

Experiment Results On LFWa

Note that these results are for LTP features. You can use the same notebook for visualizing other features (LQP, LBP) results by modifying the features directory path ('LTP-tolerance=5') with the corresponding feature directory path in the following code cells.

First set some global variables

```
In [2]: #extract configuration parameters from demo.py
import sys,os
sys.path.insert(0,'./face-rec/code')
import demo as d
reload(d)
odir,ftype=d.getConfigParameters()
print odir, ftype
```

/scratch/testing/new-experiments/ LTP

Mean Image

```
In [3]: #Mean Image:
dfmim.imread(odir+'/meanim.png')
ih=imshow(dfmim),gray()
axis('off');
```



LTP Features

$\tau = 5$

Cropping Offsets (displacement of cropping window from its center position): (X x Y)=(+1 x -4)

Feature Normalization= $\sqrt{L1}$

Without PCA

View1

```
In [4]: # Withoutout PCA and Chi-Square Simpilarlity on View1
v2s=np.load(odir+'features/LTP-tolerance=5/data/view1-LTP-chi-square-simple-threshold-model-results.npy')
print 'Accuracy of Simple Model (Without PCA, Chi-Square Distance) on View1 = %0.2f'%((v2s)*100)
```

Accuracy of Simple Model (Without PCA, Chi-Square Distance) on View1 = 73.60

View2

```
In [5]: # Without PCA and Chi-Square Simpilarlity on View2
v2s=np.load(odir+'features/LTP-tolerance=5/data/view2-LTP-chi-square-simple-threshold-model-results.npy')
print v2s
print 'Average Accuracy of Simple Model (Without PCA, Chi-Square Distance) on View2 = %0.2f'%(np.mean(v2s)*100)
```

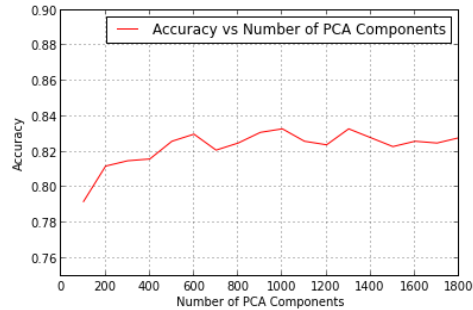
```
[ 0.75333333  0.75      0.69      0.70333333  0.72333333  0.74833333
 0.77833333  0.69166667  0.71333333  0.73333333]
```

Average Accuracy of Simple Model (Without PCA, Chi-Square Distance) on View2 = 72.85

With PCA

View1 (Cross validation for finding number of PCA components)

```
In [6]: v2pca=np.load(odir+'features/LTP-tolerance=5/data/view1-LTP-PCA-2000-cosine-results.npy')
np.set_printoptions(suppress=True)
#print 'Accuracy =', v2pca
#print 'Average Accuracy of Model With PCA (components = 500) and cosine Similarlity on View1 = %0.2f'%(v2pca[-1,1]*100)
plot(v2pca[:2,0],v2pca[:2,1],'-r',label='Accuracy vs Number of PCA Components')
ylim(0.75,0.9)
xlabel('Number of PCA Components')
ylabel('Accuracy')
legend(loc=0)
grid()
```



View2

```
In [7]: v2pca=np.load(odir+'features/LTP-tolerance=5/data/view2-LTP-PCA-1000-cosine-results.npy')
np.set_printoptions(suppress=True)
print 'Accuracy =', v2pca
print 'Average Accuracy of Model With PCA (components = 1000) and cosine Simpilarlity on View2 = %0.2f'%(mean(v2pca[:,1])*100)
```

Accuracy = [[1000. 0.84666667]

[1000.]	0.83166667]
[1000.]	0.82166667]
[1000.]	0.81333333]
[1000.]	0.83666667]
[1000.]	0.83833333]
[1000.]	0.84166667]
[1000.]	0.82333333]
[1000.]	0.81833333]
[1000.]	0.83333333]

Average Accuracy of Model With PCA (components = 1000) and cosine Simpilarlity on View2 = 83.05