UNIVERSIDADE FEDERAL DE MINAS GERAIS

RECONHECIMENTO DE PADRÕES

RELATÓRIO PCA FACES OLIVETTI

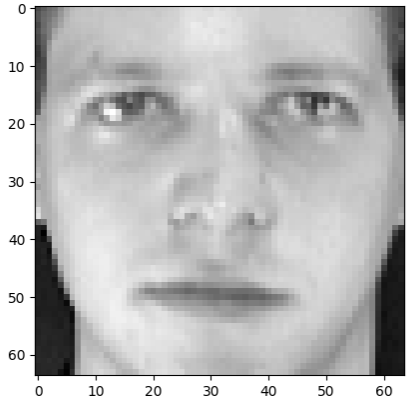
GABRIEL SARAIVA ESPESCHIT – 2015065541

04 de out de 2020

Para esse exercício, utilizou-se da base de dado Olivetti, importadas utilizando o *SkLearn:*

```
> from sklearn.datasets import fetch_olivetti_faces
> faces, _ = fetch_olivetti_faces(return_X_y=True, shuffle=False)
> n_samples, n_features = faces.shape
```

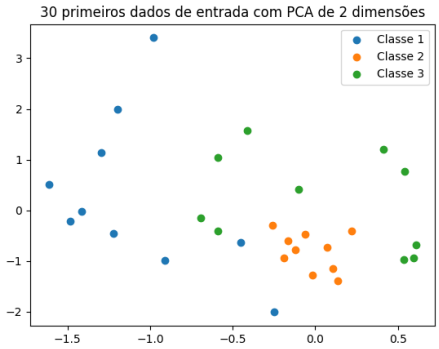
Uma das 400 faces da base de dados foi visualizada, após sua centralização:



O y foi gerado conforme explicado na guia. Em seguida, utilizando a função PCA do *SkLearn*, fez a redução de dimensionalidades do problema:

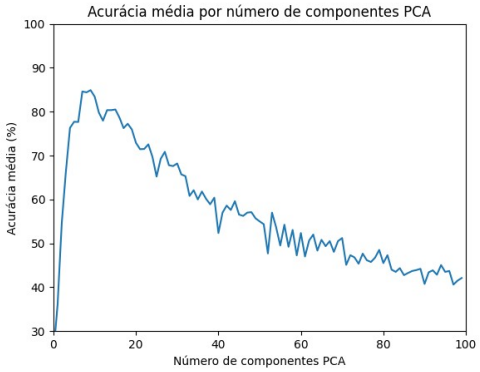
```
> from sklearn.decomposition import PCA
> pca = PCA(n_components=2)
> pcaFaces = pca.fit_transform(faces)
```

Sendo assim, foi possível visualizar os 2 atributos dos 30 primeiros componentes do nosso banco de dados:

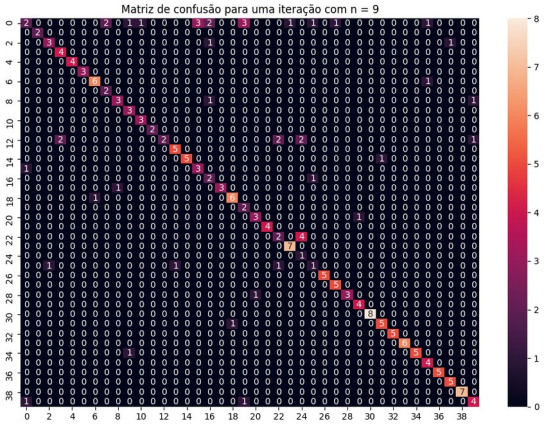


Em seguida, criou-se uma dinâmica em que se classificava as imagens. Nessa dinâmica, buscou-se estudar o impacto de como uma dimensionalidade diferente pode impactar a classificação. A classificação usada foi a KDE, desenvolvida em exercícios anteriores.

Sendo assim, criou-se um ciclo *for* em que, a cada iteração, se faz um PCA com dimensões diferentes, variando de 1 até 100. Para cada dimensão analisada, se dividia os dados em 50% para treino e 50% para teste de forma randômica e se fazia a classificação. Para cada dimensão analisada, se fez esse processo 10 vezes e salvou os resultados da acurácia média para cada dimensão do PCA. O gráfico abaixo mostra esses resultados.



Foi possível verificar que o melhor resultado foi quando utilizamos 9 dimensões. Obtivemos uma acurácia de 84,3%. A matriz de confusão para um dos resultados obtidos pode ser visualizada abaixo:



Para os atributos 21 e 1 o nosso algoritmo teve dificuldades, pois eles acabaram classificando esses atributos erroneamente algumas vezes.