Q3

June 17, 2020

In [1]:
```
import glob
from scipy.io import loadmat
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import tensorly
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from tensorly.decomposition import tucker
%matplotlib inline
```

In [2]:
```
y_train = [x[0] for x in loadmat('train_lab.mat')["train"]]
y_test = [x[0] for x in loadmat('test_lab.mat')["test"]]

train_files = glob.glob('CatsBirds/train*')
test_files = glob.glob('CatsBirds/Test*')
```

In [3]:
```
np.array(Image.open(train_files[0]).convert('L')).shape
```

Out[3]:
```
(500, 500)
```

In [4]:
```
# part 1
tnsr_train = np.zeros((500,500,len(train_files)))
tnsr_test = np.zeros((500,500,len(test_files)))


for tf in train_files:
    idx = int(tf.split("train")[1].split(".")[0]) - 1
    tnsr_train[:,:,idx] = np.array(Image.open(tf).convert('L'))

for tf in test_files:
    idx = int(tf.split("Test")[1].split(".")[0]) - 1
    tnsr_test[:,:,idx] = np.array(Image.open(tf).convert('L'))
```

In [5]:
```
#partial decomposition
G, factors = tensorly.decomposition.partial_tucker(tnsr_train, modes = [0,1],ranks=[10
```

```
        A,B =factors

        G_f = np.zeros((28,100))
        for i in range(28):
            G_f[i] = G[:,:,i].flatten()

        G_test = tensorly.tenalg.multi_mode_dot(tnsr_test, [x.T for x in factors], modes=[0,1]

        G_test_f = np.zeros((12,100))
        for i in range(12):
            G_test_f[i] = G_test[:,:,i].flatten()
```

In [6]:
```
clf = RandomForestClassifier(max_depth=2,n_estimators=100)
clf.fit(G_f, y_train)
y_hat = clf.predict(G_test_f)
# error rate
1-accuracy_score(y_test, y_hat)
```

Out[6]: 0.08333333333333337

In [7]:
```
# part 2
tnsr_train = np.zeros((500,500,3,len(train_files)))
tnsr_test = np.zeros((500,500,3,len(test_files)))

for tf in train_files:
    idx = int(tf.split("train")[1].split(".")[0]) - 1
    tnsr_train[:,:,:,idx] = np.array(Image.open(tf))

for tf in test_files:
    idx = int(tf.split("Test")[1].split(".")[0]) - 1
    tnsr_test[:,:,:,idx] = np.array(Image.open(tf))
```

In [8]:
```
#partial decomposition
G, factors = tensorly.decomposition.partial_tucker(tnsr_train, modes = [0,1,2],ranks=[
A,B,C =factors


G_f = np.zeros((28,300))
for i in range(28):
    G_f[i] = G[:,:,:,i].flatten()

G_test = tensorly.tenalg.multi_mode_dot(tnsr_test, [x.T for x in factors], modes=[0,1,

G_test_f = np.zeros((12,300))
for i in range(12):
    G_test_f[i] = G_test[:,:,:,i].flatten()
```

In [9]:
```
clf = RandomForestClassifier(max_depth=2,n_estimators=100)
clf.fit(G_f, y_train)
```

```
y_hat = clf.predict(G_test_f)
# error rate
1-accuracy_score(y_test, y_hat)
```

```
Out[9]: 0.16666666666666663
```

1 Discussion

The error rate for the grayscale and color images was 0.08 and 0.17 respectively with a random forest using 100 trees and a max depth of 2.