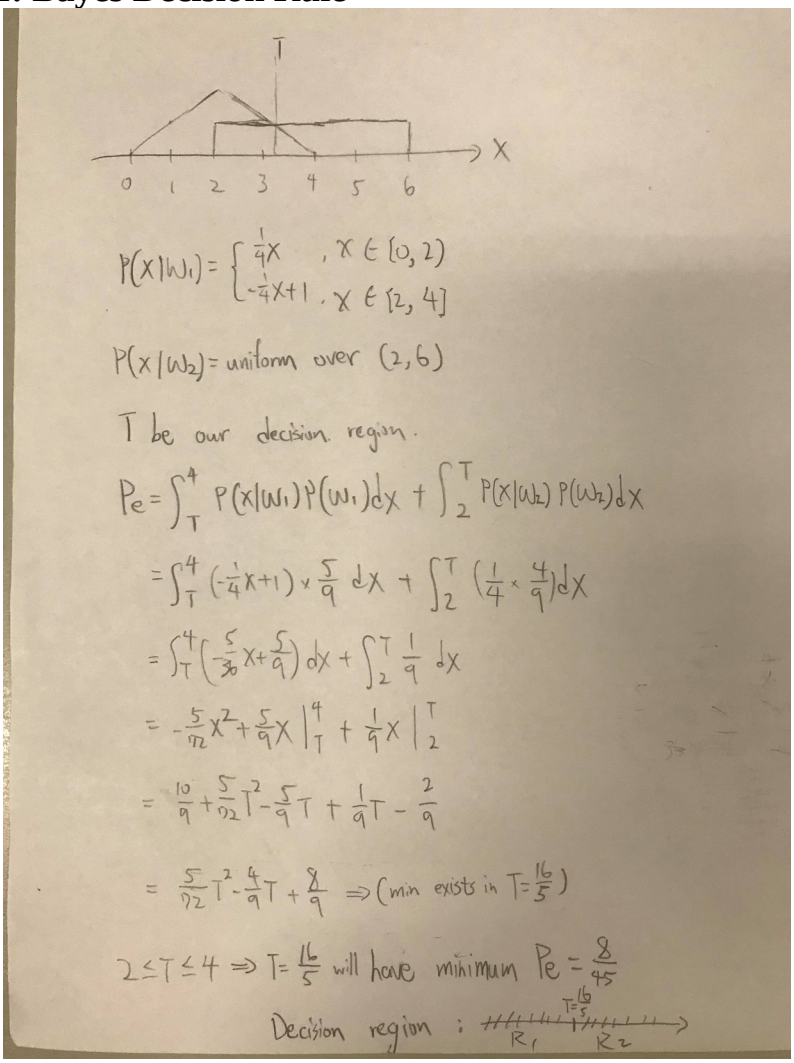
DLCV
Hw1

R07945010
許展銘

- **Problem 1: Bayes Decision Rule**



- **Problem 2: Principal Component Analysis**

1.

```
train_X = [imageio.imread("hw1_dataset/"+n) for n in train_set_name]
train_X = np.array(train_X).reshape(240,-1)
```

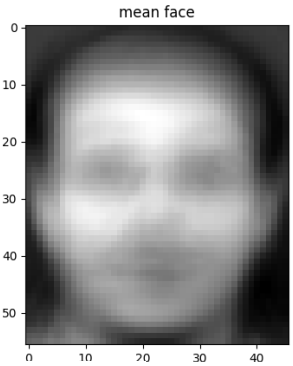
Mean face:

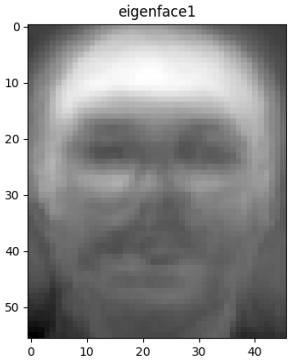
```
#mean face
mean_vector = train_X.mean(axis=0)
```

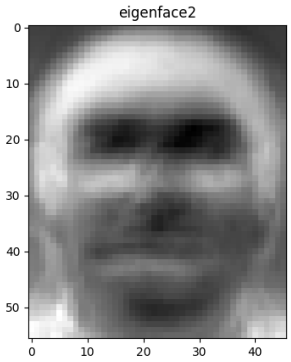
Eigenface:

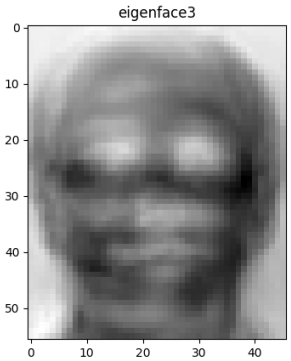
```
pca = PCA()
output = pca.fit(train_X-mean_vector)
output.components_.shape

e1 = (output.components_[0]).reshape(56,46)
e2 = (output.components_[1]).reshape(56,46)
e3 = (output.components_[2]).reshape(56,46)
e4 = (output.components_[3]).reshape(56,46)
```

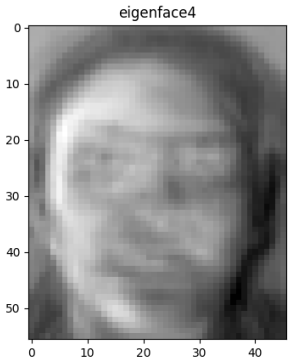
Result:







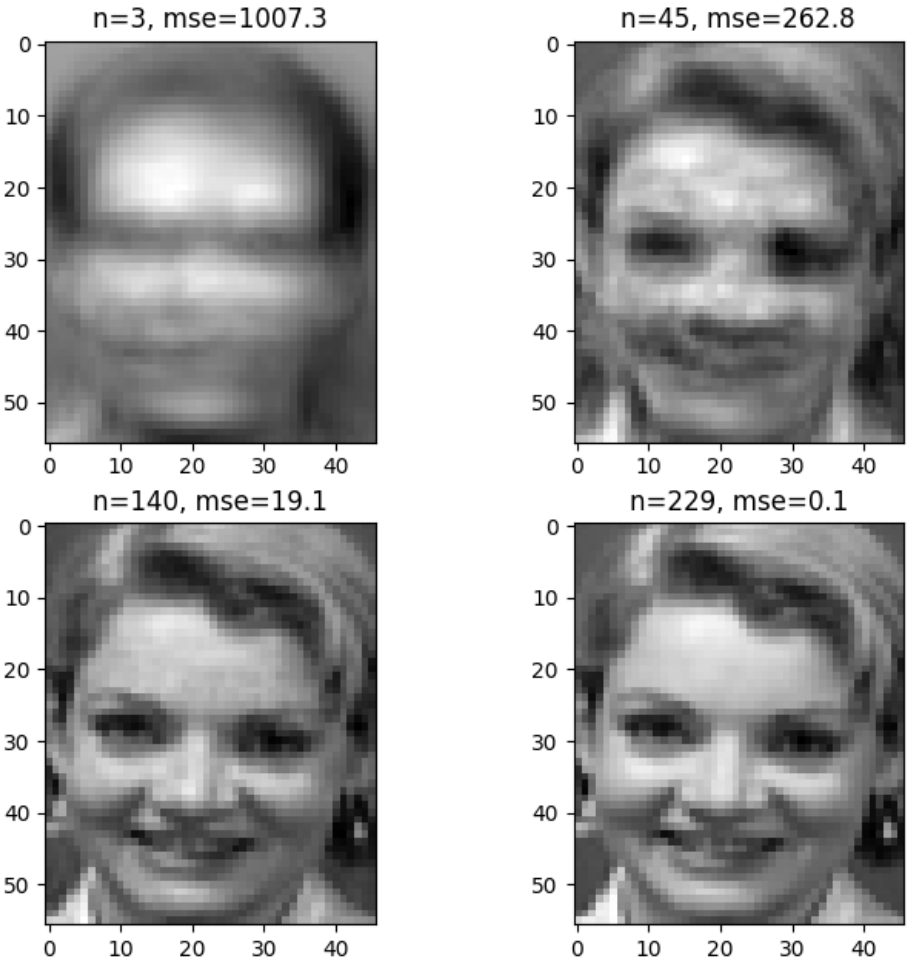




2 and 3. reconstructed images

```
projected =pca.transform(first_img-mean_vector)
```

```
figure = (projected[:,:i] @ output.components_[:i]) + mean_vector
mse = np.mean((figure - first_img)**2)
```



4. k-nearest neighbors

3-fold cross-validation(using GridSearchCV library):

```
grid = {"n_neighbors":[1,3,5]}
classify = GridSearchCV(knn, grid, cv=3)
```

Result of testing model:

```
          k=1         k=3         k=5
n=   3 [0.70416667 0.61666667 0.52083333]
n=  45 [0.92916667 0.85833333 0.79166667]
n= 140 [0.92916667 0.85833333 0.75416667]
```

Hence, we can choice k=1, n=45 or 140 since its higher score.
Here we use (k,n) = (1,45) as hyperparameters.

5. recognition rate(Accuracy on testing set)

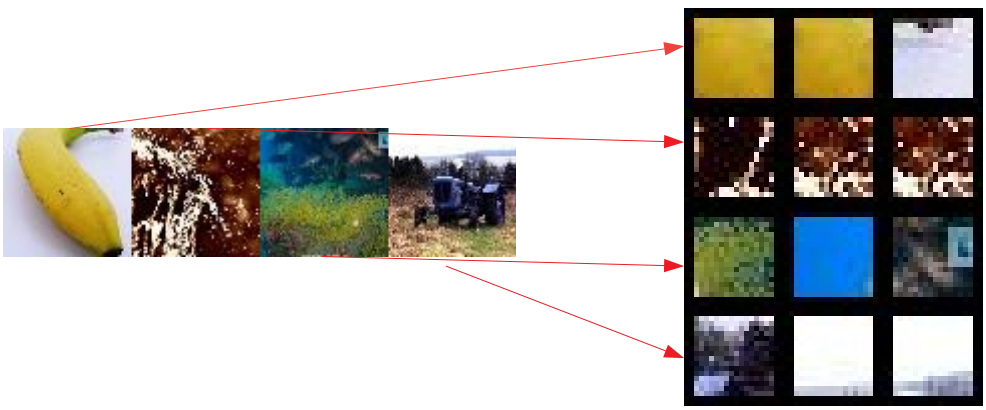
```
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(train_X_reduced[:,:45], train_y)
pred_y = knn.predict(test_X_reduced[:,:45])

acc = accuracy_score(y_pred=pred_y, y_true=test_y)
```

Accuracy:

```
(DLCV) chanminhsu@chanminhsu:~/桌面/hw1$ python pca.py
Accuracy: 0.95625
```

- **Problem 3: Visual Bag-of-Words**

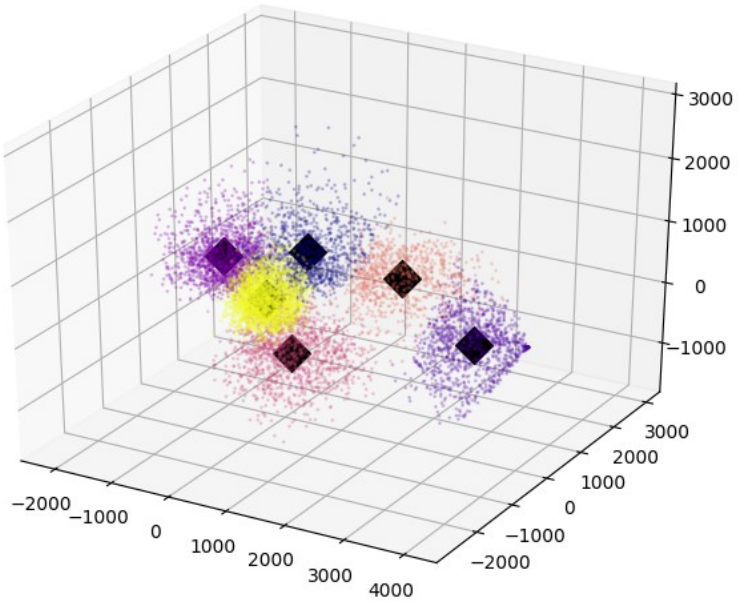
1. Picked images vs 3 patches



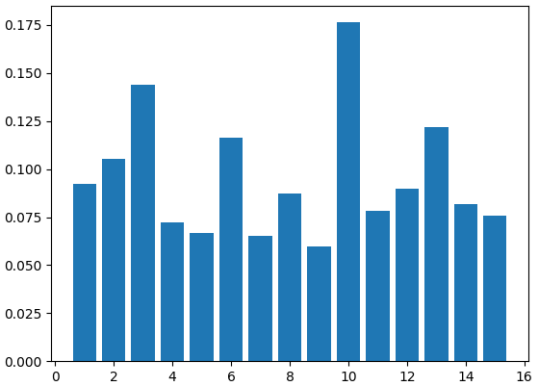
Because of the colors presented in patches(i.e yellow for banana), we easily classify each image by their patches.

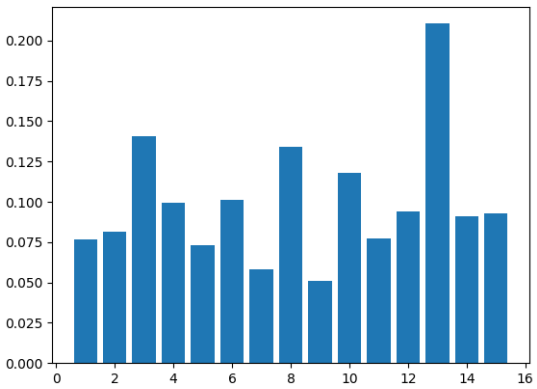
2. Kmeans and PCA

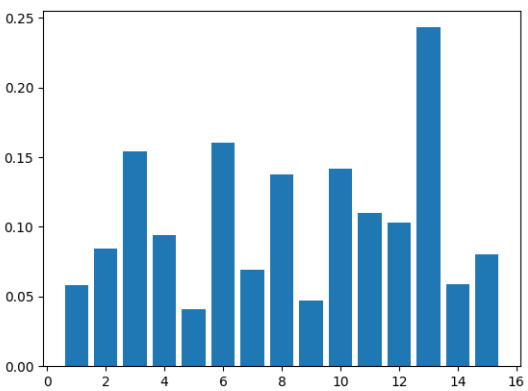
```
km = KMeans(n_clusters=CLUSTER_NUM, max_iter=5000, n_jobs=-1).fit(X_train_patches)
centers = km.cluster_centers_
# labels are the indexes of centers
labels = km.labels_
pca = PCA(n_components=3).fit(X_train_patches)
train_patches_pca = pca.transform(X_train_patches)
centers_pca = pca.transform(centers)
```

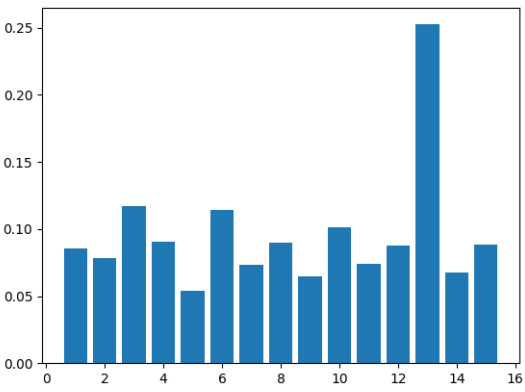


3. BoW using histogram plot(randomly choose four images from each class)
Topleft: Banana Topright: Fountain Downleft: Reef Downright: Tractor









4. Classification accuracy

```
(DLCV) chanminhsu@chanminhsu:~/桌面/hw1$ python hw1_p3.py
k-nearest neighbors classifier accuracy: 0.532
```

Accuracy: 0.532