DOCENTE: Rui Jesus

**3º Trabalho prático**

**2425SV**

|  |  |
| --- | --- |
| Aluno | Número |
| Vitor Dinis | 46327 |

*XX de XXX de 2025*

Índice

[1. Introdução 1](#_Toc194891443)

[2. Abordagem 2](#_Toc194891444)

[2.1. Deteção e normalização das faces num *frame* 2](#_Toc194891445)

[2.2. Identificação e classificação de faces 3](#_Toc194891446)

[2.3. Adição de objetos virtuais 6](#_Toc194891447)

[3. Resultados finais 8](#_Toc194891448)

[4. Conclusões 9](#_Toc194891449)

[5. Bibliografia 10](#_Toc194891450)

Índice de Figuras

[Figura 1 - Deteção da posição dos olhos numa face, com o MediaPipe 2](#_Toc194891451)

[Figura 2 - Comparação do resultado da reconstrução com a imagem original 4](#_Toc194891452)

[Figura 3 – Teste de classificação de imagem 6](#_Toc194891453)

[Figura 4 – Teste com todos os objetos virtuais aplicados a uma face 7](#_Toc194891454)

[Figura 5 - Teste ao desenho de objetos virtuais em faces com inclinação 7](#_Toc194891455)

# Introdução

No presente relatório consta um resumo de todo o trabalho realizado ao longo do 1º Trabalho prático da Unidade Curricular de Visão Artificial e Realidade Mista.

O principal objetivo deste primeiro trabalho prático foi a implementação de um *Vocoder*.

Para tal, recorreu-se à linguagem de programação Python [1], assim como às bibliotecas

Exemplo VARM: OpenCV (Open-Source Computer Vision) [2], para processamento de imagem, e dlib [3], para reconhecimento facial.

# Aplicação

Escrever algo aqui …

# Descrição app

Escrever algo aqui …

# Tarefas

Escrever algo aqui …

# Peritos

Escrever algo aqui …

# Contribuição 1 – Nome1

Escrever algo aqui …

# Contribuição 2 – Nome2

Escrever algo aqui …

# Contribuição 3 – Nome3

Escrever algo aqui …

# Peritos

Escrever algo aqui …

# Contribuição 1 – Nome1

Escrever algo aqui …

# Contribuição 2 – Nome2

Escrever algo aqui …

# Contribuição 3 – Nome3

Escrever algo aqui …

# Conclusões

Podemos concluir que foram atingidos os objetivos propostos, foi implementado um Vocoder simples capaz de sintetizar fala de forma satisfatória …

Exemplo VARM:

a aplicação implementada cumpre com os objetivos propostos no enunciado do trabalho. As minhas maiores dificuldades estiveram na implementação dos algoritmos para gerar as *Eigenfaces* e *Fisherfaces*, na utilização das suas matrizes para construir o classificador e no treino do modelo.

Para o futuro, gostaria de melhorar o treino do modelo para a identificação facial, para tentar obter resultados mais consistentes. Além disso, seria interessante testar implementar o modelo de deteção facial do MediaPipe, para ver se teria alguma melhoria significativa na performance, e tentar otimizar a aplicação na componente de desenho dos objetos virtuais.

# Bibliografia

|  |  |
| --- | --- |
| [1] | Python Software Foundation, “Python,” 2001-2025. [Online]. Available: https://www.python.org/. |
| [2] | OpenCV team, “OpenCV - Open Computer Vision Library,” 2025. [Online]. Available: https://opencv.org/. [Acedido em Feb. 2025]. |
| [3] | Dlib, “dlib c++ Library,” 8 Mai. 2022. [Online]. Available: https://dlib.net/. [Acedido em Fev. 2025]. |
| [4] | Google, “Face detection guide | Google AI,” Jan. 2025. [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/vision/face\_detector. [Acedido em Fev. 2025]. |
| [5] | A. J. Abrantes, *Face Recognition,* 2025. |
| [6] | “scikit-learn: machine learning in Python,” [Online]. Available: https://scikit-learn.org/stable/. [Acedido em Abr. 2025]. |