.MAT:** Dataset that contains images with pose variation. 68 subjects under 13 different poses.
- **ILLUMINATION.MAT:** Dataset that contains images with illumination variation. 68 subjects across 21 illuminations.

3. Classifier Implementations

3.1 ML estimation with Gaussian assumption followed by Bayes rule

- Calculate the sample mean for each class $\mu_i, i = 1 \dots c$
- Calculate the sample variance for each class $\Sigma_i, i = 1 \dots c$
- If Σ_i is singular, add a small constant diagonal matrix to make it non-singular
- Discriminant functions: $g_i(x) = x^t W_i x + w_i^t x + w_{i0}$
- $W_i = -\frac{1}{2}\Sigma_i^{-1}$,
- $w_i = \Sigma_i^{-1}\mu_i$,
- $w_{i0} = -\frac{1}{2}\mu_i^t\Sigma_i^{-1}\mu_i - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)$. The last term $\ln P(\omega_i)$ can be ignored assuming each class has the same prior probability.
- Decision rule: assign to ω_k if $\text{argmax}_i\ g_i(x) = k$

3.2 Nearest neighbor (NN) rule

- For each test data x, calculate its Euclidean distance $d(x_i, x)$ for all training data.
- Decision rule: assign to ω_k if $\text{argmin}_i\ d(x_i, x) = k$

3.3 PCA followed by Bayes classifier and NN rule

- Assume we want c-1 principle components (The same number of components as LDA).
- Use Singular Vector Decomposition: [W, S, V] = svds(D, c-1) , a function in Matlab, to find the transformation matrix W.
- $W_{d*(c-1)}$ transforms the training data matrix X_{d*n} to a new data matrix $Y_{(c-1)*n} = W'X$
- Build Bayes classifier or NN classifier based on the new training data matrix $Y_{(c-1)*n}$
- Use the same $W_{d*(c-1)}$ to transform the test data to c-1 dimensions.
- Classify the test data using the Bayes classifier or NN classifier.

3.4 LDA followed by Bayes classifier and NN rule

- Calculate the sample mean for each class $m_i, i = 1 \dots c$
- Calculate the total sample mean m
- Calculate the sample variance matrix for each class $\Sigma_i, i = 1 \dots c$
- If Σ_i is singular, add a small constant diagonal matrix to make it non-singular
- Calculate the within-scatter matrix $S_W = \sum_{i=1}^{c} \Sigma_i$
- Calculate the between-scatter matrix $S_B = \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^t$
- Generalized eigenvector decomposition: [W, EV] = eigs(SB, SW, c-1), a function in Matlab
- $W_{d*(c-1)}$ transforms the training data matrix X_{d*n} to a new data matrix $Y_{(c-1)*n} = W'X$
- Build Bayes classifier or NN classifier based on the new training data matrix $Y_{(c-1)*n}$
- Use the same $W_{d*(c-1)}$ to transform the test data to c-1 dimensions.
- Classify the test data using the Bayes classifier or NN classifier.

4. Experiments

4.1 Test the impact of facial expressions, illumination variations

- Dataset used: data.mat
- Classifier used: Bayes classifier, NN classifier, PCA+Bayes classifier, PCA+NN classifier, LDA+Bayes classifier, LDA+NN classifier
- Training dataset: select 2 out of 3 images for each subject for training. For example, if we want to test the impact of facial expressions, then we select the neutral face and the face with illumination variations as training dataset.
- Test dataset: select the remaining 1 out of 3 image for each subject for testing

- Results:

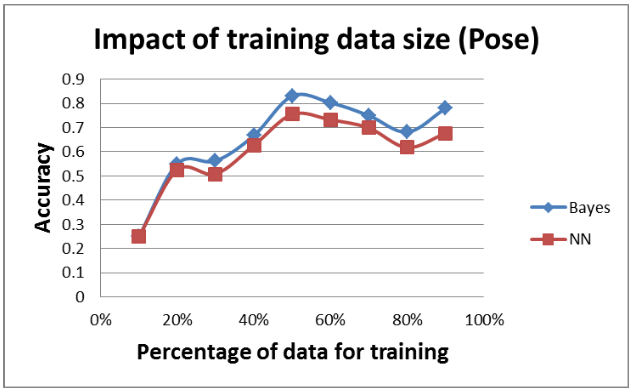
Case	Training dataset	Test dataset	Bayes	NN	PCA +Bayes	PCA +NN	LDA +Bayes	LDA +NN	Average
1	Neural Expression	Illumination	0.64	0.595	0.635	0.585	0.66	0.63	0.624
2	Neural Illumination	Expression	0.665	0.65	0.66	0.65	0.78	0.78	0.698
3	Expression Illumination	Neural	0.72	0.555	0.71	0.555	0.885	0.885	0.718
Average			0.675	0.6	0.668	0.597	0.775	0.765	-

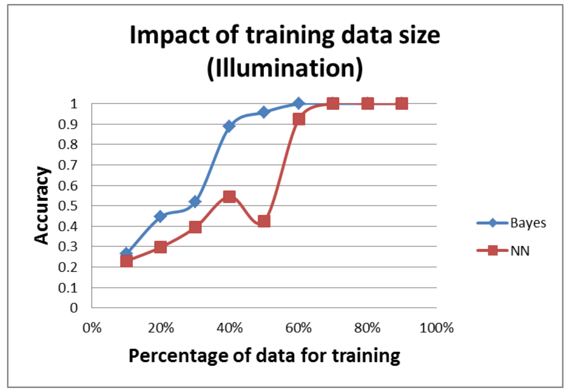
- Discussion:
  - Comparing three different cases for training and testing, we can observe that on average, case 3 (the faces with different expressions and illumination variations are used to train the model and the neural faces are tested) achieves the best accuracy. This may indicates that if the training image has enough variance such as illumination and expression, then the accuracy of classifying a new images will be higher.
  - Comparing six different classifiers, we can first observe that the testing accuracy for the "Bayes" and "NN" classifier is low, on average only 67.5% and 60% respectively. This may due to the lack of enough training data. Secondly, "LDA+Bayes" and "LDA+NN" classifiers perform the best, which obtain 88.5% accuracy, which seems to contradict with the intuition that the true error can't be decreased if we project the data distribution to a lower dimension. It may due to the fact that we lack enough number of training data and we have to manually add a constant diagonal matrix to make the sample covariance matrix non-singular. Thirdly, the result of "PCA+Bayes" and "PCA+NN" are worse than the result without PCA and the result with LDA. This indicates that PCA as an unsupervised dimensionality reduction technique would undermine the testing accuracy when it's used for supervised learning. Lastly, "Bayes" classifier performs better than "NN" classifier, which is in agreement with the textbook analysis.

4.2 Test the impact of training data size

- Dataset used: pose.mat, illumination.mat
- Classifier used: Bayes classifier and NN classifer
- Experiment: Vary the percentage of data used for training from 10% to 90%, use the remaining data for testing and investigate its impact on accuracy
- Results:

Dataset	Classifier	Percentage of data for training								
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Pose	**Bayes**	0.2488	0.55	0.5621	0.6673	0.8309	0.8029	0.75	0.6814	0.7794
	NN	0.2488	0.525	0.5082	0.6268	0.7549	0.7324	0.6985	0.6176	0.6765
Illumination	**Bayes**	0.2647	0.4455	0.5196	0.8869	0.9574	0.9982	1	1	1
	NN	0.2299	0.2976	0.3941	0.5452	0.4250	0.9228	1	1	1





- Discussion:
    - For the pose.mat dataset, we can observe that when the training data size increases from 10% to 50%, the testing accuracy increases for both the "Bayes" and "NN" classifiers. The maximum accuracy occurs when the percentage of data for training is 50%. When the training data size is further increased, the testing accuracy becomes saturated and even dropped. This is probably because that when training data size is small (<50%), the model is under-fitting and thus can't obtain high accuracy for the test data. On the other hand, when the training data size is large (>50%), some training data may be outliers and undermine the trained model.
    - For the illumination.mat dataset, we can observe that the testing accuracy increases as the percentage of data for training increases. The accuracy can achieve 100% when the percentage of data for training is larger than 70%.
    - For both datasets, we can observe that "Bayes" classifier performs better than "NN" classifier, which is in agreement with the textbook analysis.

5. Conclusion

In this project, two types of classifier are implemented: the Bayes classifier and the NN classifier. Based on that, two dimensionality reduction techniques (PCA and LDA) are implemented and applied on the Bayes and the NN classifier. The result for the impact of facial expressions, illumination variations shows that if the training image has enough variance, then the accuracy of classifying a new images will be higher. The result for the impact of training data size shows that more training data can normally increase the accuracy.