**Introdução**

Olá nos somos, os (nossos nomes) e somos No name is still th best name,

E vamos apresentar a reconstrução facial utilizando eigenfaces.

Neste trabalho foi proposto que o grupo construísse um projeto em java para fazer reconhecimento facial para a **European inteligence agency**, esta linguagem de programação permite que esta organização pudesse introduzir a sua base de dados de imagens e introduzindo de uma nova imagem deveria poder identificar a imagem da pessoa na sua base de imagens que se assemelha melhor, permitindo assim a que com uma imagem nova fosse possível identificar a identidade dessa pessoa.

Neste trabalho tivemos uma contextualização teórica em que nos foram dadas formulas como a da decomposição própria de uma matriz, reconstrução de imagens e reconhecimento facial. Todas as imagens que estão na nossa base imagens formam matrizes pois estão numa escala de cinzento de 0 até 255 em termos de cor isto permitindo que seja criada uma matriz com a quantidade de pixels das imagens

**Teoria**

**Decomposição própria:** As fórmulas que utilizamos nesta primeira funcionalidade da decomposição própria foram estas. A primeira formula representa a decomposição própria em valores e vetores próprios.

Neste momento torna-se importante referir o que são valores e vetores próprios, os valores próprios dizem quais são as partes mais importantes de uma cara e os vetores próprios mostram os padrões que identificam as diferenças em cada uma das caras.

Esta formula pode ser usada de duas maneiras, pois o utilizador tem a possibilidade de escolher a quantidade e valores próprios que quer usar sendo que estas podem ser representados como K , esta segunda formula vai ser usada para calcular o Erro Absoluto medio este é a subtração que é feita para todos as imagens em que utilizados a formula anterior com k eigenfaces e a imagem sem esta restrição, isto quer dizer que se uma matriz com colunas e linhas com valor de 64 quanto menor for o K introduzido pelo utilizador maior vai ser o erro absoluto medio.

**Reconstrução de imagens:** A reconstrução facial é a criação de um rosto usando apenas as informações mais importantes, isto é, escolher quais dos valores próprios são os mais importantes. O processo passa pelo cálculo da face media e a centralização de imagens em que o objetivo é juntar todas as imagens da base de imagens num só sítio. A matriz de covariâncias vai criar um valor novo que vem da subtração entre a centralização da imagem nova na face media da base de imagens, o peso associado representa a contribuição de cada eigenface para descrever uma imagem especifica

**Reconhecimento facial:** É como o sistema identifica e compara a imagem nova que foi introduzida e avalia qual é a imagem mais próxima na base de imagens disponível. Vai fazer a soma da imagem media com os pesos de todas as imagens que foi reconstruída com as restantes imagens, e por fim utilizar a distância euclidiana com os pesos de todas as imagens para encontrar aquele que é mais próxima

**Funcionalidades**

Em java as funcionalidades foram implementar a funcionar desta maneira:

**Funcionalidade 1:** utiliza a base de imagens que estão como CSV de forma a introduzir os dados de todas na aplicação estrai todos os valores e vetores próprios, reconstrói a matriz e calcula o erro absoluto medio,

**Funcionalidade 2:** O processo da funcionalidade 2 vai utilizar todas as imagens que já foram lidas e processadas na funcionalidade 1 para criar a face media centralizar as imagens, calcular os valores e vetores próprios normalizar os seus valores, ou seja, fazer com que os valores dentro das imagens fiquem entre 0 e 255 que tal como foi dito é como são os valores com que as imagens são introduzidas e ver os pesos de todas as imagens na base de imagens

**Conclusão**

Como conclusão decidimos fazer algo diferente identificando as palavras que sentimos que definem este trabalho.

**Eigenfaces,** este processo permitiu conhecer mais sobre um dos processos utilizados para reconhecimento facial, e que levou o grupo a ter contacto com algo que tendo vindo a ser cada vez mais usado.

**Álgebra:** as eigenfaces estariam incompletas sem que tivéssemos aprendido o processo algébrico de como funcionam, serviu para que tivéssemos um conhecimento teórico da utilização deste processo e de com funciona com uma grande profundidade teórica

Github e scrum: decidimos também utilizar como palavras que definem este trabalho o github e o scrum pois sem esta ferramenta e este método de trabalho teria sido muito difícil trabalhar em equipa, e aplicar todos os conhecimentos conjuntos que o grupo possui.

JAVA: esta linguagem foi a que foi mais utilizada e apesar de ser uma linguagem difícil, conseguimos utilizada para levar este projeto até ao final e com sucesso, sendo que, ao utilizá-la ganhamos maior conhecimento da sua aplicação no mundo real e ferramentas para que a consigamos implementar em projetos futuros.

Dificuldades: Ao longo do projeto, todo o grupo passou por dificuldades constantes, em que todo o projeto parecia, ser de uma grande complexidade, e aplicar o conceitos de álgebra em java parecia não ser possível. Apesar disso e com a aplicação do Scrum, pesquisa, e aplicação de conhecimentos aprendidos ao longo do semestre estas dificuldades foram diminuindo, e o projeto foi sendo desenvolvido até estar terminado.

Isto serve para dizer que conseguimos ultrapassar as dificuldades e fazer o que foi proposto pelo cliente da melhor maneira possível.