

Relatório Trabalho Final

Uso de Eigenfaces para reconhecimento facial

Sumário

1. Introdução	3
2. Aplicações.....	3
2.1 Sobre o dataset	3
2.2 Comportamento do PCA com 2 pessoas	4
2.3 PCA aplicado em todo dataset	7
2.4 Comentários e melhorias	10
3. Conclusão	11

1. Introdução

Esse relatório busca explicar a aplicação prática do Trabalho Final da disciplina de Computação Científica e Análise de Dados, no qual buscou-se explorar a aplicação do PCA em técnicas de reconhecimento facial. As seções abaixo detalham sobre as formas de aplicação desenvolvidas e a teoria que foi utilizada. A aplicação essencialmente foi dividida em dois testes, o primeiro sendo a aplicação do PCA em um dataset menor, com imagens de apenas 2 pessoas, na qual o objetivo é gerar visualizações em 2D mais interessantes, enquanto a segunda parte do trabalho foi responsável por tentar aplicar o reconhecimento facial em todo dataset e avaliar os resultados obtidos pelo algoritmo.

Para acessar os códigos desenvolvidos, olhar o anexo enviado no classroom ou acessar o link do repositório no github disponibilizado [aqui](#).

2. Aplicações

2.1 Sobre o dataset

O dataset escolhido para essa aplicação foi o Olivetti Faces, obtido através da própria biblioteca `scikit-learn`, a qual foi a mesma biblioteca que foi utilizada a função do PCA. Esse é um dataset bem conhecido e estruturado, e fortemente recomendado para aplicações de reconhecimento facial. No dataset há um total de **400** imagens de **40** pessoas diferentes, ou seja, há **10** imagens de cada pessoa. As imagens estão todas em preto e branco e estão em uma resolução de **64 x 64** pixels, sendo que cada pixel da escala cinza já vem previamente convertidos para valores de ponto flutuante entre **0** e **1**.

A escolha desse conjunto de dados foi feita com base na forma como os rostos são dispostos no dataset, uma vez que todas as imagens são apenas das faces das pessoas, em sua maioria em um ângulo reto com a câmera, porém algumas ainda apresentam algumas variações de posicionamento do rosto, que foram imagens interessantes de analisar o comportamento do PCA diante disso. Abaixo está uma breve visualização de apenas algumas imagens do dataset, na qual é possível ver o formato em que as imagens foram capturadas.



Imagem 1: Fotos das pessoas do conjunto de dados

A biblioteca `scikit-learn` disponibiliza as imagens em duas formas distintas, a primeira é através de uma matriz 3D, de dimensões **400 x 64 x 64**, na qual cada linha desse “paralelepípedo” representa uma imagem **64 x 64** pixels. Esse é um bom conjunto de dados para plotar as imagens e mostrar alguns resultados, mas o formato que foi utilizado para todas as análises feitas foi uma matriz **400 x 4096**, na qual todas as imagens foram convertidas de uma matriz **64 x 64** para um vetor **1 x 4096**, e todos eles foram agrupados nas linhas da matriz que foi chamada de `data`. Por fim, outro conjunto de dados importante foi um vetor, aqui chamado `targets`, com os identificadores de cada linha da matriz `data`, na qual cada entrada do vetor `targets` indicaria o ID da pessoa a qual a linha na matriz `data` se refere.

Abaixo está representada a matriz `data`, a qual será a protagonista nos experimentos:

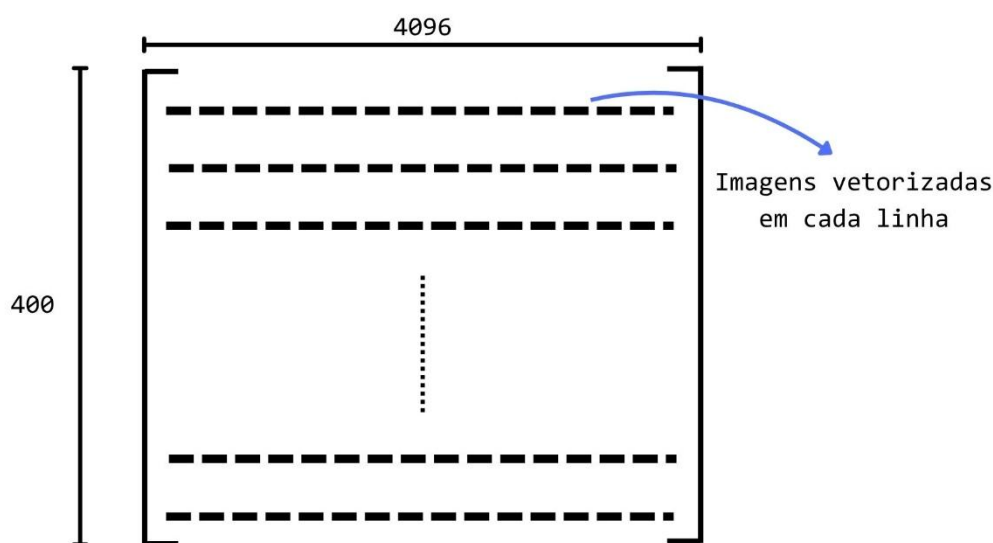


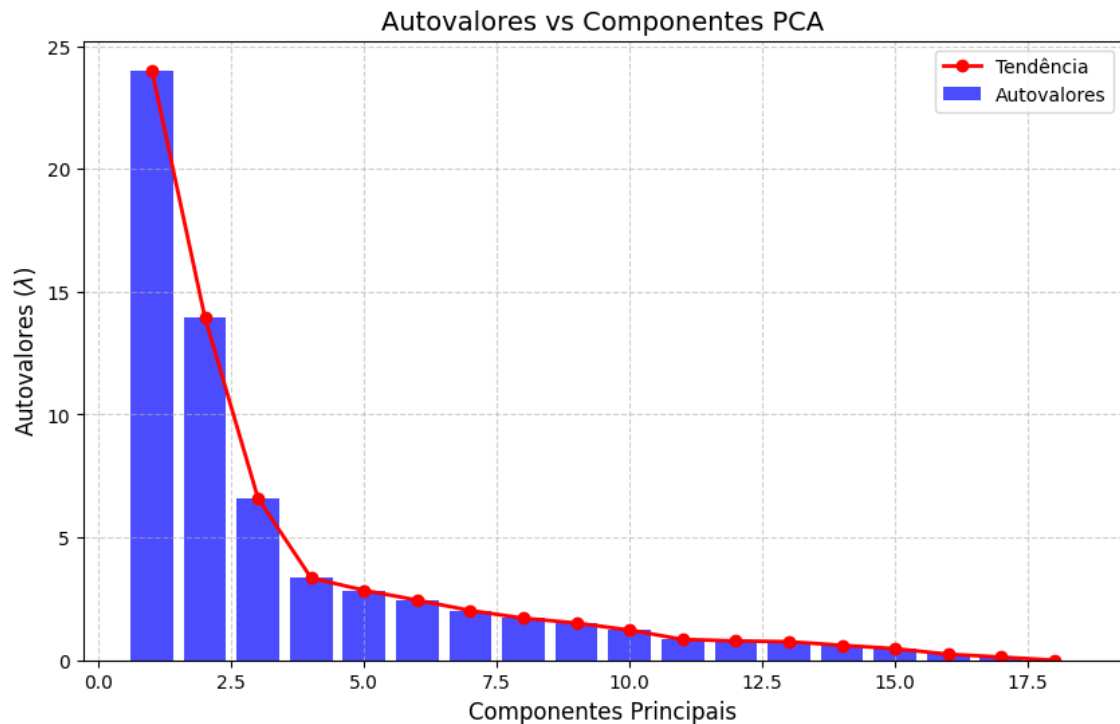
Imagem 2: Representação da matriz que contém as imagens vetorizadas

2.2 Comportamento do PCA com 2 pessoas

Nessa primeira parte do trabalho o objetivo é ter uma análise em dimensão 2 em um subconjunto do dataset original, no qual seria composto por imagens de apenas 2 pessoas do dataset.

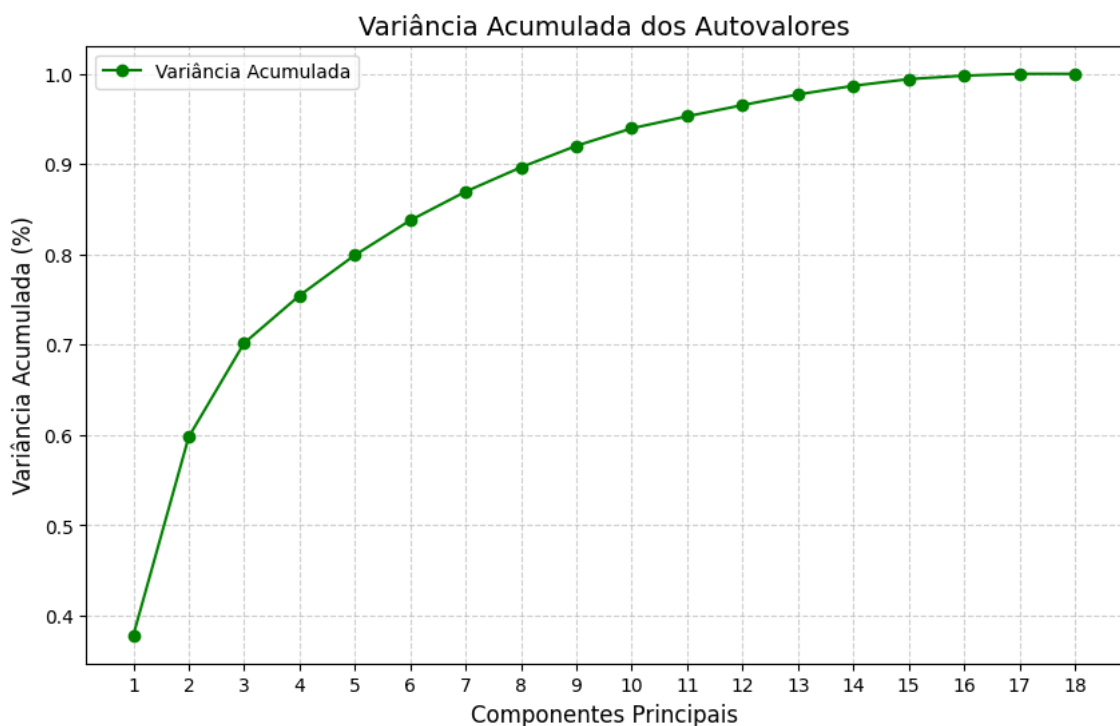
De início, foram escolhidas aleatoriamente 2 IDs distintos do dataset e foram obtidas as imagens das respectivas pessoas pertencentes aos IDs. Além disso, de cada pessoa foi separada uma imagem aleatória para realizar um teste se o PCA seria capaz de identificar de reconhecer de quem seria aquela imagem nesse subconjunto. Dessa forma, nesse subconjunto temos uma matriz **18 x 4096**, com 18 imagens de duas pessoas escolhidas aleatoriamente e separadas estão dois vetores **1 x 4096** para o teste futuro.

Apesar desse experimento abordar apenas o PCA 2, para projetar o conjunto de dados em dimensão 2, foi feita uma análise para observar o quanto dos dados está sendo capturado com as 2 primeiras componentes principais. Dessa forma, no gráfico abaixo foi plotado o “gráfico cotovelo” do valor dos autovalores pelo respectivo número de componentes.



Através do gráfico já é possível ter uma breve noção de que as 2 primeiras componentes capturam uma grande quantidade dos dados originais. Entretanto, é interessante notar que a partir da quarta componente o decrescimento dos autovalores se dá de uma forma mais leve, ou seja, aquela componente possivelmente é um cotovelo do nosso gráfico.

Além do gráfico dos autovalores propriamente ditos, é mais interessante analisar percentualmente o quanto cada componentes capta dos dados originais, sendo assim, o gráfico abaixo representa o percentual acumulado pelo número de componentes selecionado.



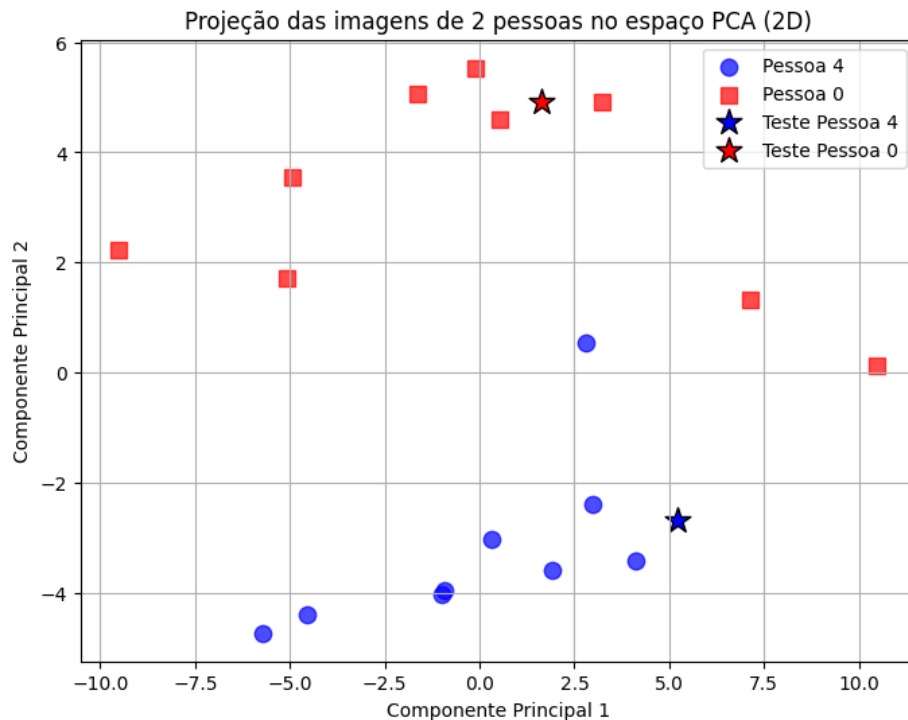
Através do gráfico já é possível perceber que o PCA 2 será capaz de capturar por volta de 60% dos dados originais, o que para essa primeira análise é aceitável. Para confirmação desse valor foi calculado a soma dos dois maiores autovalores e feito a razão dessa soma com a soma de todos os autovalores e isso nos deu um resultado de 59.78%, o que bate com o nosso resultado esperado pelo gráfico.

Agora que já analisamos quando que o PCA 2 capturará dos dados originais, foi utilizado a classe PCA da biblioteca `scikit-learn` para obter as duas componentes principais. Essa classe recebe o número de componentes principais que desejamos obter e possui métodos para projetar o nosso conjunto no espaço PCA 2, além de obter os autovetores e autovalores. Note que as duas primeiras componentes principais obtidas pelo PCA 2 são os 2 autovetores associados aos 2 maiores autovalores, que no contexto da nossa aplicação seriam autofaces, ou em inglês *eigenfaces*. Dessa forma, abaixo está representado as duas *eigenfaces* obtidos pelo PCA 2.



Imagem 3: Eigenfaces obtidas pelo PCA 2

Agora, com as *eigenfaces* obtidas, é necessário projetar todas as linhas do dataset no espaço gerado por elas duas, e após isso, projetar as duas imagens que foram separadas do dataset, mas que se referem as pessoas que estão nele, para testar se o PCA é capaz de identificar quem é quem. Para efeito de explicação, na legenda do gráfico a seguir foi explicitado o ID das pessoas que foram selecionadas de forma aleatório previamente, a pessoa 4 foi o primeiro ID aleatório e a pessoa 0 foi o segundo. Para visualizar essa projeção o gráfico abaixo foi elaborado, no qual está devidamente destacado cada pessoa que estava no dataset e a imagem teste de cada pessoa.



Através do gráfico acima foi possível concluir algumas coisas. A primeira é que o PCA 2 foi capaz de identificar cada imagem de teste próximo ao grupo respectivo das imagens da pessoa que ela pertence, sendo assim, ele foi capaz de reconhecer quem eram aquelas pessoas. A segunda observação percebida, foi a disposição de alguns pontos nesse gráfico, note que apesar das projeções da Pessoa 4 terem ficado bem próximas, 1 projeção se destacou ficando mais distante, o que provavelmente representaria uma imagem dessa pessoa com uma angulação um pouco diferente. Além disso, as projeções da Pessoa 0, apesar de terem ficado relativamente próximas também apresentou certo espalhamento irregular na parte de cima do gráfico.

Essas observações são interessantes de se notar, pois essas são algumas limitações que vamos enfrentar na segunda parte desse trabalho, na qual tentaremos aplicar um reconhecimento facial em todos dataset. Abaixo é mostrado as imagens das pessoas que estavam no dataset e a imagem de teste que foi identificada como pertencente da respectiva pessoa.



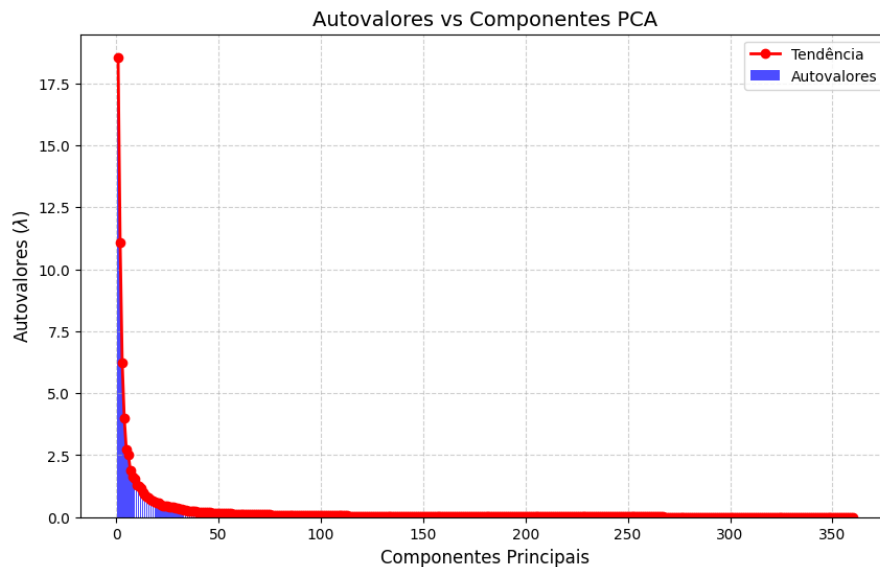
Imagem 4: Imagens do dataset e de teste

2.3 PCA aplicado em todo dataset

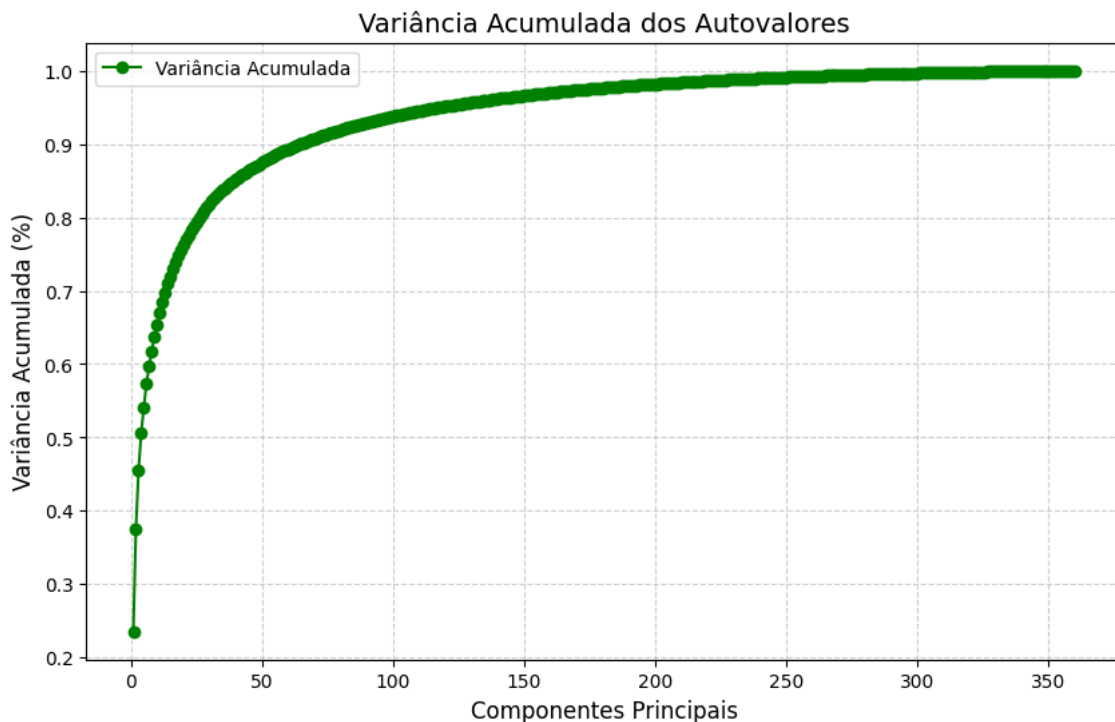
Para a segunda parte do trabalho, o objetivo é selecionar do dataset original algumas fotos para servir de testes de reconhecimento no final. Para isso, será aplicado o PCA em cima do dataset sem as imagens de teste. Além disso, para esse teste, em contrapartida do anterior

no qual definimos 2 componentes principais, nesse buscaremos o número de componentes necessários para manter 95% de integridade dos dados originais.

De início, foi feita a separação de 40 das 400 imagens do dataset original. Cada imagem pertence a uma pessoa do dataset, e não há repetição de pessoas, então serão 40 imagens de pessoas diferentes. Feito essa separação, começa a busca pelo número de componentes principais ideal para capturar 95% do dado original, e para isso, novamente tentaremos analisar os gráficos de autovalores e de variância acumulada. Abaixo está o plot do gráfico de autovalores.



Esse gráfico, diferente da primeira parte desse experimento, não nos auxilia muito, uma vez que há muitas componentes e os valores não são muito visuais de se identificar. Para tal, o gráfico da variância acumulada foi feito também.



Esse gráfico, mesmo não estando com a mesma precisão do gráfico da primeira parte do experimento, devido ao grande número de componentes, já é capaz de nos dizer que o número de componentes para captar cerca 95% dos dados é bem próximo de 100 componentes, então já temos um valor esperado para esse número.

Para validar isso, foi feita a soma percentual de cada componente até obter um acúmulo de 0.95, e a partir disso, foi obtido o valor de 118 componentes para esse nível de captura dos dados originais. Sendo assim, para essa parte do trabalho, trabalharemos com 118 componentes principais.

Sendo assim, o procedimento para obtenção das *eigenfaces* é fornecer o valor $k=118$ a classe do PCA que utilizamos e receber as componentes principais e os autovalores. Para um efeito de visualização, abaixo estão as 4 primeiras *eigenfaces* que foram obtidas.

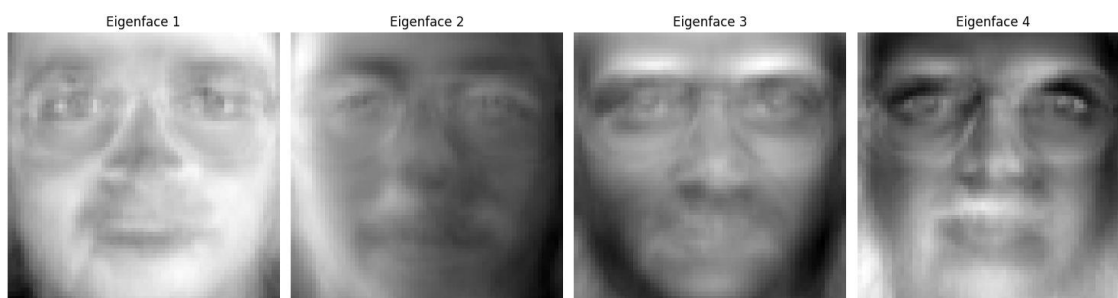


Imagem 5: 4 primeiras eigenfaces

Com o espaço PCA obtido, agora o que deve ser feito é projetar as imagens do dataset nele e após isso projetar as imagens de teste que foram previamente selecionadas. Após isso, a forma que será feita para se identificar a quem pertence àquela é imagem é através da norma euclidiana entre a imagem de teste já projetada no espaço PCA e todas as imagens do dataset que já foram projetadas nele também. A menor distância que for obtida indicará a pessoa a qual aquela imagem de teste está se referindo e para um efeito comparativo será mostrado a imagem mais próxima que foi identificada.

Como foram selecionadas 40 imagens de teste, e esse plot ficaria muito extenso, mostrarei as 10 primeiras imagens de teste e seus pares mais próximos identificados. Nesses 10 primeiros já será possível abordar algumas questões da aplicação do PCA para reconhecimento facial. Sendo assim, abaixo está as 10 primeiras imagens de teste casadas com as imagens mais próximas a elas associadas no espaço PCA.

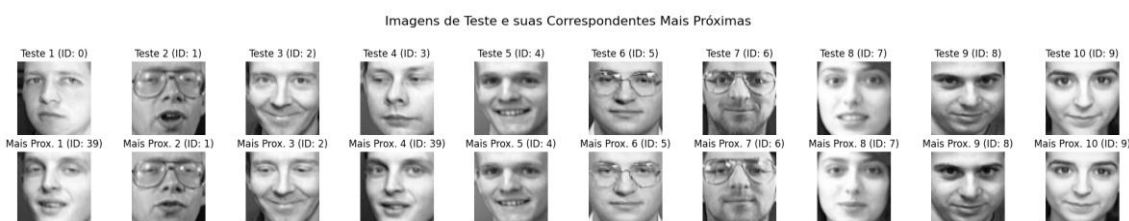


Imagem 6: Pares de imagem teste e imagem mais próxima reconhecida

De antemão, vale destacar que os testes foram feitos com todas as imagens, e o algoritmo de reconhecimento facial acertou corretamente as pessoas em 36 das 40 imagens. Nas imagens mostradas acima, é possível notar que 2 dos erros cometidos pelo PCA estão presentes, porém foram erros interessantes de se notar.

Os testes 0 e 3 erroneamente foram pareados com imagens incorretas, porém vale um destaque maior para esses casos pois apesar do reconhecimento facial ter sido incorreto, a afirmação de que o PCA errou não está totalmente correta. Abaixo estão imagem ampliadas desses dois casos.



Em ambos os casos, as imagens de teste 1 e 4 foram confundidas com imagens da pessoa de ID = 39, porém mais que isso, ambas as imagens têm um certo formato em comum, que é o rosto estar localizado mais a direita da imagem, deixando assim um espaço mais escuro a esquerda, o qual seria o fundo. E isso é identificado pelo PCA em ambos os testes, tanto o posicionamento do rosto quanto a presença desse espaço vazio a esquerda muito provavelmente foram fatores que fizeram com que essas imagens fossem erroneamente identificadas como parecidas. Além disso, o fato de utilizarmos apenas a distância euclidiana para definir essa similaridade com certeza é um fator importante nisso também.

2.4 Comentários e melhorias

Essa sessão é destinada a alguns breves comentários sobre como foi fazer esse trabalho e algumas possíveis melhorias a serem implementadas futuramente. Explorar essa aplicação de reconhecimento facial através das *eigenfaces* foi uma ótima experiência para aprofundar meu conhecimento sobre PCA, autovetores e autovalores. Sinto que todas as plots e as considerações que fui fazendo foram muito associadas às aulas que tivemos e acredito que consegui replicar bastante as técnicas ensinadas pelo professor João Paixão. Foi um ótimo momento para analisar de forma crítica também, haja vista os erros que o PCA cometeu no reconhecimento facial e buscar entender o porquê desses erros estarem acontecendo.

Ademais, acredito que esse trabalho apesar de abordar bastante conteúdo, ainda poderia ser incrementado com uma definição melhor de um limiar de similaridade entre as imagens para assim ser possível definir de maneira mais correta a quem pertence tal imagem de teste. Além de, é claro, permitir o teste com imagens de pessoas fora do dataset, para que o algoritmo seja capaz não só de identificar bem alguém que está no dataset, mas conseguir saber quando alguém não está presente também, uma função bem útil pensando em aplicações de reconhecimento facial.

3. Conclusão

Por fim, o resultado desse projeto, por mim, é considerado um sucesso. Foi possível explorar a aplicação de um algoritmo tão valioso como é o PCA nas técnicas de reconhecimento facial e foi possível entender também as limitações que ele possui, uma vez que ele apesar de ter acertado em 90% dos casos ainda cometeu erros devido a estruturação das imagens do dataset. Então, é um algoritmo bem robusto e foi possível explorar excelentemente a sua aplicação.