



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Bauru

Carlos Eduardo Rossi Cubas da Silva

Animação facial baseada em *blendshapes* utilizando Floresta de Caminhos
Ótimos e Floresta Aleatória

Bauru
2022

Carlos Eduardo Rossi Cubas da Silva

Animação facial baseada em *blendshapes* utilizando Floresta de Caminhos
Ótimos e Floresta Aleatória

Tese de doutorado apresentado ao Programa de Pós-Graduação em Mídia e Tecnologia da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de Bauru como requisito à obtenção do título de “Doutor em Mídia e Tecnologia”.

Orientador: Dr. Antonio Carlos Sementille

Bauru
2022

Silva, Carlos Eduardo Rossi Cubas da
Animação facial baseada em *blendshapes* utilizando
Floresta de Caminhos Ótimos e Floresta Aleatória / Carlos
Eduardo Rossi Cubas da Silva - Bauru, 2022.
124f. : il., tabs.

Orientador: Dr. Antonio Carlos Sementille.

Tese (Doutorado) – Universidade Estadual Paulista
(Unesp). Faculdade de Arquitetura, Artes, Comunicação e
Design, Bauru, 2022.

1. Captura de movimentos faciais. 2. *Blendshape*. 3.
Aprendizado de máquina. 4. Floresta aleatória 5. Floresta de
caminhos ótimos. 6. Mídia e tecnologias. I. Dr. Antonio Car-
los Sementille. III. Universidade Estadual Paulista (Unesp).
Faculdade de Arquitetura, Artes, Comunicação e Design,
Bauru, 2022. II. Título.

ATA DA DEFESA PÚBLICA DA TESE DE DOUTORADO DE CARLOS EDUARDO ROSSI CUBAS DA SILVA, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM MÍDIA E TECNOLOGIA, DA FACULDADE DE ARQUITETURA, ARTES, COMUNICAÇÃO E DESIGN.

Aos 31 dias do mês de outubro do ano de 2022, às 14:30 horas, no(a) Auditório da Seção Técnica de Pós-Graduação da FAAC, realizou-se a defesa de TESE DE DOUTORADO de CARLOS EDUARDO ROSSI CUBAS DA SILVA, intitulada **Animação facial baseada em blendshapes utilizando Floresta de Caminhos Ótimos e Floresta Aleatória**. A Comissão Examinadora foi constituída pelos seguintes membros: Professor Associado ANTONIO CARLOS SEMENTILLE (Orientador(a) - Participação Presencial) do(a) Departamento de Computação / UNESP/Câmpus de Bauru, Prof. Dr. WILSON MASSASHIRO YONEZAWA (Participação Presencial) do(a) Departamento de Computação da Faculdade de Ciências / Universidade Estadual Paulista, Prof. Dr. SILVIO RICARDO RODRIGUES SANCHES (Participação Virtual) do(a) Departamento Acadêmico de Computação / Universidade Tecnológica Federal do Paraná. Após a exposição pelo doutorando e arguição pelos membros da Comissão Examinadora que participaram do ato, de forma presencial e/ou virtual, o discente recebeu o conceito final: APROVADO. Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelo(a) Presidente(a) da Comissão Examinadora.

Professor Associado ANTONIO CARLOS SEMENTILLE

Antonio C. Sementille

Agradecimentos

Agradeço, primeiramente, à minha família por toda a dedicação e paciência contribuindo diretamente para que eu pudesse ter um caminho mais fácil e prazeroso durante esses anos de estudo.

Agradeço ao meu orientador, professor Antonio Carlos Sementille, que soube conduzir com maestria o meu doutorado e me ajudou a finalizar este trabalho. Sem ele, isso seria impossível. Além disso, todos os ensinamentos, as aulas, as reuniões, eu sentirei saudade.

Aos meus companheiros (as) do PPGMiT, que compartilharam comigo essa experiência acadêmica (estudantes, funcionários e professores), muito obrigado pela parceria. Agradeço aos colegas do laboratório *Recogna - Biometric amp; Pattern Recognition*, em especial aos professores João Paulo Papa e Clayton, que me ajudaram muito dando suporte, sugestões e ajudando com equipamentos do laboratório.

Agradeço também aos colegas do laboratório, SACI (Sistemas Adaptativos e Computação Inteligente), em especial, ao meu amigo Everton, que foi um colega próximo das longas noites de pesquisa e todos que por ali passaram e me ajudaram com dicas de suporte, para as implementações feitas na pesquisa.

Aproveito para agradecer ao professor Robson Fernandes, que me ajudou muito no começo das pesquisas me guiando em um tema que para mim era distante, o de redes neurais e classificadores. Com a sua ajuda, consegui seguir em frente e, com certeza, foram vitais para o desenvolvimento desta pesquisa.

Uma agradecimento especial a Gisele, minha companheira, que me deu forças para continuar e esteve ao meu lado em todos os momentos. Muito grato pela ajuda e compreensão.

*“Deus nunca disse que a jornada seria fácil,
mas Ele disse que a chegada valeria a pena.”*
(Max Lucado)

Resumo

Nas últimas décadas, o interesse quanto à captura de movimentos da face humana e à identificação de suas expressões, com a finalidade de geração de animações faciais realistas, tem aumentado, tanto na comunidade científica quanto na indústria do entretenimento. A alta acurácia nesse processo é necessária, pois os humanos são naturalmente treinados para identificar expressões faciais, detectando facilmente pequenas imperfeições na animação de uma face virtual. O redirecionamento de movimento facial é uma das técnicas utilizadas para gerar animações realistas, principalmente em filmes. Muitos sistemas foram desenvolvidos e consistem, de maneira geral, de uma fase de modelagem, captura de movimentos e identificação de expressões faciais, seguido de um procedimento de redirecionamento de expressões. Para o redirecionamento são aplicadas várias técnicas, onde algoritmos de cálculo de distância ou redes neurais são utilizados, fazendo com que a face final gerada no processo seja próxima a da captura. Nesse sentido, a utilização de novas técnicas é extremamente útil permitindo que a acurácia do processo e a velocidade sejam melhoradas. Considerando o contexto exposto, o presente trabalho tem como principal objetivo implementar e validar um sistema de animação facial, o qual utiliza um modelo de classificador supervisionado capaz de identificar as expressões mais comuns da face humana e redirecioná-las, de forma automática, a um modelo de face gerado a partir da combinação de um pequeno número de *blendshapes* (mistura de formas). Com isso, foi gerado um sistema de animação facial que, através da captura de faces com câmeras RGB e usando como referências apenas informações 2D, conseguiu gerar expressões próximas a da captura utilizando classificadores mais leves, possibilitando o uso em dispositivos acessíveis com *tablets*, celulares e *desktops* de baixo custo.

Palavras-chaves: Captura de movimentos faciais, *Blendshapes*, *OPF*, Aprendizado de máquina, Floresta aleatória, Floresta de caminhos ótimos, *Retarget*.

Abstract

In the last decades, the interest in capturing human face movements and identifying their expressions, in order to generate realistic facial animations, has increased, both in the scientific community and in the entertainment industry. High accuracy in this process is necessary, as humans are naturally trained to identify facial expressions, easily detecting small imperfections in the animation of a virtual face. Facial motion redirection is one of the techniques used to generate realistic animations, especially in movies. Many systems have been developed and generally consist of a modeling, motion capture and facial expression identification phase, followed by an expression redirection procedure. For the redirection, several techniques are applied, where distance calculation algorithms or neural networks are used, making the final face generated in the process close to the capture. In this sense, the use of new techniques is extremely useful, allowing the process accuracy and speed to be improved. Considering the above context, the main objective of this work is to implement and validate a facial animation system, which uses a supervised classifier model capable of identifying the most common expressions of the human face and automatically redirecting them to a face model generated from the combination of a small number of blendshapes. With this, a facial animation system was generated that, through the capture of faces with RGB cameras and using only 2D information as references, managed to generate expressions close to the capture using lighter classifiers, allowing the use in accessible devices with tablets, cell phones and low-end desktops.

Keywords: Facial motion capture, *Blendshapes*, *OPF*, Machine Learning, *Random Forest*, *Retarget*, *Optimum Path Forest*.

Lista de ilustrações

Figura 1.1 – Personagem Gollum da franquia Senhor dos Anéis.	18
Figura 1.2 – Exemplo da deformação de malha.	19
Figura 1.3 – Exemplo de <i>blendshapes</i>	20
Figura 1.4 – Exemplo da criação de uma face utilizando a combinação de vários <i>blendshapes</i> . A primeira face, da esquerda, está em expressão neutra e as seguintes, possuem expressões diferentes. A última, da direita, mostra o resultado final da soma entre elas.	20
Figura 1.5 – Exemplo da utilização de marcadores com tinta refletiva.	21
Figura 1.6 – Exemplo da utilização de marcadores com tinta refletiva.	21
Figura 2.1 – Conjunto de expressões para o refinamento dos <i>blendshapes</i>	27
Figura 2.2 – Formas de <i>blendshapes</i>	28
Figura 2.3 – Matriz-vetor que corresponde às expressões do modelos de <i>blendshapes</i>	29
Figura 2.4 – Captura de uma face usando um dispositivo de captura 3D.	30
Figura 2.5 – Exemplo de modelo morfológico. (a) Foto de uma face 2D; (b) Modelo 3D gerado em posição frontal; (c) Modelo 3D gerado em posição lateral.	30
Figura 2.6 – Transferência de expressões entre modelos 3D.	31
Figura 2.7 – Módulos de uma arquitetura de <i>retarget</i>	31
Figura 2.8 – Exemplo de um conjunto de expressões gerado na fase <i>offline</i>	32
Figura 2.9 – Desenvolvido pelos psicólogos Paul Ekman e Wallace Friesen, os <i>FACS</i> possuem trinta e três unidades de ação (<i>AUs</i>) (por exemplo, as rugas da testa são uma unidade de ação).	32
Figura 2.10 – Marcadores faciais extraídos do <i>Dataset</i> na fase <i>offline</i>	33
Figura 2.11 – Faces virtuais geradas pelo processo de redirecionamento de movimento facial.	33

Figura 3.1 – (a) Grafo completo ponderado nas arestas, para um determinado conjunto de treinamento. (b) MST do grafo completo. (c) Protótipos escolhidos como sendo os elementos adjacentes de classes diferentes na MST (nós ponto estrela). (d) Floresta de caminhos ótimos resultante para a função de valor de caminho f_{max} e dois protótipos. Os identificadores (x, y) , acima dos nós, são respectivamente, o custo e o rótulo dos mesmos. A seta indica o nó predecessor no caminho ótimo. (e) Uma amostra de teste (pergaminho) é conectada aos demais nós de ambas as classes (linhas pontilhadas), com os nós do conjunto de treinamento. (f) O caminho ótimo do protótipo mais fortemente conexo, seu rótulo 2 e o custo de classificação 0.4 são associados a amostra de teste. Note que, mesmo a mostra de teste estando mais próxima de um nó da classe 1, ela foi classificada como sendo da classe 2.	38
Figura 3.2 – Ilustração da lógica por trás do algoritmo Florestas Aleatórias.	43
Figura 3.3 – Etapas utilizadas pelo algoritmo de Florestas Aleatórias.	44
Figura 3.4 – Exemplo de uma matriz de confusão.	45
Figura 4.1 – Visão geral do método proposto por Weise et al. (2011b). Os pesos dos blendshapes são estimados pela correspondência da textura da imagem 2D e o mapa de profundidade 3D.	47
Figura 4.2 – Visão geral do processo.	48
Figura 4.3 – Visão geral do processo.	49
Figura 4.4 – Visão geral do processo.	50
Figura 4.5 – Visão geral do método proposto.	51
Figura 4.6 – Visão geral do método proposto.	52
Figura 4.7 – Visão geral da estrutura de módulos. As linhas em azul representam os módulos usados desenvolvidos.	54
Figura 4.8 – Interface do módulo ajuste dos AUS.	55
Figura 4.9 – Resultado do método proposto por Chaudhuri, Vesdapunt e Wang (2019).	56
Figura 4.10–Visão geral do método proposto por Nishimura et al. (2019).	57
Figura 4.11–Visão geral do método proposto por Abdelaziz et al. (2020).	58
Figura 5.1 – Estruturação do método proposto desenvolvido.	61
Figura 5.2 – Etapas para a criação da base de dados.	63
Figura 5.3 – Exemplos de algumas imagens coletadas de vídeos da base de dados FEAF+.	63
Figura 5.4 – Exemplos de algumas imagens coletadas de vídeos da base de dados MUG.	64
Figura 5.5 – Exemplos de algumas imagens coletadas de vídeos da base de dados KDEF.	65
Figura 5.6 – Exemplos de algumas imagens coletadas de vídeos da base de dados CK+.	65

Figura 5.7 – Pré-processamento da base de dado de imagens.	66
Figura 5.8 – Exemplo dos pontos detectados pela biblioteca <i>Dlib</i> na extração dos marcadores faciais.	67
Figura 5.9 – Exemplo da extração dos marcadores faciais nas imagens das bases de dados pré-processadas.	67
Figura 5.10–Exemplo de desalinhamento das faces na base de imagens pré processadas.	68
Figura 5.11–Translação, alinhamento e escala do conjunto de dados usando como referência uma das amostras.	69
Figura 5.12–Posição média dos marcadores faciais gerados pelo processo de normalização utilizada pelo algoritmo de Procrustes.	69
Figura 5.13–As três regiões da face e os pontos escolhidos em cada região.	70
Figura 5.14–Características definidas através do cálculo da distância entre um ponto fixo e os demais pontos das áreas da face.	71
Figura 5.15–Geração dos modelos não supervisionados das características das imagens. Foram gerados 3 modelos, um para cada região da face.	72
Figura 5.16–Etapas para a geração da base de dados características de <i>blendshapes</i> .	73
Figura 5.17– <i>Software FaceGen Modeller 3.3</i>	74
Figura 5.18–Base de dados de <i>blendshapes</i> . Algumas expressões são baseadas em AUS, enquanto outras, são expressões básicas ou combinações de AUS.	74
Figura 5.19–Face 3D customizada gerada a partir de uma única foto.	75
Figura 5.20–Imagens exportadas dos <i>blendshapes</i> usando a textura da imagem de entrada.	76
Figura 5.21–Imagens selecionadas por região da face. (a) Imagem que corresponde a região da boca; (b) Imagem que corresponde a região dos olhos; (c) Imagem que corresponde a região das sobrancelhas.	76
Figura 5.22–Processo de rotulação da base de dados de amostras de imagens baseado nos rótulos dos <i>blendshapes</i>	78
Figura 5.23–Etapas para a criação da base de dados supervisionada.	79
Figura 5.24–Aplicando o modelo treinado à captura em tempo real.	80
Figura 6.1 – <i>Blendshapes</i> classificados para região da boca.	83
Figura 6.2 – <i>Blendshapes</i> classificados para região dos olhos.	83
Figura 6.3 – <i>Blendshapes</i> classificados para região das sobrancelhas.	84
Figura 6.4 – Matriz de confusão do modelo da região da boca do classificador de <i>Floresta Aleatória</i>	84
Figura 6.5 – Proximidade entre características das classes AU14 e AU24.	85
Figura 6.6 – Semelhanças entre as classes AU16 e AU25.	86
Figura 6.7 – Matriz de confusão do modelo da região dos olhos do classificador de <i>Floresta Aleatória</i>	87
Figura 6.8 – Semelhanças entre as classes AU06 e AU07.	88

Figura 6.9 – Matriz de confusão do modelo da região das sobrancelhas do classificador de Floresta Aleatória.	89
Figura 6.10–Matriz de confusão do modelo da região da boca do classificador OPF.	90
Figura 6.11–Matriz de confusão do modelo da região dos olhos do classificador OPF.	92
Figura 6.12–Matriz de confusão do modelo da região das sobrancelhas do classificador OPF.	93
Figura 6.13–Exemplo de comparação entre a face 3D de referência (<i>ground truth</i>) e a face 3D gerada pelo <i>pipeline</i> desenvolvido.	95
Figura 6.14–Exemplo de comparação entre a imagem da face do ator (<i>ground truth</i>) e a imagem do <i>blendshape</i> gerado pelo <i>pipeline</i>	96
Figura 6.15–Desempenhos dos classificadores por expressões.	97
Figura 6.16–Desempenhos dos classificadores pela expressão de raiva.	98
Figura 6.17–Desempenhos dos classificadores pela expressão de desgosto.	99
Figura 6.18–Desempenhos dos classificadores pela expressão de medo.	100
Figura 6.19–Desempenhos dos classificadores pela expressão de felicidade.	100
Figura 6.20–Desempenhos dos classificadores pela expressão de tristeza.	101
Figura 6.21–Desempenhos dos classificadores pela expressão de surpresa.	101
Figura 6.22–Capturas dos quadros dos vídeos de medo e surpresa.	102
Figura 6.23–Capturas dos quadros dos vídeos da expressão raiva.	103
Figura 6.24–Exemplo de desempenhos do classificador OPF na região da boca.	103
Figura 6.25–Comparação do tempo de processamento entre os classificadores de Floresta Aleatória e OPF.	104
Figura 6.26–Comparação do tempo de processamento entre os classificadores de Floresta Aleatória e OPF para a região da boca.	104
Figura 6.27–Comparação do tempo de processamento entre os classificadores de Floresta Aleatória e OPF para a região dos olhos.	105
Figura 6.28–Comparação do tempo de processamento entre os classificadores de Floresta Aleatória e OPF para a região das sobrancelhas.	105
Figura 6.29–Exemplo de captura e saídas diferentes utilizando apenas um <i>blendshape</i> por área da face.	106
Figura 6.30–Exemplos de capturas de uma faces feminina para um <i>blendshape</i> masculino.	107
Figura 6.31–Exemplo de uma captura de uma face feminina oriental para um <i>blendshape</i> masculino ocidental.	107
Figura 6.32–Exemplo de uma captura de uma face masculina oriental para um <i>blendshape</i> masculino ocidental.	108

Lista de tabelas

Tabela 1 – Características principais dos métodos levantados.	58
Tabela 2 – Separação dos <i>Blendshapes</i> por região usando a referência das AUS como rótulos.	77
Tabela 3 – Quantidade de amostras rotuladas por região.	78
Tabela 4 – Métricas do modelo para região da boca do classificador de Floresta Aleatória.	86
Tabela 5 – Métricas do modelo da região da olhos do classificador de Floresta Aleatória.	88
Tabela 6 – Métricas do modelo da região das sobrancelhas do classificador de Floresta Aleatória.	89
Tabela 7 – Métricas da região da boca do classificador <i>OPF</i>	91
Tabela 8 – Métricas da região da olhos do classificador <i>OPF</i>	91
Tabela 9 – Métricas da região das sobrancelhas do classificador <i>OPF</i>	92
Tabela 10 – Comparação da métrica de <i>recall</i> entre os classificadores para a região da boca.	93
Tabela 11 – Comparação da métrica de <i>recall</i> entre os classificadores para a região dos olhos.	94
Tabela 12 – Comparação da métrica de <i>recall</i> entre os classificadores para a região das sobrancelhas.	94
Tabela 13 – Comparação do desempenho dos classificadores através da análise da porcentagem dos quadros melhores.	102
Tabela 14 – Análise de desempenho dos classificadores dos quadros do centro do vídeo.	103

Lista de abreviaturas e siglas

2D	<i>Bi-Dimensional</i>
3D	<i>Tri-Dimensional</i>
3DMM	<i>3D morphable model</i>
AAM	<i>Active Appearance Model</i>
AI	<i>Artificial intelligence</i>
AVI	<i>Audio Video Interleave</i>
AU	<i>Action Unit</i>
AUS	<i>Action Unit System</i>
CNN	<i>Convolutional Neural Networks</i>
DE	<i>Distância Euclidiana</i>
DNN	<i>Deep neural networks</i>
FACS	<i>Facial Action Coding System</i>
FCP	<i>Facial Characteristic Point</i>
FPS	<i>Frame per second</i>
GRU	<i>Gated Recurrent Unit</i>
LSTM	<i>Long Short Term Memory</i>
IA	<i>Inteligência Artificial</i>
MPPCA	<i>Probabilistic Principal Component Analyzers</i>
MLP	<i>Multi-Layer Perceptron</i>
NME	<i>Normalized Mean Error</i>
OPF	<i>Optimum-Path Forest</i>
PCA	<i>Principal Component Analysis</i>
P-OPF	<i>Probabilistic Optimum-Path Forest</i>

RNN	<i>Recurrent Neural Networks</i>
RBF	<i>Radial basis function</i>
RF	<i>Random Forest</i>
RGB	<i>Red, Green, Blue</i>
RGB-D	<i>Red, Green, Blue and Depth</i>
SDM	<i>Supervised Descent Method</i>
SVM	<i>Support Vector Machines</i>

Sumário

1	INTRODUÇÃO	18
1.1	Os problemas	22
1.2	Hipóteses	23
1.3	Objetivos	24
1.3.1	Objetivo geral	24
1.3.2	Objetivos específicos	24
1.4	Contribuição da pesquisa	24
1.4.1	Escopo da pesquisa	25
1.5	Organização da tese	25
2	BLENDSHAPES E SISTEMAS DE ANIMAÇÃO FACIAL BASE- ADO EM PERFORMANCE	27
2.1	Blendshapes	27
2.1.1	Criação de expressões com <i>blendshapes</i>	28
2.1.2	Construção de <i>blendshapes</i>	29
2.2	Sistemas de animação facial baseado em performance	30
2.2.1	Processo <i>offline</i>	31
2.2.2	Processo <i>online</i>	33
3	APRENDIZADO DE MÁQUINA, CLASSIFICADOR DE FLORESTA DE CAMINHOS ÓTIMOS E FLORESTA ALEATÓRIA	34
3.1	Aprendizado de máquina	34
3.1.1	Classificação multiclasse	34
3.2	Classificador de padrões baseado em Florestas de Caminhos Ótimos - OPF	35
3.2.1	Classificação supervisionada	35
3.2.2	Fundamentação Teórica	35
3.2.3	Algoritmo de poda do conjunto de treinamento	39
3.3	Classificador <i>k-Means</i>	41
3.4	Algoritmos de Árvore de decisão	42
3.4.1	Classificador de Florestas Aleatórias (<i>Random Forest</i>)	42
3.5	Métricas de validação para os classificadores	43
4	TRABALHOS RELACIONADOS	47
4.1	Método de Weise et al. (2011b)	47
4.2	Método de Li et al. (2013b)	48

4.3	Método de Seol, Ma e Lewis (2016)	48
4.4	Método de Pham et al. (2016)	50
4.5	Método de Aneja et al. (2018)	51
4.6	Método de Ekmen e Ekenel (2018)	52
4.7	Método de Struijk et al. (2018)	53
4.8	Método de Chaudhuri, Vesdapunt e Wang (2019)	55
4.9	Método de Nishimura et al. (2019)	56
4.10	Método de Abdelaziz et al. (2020)	57
4.11	Considerações finais sobre os métodos apresentados	58
5	O SISTEMA DE ANIMAÇÃO FACIAL PROPOSTO	60
5.1	Visão geral do método desenvolvido	60
5.2	Visão detalhada do método	62
5.2.1	Criação da base de dados de marcadores faciais	62
5.2.1.1	Seleção das bases de dados de faces	62
5.2.1.2	Extração dos marcadores faciais	66
5.2.1.3	Normalização dos dados	67
5.2.1.4	Definição das características	68
5.2.1.5	Agrupamento das amostras	72
5.3	Criação da base de dados de <i>Blendshapes</i>	73
5.3.1	Geração da base de dados de <i>blendshapes</i>	73
5.3.2	Geração da base de dados de imagens de <i>blendshapes</i>	75
5.3.3	Separação das imagens dos <i>blendshapes</i> por região da face	75
5.3.4	Geração da base de dados de amostras de <i>blendshapes</i>	77
5.4	Rotulação das características das imagens	77
5.5	Treinamento dos modelos supervisionados	79
5.6	Utilizando o modelo supervisionado	79
6	EXPERIMENTOS E ANÁLISES DOS RESULTADOS	81
6.1	Ambiente experimental	81
6.1.1	Hardware	81
6.1.2	Software	81
6.2	Métrica de validação	82
6.3	Análise dos classificadores de Floresta Aleatória e OPF	82
6.3.1	Análise do classificador de Florestas Aleatórias supervisionado	84
6.3.2	Análise do classificador <i>OPF</i> supervisionado	89
6.3.3	Comparação entre os classificadores de Floresta Aleatória e OPF supervisionados	92
6.4	Experimentos	94
6.4.1	Desempenho dos classificadores	95
6.4.1.1	Comparação de desempenho através de expressões 3D	96

6.4.1.2	Comparação de desempenho através de vídeos 2D quadro a quadro	97
6.4.1.3	Comparando os classificadores em função do tempo	103
6.4.1.4	Comparação entre a face da entrada e um <i>blendshape</i> diferente na saída	105
7	CONCLUSÕES	109
	REFERÊNCIAS	111
A	APÊNDICE	119
A.1	Amostra da base de dados de <i>blendshapes</i>.	119
A	ANEXOS	123
A.1	Procrustes	123
A.2	Distância Euclidiana	124

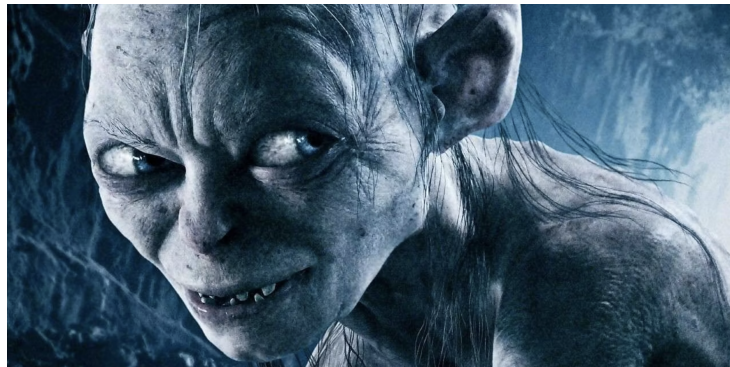
1 Introdução

Nas últimas décadas, o interesse quanto à captura de movimentos da face e a identificação de suas expressões, com a finalidade de gerar animações faciais realistas, em personagens virtuais 2D e 3D, tem aumentado, tanto na comunidade científica quanto na indústria do entretenimento. Com uma animação facial de qualidade, é possível fazer com que personagens virtuais transmitam emoções quase reais. No entanto, esse tipo de animação é uma tarefa complexa, pois lida com inúmeras variações de movimento do rosto.

A grande complexidade e sofisticação da estrutura da cabeça humana aumenta a dificuldade da reprodução de uma animação facial convincente. Uma alta acurácia é necessária, porque os humanos são treinados para observar e decodificar expressões faciais desde o nascimento, tornando-os especialistas na detecção de pequenos erros na animação de uma face virtual (FRATARCANGELI, 2013).

O cinema é um dos grandes beneficiados com a captura de movimentos faciais. O personagem Gollum da franquia Senhor dos Anéis (Figura 1.1) e os filmes King Kong e Avatar, são exemplos de produções cinematográficas que utilizaram esse tipo de animação (SEOL et al., 2011).

Figura 1.1 – Personagem Gollum da franquia Senhor dos Anéis.



Fonte: GELMAN (2020).

Já, no mundo dos jogos, essa tecnologia pode ser usada para controlar avatares, representando a face do usuário, permitindo que os seus amigos reais percebam as expressões do usuário refletidas no mundo virtual, em tempo real.

A medicina, também podem se beneficiar dessa tecnologia, nas quais novas formas de terapia interativa são possíveis. Por exemplo: sessões de treinamento baseadas em avatares podem ser criadas para pessoas com autismo ou outros transtornos do desenvolvimento neural (PAULY, 2013).

Um ambiente imersivo onde esta tecnologia também pode ser aplicada é o metaverso que, segundo Lee et al. (2021), é um ambiente virtual que mistura o físico e o digital onde os usuários possuem seus respectivos avatares e experimentam uma alternativa a vida física em uma realidade virtual. Já para Chen et al. (2022), o metaverso é um espaço de vida e ciberespaço que realiza o processo de virtualização e digitalização do mundo real. Ele integra uma infinidade de tecnologias existentes com o objetivo de poder mapear o mundo real, mesmo além do mundo real.

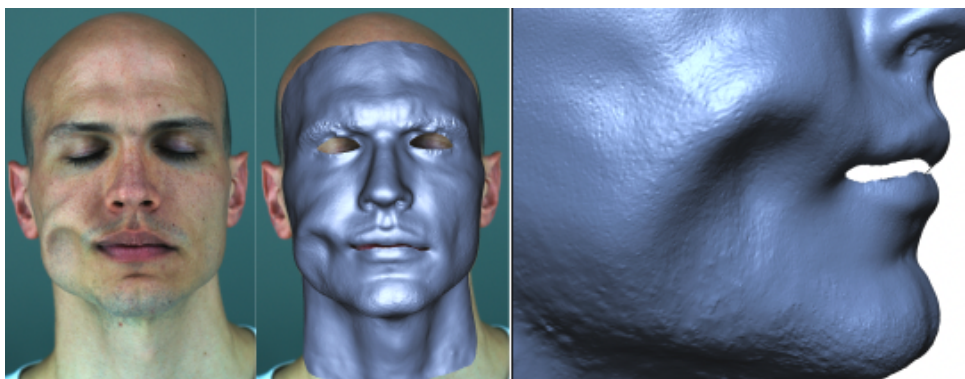
Com isso, o uso de avatares no metaverso torna-se uma necessidade, proporcionando aos usuários a experiência de viver em um mundo paralelo virtual, onde ele poderá se comunicar com outros avatares através de texto, voz e gestos, criando, assim, uma sociedade totalmente virtual. Desta forma, o uso das tecnologias de captura dos movimentos faciais no metaverso, podem proporcionar, além da comunicação verbal, uma comunicação através das mudanças de expressões nas faces virtuais dos avatares em tempo real, proporcionando uma experiência mais imersiva e mais próxima da humana no ambiente físico.

O redirecionamento de movimento facial é umas das técnicas utilizadas para a geração de animações faciais realistas. Ele é composto de três principais etapas: modelagem, captura e *retarget*. Cada etapa é responsável pelo processamento de várias informações, como a geração de um banco de dados de expressões, captura da face do ator em tempo real, cálculo para gerar a nova face e transferência para uma face virtual (SHAKIR; AL-AZZA, 2022).

Na etapa da modelagem são construídos modelos digitais de faces que podem ser armazenadas, exibidas e modificadas digitalmente. Duas abordagens se destacam: a deformação de malha e os *blendshapes*.

Na deformação de malha, uma única malha é deformada ao longo da captura, como mostrado na Figura 1.2. Um exemplo pode ser visto no trabalho de Wu et al. (2016).

Figura 1.2 – Exemplo da deformação de malha.



Fonte: Wu et al. (2016)

Nos *blendshapes* se utilizam modelos faciais, onde vetores representam expressões

individuais, para representar as expressões (Figura 1.3).

Figura 1.3 – Exemplo de *blendshapes*.



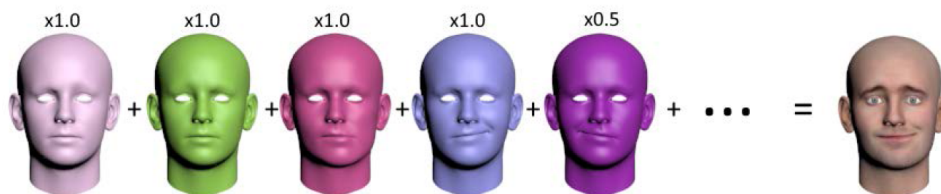
Fonte: Lewis et al. (2014).

Para a geração de uma nova face virtual são utilizadas combinações de vários *blendshapes*, por meio do cálculo da relevância das expressões armazenadas em uma base de dados, com relação à face real capturada. Esse cálculo resulta em pesos, valores que serão multiplicados pelas expressões armazenadas e somadas a uma das expressões base, gerando a face virtual final.

Ao variar as combinações, através de uma soma de vetores, uma gama de expressões faciais pode ser criada com pequeno esforço computacional (WEISE et al., 2011b). Um exemplo pode ser visto na Figura 1.4, onde vários *blendshapes* são somados, cada uma com uma relevância, resultando no final em uma nova face.

Figura 1.4 – Exemplo da criação de uma face utilizando a combinação de vários *blendshapes*.

A primeira face, da esquerda, está em expressão neutra e as seguintes, possuem expressões diferentes. A última, da direita, mostra o resultado final da soma entre elas.



Fonte: Costigan et al. (2016).

Na etapa de captura, existem vários métodos que podem ser usados para realizar o redirecionamento de movimento facial. Um método muito usado é o método óptico, onde câmeras especiais são colocadas à frente do rosto do ator e marcadores com tinta refletiva são pintados em pontos estratégicos de sua face (Figura 1.5). Porém, a transposição desses movimentos para a face de um personagem virtual não é trivial e, geralmente, exige a

execução de vários procedimentos durante a fase de pós-produção, sendo um processo quase artesanal (CUBAS; SEMENTILLE, 2017).

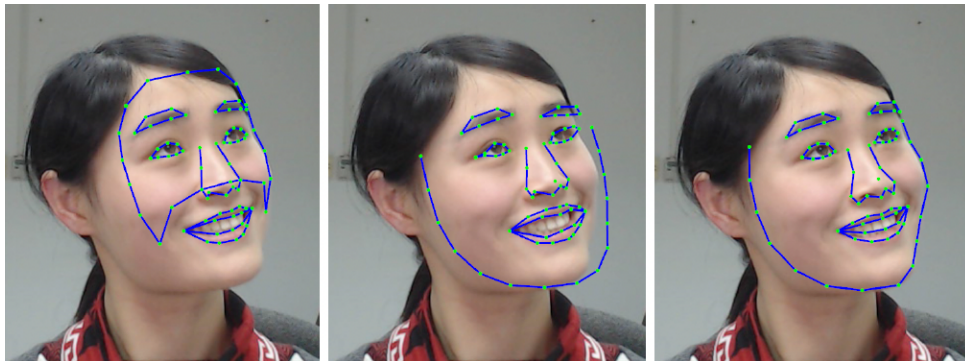
Figura 1.5 – Exemplo da utilização de marcadores com tinta refletiva.



Fonte: Pennington (2018).

Outro método utiliza a captura de forma automática, através de câmeras de captura de vídeo, como as *webcams*¹. Elas registram a face do ator e capturam os marcadores faciais, que são informações de pontos específicos, como o contorno da boca, dos olhos, do nariz, das sobrancelhas, entre outros, que servirão de referências para a construção da nova face (Figura 1.6).

Figura 1.6 – Exemplo da utilização de marcadores com tinta refletiva.



Fonte: Cao, Hou e Zhou (2014).

Nessa etapa, alguns trabalhos como os de, Weise et al. (2011b) ou Li et al. (2013b) utilizam câmeras especiais com informações RGB-D, que capturam, além das imagens 2D, a informação de profundidade associada a cada *pixel* com a câmera RGB-D *RealSense* da Intel. Outros trabalhos, como os de Saito, Li e Li (2016) e Cao, Hou e Zhou (2014), utilizam informações de câmeras RGB que contêm informações 2D.

¹ Pequena câmara de vídeo digital que se liga ao computador e permite a captação e o envio de imagens em tempo real, através da internet (PORTO, 2003).

Na etapa de *retarget*, os parâmetros obtidos na etapa de captura são utilizados para animar os personagens virtuais. Essa tarefa é altamente não trivial, especialmente, quando o personagem alvo não é uma réplica próxima da face capturada e tem proporções diferentes dela.

Existem várias abordagens para o *retarget*, como a utilização do PCA (*Principal Component Analysis*)² (BISHOP, 2006), AAM (*Active Appearance Model*)³ (ZHOU et al., 2010) ou redes neurais como a CNN (*Convolution Neural Networks*)⁴ Guo et al. (2017).

Outras abordagens são as técnicas de aprendizado de máquina. Segundo Monard e Baranauskas (2003), Aprendizado de Máquina é uma área da Inteligência Artificial (IA), cujo objetivo, é o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como, a construção de sistemas capazes de adquirir conhecimento de forma automática.

No aprendizado de máquina existem alguns tipos de classificação. Uma delas é a classificação multiclasse ou multinomial, a qual se preocupa em classificar instâncias em várias classes. Exemplos de classificadores multiclasse são, o classificador de padrões baseado em Florestas de Caminhos Ótimos, OPF, que utiliza grafos para classificar as amostras e o classificador de Florestas Aleatórias, que utiliza uma árvore de decisão, para identificar as classes que as amostras pertencem.

1.1 Os problemas

Nesta tese se escolheu a representação da face humana por meio de *blendshapes*, porque tal método, tem se tornado muito popular, principalmente, na indústria de entretenimento, devido à sua facilidade de uso e intuitividade por parte dos artistas digitais. No entanto, o uso desse tipo de representação, apresenta diversos problemas, no que diz respeito ao processo de animação facial baseado em performance, ou seja, quando a animação utiliza a captura das expressões faciais diretamente do rosto de um ator. Nesse aspecto, nesta tese, o interesse se focou nos problemas identificados, a partir do levantamento dos trabalhos de estado da arte:

- i. Não há um mapeamento claro entre marcadores extraídos da face do ator e os pesos dos *blendshapes*. Esse mapeamento se torna ainda mais complexo, quando é

² O PCA é uma técnica de análise estatística usada para compressão, visualização e classificação de dados. Ela reduz a dimensionalidade de um conjunto de dados, com o mínimo de perda de informação. (BISHOP, 2006)

³ Algoritmo de visão computacional, que busca ajustar uma máscara de pontos generalistas, a uma classe de objeto. A construção do AAM é estruturada com base em um modelo de forma e um modelo de textura que, combinados, buscam descrever estatisticamente uma classe de objetos (PANCERI, 2017)

⁴ CNN é uma rede neural de aprendizado profundo, que pode captar uma imagem de entrada, atribuir importância (pesos e vieses que podem ser aprendidos) a vários aspectos da imagem e ser capaz de diferenciar um do outro. (INDOLIA et al., 2018)

utilizada uma única câmera RGB na etapa de extração dos marcadores, pois tais equipamentos, não permitem a obtenção da informação de profundidade;

- ii. A dificuldade de se reproduzir, de maneira satisfatória, as expressões obtidas a partir da captura da face de um ator, em uma face de saída que não seja, estruturalmente, idêntica à dele;
- iii. As técnicas de Aprendizado de Máquina, encontradas nos trabalhos levantados, normalmente, exigem um alto poder computacional, como é o caso das Redes Neurais Convolucionais, o que dificulta a implementação do sistema de animação facial em equipamentos mais simples; e
- iv. Utilizar-se, de maneira automática, um número reduzido de blendshapes, mas, ainda assim, obter-se uma animação facial satisfatória para alguns tipos de produções midiáticas mais simples.

1.2 Hipóteses

Considerando-se os problemas levantados na seção anterior, foram propostas, nesta tese, as seguintes hipóteses:

A primeira hipótese é que seja possível treinar, utilizando-se técnicas de Aprendizado de Máquina, um modelo de classificador mais genérico para os movimentos da face humana, ou seja, que considere as características minimamente relevantes das expressões faciais, a partir do qual seja viável realizar animações críveis em faces virtuais, em situações nas quais, a estrutura da face do ator não corresponda exatamente à da face de saída.

A segunda hipótese é que, seja possível se utilizar técnicas de Aprendizado de Máquina mais simples do que as redes neurais artificiais, como a de Florestas Aleatórias e OPF, a fim de se obter um *pipeline* de animação facial que exija menor poder computacional.

Finalmente, a terceira hipótese, é que seja possível a utilização de um pequeno número de *blendshapes*, correspondentes às três áreas de movimentos independentes da face (áreas da boca, dos olhos e das sobrancelhas, para se compor a face virtual final, a partir de uma face neutra. Em outras palavras, ao invés de se trabalhar, convencionalmente, com os pesos dos *blendshapes*, em uma abordagem, tradicionalmente, mais contínua, a ideia seria trabalhar com os próprios *blendshapes* individualizados por áreas, utilizando-se uma abordagem alternativamente mais discreta.

1.3 Objetivos

1.3.1 Objetivo geral

O objetivo geral desta tese de doutorado é propor, implementar e validar um sistema de animação facial, o qual utiliza um modelo de classificador supervisionado, capaz de identificar as expressões mais comuns da face humana e redirecioná-las, de forma automática, a um modelo de face gerado a partir da combinação de um pequeno número de *blendshapes*. Para a geração do classificador foram utilizados os algoritmos Floresta de Caminhos Ótimos e Floresta Aleatória (*Random Forest*). Adicionalmente, o sistema proposto se baseia na captura da face de um ator realizada por uma única câmera RGB.

1.3.2 Objetivos específicos

São objetivos específicos desta tese de doutorado:

- i. A estruturação e implementação de um protótipo em *software* para o sistema de animação facial baseado em *blendshapes* proposto;
- ii. A determinação das características relevantes que viabilizem o treinamento dos modelos supervisionados dos classificadores. As características levaram em consideração as informações obtidas dos marcadores extraídos das imagens capturadas da face do ator;
- iii. A geração de uma base de imagens de faces, construída a partir de bases de imagens do estado da arte, bem como, de uma base de amostras, considerando as características relevantes extraídas da base de imagens;
- iv. A geração de uma base de *blendshapes* e de uma base de amostras adequadas para o sistema proposto;
- v. A verificação da viabilidade da utilização dos classificadores baseados nos algoritmos Floresta de Caminhos Ótimos e Floresta Aleatória na construção do sistema de animação facial proposto; e
- vi. A análise do desempenho do sistema proposto, no que diz respeito ao tempo de processamento para o redirecionamento facial, visando sua possível utilização em aplicações de tempo real.

1.4 Contribuição da pesquisa

São contribuições desta pesquisa, além, do atingimento dos objetivos propostos:

- i. Disponibilização de um *pipeline* flexível, que possibilite o teste de outras técnicas, além, das investigadas nesta tese, para o redirecionamento facial;
- ii. Propor uma forma mais discreta para o uso dos *blendshapes*; e
- iii. Desenvolver um método de redirecionamento de movimento facial, onde se utilize classificadores mais leves, que se possam executar em dispositivos com baixo poder computacional como *tablets*, telefones celulares e computadores *desktops*, sem um *hardware* específico.

1.4.1 Escopo da pesquisa

O processo de transferência possui várias etapas, sendo que cada uma delas apresenta desafios únicos e complexos. Por isso, o foco deste trabalho foi a criação de um processo, sendo baseado na hipótese de que um classificador treinado poderia gerar, em tempo real, uma face virtual convincente.

Nesse processo foram analisados apenas os resultados que envolveram a soma de *blendshapes* para a geração de uma face virtual, sem a realização de manipulações nas malhas como ajustes e outras etapas de refinamento.

O processo trabalhou com faces virtuais que têm aparência humana, sendo que não foram utilizados personagens humanoides ou cartunescos.

O sistema implementado trabalhou com áreas diferentes da face. A área identificada com a região dos olhos, na verdade, trata-se da região que abrange as pálpebras porque o movimento do globo ocular, que seria os olhos de fato, e do cabelo foi ignorado. Além disso, todas as comparações para validação das métricas levam em conta uma captura de face em posição frontal.

Para a captura foi utilizada uma câmera RGB tipo *webcam*. Não foram utilizados equipamentos de captura RGB-D ou similares. Com isso, as características usadas para o treinamento foram os marcadores faciais.

1.5 Organização da tese

Esta tese de doutorado está organizada da forma que segue.

No capítulo 2 são apresentados os modelos de *blendshapes*, como podem ser criados e os detalhes do funcionamento de um sistema de animação facial.

Uma introdução ao aprendizado de máquina e classificação multiclasse, bem como, as características do classificador de padrões baseado em Floresta de Caminhos Ótimos - OPF, são apresentados no capítulo 3.

O capítulo 4 apresenta os trabalhos correlatos mais representativos encontrados na literatura.

No capítulo 5 se tem a descrição detalhada do método desenvolvido, validação do método e restrições adotadas.

Os resultados dos experimentos são apresentados no Capítulo 6 e, no Capítulo 7, tem-se as conclusões sobre o trabalho desenvolvido.

No apêndice A.1, é mostrado um conjunto de modelos da base de dados de *blendshapes*.

Por fim, no Anexo A.1 são vistos os fundamentos do algoritmo de Procrustes e no Anexo A.2 são apresentados conceitos da distância euclidiana.

2 *Blendshapes* e sistemas de animação facial baseado em performance

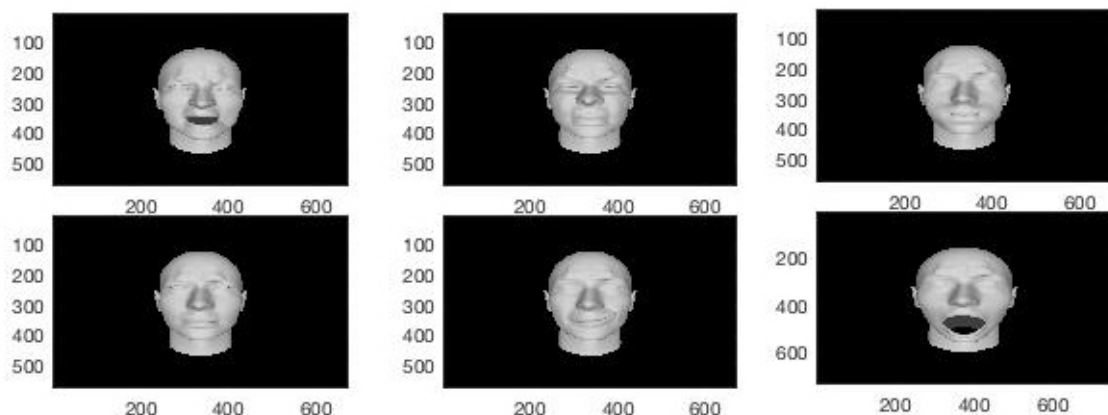
Nesse capítulo são apresentados os modelos dos *blendshapes*, como podem ser gerados e como podem ser usados nos sistemas de animação facial baseado em performance.

2.1 Blendshapes

O termo *blendshape* foi introduzido pela indústria gráfica na década de 80, quando se tornou popular em *softwares* comerciais. Lewis et al. (2014) definiu *blendshapes* como sendo modelos faciais, em que os vetores representam expressões faciais individuais. Isso consiste, em criar poses da face em várias malhas. Cada malha é designada a uma forma. Uma das malhas é a forma base, enquanto que as outras, são chamadas de formas alvo. A diferença entre a forma base e a forma alvo é representada por vetores de configuração. Cada vetor corresponde à diferença entre a base e o alvo (NENDYA; YUNIARNO; GANDANG, 2014).

Um modelo de *blendshape* gera uma pose facial, como uma combinação linear de um número de expressões faciais, os *blendshapes* "alvos". Ao variar os pesos da combinação linear, uma gama de expressões faciais pode ser expressa com pequeno esforço computacional. O conjunto de formas pode ser estendido conforme desejado para refinar a gama de expressões, que o personagem pode produzir (Figura 2.1) (LEWIS et al., 2014).

Figura 2.1 – Conjunto de expressões para o refinamento dos *blendshapes*.



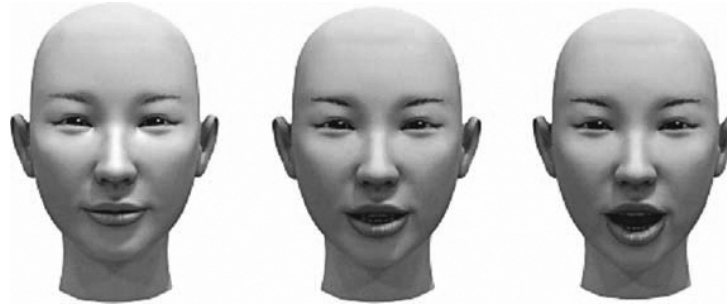
Fonte: elaborada pelo autor.

As expressões alvo podem ser criadas por artistas ou adquiridas por meio de dispositivos de captura de alta resolução, como scanners 3D. Alguns trabalhos como os de Sumner e Popović (2004), apresentam métodos para a transferência de expressões faciais,

entre uma expressão esculpida na forma neutra em um *blendshape*, para um conjunto de expressões preexistentes.

Um novo *blendshape* pode ser completamente criado usando esta técnica (Figura 2.2), entretanto, é necessária a criação de um grande número de *blendshapes*, para prover os controles em todas as regiões da face. *Stuart Little*, *Star Wars* e *Senhor dos Anéis* são exemplos de uso de *blendshapes* (DENG; NEUMANN, 2008).

Figura 2.2 – Formas de *blendshapes*.



Fonte: Deng e Neumann (2008).

2.1.1 Criação de expressões com *blendshapes*

Segundo Lewis et al. (2014), a criação de expressões utilizando *blendshapes* pode ser descrita como uma simples soma de vetores. Como exemplo, é considerado um conjunto de $n = 100$ expressões, cada uma com $p = 1000$ vértices, cada vértice possui 3 componentes x, y, z . Cada *blendshape* é transformado em um vetor b_k com 30.000 posições, que correspondem a 1.000×3 tendo seus elementos dispostos na ordem de $xxxyyyzzz$ ou $xyzxyzxyz$ (Figura 2.3).

O modelo de *blendshape* tem a seguinte equação:

$$f = \sum_{k=0}^n W_k B_k \quad (2.1)$$

,

ou, em notação de matriz

$$f = B_W \quad (2.2)$$

onde f é a face resultante, B é uma matriz $m = 30.000 \times 100$, em que, cada coluna do vetor b_k , corresponde a um *blendshape* individual e w é o peso aplicado em cada forma.

Figura 2.3 – Matriz-vetor que corresponde às expressões do modelos de *blendshapes*.

$$\begin{bmatrix} f_{1x} \\ f_{1y} \\ f_{1z} \\ f_{2x} \\ f_{2y} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_{nz} \end{bmatrix} = \begin{bmatrix} x & x & \cdots & x \\ y & y & \cdots & y \\ z & z & \cdots & z \\ x & x & \cdots & x \\ y & y & \cdots & y \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_n \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

Fonte: Lewis et al. (2014).

Uma face b_0 , tipicamente em expressão de repouso, é designada como face neutra e as faces restantes, b_k , $K = 1 \dots N$ são substituídas pela diferença entre $b_k - b_0$ e, entre, a enésima K faces alvos e a face neutra:

$$f = b_0 + \sum_{k=1}^n W_k(b_k - b_0) \quad (2.3)$$

onde b_0 sendo a forma neutra. Denota-se isso, como:

$$f = b_0 + B_w \quad (2.4)$$

Nessa formulação, os pesos w são limitados entre $[0, 1]$

2.1.2 Construção de *blendshapes*

Existem várias abordagens para a construção de *blendshapes*. Uma forma tradicional é a deformação de uma malha, por um artista digital habilidoso. Outra alternativa é o escaneamento da face de um ator real, utilizando equipamentos como *kinect*¹ da *Microsoft* ou a *RealSense*, que é uma câmera desenvolvida pela *Intel*, que captura imagens em RGB e RGB-D. (Figure 2.4).

Blanz e Vetter (1999) desenvolveram uma técnica para modelar faces 3D, automaticamente, a partir de uma ou mais fotografias. Através de modelos previamente capturados, é realizado um processamento para gerar a face final, adaptada para a geometria da face

¹ O *kinect* é uma câmera que permite que o computador detecte. Diretamente, a terceira dimensão (profundidade) do ambiente (ZHANG, 2012).

Figura 2.4 – Captura de uma face usando um dispositivo de captura 3D.



Fonte: elaborada pelo autor.

das fotos. Isso é feito usando uma base de dados de mais de 200 faces capturadas em alta resolução, como pode ser visto na figura 2.5.

Figura 2.5 – Exemplo de modelo morfológico. (a) Foto de uma face 2D; (b) Modelo 3D gerado em posição frontal; (c) Modelo 3D gerado em posição lateral.



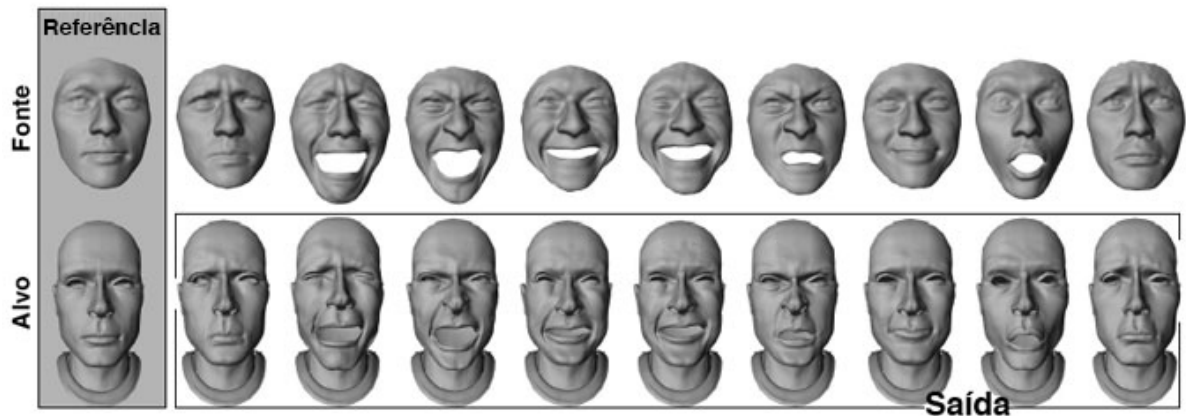
Fonte: elaborada pelo autor.

Sumner e Popović (2004) desenvolveram uma técnica de modelagem de *blendshapes* que usa transferências de expressões, entre um modelo padrão e um conjunto de *blendshapes*, como mostrado na figura 2.6.

2.2 Sistemas de animação facial baseado em performance

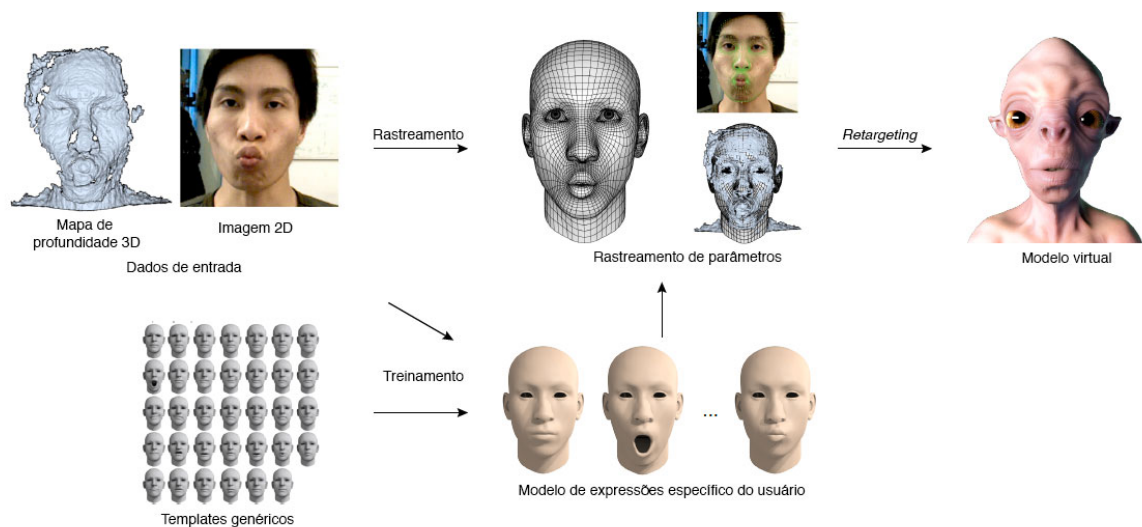
O redirecionamento de movimento facial, também conhecido como *retarget*, introduz a ideia da captura da face de um ator real e seu redirecionamento para um ator virtual. Para Li et al. (2013a), o *retarget* refere-se ao problema de mapeamento de expressões faciais de um ator para um avatar digital, de maneira realista e compatível, com o desempenho de entrada. É normalmente constituída por uma etapa de rastreamento da face, seguida por um procedimento de síntese de expressão. A maioria dos métodos de redirecionamento utiliza um sistema modular, onde os processos são desenvolvidos em várias etapas, no

Figura 2.6 – Transferência de expressões entre modelos 3D.



Fonte: Sumner e Popović (2004).

geral, sendo uma *offline* e as demais *online*. Na Figura 2.7, pode ser visto um sistema genérico.

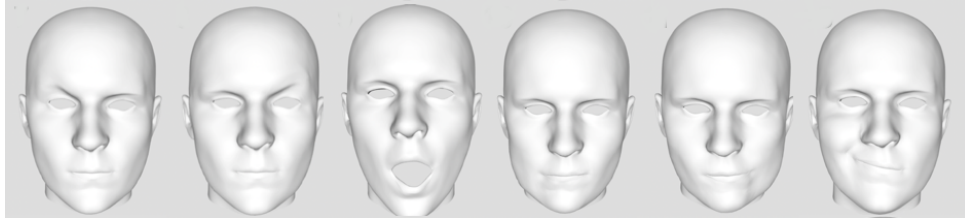
Figura 2.7 – Módulos de uma arquitetura de *retarget*.

Fonte: Pauly (2013).

2.2.1 Processo *offline*

No processo *offline* é realizada uma captura inicial do ator em expressão neutra. Essa captura, servirá para geração de um conjunto de *blendshapes* que contém expressões faciais como: alegria, tristeza, espanto, raiva e surpresa. Um exemplo do conjunto gerado no processo *offline* pode ser visto na Figura 2.8.

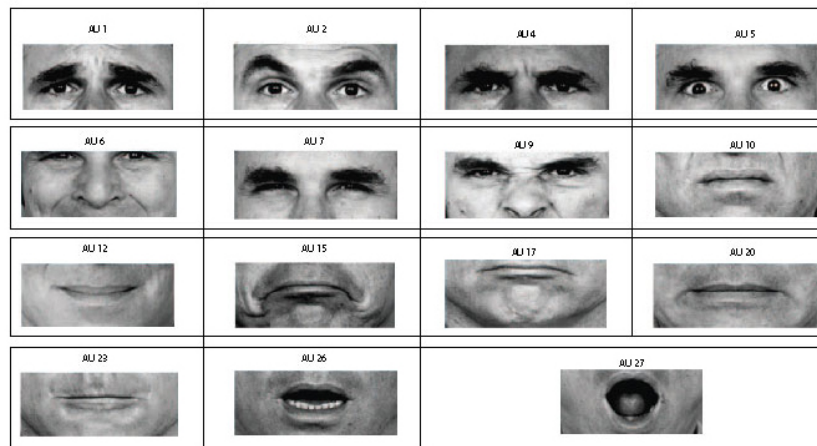
Figura 2.8 – Exemplo de um conjunto de expressões gerado na fase *offline*.



Fonte: Yan et al. (2019)

As expressões seguem a classificação proposta por Ekman e Rosenberg (1997), as (*FACS – Facial Action Coding System*). Eles dividiram a face em diferentes grupos musculares chamados Unidades de Ação (*AU – Action Unit*). Um exemplo pode ser visto na Figura 2.9.

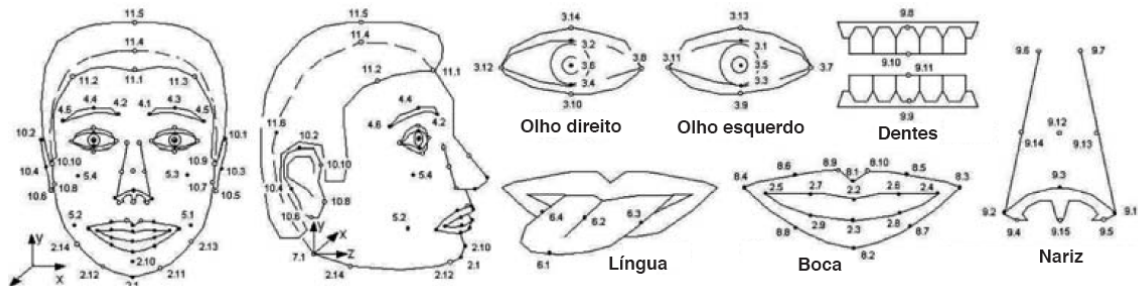
Figura 2.9 – Desenvolvido pelos psicólogos Paul Ekman e Wallace Friesen, os *FACS* possuem trinta e três unidades de ação (*AUs*) (por exemplo, as rugas da testa são uma unidade de ação).



Fonte: Velusamy et al. (2011).

No módulo *offline*, são extraídos os marcadores faciais dos *blendshapes* denominados *FCP (Facial Characteristic Point)*: pontos que marcam áreas de destaque na face como o contorno dos olhos, das sobrancelhas, da boca e do nariz. Eles são usados, também, para delimitar o contorno da face (MARAR et al., 2011).

Os marcadores faciais são baseados no padrão O MPEG-4 (*Moving Picture Experts Group*), que é utilizado para animação facial e sugere um conjunto de 84 pontos localizados na face. Ele foi desenvolvido com base no sistema de codificações de ações de Ekman. Um exemplo pode ser visto na Figura 2.10

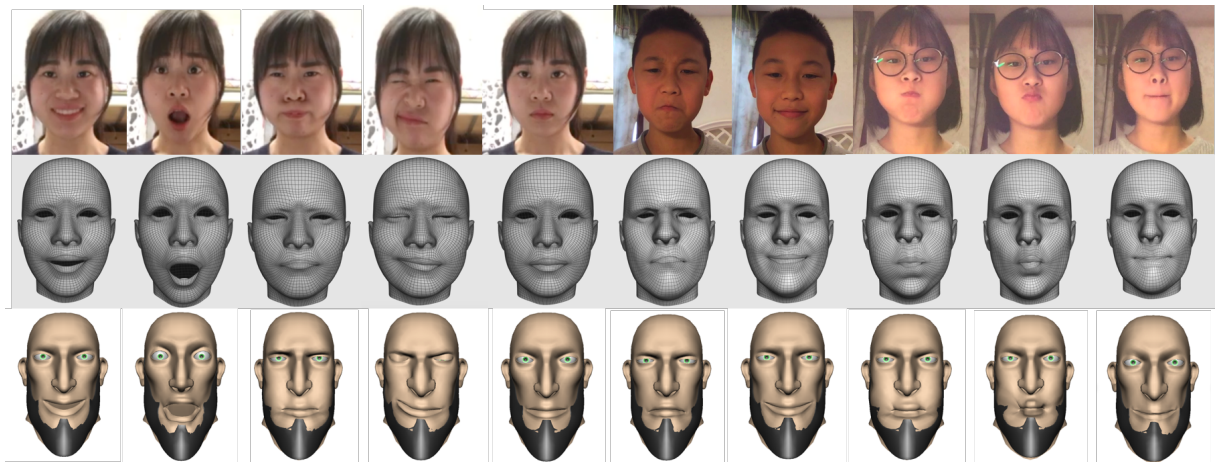
Figura 2.10 – Marcadores faciais extraídos do *Dataset* na fase *offline*.

Fonte: Braun (2014 apud PANDZIC; FORCHHEIMER, 2002).

2.2.2 Processo *online*

No processo *online*, uma câmera de vídeo captura a face do ator e seus marcadores faciais. É, através deles, que será identificado o quão próximo da face do ator, está das faces criadas no conjunto de *blendshapes*. Isso identifica o peso, ou seja, a proximidade que cada face gerada no processo *offline* terá na face final. O peso é aplicado aos *blendshapes* gerando a face virtual. Com isso, é possível a criação de vários tipos de expressões com a combinação de *blendshapes*. Todo este processo deverá acontecer em tempo real, sendo que o uso de *blendshapes* é mais recomendado, pois computacionalmente significa apenas uma soma de vetores. Um exemplo do processo final pode ser visto na Figura 2.11

Figura 2.11 – Faces virtuais geradas pelo processo de redirecionamento de movimento facial.



Fonte: Yan et al. (2019).

3 Aprendizado de máquina, classificador de Floresta de Caminhos Ótimos e Floresta Aleatória

Neste capítulo são apresentados conceitos fundamentais para a elaboração deste trabalho, tais como, as formas de aprendizado de máquina, classificadores multiclasse, classificador de Floresta de Caminhos Ótimos e classificador de Floresta Aleatória.

3.1 Aprendizado de máquina

Aprendizado de Máquina é uma área da Inteligência Artificial (IA), cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como, a construção de sistemas capazes de adquirir conhecimento de forma automática. Os diversos sistemas de aprendizado de máquina possuem características particulares e comuns, que possibilitam sua classificação quanto à linguagem de descrição, modo, paradigma e forma de aprendizado utilizado (MONARD; BARANAUSKAS, 2003). Existem dois tipos de aprendizado de máquina: o supervisionado e o não supervisionado.

No aprendizado supervisionado é fornecido ao algoritmo de IA, um conjunto de exemplos de treinamento, para os quais, o rótulo da classe associada é conhecido. O objetivo do algoritmo de aprendizado supervisionado é construir um classificador que possa determinar corretamente a classe de novos exemplos ainda não rotulados, ou seja, exemplos que não tenham o rótulo da classe.

No aprendizado não supervisionado, o algoritmo analisa os exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados. Após, a determinação dos agrupamentos, normalmente é necessária uma verificação, para determinar o que cada agrupamento significa no contexto do problema que está sendo analisado.

3.1.1 Classificação multiclasse

No aprendizado de máquina existem alguns tipos de classificação. Uma delas é a binária, onde o problema possui dois tipos de classes, tipo gato ou cachorro, homem ou mulher. Já, a classificação multiclasse ou multinomial, é o problema de classificar instâncias em mais de duas classes. Essa classificação, não deve ser confundida com multirrótulo, onde uma classe pode pertencer a vários tipos de rótulos, por exemplo, classificar um conjunto de imagens de frutas, que pode ser laranja, maçã ou pera. A classificação de

várias classes pressupõe que, cada amostra, seja atribuída a um e apenas um rótulo: uma fruta pode ser uma maçã ou uma pera, mas não as duas ao mesmo tempo.

3.2 Classificador de padrões baseado em Florestas de Caminhos Ótimos - OPF

O classificador baseado em Florestas de Caminhos Ótimos (OPF) modela o problema de reconhecimento de padrões, como um problema de Floresta de Caminhos Ótimos, em um grafo definido no espaço de atributos, onde os nós são as amostras, as quais são representadas pelos seus respectivos vetores de atributos, e os arcos são definidos de acordo com uma relação de adjacência preestabelecida (PEREIRA, 2012). Ele possui versões de classificador supervisionado, quando as amostras são rotuladas e não supervisionados, quando não se conhece os tipos das amostras. Atualmente, existem duas versões distintas do OPF para aprendizado supervisionado: (i) uma que faz o uso de um grafo completo (PAPA; FALCAO; SUZUKI, 2009), e (ii) outra versão que utiliza um grafo k-vizinhos mais próximos (PAPA; FERNANDES; FALCAO, 2017).

3.2.1 Classificação supervisionada

O algoritmo OPF com grafo completo foi, primeiramente, apresentado por Papa, Falcão e Suzuki (2009) e tem sido amplamente utilizado em diversas aplicações. Esse algoritmo modela as amostras como sendo os nós de um grafo completo, onde os elementos mais representativos de cada classe do conjunto de treinamento, isto é, os protótipos, são escolhidos como sendo os elementos pertencentes às regiões de fronteira entre as classes.

Os protótipos participam de um processo de competição disputando as outras amostras, oferecendo-lhes caminhos de menor custo e seus respectivos rótulos. Ao final desse processo, obtém-se um conjunto de treinamento particionado em árvores de caminhos ótimos, sendo que, a união das mesmas remete a uma Floresta de Caminhos Ótimos. Essa abordagem apresenta vários benefícios com relação a outros métodos de classificação de padrões supervisionados: (i) é livre de parâmetros, (ii) possui tratamento nativo de problemas multiclases e (iii) não faz alusão sobre forma e/ou separabilidade das classes (PEREIRA, 2012).

3.2.2 Fundamentação Teórica

Seja $Z = Z_1 \cup Z_2 \cup Z_3$ um conjunto de dados, tais que Z_1 , Z_2 e Z_3 denotam os conjuntos de treinamento, avaliação e teste, respectivamente. Seja (Z_1, A) um grafo completo, cujos nós são amostras em Z_1 e qualquer par de amostras define um arco em $A = Z_1 \times Z_1$. Os arcos não precisam ser armazenados, sendo a representação do grafo de

maneira implícita. Define-se também o caminho como uma sequência de amostras distintas $\pi_t = \langle s_1, s_2, \dots, s_{k-1}, s_k \rangle$ com término t , onde $(s_i, s_{i+1}) \in A$ para $1 \leq i \leq k-1$.

O caminho é dito trivial se $\pi_t = \langle t \rangle$. Associa-se a cada caminho π_t um custo $f(\pi_t)$ dado pela função de conectividade f . Um caminho π_t é considerado ótimo se $f(\pi_t) \leq f(\tau_t)$ para qualquer outro caminho τ_t . Denota-se também, $\pi_s \cdot \langle s, t \rangle$ a concatenação de um caminho π_s e arco (s, t) .

Dado com um conjunto de amostras *protótipos* $S \in Z_1$, a fase de treinamento do OPF consiste em calcular uma Floresta de Caminhos Ótimos sobre o conjunto de treinamento. Essa floresta é, essencialmente, a coleção de árvores de caminhos ótimos, com raízes em cada protótipo. A partição (árvore) da grafo completo da Floresta de Caminhos Ótimos é computado nos $\mathbb{R}^n$ pelo algoritmo da transformada imagem-floresta (IFT) (FALCÃO; STOLFI; LOTUFO, 2004).

Considerando $R(s) \in S$ como a raiz de $s \in Z_1$, dizemos que s é mais fortemente conectada com $R(s)$, do que qualquer outra raiz $u \in S$ em Z_1 . Isto significa que, uma vez s é conquistada por alguma amostra, a qual pode ser $R(s)$ ou alguma outra amostra $v \in Z_1$, tal que, $R(v) = R(s)$, s pertence à mesma árvore de caminhos ótimos de $R(s)$.

Como dito, anteriormente, as amostras em S são escolhidas como sendo as mais próximas com rótulos diferentes em Z_1 . Para encontrar tais protótipos, Papa, Falcão e Suzuki (2009) propuseram computar uma Árvore de Espalhamento Mínima (*Minimum Spanning Tree* - MST) em Z_1 e marcar as amostras conexas com rótulos diferentes como sendo os protótipos. O algoritmo1 implementa a fase de treinamento do OPF.

Algoritmo 1 – TREINAMENTO POR FLORESTA DE CAMINHOS ÓTIMOS

ENTRADA: Um conjunto de treinamento Z_1 λ -rotulado e o par (v, d) composto pelo vetor de características e as distâncias computadas.

SAÍDA: Floresta de Caminhos Ótimos P_1 , mapa de custo C_1 , mapa de rótulos L_1 , e o conjunto ordenado Z'_1 .

AUXILIARES: Fila de prioridade Q , conjunto S de protótipos, e variável de custo cst .

1. $Z'_1 \leftarrow \emptyset$ e compute via MST o conjunto de protótipos $S \subset Z_1$.
2. Para cada $s \in Z_1 \setminus S$, faça $C_1(s) \leftarrow +\infty$.
3. Para cada $s \in S$, faça
 4. $\perp$ $C_1(s) \leftarrow 0$, $P_1(s) \leftarrow \text{nil}$, $L_1(s) \leftarrow \lambda(s)$, insira s em Q .
 5. Enquanto $Q \neq \emptyset$, faça
 6. $\perp$ Remova de Q uma amostra s tal que $C_1(s)$ é mínimo.
 7. $\perp$ Insira s em Z'_1 .
 8. $\perp$ Para cada $t \in Z_1$ tal que $C_1(t) > C_1(s)$, faça
 9. $\perp$ $\perp$ Compute $cst \leftarrow \max\{C_1(s), d(s, t)\}$.
 10. $\perp$ $\perp$ Se $cst < C_1(t)$, então

11. $\left[\begin{array}{l} \left[\begin{array}{l} \left[\begin{array}{l} \text{Se } C_1(t) \neq +\infty, \text{ então remova } t \text{ de } Q. \\ P_1(t) \leftarrow s, L_1(t) \leftarrow L_1(s), C_1(t) \leftarrow cst. \\ \text{Insira } t \text{ em } Q. \end{array} \right] \end{array} \right] \end{array} \right]$
- 12.
- 13.
14. Retorne o classificador $[P_1, C_1, L_1, Z'_1]$.

Após a fase de treinamento, o processo de classificação encontra o caminho ótimo entre uma amostra de teste $t \in Z_3$ (um procedimento similar é aplicado a amostras de Z_2) e um nó de treinamento, e t recebe o rótulo da amostra em Z_1 que o conquistou, isto é, $L(t) \leftarrow L(P(t))$, onde $P(t)$ denota o predecessor de t no caminho ótimo até $R(t)$. Em outras palavras, $P(t)$ denota a amostra que conquistou t . O algoritmo 2 apresenta esse procedimento.

Algoritmo 2 – CLASSIFICAÇÃO POR FLORESTA DE CAMINHOS ÓTIMOS

ENTRADA: Classificador $[P_1, C_1, L_1, Z'_1]$, conjunto de teste Z_3 , e o par (v, d) composto pelo vetor de características e as distâncias computadas.

SAÍDA: Rótulo L_2 e predecessor P_2 mapas definidos por Z_3 , e valor de acurácia Acc .

AUXILIARES: Variáveis de custo tmp e $mincost$.

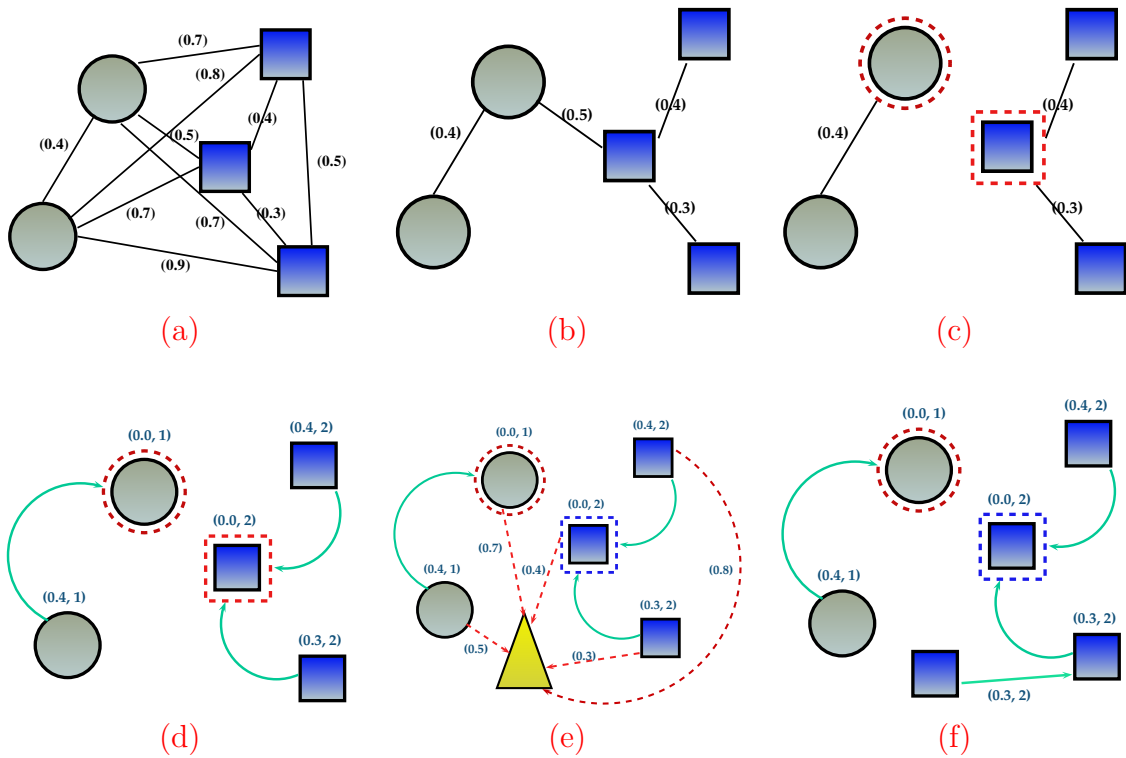
1. Para cada $t \in Z_3$, faça
2. $i \leftarrow 1, mincost \leftarrow \max\{C_1(k_i), d(k_i, t)\}.$
3. $L_2(t) \leftarrow L_1(k_i)$ e $P_2(t) \leftarrow k_i.$
4. Enquanto $i < |Z'_1|$ e $mincost > C_1(k_{i+1})$, faça
5. $tmp \leftarrow \max\{C_1(k_{i+1}), d(k_{i+1}, t)\}.$
6. Se $tmp < mincost$, então
7. $mincost \leftarrow tmp.$
8. $L_2(t) \leftarrow L(k_{i+1})$ e $P_2(t) \leftarrow k_{i+1}.$
9. $i \leftarrow i + 1.$
10. Calcule a acurácia Acc de acordo com (PAPA; FALCÃO; SUZUKI, 2009).
11. Retorne $[L_2, P_2, Acc]$.

O algoritmo baseado em OPF pode ser utilizado com qualquer função de custo suave que agrupa amostras com propriedades similares (FALCÃO; STOLFI; LOTUFO, 2004). Na versão OPF, com grafo completo, a função de custo abordada foi a f_{max} (Equação 3.1). O algoritmo baseado em OPF associa um caminho ótimo $P^*(s)$ de S a toda amostra $s \in Z_1$, formando uma Floresta de Caminhos Ótimos P (uma função sem ciclos, a qual associa a todo $s \in Z_1$ seu predecessor $P(s)$ em $P^*(s)$, ou uma marca *nil* quando $s \in S$, como mostrado na Figura 3.1d). Seja $R(s) \in S$ a raiz de $P^*(s)$, a qual pode ser alcançada por $P(s)$.

$$\begin{aligned}
 f_{max}(\langle s \rangle) &= \begin{cases} 0 & \text{se } s \in S, \\ +\infty & \text{caso contrário} \end{cases} \\
 f_{max}(\pi \cdot \langle s, t \rangle) &= \max\{f_{max}(\pi), d(s, t)\},
 \end{aligned} \tag{3.1}$$

Sendo que $f_{max}(\pi)$ computa a distância máxima entre amostras adjacentes em π , quando π não é um caminho trivial. A Figura 3.1 ilustra todos os passos executados pelos algoritmos de treinamento, classificação e testes do classificador de padrões OPF.

Figura 3.1 – (a) Grafo completo ponderado nas arestas, para um determinado conjunto de treinamento. (b) MST do grafo completo. (c) Protótipos escolhidos como sendo os elementos adjacentes de classes diferentes na MST (nós ponto estrela). (d) Floresta de caminhos ótimos resultante para a função de valor de caminho f_{max} e dois protótipos. Os identificadores (x, y) , acima dos nós, são respectivamente, o custo e o rótulo dos mesmos. A seta indica o nó predecessor no caminho ótimo. (e) Uma amostra de teste (pergaminho) é conectada aos demais nós de ambas as classes (linhas pontilhadas), com os nós do conjunto de treinamento. (f) O caminho ótimo do protótipo mais fortemente conexo, seu rótulo 2 e o custo de classificação 0.4 são associados a amostra de teste. Note que, mesmo a mostra de teste estando mais próxima de um nó da classe 1, ela foi classificada como sendo da classe 2.



Fonte: Pereira (2012).

3.2.3 Algoritmo de poda do conjunto de treinamento

Grandes bases de dados usualmente apresentam redundância. Pelo menos em teoria, seria possível estimar um conjunto reduzido de treinamento (Z_1) com os padrões mais relevantes para classificação. Nesse sentido, a utilização de um conjunto de treinamento e de avaliação (Z_2) tem permitido ao OPF aprender as amostras mais relevantes do conjunto de treinamento, para a classificação de erros no conjunto de avaliação, trocando amostras de Z_2 classificadas erroneamente entre amostras não protótipos de Z_1 (PAPA; FALCÃO; SUZUKI, 2009). Nesta estratégia de aprendizado, Z_1 permanece com o mesmo tamanho e, a instância do classificador, com maior acurácia em Z_2 é selecionada para ser avaliada no conjunto de teste Z_3 . O Algoritmo 3 implementa esta ideia.

Algoritmo 3 – ALGORITMO DE APRENDIZADO POR FLORESTA DE CAMINHOS ÓTIMOS

ENTRADA: Conjuntos de treinamento e avaliação Z_1 e Z_2 , respectivamente, λ -rotulados, número T de iterações, e o par (v, d) composto pelo vetor de características e as distâncias computadas.

SAÍDA: Floresta de Caminhos Ótimos P_1 , mapa de custos C_1 , mapa de rótulos L_1 , conjunto ordenado Z'_1 e $MaxAcc$.

AUXILIARES: Vetores FP e FN de tamanho c para falso positivo e falso negativo, conjunto S de protótipos, e lista LM de amostras classificadas erroneamente.

1. Determine $MaxAcc \leftarrow -1$.
2. Para cada iteração $I = 1, 2, \dots, T$, faça
 3. $LM \leftarrow \emptyset$ e compute o conjunto $S \subset Z_1$ de protótipos.
 4. $[P_1, C_1, L_1, Z'_1] \leftarrow \text{Algoritmo 1}(Z_1, S, (v, d))$.
 5. Para cada classe $i = 1, 2, \dots, c$, faça
 6. $FP(i) \leftarrow 0$ e $FN(i) \leftarrow 0$.
 7. $[L_2, P_2, Acc] \leftarrow \text{Algoritmo 2}([P_1, C_1, L_1, Z'_1], Z_2, (v, d))$
 8. Se $Acc > MaxAcc$ então
 9. Salve a atual instância $[P_1, C_1, L_1, Z'_1]$
 10. do classificador e conjunto $MaxAcc \leftarrow Acc$.
 11. Enquanto $LM \neq \emptyset$
 12. $LM \leftarrow LM \setminus t$.
 13. Substitua t por uma amostra não-protótipo,
 14. randomicamente selecionada de Z_1 .
15. Retorne a instância do classificador $[P_1, C_1, L_1, Z'_1]$ com maior
16. acurácia em Z_2 , em seu valor $MaxAcc$.

Além disso, Papa et al. (2010) propuseram um algoritmo que tem por objetivo identificar as amostras mais relevantes de Z_1 , por meio, da classificação do conjunto de

avaliação Z_2 . A ideia consiste, basicamente, em marcar cada amostra do conjunto de treinamento até sua raiz na árvore de caminhos ótimos, que classificou uma dada amostra de Z_2 . Este procedimento é repetido até que um critério de convergência tenha sido satisfeito. Ao final do processo, as amostras não marcadas são removidas do conjunto de treinamento.

Essa abordagem visa aprender as amostras mais relevantes do conjunto de treinamento e também reduzir o seu tamanho. Como a complexidade computacional do OPF para classificação é proporcional ao número de amostras de treinamento, podemos acelerar o processo diminuindo o tamanho do conjunto de treinamento. Embora a acurácia no conjunto de teste possa ser afetada, em algumas aplicações não é percebido tal comportamento (PAPA et al., 2010) (PISANI et al., 2011) (PEREIRA et al., 2011).

O critério de parada é dado pela variável $MLoss$, a qual é definida como a diferença absoluta entre a taxa de acerto no conjunto de avaliação, utilizando o conjunto de treinamento original, isto é, sem poda, e a taxa de acerto, utilizando o conjunto de treinamento reduzido. O Algoritmo 4 implementa esse processo.

Algoritmo 4 – ALGORITMO DE APRENDIZADO COM PODA POR FLORESTA DE CAMINHOS ÓTIMOS

ENTRADA: Conjunto de treinamento e avaliação, Z_1 e Z_2 , λ -rotulados, o par (v, d) composto pelo vetor de características e as distâncias computadas, variável $MLoss$ e número T de iterações.

SAÍDA: Classificador $[P_1, C_1, L_1, Z'_1]$ com conjunto de treinamento reduzido.

AUXILIARES: Conjunto R de amostras relevantes, e variáveis Acc e tmp .

1. $[P_1, C_1, L_1, Z'_1] \leftarrow \text{Algoritmo } 3(Z_1, Z_2, T, (v, d))$.
2. $[L_2, P_2, Acc] \leftarrow \text{Algoritmo } 2([P_1, C_1, L_1, Z'_1], Z_2, (v, d))$.
3. $tmp \leftarrow Acc$ e $R \leftarrow \emptyset$.
4. Enquanto $|Acc - tmp| \leq MLoss$ e $R \neq Z_1$ faça
 5. $R \leftarrow \emptyset$.
 6. Para cada amostra $t \in Z_2$, faça
 7. $s \leftarrow P_2(t) \in Z_1$.
 8. Enquanto $s \neq nil$, faça
 9. $R \leftarrow R \cup s$.
 10. $s \leftarrow P_1(s)$.
 11. Mova as amostras de $Z_1 \setminus R$ para Z_2 .
 12. $[P_1, C_1, L_1, Z'_1] \leftarrow \text{Algoritmo } 3(Z_1, Z_2, T, (v, d))$.
 13. $[L_2, P_2, tmp] \leftarrow \text{Algoritmo } 2([P_1, C_1, L_1, Z'_1], Z_2, (v, d))$.
14. Retorne $[P_1, C_1, L_1, Z'_1]$.

3.3 Classificador *k-Means*

k-Means é um método não supervisionado de clusterização. Ele tenta separar os dados em K (um número predefinido) *clusters*, de acordo com a distância de cada ponto, até algo chamado de centroide. Um centroide é como um protótipo para um *cluster*. Na maioria das técnicas, K pontos aleatórios das amostras, são escolhidos para serem as coordenadas dos centroides iniciais. Então, cada instância desses dados, é atribuída ao centroide mais próximo. Nas demais iterações, a posição dos centroides é calculada, através, da distância média entre todos os pontos atribuídos àquele centroide na última iteração. O algoritmo termina, quando a posição dos centroides não é mais modificada ou a distância dessa mudança é menor que um *threshold* predefinido.

Segundo (KANUNGO et al., 2002), o agrupamento baseado no *k-Means* está intimamente relacionado a vários outros problemas de agrupamento e localização. Estes incluem as *k*-medianas euclidianas (ARORA; RAGHAVAN; RAO, 1998), em que o objetivo é minimizar a soma das distâncias ao centro mais próximo e o problema geométrico de *k*-center (AGARWAL; PROCOPIUC, 2002) em que o objetivo é minimizar a distância máxima de cada ponto ao seu centro mais próximo.

Trevino (2016) diz que, o algoritmo de agrupamento *k-Means* usa refinamento iterativo para produzir um resultado final. As entradas do algoritmo são o número de *cluster* k e o conjunto de dados. O conjunto de dados é uma coleção de recursos para cada ponto de dados. Os algoritmos começam com estimativas iniciais para os centroides K , que podem ser gerados, aleatoriamente, ou selecionados, aleatoriamente, do conjunto de dados. O algoritmo, então, itera entre duas etapas:

1. Etapa de atribuição de dados:

Cada centroide define um dos *cluster*. Nesta etapa, cada ponto de dados é atribuído ao seu centroide mais próximo, com base no quadrado da distância euclidiana. Mais, formalmente, se c_i é a coleção de centroides no conjunto C , então, cada ponto de dados x é atribuído a um cluster baseado em

$$\arg \min_{c_i \in C} \text{dist}(c_i, x)^2$$

onde *dist* é a distância euclidiana padrão (L2). Seja o conjunto de designações de pontos de dados para cada centroide de *cluster* i -ésimo ser S_i .

2. Etapa de atualização do centroide:

Nessa etapa, os centroides são recalculados. Isso é feito, tomando a média de todos os pontos de dados atribuídos ao *cluster*, desse centroide.

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i$$

O algoritmo itera entre as etapas um e dois, até que um critério de parada seja atendido (ou seja, nenhum ponto de dados muda de *cluster*, a soma das distâncias é minimizada ou algum número máximo de iterações é alcançado).

Esse algoritmo é garantido para convergir para um resultado. O mesmo, pode ser um ótimo local (ou seja, não necessariamente, o melhor resultado possível), o que significa que, avaliar mais de uma execução do algoritmo com centroides iniciais aleatórios, pode fornecer um resultado melhor.

3.4 Algoritmos de Árvore de decisão

3.4.1 Classificador de Florestas Aleatórias (*Random Forest*)

O classificador de aprendizado supervisionado de Florestas Aleatórias (*Random Forest*) cria árvores de decisão em amostras de dados selecionadas aleatoriamente, obtêm previsões de cada árvore e seleciona a melhor solução, por meio de votação, usando um subconjunto de atributos selecionados aleatoriamente possibilitando assim a melhor análise dos dados (STATISTICS; BREIMAN, 2001). O método foi definido em 1995, por Tim Kam Ho, no artigo *Random Decision Forests* (HO, 1995). Ele propôs esse algoritmo devido às limitações existentes nas árvores de decisão, como por exemplo, o fato de árvores muito complexas tenderem a sofrer o fenômeno de *overfitting*¹.

A Floresta Aleatória possui características que se destacam referentes às outras técnicas, algumas delas são:

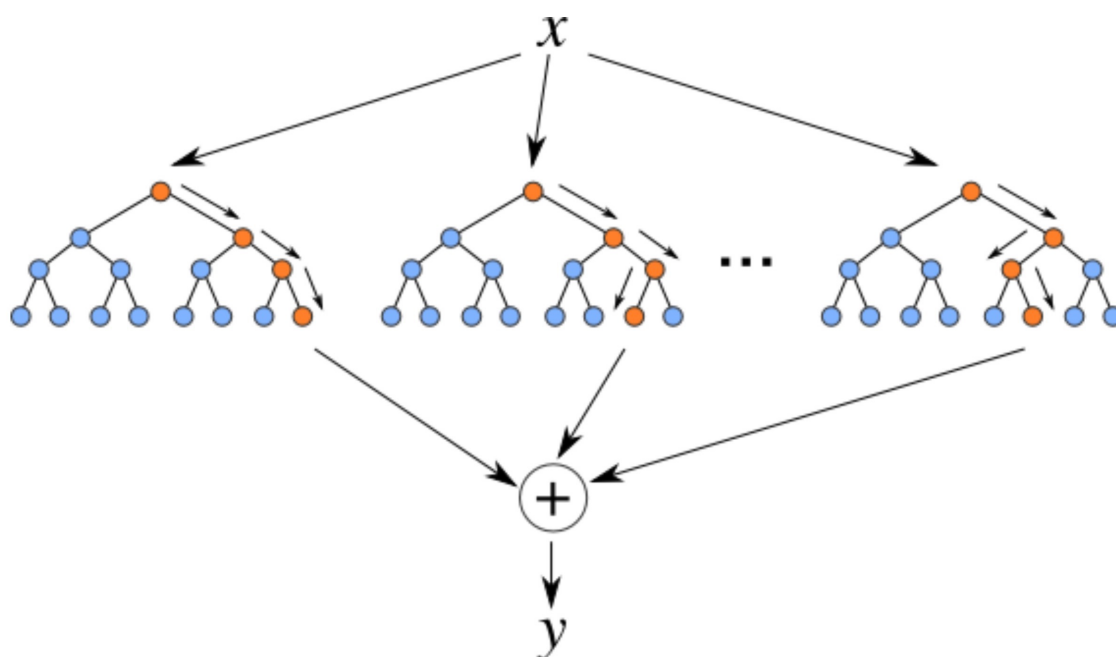
- Algoritmo mais poderoso do que comparado somente a uma árvore de decisão;
- Possui boa taxa de acerto quando testado em diferentes conjuntos de dados;
- Técnica exata;
- Evitam sobre ajuste (*overfitting*);
- Menos sensível a ruídos; e
- Classificação aleatória das árvores sem intervenção humana

Partindo de um elemento X (Figura 3.2), no caso uma base de dados, geram-se várias Florestas Aleatórias. Neste ponto cada uma gera várias regras e nelas, a possibilidade

¹ Termo usado em estatística, para descrever quando um modelo estatístico se ajusta muito bem ao conjunto de dados anteriormente observado, mas se mostra ineficaz, para prever novos resultados

de descoberta de novos padrões, que poderão ser essenciais na tomada de decisão correta. Após as florestas criadas, o próximo passo é calcular qual delas contém as regras mais exatas. Com a escolha feita, ela é aplicada na base de dados e, assim, chegando a um resultado (LORENZETT; TELOKEN, 2016).

Figura 3.2 – Ilustração da lógica por trás do algoritmo Florestas Aleatórias.



Fonte: Lorenzett e Teloken (2016).

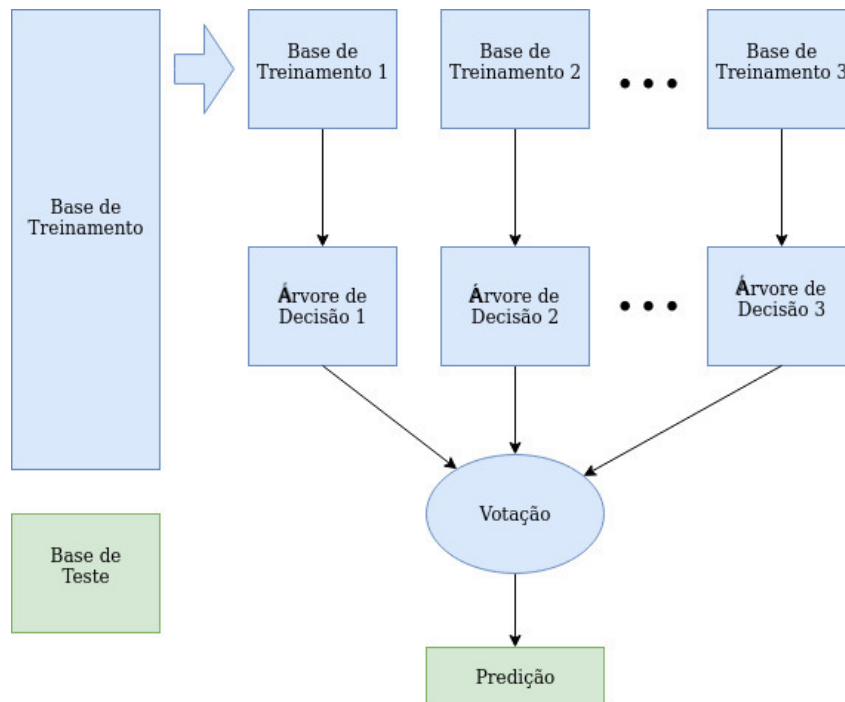
Em resumo, o classificador segue quatro etapas (Figura 3.3):

- Seleciona amostras aleatórias de um determinado conjunto de dados;
- Constrói uma árvore de decisão para cada amostra e obtém um resultado de previsão de cada árvore de decisão;
- Faz uma votação para cada resultado previsto; e
- Seleciona o resultado da previsão com mais votos, sendo a previsão final.

3.5 Métricas de validação para os classificadores

Para a comparação dos classificadores são utilizadas métricas de validação, com a finalidade de se aferir o desempenho dos modelos treinados.

Figura 3.3 – Etapas utilizadas pelo algoritmo de Florestas Aleatórias.



Fonte: Navlani (2018).

A maioria das métricas são baseadas na matriz de confusão, que é uma matriz que traz a informação de todos os acertos e erros do modelo ao prever as classes (SOUZA, 2020).

Ela é uma matriz quadrada, em que se comparam os verdadeiros valores de uma classificação com os valores preditos, através, de algum modelo. Sua diagonal é composta pelos acertos do modelo e, os demais valores, são os erros cometidos (Figura 3.4).

A Matriz de Confusão é composta por 4 valores:

- Verdadeiro Positivo (VP): Valores classificados como positivos nos dados originais e, corretamente, previstos como positivos no modelo.
- Verdadeiro Negativo (VN): Valores classificados como negativos nos dados originais e, corretamente, previstos como negativos no modelo.
- Falso Positivo (FP): Valores classificados como negativos nos dados originais e, erroneamente, previstos como positivos no modelo.
- Falso Negativo (FN): Valores classificados como positivos nos dados originais, e, erroneamente, previstos como negativos no modelo.

Chauhan (2020) apresenta algumas métricas que podem ser utilizadas para validação dos modelos.

Figura 3.4 – Exemplo de uma matriz de confusão.

		Valor predito	
		Positivo	Negativo
Valor real	Positivo	VP	FN
	Negativo	FP	VN

Fonte: elaborada pelo autor.

A acurácia é uma delas. Ela representa o número de previsões corretas do modelo. Quanto maior a acurácia do modelo, maior será a quantidade de acertos e menor será a quantidade de erros cometidos. O seu cálculo pode ser visto na Equação 3.2.

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (3.2)$$

Porém, não é interessante utilizar a acurácia quando se tem classes desbalanceadas, pois, pode causar uma falsa impressão de bom desempenho.

Outra métrica importante é a precisão, também conhecida como valor preditivo positivo (VPP), a qual representa a quantidade de observações classificadas como positivas, que realmente são positivas, ou seja, entre todas as observações identificadas como positivas, quantas foram identificadas corretamente. O seu cálculo pode ser feito através da Equação 3.3:

$$Precisão = \frac{VP}{VP + FP} \quad (3.3)$$

Já, a métrica especificidade, calculada conforme a Equação 3.4, é a proporção dos

Verdadeiros Negativos, entre todas as observações que, realmente, são negativas no seu conjunto de dados, ou seja, entre todas as observações que são negativas, quantas o modelo conseguiu prever como negativas. Representa a capacidade de um modelo em prever a classe negativa.

$$Especificidade = \frac{VN}{VN + FP} \quad (3.4)$$

O *recall* ou sensibilidade, também é uma métrica que indica a proporção dos verdadeiros positivos entre todas as observações que, realmente, são positivas no seu conjunto de dados, ou seja, entre todas as observações que são positivas, quantas o modelo conseguiu identificar como positivas. Representa a capacidade de um modelo em prever a classe positiva (Equação 3.5).

$$Recall = \frac{VP}{VP + FN} \quad (3.5)$$

E, por fim, a métrica *F1-Score* que é a média harmônica entre o *recall* e a precisão. Utilizada quando temos classes desbalanceadas. Seu cálculo pode ser visto na Equação 3.6.

$$F1 - Score = \frac{2 * (precisão * recall)}{(precisão + recall)} \quad (3.6)$$

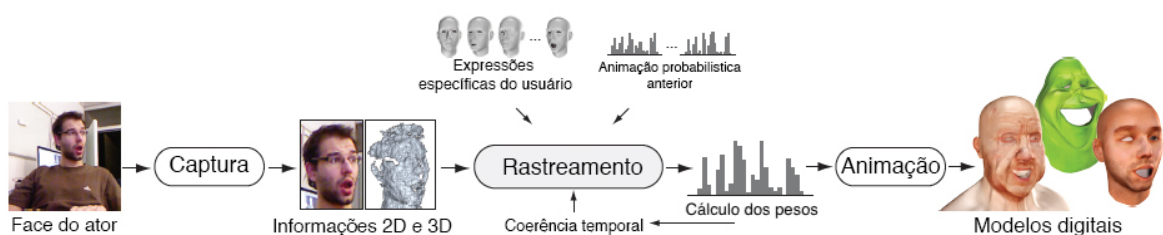
4 Trabalhos relacionados

Neste capítulo, são apresentados trabalhos que demonstram sistemas de animação facial desenvolvidos de maneira diferente. São apresentados sistemas modulares e sistemas que utilizando o algoritmo de floresta aleatória (*Random Forest*). Também são discutidos trabalhos que usam redes neurais do tipo CNN (*Convolutional Neural Network*) para animação facial em múltiplas faces e, para finalizar, trabalhos que vem despontando na área animação facial que utilizam a fala como entrada de dados.

4.1 Método de Weise et al. (2011b)

Weise et al. (2011b) desenvolveram um método para redirecionamento de movimento facial fazendo uso do *Kinect* da *Microsoft*, que combina a geometria 3D capturada por uma câmera RGB-D e a textura 2D. Na obtenção do resultado final, são utilizados *blendshapes*. Esse método, possui uma etapa *offline* e outra *online*. Na fase *offline*, são capturadas cerca de 15 expressões de um ator, para a criação de uma base de dados de expressões. Para melhorar a acurácia, são extraídos os marcadores faciais da região dos olhos e da boca. O processo pode ser visto na Figura 4.1.

Figura 4.1 – Visão geral do método proposto por Weise et al. (2011b). Os pesos dos blendshapes são estimados pela correspondência da textura da imagem 2D e o mapa de profundidade 3D.



Fonte: Weise et al. (2011b).

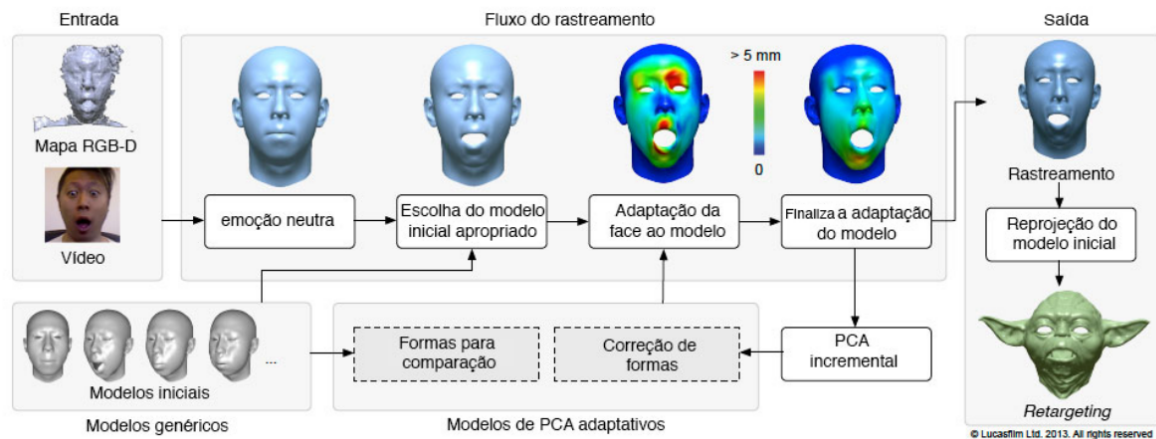
A fase *online* tem início com uma captura da face do ator em tempo real para a leitura das informações 2D e 3D, seguindo por uma etapa para determinar os pesos aplicados aos *blendshapes* e, no final, a transferência dos pesos obtidos para a animação. Para a determinação dos pesos, de forma mais realista, o autor explora uma coerência temporal entre quadros capturados consecutivos, produzindo um resultado mais elaborado, tanto para a geometria quanto para o movimento do usuário rastreado. Para determinação dos pesos é utilizado a MPPCA (*Probabilistic Principal Component Analyzers*) (TIPPING; BISHOP, 1999).

4.2 Método de Li et al. (2013b)

O método apresentado por Li et al. (2013b), captura a face através do *Kinect*. São utilizadas imagens RGB-D e vídeos 2D em tempo real. Ele possui uma etapa de pré-processamento feito de forma *offline* e, o processo *online*, feito em tempo real. Em ambos, alguns algoritmos e métodos, para ajuste da malha e redirecionamento, são utilizados como o ICP ¹ (*Iterative closest point*) e o PCA ² (*Principal Component Analysis*).

O método é iniciado por uma captura da face na posição neutra. São usadas as informações RGB-D e 40 marcadores faciais, capturados automaticamente, que correspondem aos pontos ao redor da boca e dos olhos. Uma visão geral do método, pode ser vista na Figura 4.2.

Figura 4.2 – Visão geral do processo.



Fonte: Li et al. (2013b).

No processo *online*, é iniciada a captura do ator, por meio, do *Kinect* que traz a imagem de profundidade e o vídeo 2D. Nessa etapa, é usado o ICP pra ajustar a malha 3D. Após o ajuste, o módulo de cálculo dos pesos indica qual é o modelo inicial da base de dados de *blendshape* que deverá ser utilizado. Na etapa seguinte, é feita uma adaptação ao modelo, usando um ICP não rígido. No ajuste final é usada uma técnica de PCA incremental, refinando os coeficientes usados para o redirecionamento.

4.3 Método de Seol, Ma e Lewis (2016)

Seol, Ma e Lewis (2016) propuseram um método que transfere faces para um objeto 3D de maneira semiautomática. Ele depende de blendshapes e de uma face pré-configurada

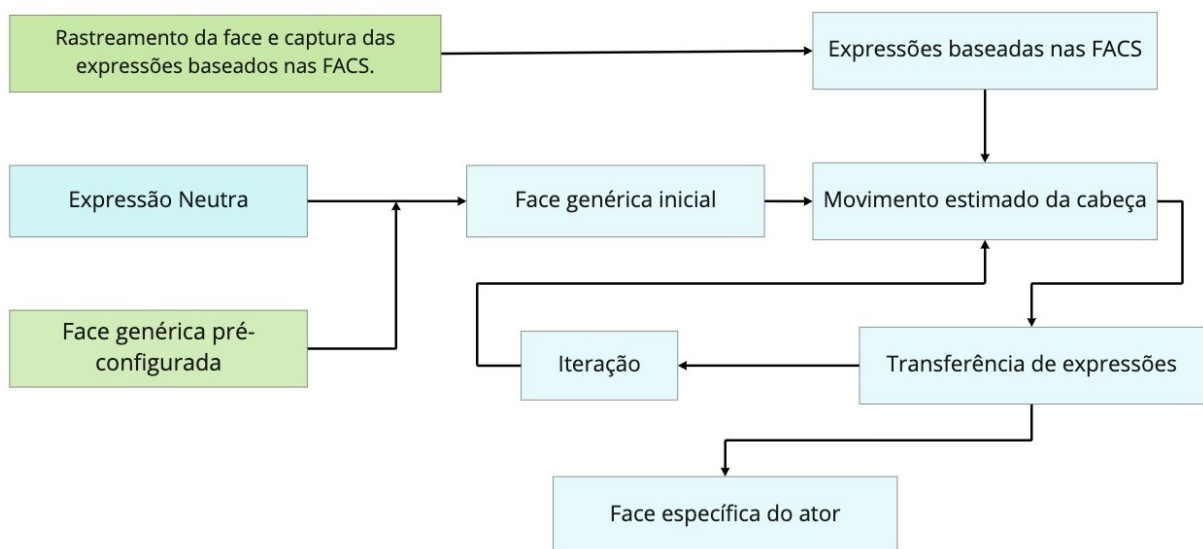
¹ Este método, tornou-se dominante no alinhamento tridimensional de modelos baseados em geometria e malhas. Ele é usado neste trabalho para ajustar a captura da malha nas fases iniciais do método, corrigindo as suas imperfeições (RUSINKIEWICZ; LEVOY, 2001).

² Método amplamente utilizado para a redução de dimensionalidade das informações, compressão de dados, extração de características relevantes e visualização de informação (BRAUN, 2014).

do ator capturado. O *pipeline* é dividido em 4 etapas:

- Captura e detecção da expressão facial;
- Criação da face customizada do ator com as capturas na primeira etapa;
- Estimativa de movimento da cabeça; e
- Transferência da expressão capturada para a face 3D.

Figura 4.3 – Visão geral do processo.



Fonte: Seol, Ma e Lewis (2016).

Na primeira etapa são capturadas do ator, as expressões baseadas nas FACS incluindo uma expressão da face neutra e os movimentos básicos como a boca aberta, olhos abertos, entre outros. Após essa etapa, são construídas as faces 3D, através de uma malha de pontos.

Na segunda etapa são transferidas todas as faces criadas na primeira, para uma face pré-configurada, onde os movimentos capturados podem ser manipulados. Isso cria a face inicial, que se parece com o ator e, tem expressões base para a manipulação. No entanto, essa face ainda não é específica do ator.

Na terceira etapa, para cada expressão rastreada, é feita uma normalização da posição da face, por meio, do algoritmo de Procrustes (Anexo A.1).

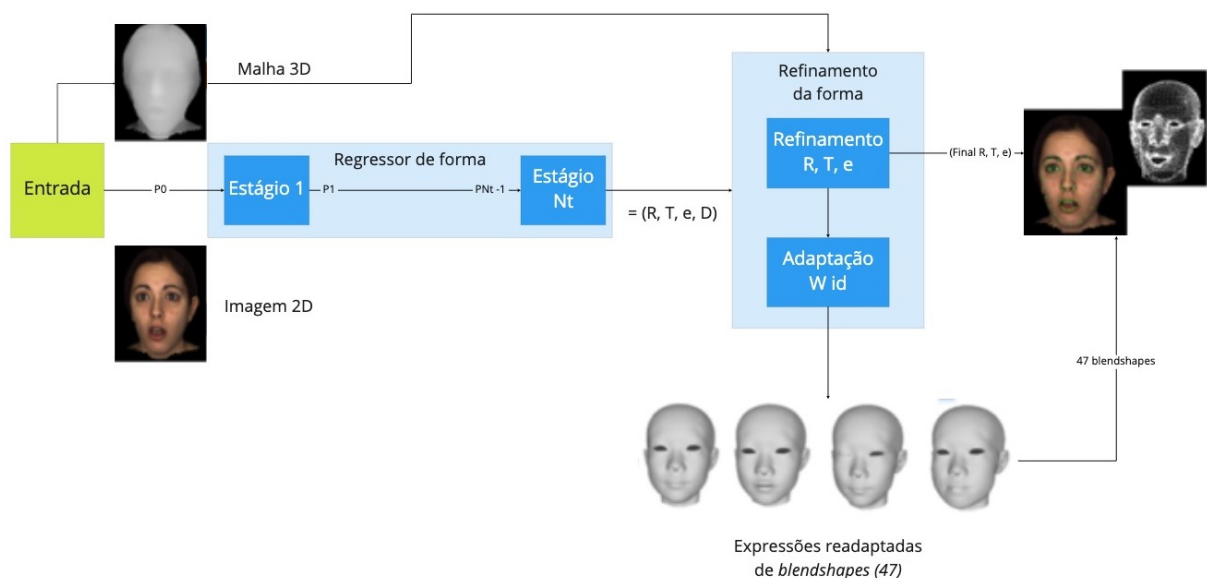
Na última etapa é feita a transferência da expressão capturada do ator de forma *online*, para a face inicial, gerada anteriormente, através do cálculo da diferença entre a face capturada e as expressões criadas anteriormente.

4.4 Método de Pham et al. (2016)

Pham et al. (2016) propôs um método para redirecionamento da animação facial, que combina as vantagens da regressão de imagens RGB e o ponto 3D capturados por uma câmera RGB-D. Ele usa o algoritmo de Floresta Aleatória (subseção 3.4.1) para treinar uma rede que calcula os pesos a serem aplicados ao *blendshapes*.

Esse método traz inovação com possibilidade de que as capturas para o redirecionamento sejam feitas em uma distância maior do que outros trabalhos. Dessa maneira, esse método foca em problemas onde o ator não está próximo como, por exemplo, uma sala de vídeo conferência. Na Figura 4.4 pode ser visto o sistema proposto.

Figura 4.4 – Visão geral do processo.



Fonte: Pham et al. (2016).

O sistema proposto consiste em dois módulos principais: o módulo de regressão e o módulo de refinamento. O módulo de regressão, executa o primeiro estágio de otimização das características, através de uma rede treinada com as informações 2D capturadas da imagem de uma base de dados a *Face Warehouse* (CAO et al., 2014). O refinamento utiliza, além dos dados 2D, os dados 3D da nuvem de pontos encontrados nas imagens RGB-D.

Para a geração dos pesos, a rede faz uma predição da possibilidade da captura pertencer a alguma expressão treinada. Isso faz, com que seja gerado um vetor de características com os valores normalizados entre 0 e 1. Este vetor é usado para a construção da face final, como citado no seção 2.1.

4.5 Método de Aneja et al. (2018)

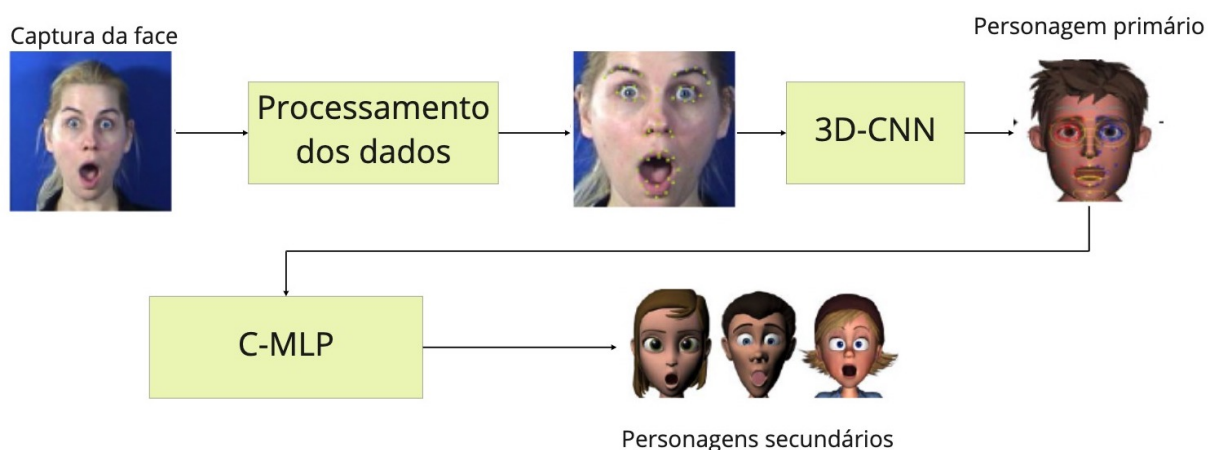
O trabalho apresentado por Aneja et al. (2018), desenvolveu um método semi-supervisionado de redirecionamento de movimento facial que, através do mapeamento 2D de um rosto humano, prevê os parâmetros necessários para animação da face de personagens 3D estilizados.

O sistema que foi proposto utiliza bases de dados de expressões faciais divididas em seis categorias: raiva, nojo, medo, alegria, tristeza, surpresa e a expressão neutra. Essas expressões ocorrem em uma ampla gama de intensidade e podem se misturar para criar expressões adicionais.

Através da análise de similaridade, é treinada uma rede neural convolucional CNN (INDOLIA et al., 2018) que faz a análise características 2D e transferência de expressões da face do ator para o personagem estilizado 3D.

Na etapa de captura da face, são extraídos 49 marcadores faciais. Os marcadores são usados como pontos de referência para extrair as características geométricas da face, incluindo a altura da sobrancelha esquerda, altura da sobrancelha direita, a distância entre o topo da sobrancelha, o centro do olho, altura da pálpebra esquerda, altura da pálpebra direita, largura do nariz, medida de boca fechada, altura do lábio esquerdo e altura do lábio direito. As distâncias das áreas analisadas são normalizadas pelos marcadores faciais do contorno da face. Na Figura 4.5 pode ser visto o *pipeline* do método proposto.

Figura 4.5 – Visão geral do método proposto.



Fonte: Aneja et al. (2018).

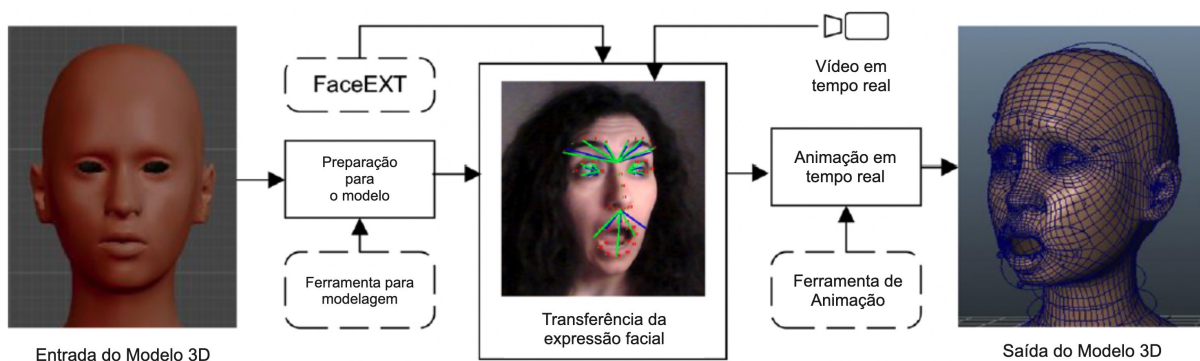
Na primeira etapa é feita a captura da face do ator. Após a captura, é feito o processamento por meio da extração dos marcadores faciais. Os marcadores faciais obtidos na etapa de captura são utilizados para rotular uma base de dado de personagens a FERG-DB (ANEJA et al., 2016), para encontrar a semelhança entre eles, através da distância euclidiana (Anexo A.2). Com isso a base é rotulada.

Após essa etapa, uma rede do tipo CNN é treinada para detectar a similaridade da captura da face com os personagens. No final, uma rede do tipo MLP, *Multi-Layer Perceptron* (CHOI; PARK, 2021) é utilizada para fazer a transferência da expressão para o personagem.

4.6 Método de Ekmen e Ekenel (2018)

Neste artigo, Ekmen e Ekenel (2018) apresentam um método de transferência facial que cria animações realistas, através do rastreamento de expressões de um rosto humano em 2D e transferindo-as para um Modelo 3D em tempo real. O método não necessita de calibração e não requer treinamento. O rastreamento é realizado usando uma única câmera RGB. Na primeira etapa, uma face 3D, baseada em articulações, é anexada ao modelo. A segunda etapa abrange o rastreamento dos marcadores faciais, a partir de uma única imagem e a transferência do movimento relativo 3D, para mover as respectivas juntas, no modelo. A última etapa inclui a gravação de animação que usa uma técnica de *keyframe*³ parcialmente automatizada.

Figura 4.6 – Visão geral do método proposto.



Fonte: Ekmen e Ekenel (2018).

O modelo usado para a transferência facial, foi baseado em pseudo-músculos, pois, segundo o autor, resulta em animações mais realistas imitando as expressões humanas. Com isso, as articulações, que estão localizadas no rosto 3D, considerando os recursos de rastreamento facial e movimentos musculares, controlam os movimentos e expressões faciais.

Para a animação facial, o sistema transfere as coordenadas de rastreamento 2D para 3D, obtidos na entrada de movimento quadro a quadro. Essa transferência altera

³ *keyframe*, este é o momento mais importante de uma animação. Como o nome diz, essas etapas são a "chave" (key) de todos os quadros de uma animação. São elas, as que apresentam diferenças nas ações de um personagem, uma mudança crucial no movimento, ou até uma grande pose. (SILVEIRA, 2017)

a expressão do modelo 3D produzindo expressões, o mais próximo da captura do vídeo, fornecendo precisão e realismo.

É utilizado um sistema de cliente servidor onde a captura é feita no cliente e as informações são transferidas, via rede, para um sistema baseado no *Maya* da *Autodesk* (RITCHIE; CALLERY; BIRI, 2005).

Para a transferência da expressão capturada para o modelo 3D, o sistema compara o deslocamento dos pontos capturados, quadro a quadro, com a face neutra do ator capturado previamente. Essa comparação é feita através da distância euclidiana (Anexo A.2).

4.7 Método de Struijk et al. (2018)

Struijk et al. (2018) desenvolveu um *framework*⁴ chamado *FACSVatar* criado para possibilitar a animação facial baseado nas FACS em *softwares* como *Unity 3D* ou *Blender*. Ele utiliza implementações de código aberto para compor os módulos propostos. A *pipeline* implementada para o *framework* pode ser vista na Figura 4.7

Todos os módulos do *framework* se comunicam de forma assíncrona, trocando mensagens entre si. Os módulos efetivados e testados foram:

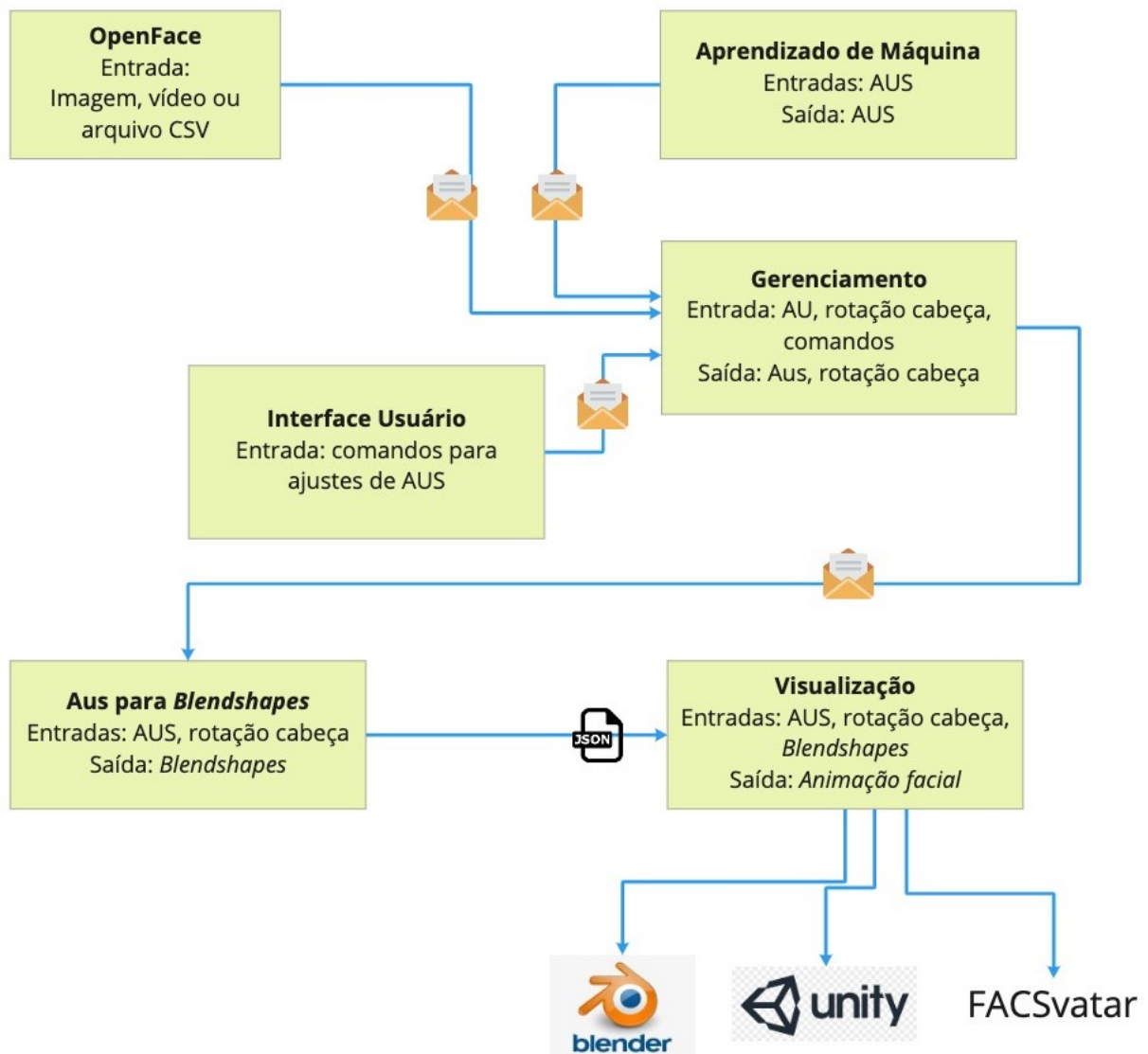
- Entrada dos FACS, movimento dos olhos e rotação da cabeça em tempo real e *offline*;
- Gerenciamentos das mensagens assíncronas;
- Interface gráfica para ajustes das AUS;
- Rede Neural para geração da animação facial;
- Conversão dos AUS para os *blendshapes*; e
- Visualização dos resultados *Blender* (COMMUNITY, 2018), *Unity3D* (HAAS, 2014) ou *FACSHuman* (GILBERT; DEMARCHI; URDAPILLETA, 2018).

Para o módulo inicial foi utilizada a implementação *OpenFace 2.0*, uma ferramenta para pesquisadores de visão computacional e aprendizado de máquina, comunidade de computação afetiva e pessoas interessadas em construir aplicativos interativos baseados em análise de comportamento facial, segundo Baltrusaitis et al. (2018), modificado para enviar os AU da face capturada, a posição dos olhos e a rotação da cabeça em tempo real.

O controle de informações entre os módulos é feito de maneira assíncrona, por meio, da troca de mensagens. Foi desenvolvido um gerenciador de mensagens que funciona

⁴ *framework*, que é uma coleção de objetos e padrões dedicados a resolver determinados problemas em uma arquitetura flexível e extensível.

Figura 4.7 – Visão geral da estrutura de módulos. As linhas em azul representam os módulos usados desenvolvidos.



Fonte: Struijk et al. (2018).

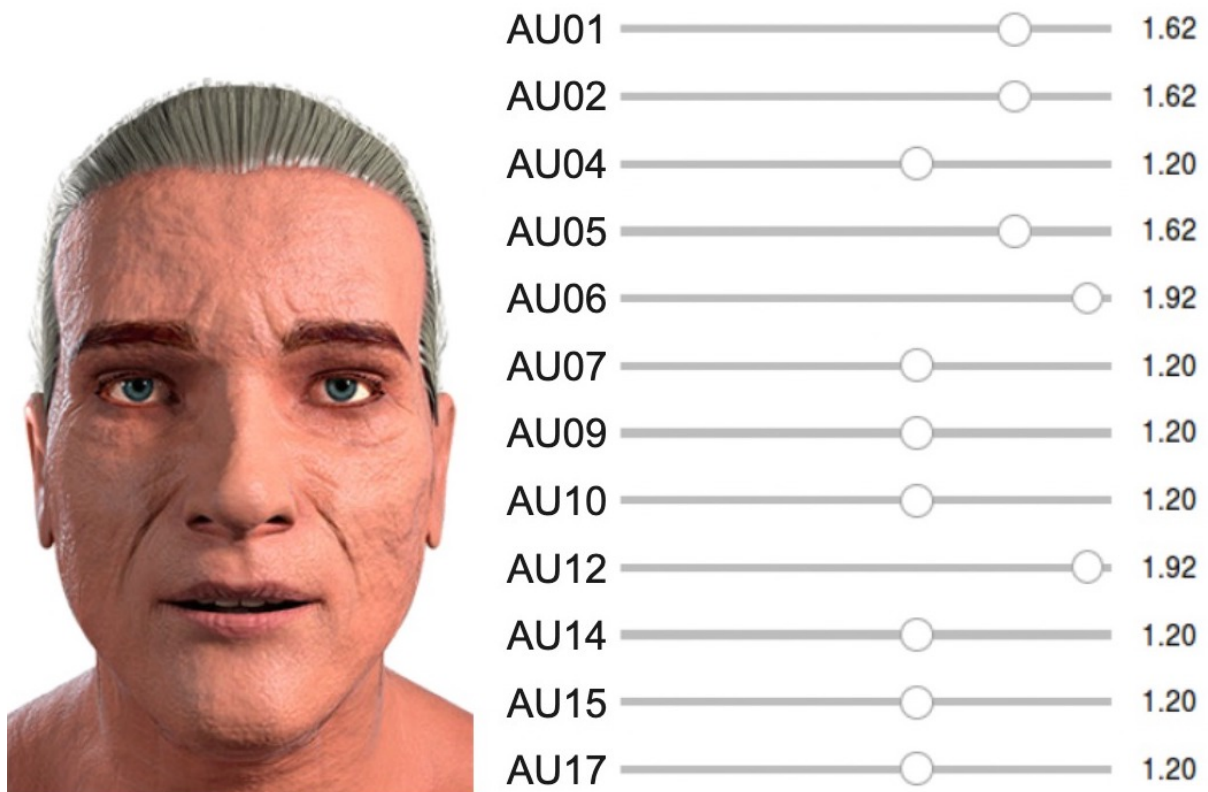
como uma central de envio e recebimento de informações, que são repassadas aos outros módulos do *pipeline*.

A interface gráfica para ajuste das AUS foi desenvolvida em um módulo separado, através de comandos horizontais, que representam os tipos de AUS (Figura 4.8).

O módulo de aprendizado de máquina utilizou uma rede neural *Gated Recurrent Unit* - GRU (CHO et al., 2014) implementada em Keras (CHOLLET et al., 2015), que calcula a animação facial através da análise das AUS. A saída se dá, por meio, dos valores de 17 diferentes AUS indicando a sua relevância para o redirecionamento. Os dados da base de dados MAHNOB Mimicry (BILAKHIA et al., 2015) foram usados para o treinamento.

No módulo de conversão dos AUS, para os *Blendshapes*, é feita uma análise de todas

Figura 4.8 – Interface do módulo ajuste dos AUS.



Fonte: Struijk et al. (2018).

as possibilidades disponíveis encontrando assim, a melhor combinação entre os *blendshapes* para às expressões identificadas pelos AUS.

Como o *framework* foi desenvolvido para ser um *plugin*, o resultado final se dá nos softwares *Unity 3D*, *FACSHuman* ou *Blender*.

4.8 Método de Chaudhuri, Vesdapunt e Wang (2019)

Chaudhuri, Vesdapunt e Wang (2019) propuseram um método que prevê o redirecionamento da animação facial para múltiplas faces, diferente de outros métodos que trabalham com uma única face, através de uma rede neural convolucional (CNN *Convolutional Neural Network*). Para treinar essa rede são usados os parâmetros da 3DMM⁵ (BLANZ; VETTER, 1999) para a extração das características.

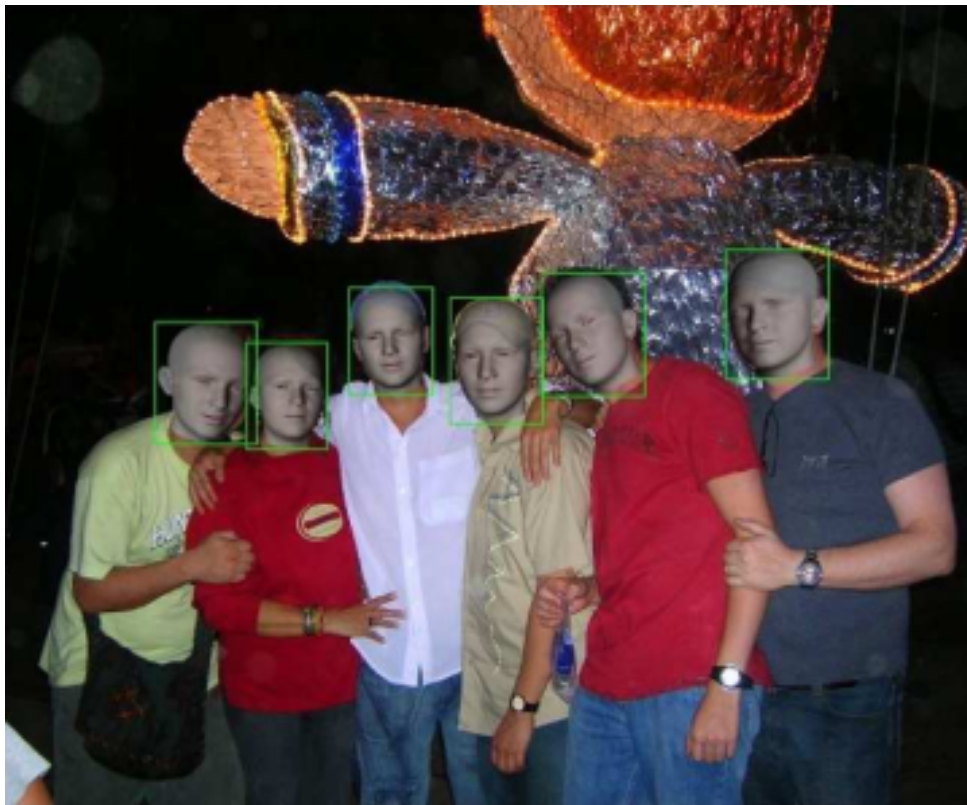
O método é dividido em duas partes. Na primeira parte, é treinada uma rede utilizando os parâmetros 3DMM para detecção de uma única face chamada SFN (*Single Face Network*). São eles: parâmetros de formato da face, parâmetros que capturam a

⁵ 3DMM que é uma ferramenta para reconstruir a face 3D a partir de uma única imagem de face 2D. No processo de ajuste de forma, o 3DMM estima a correspondência entre os marcadores 2D e 3D. (YANGA; WU; KITTLER, 2018)

expressão facial, parâmetros da rotação 3D da cabeça e parâmetros de escala que vinculam a face 3D à imagem 2D.

Na segunda parte, o processo é expandido para detectar múltiplas faces em uma única imagem. Um exemplo do método pode ser visto na Figura 4.9

Figura 4.9 – Resultado do método proposto por Chaudhuri, Vedapunt e Wang (2019).



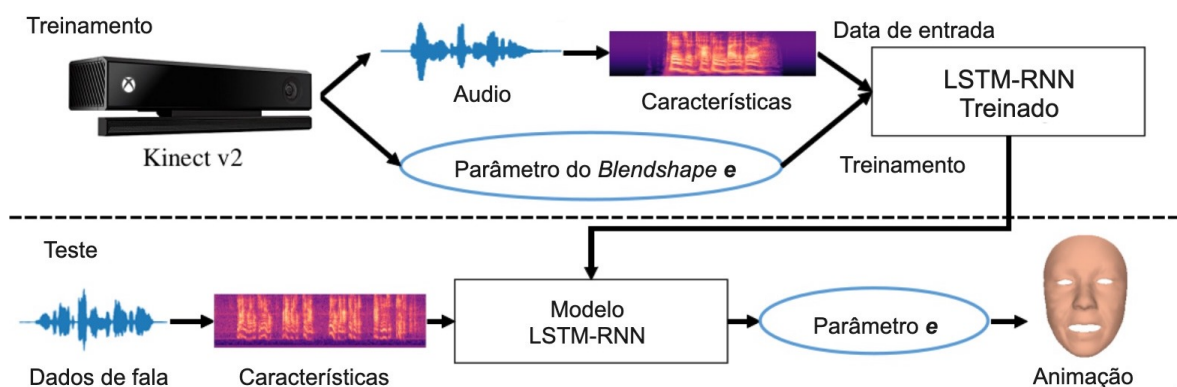
Fonte: Chaudhuri, Vedapunt e Wang (2019).

Para o redirecionamento da animação facial final, a rede treinada gera os pesos finais baseados nas entradas extraídas dos parâmetros 3DMM.

4.9 Método de Nishimura et al. (2019)

Nishimura et al. (2019) apresentaram um método para redirecionamento da animação facial que utiliza a fala. Segundo os autores, os modelos tradicionais que usam uma câmera convencional podem causar limitações como, por exemplo, o ângulo da captura que, dependendo da posição da face, pode comprometer o processo. Nesse método, é utilizado LSTM (*Long Short Term Memory*), que é uma variação da RNN (*Recurrent Neural Networks*) (ACADEMY, 2019) onde a entrada são os dados de análise da fala e a saída os pesos para os *blendshapes*. O sistema usa o *Kinect* para a captura dos pontos da face e geração dos *blendshapes*. Uma visão geral do método pode ser visto na Figura 4.10.

Figura 4.10 – Visão geral do método proposto por Nishimura et al. (2019).



Fonte: Nishimura et al. (2019).

O sistema proposto estima expressões faciais de fala por LSTM-RNN, porque os dados de áudio são um sinal com séries temporais possuindo uma característica que se propaga, sequencialmente, ao longo do tempo. A rede LSTM-RNN, através dos dados de áudio, gera os pesos dos *blendshapes* em valores entre 0 e 1.

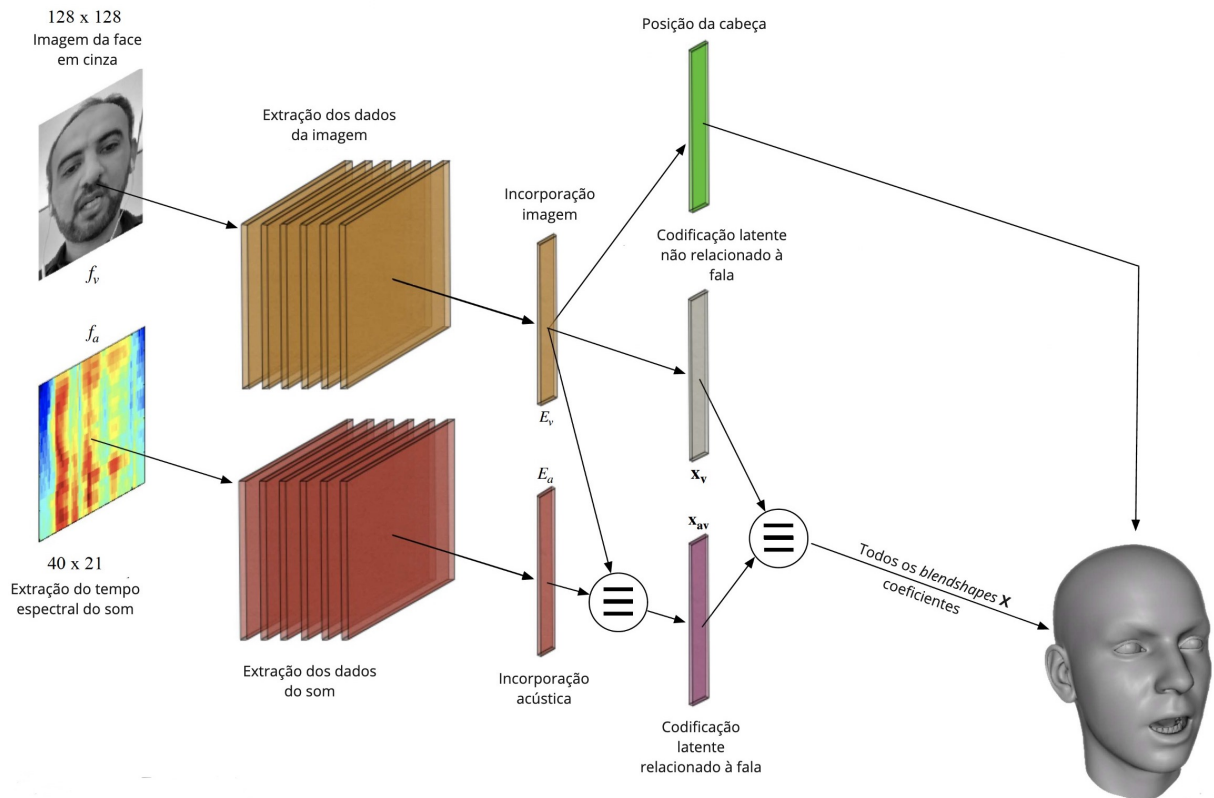
4.10 Método de Abdelaziz et al. (2020)

O método de redirecionamento facial proposto por Abdelaziz et al. (2020) utiliza como entrada de dados informações acústicas e visuais. O modelo proposto roda em tempo real, em *hardware* com recursos limitados como, por exemplo, um telefone celular e, é independente, podendo funcionar com qualquer tipo de usuário. A saída são pesos para os *blendshapes* baseados nas FACS.

A Figura 4.11 apresenta uma visão geral do *pipeline* para rastreamento e animação. Os recursos visuais de entrada fv são imagens em escala de cinza. As faces são detectadas, através, de uma rede neural pré-treinada quadro a quadro. Os recursos acústicos de entrada fa são extraídos das amostras de fala bruta. São capturados, simultaneamente e, no mesmo dispositivo, informações de áudio e vídeo, para que os sinais sejam sincronizados no momento da captura.

Para o cálculo dos pesos utilizados na geração da face final, o método proposto treinou uma DNN (*Deep neural networks*) (GAWLIKOWSKI et al., 2021), para mapear imagens da face de entrada para coeficientes de *blendshape*. A rotulagem da base de dados para treinar a rede, foi feita através de um método proposto por Weise et al. (2011a), que gerou os coeficientes, a partir de uma série de modelos personalizados. Esses modelos foram criados adaptando um modelo genérico de *blendshape*.

Figura 4.11 – Visão geral do método proposto por Abdelaziz et al. (2020).



Fonte: Abdelaziz et al. (2020).

4.11 Considerações finais sobre os métodos apresentados

A Tabela 1 sintetiza as principais características dos métodos levantados.

Tabela 1 – Características principais dos métodos levantados.

Método	Captura	Algoritmo	Geração dos pesos
Weise et al. (2011b)	RGB-D	MPPCA	Dados 3D e 2D
Li et al. (2013b)	RGB-D	ICP e PCA	Dados 3D e 2D
Seol, Ma e Lewis (2016)	RGB	Deformação Malha	Dados 3D e 2D
Pham et al. (2016)	RGB e RGBD	RF	Dados 2D e 3D
Ekmen e Ekenel (2018)	RGB	SDM	Dados 2D
Aneja et al. (2018)	RGB	CNN	Dados 2D
Struijk et al. (2018)	RGB	GRU	Dados 2D
Chaudhuri, Vesdapunt e Wang (2019)	RGB	CNN	Parâmetros 3DMM
Nishimura et al. (2019)	Som	LSTM	Fala
Abdelaziz et al. (2020)	Som e Vídeo	DNN	Dados 3D e 2D

Fonte: elaborada pelo autor.

O método apresentado por Weise et al. (2011b) foi pioneiro no uso de câmeras RGB-D para o redirecionamento facial, mas o processo para a geração da base de expressões é manual. Já, Li et al. (2013b) também usa os dados RGB-D, mas precisa que seja estimada

uma face neutra do ator, por meio, da captura de um quadro no começo do processo *online*.

Seol, Ma e Lewis (2016) usa uma pré-configuração genérica, em que, ajusta as malhas específicas do ator, depois da geração. Para a captura se usa câmera RGB.

O método apresentado por Pham et al. (2016) propôs o redirecionamento de faces a longas distâncias, mas o processo possui configurações em duas etapas dependentes uma da outra, dificultando a configuração inicial para o redirecionamento.

Já, o método apresentado por Ekmen e Ekenel (2018) possui um mecanismo que foi dividido em 3 etapas, sendo a primeira, a entrada do modelo 3D, a segunda, a captura dos marcadores faciais e seus processamento e, no final, a saída do modelo 3D. Mesmo não utilizando *blendshapes*, o método, usa referências 2D e câmera RGB no seu processo, sendo adaptado para uso em outros tipos de saída.

Aneja et al. (2018) apresentou um método de animação facial para personagens estilizados. São utilizados marcadores faciais para identificar as características e treinamento da base de dados, de forma não supervisionada. Após, a identificação dos personagens, é treinada uma rede do tipo CNN para a transferência das expressões entre a face capturada e o personagem estilizado.

O trabalho apresentado por Struijk et al. (2018) mostra o desenvolvimento de um *framework* para redirecionamento facial. A ideia principal foi o desenvolvimento de um modelo flexível, com módulos se comunicando de forma assíncrona e utilizando projetos de código aberto. Sendo seu funcionamento um *plugin*⁶, a saída pode ser redirecionada para os *softwares* *Blender*, *Unity3D* ou *FACSHuman*

Chaudhuri, Vedapunt e Wang (2019) também apresentam um método inovador prevendo o redirecionamento facial para várias capturas na mesma imagem, mas o processo de treinamento para geração dos pesos passa por duas etapas, sendo a primeira, para uma única foto e a segunda, para várias fotos, levando em conta algumas informações do primeiro processo. Como são processos complementares, eles compartilham informações entre si, dificultando a calibração inicial e o redirecionamento da face.

O método de Nishimura et al. (2019) sugere o redirecionamento facial através da captura da fala, visto que, alguns métodos mais tradicionais que utilizam a captura da face, podem sofrer problemas quando as obstruções como por exemplo, com o cabelo na frente dos olhos ou sobrancelhas e a posição da cabeça. Para melhoria, esse método aponta problemas na sincronização das expressões, principalmente, no movimento da boca.

Finalmente, o método proposto por Abdelaziz et al. (2020) utiliza a fala e imagem sincronizadas e uma rede DNN para o cálculo dos pesos dos *blendshapes*.

⁶ *Plugin* e todo programa, ferramenta ou extensão que se encaixa a outro programa principal, para adicionar mais funções e recursos a ele.

5 O sistema de animação facial proposto

Neste capítulo são apresentados os detalhes do sistema de animação facial baseado em performance utilizando *blendshapes* desenvolvido nessa tese. Primeiramente é apresentada uma visão geral e, depois, uma visão detalhada do sistema onde serão discutidas as soluções encontradas.

5.1 Visão geral do método desenvolvido

O método desenvolvido consiste na criação de um *pipeline* para um sistema de animação facial baseado em *blendshapes*, que utilizou os classificadores OPF e Floresta Aleatória (*Random Forest*) para a fase de *retarget*. A sua estrutura é mostrada na Figura 5.1 onde pode ser vista uma visão geral dos módulos existentes.

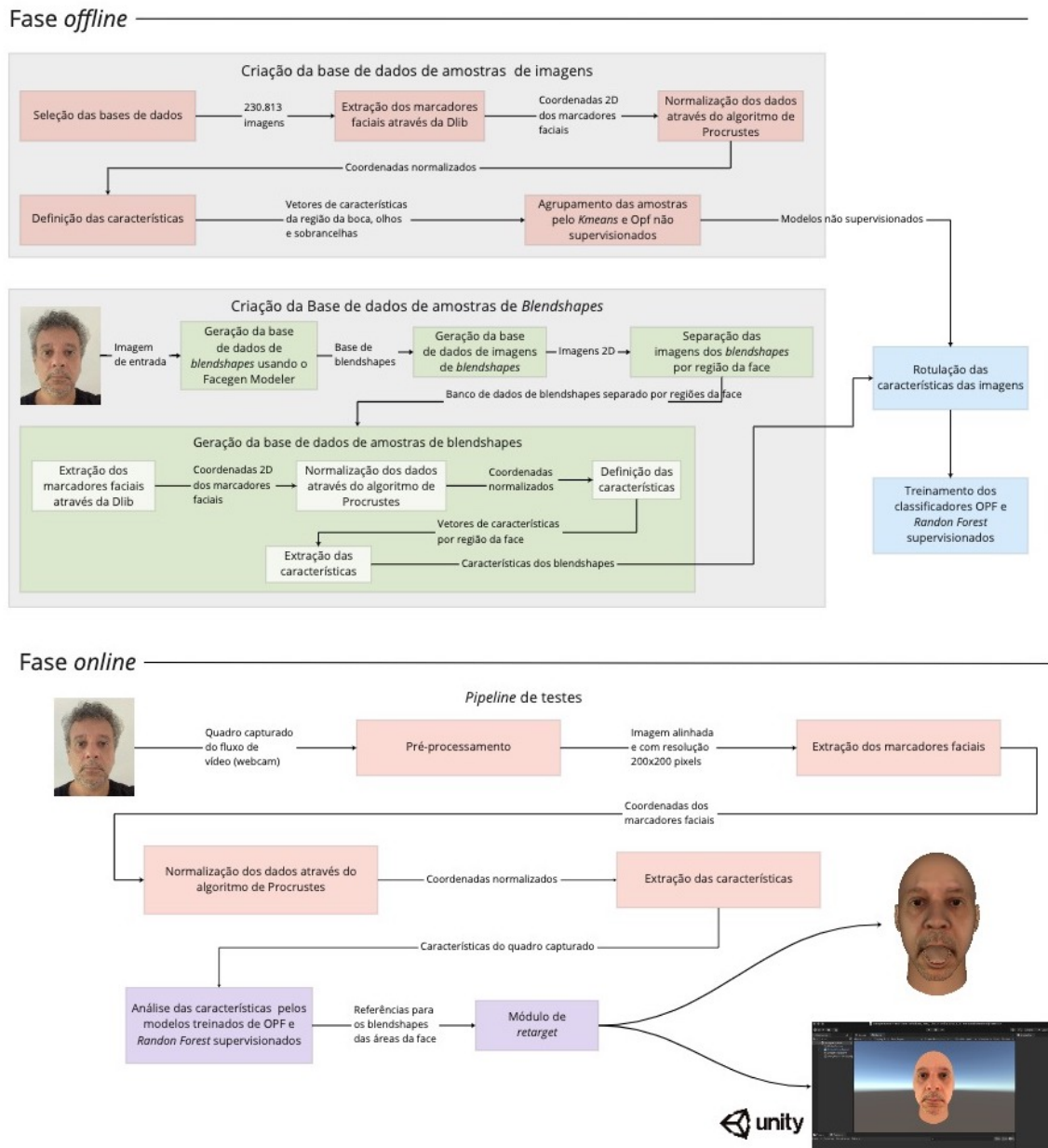
Na fase *offline* ocorrem dois processos: a criação da base de dados de amostras de imagens e a criação da base de dados de amostras de *blendshapes*. Em ambos os processos, são criadas as bases de dados de amostras para o treinamento dos classificadores. Na etapa de criação da base de dados de amostras de imagens, o processo se inicia com a seleção de bases de dados de expressões. Foram selecionadas as bases FEAFa, MUG, KDEF e CK+. Em comum, essas bases possuem faces de diferentes indivíduos, com diferentes tipos de expressões.

Já, no processo da criação da base de dados de amostras de *blendshapes*, o processo se inicia com a geração de uma base de dados de *blendshapes*, através do *software Face Gen Modeler*, que, usando uma única foto, consegue gerar cerca de 125 diferentes tipos de expressões em 3D. Esse processo é fundamental para a pesquisa, pois a saída final do *pipeline* é um *blendshape* com a face capturada do ator. Além da geração dos *blendshapes*, nessa etapa ocorre a exportação dos objetos 3D para imagens 2D, sendo que, a pesquisa proposta, para fazer o redirecionamento da face usa as informações 2D e não as informações 3D. No próximo passo, que é a separação das imagens de *blendshape* por região, são selecionadas apenas as imagens dos *blendshapes* que possuem relevância para as 3 áreas que serão analisadas pelos classificadores (boca, olhos e sobrancelhas).

Ao final dos processos citados, tanto a criação da base de dados de amostras de imagens quanto a criação da base de dados de amostras de *blendshapes* compartilham etapas em comum, que são:

- Extração dos marcadores faciais, que são utilizados na definição das características de ambas as bases, por meio, das suas coordenadas 2D;

Figura 5.1 – Estruturação do método proposto desenvolvido.



Fonte: elaborada pelo autor.

- Normalização dos dados que, tanto as imagens das bases de dados de expressões quanto às imagens dos *blendshapes* devem estar com a mesma especificação, quanto a tamanho e posicionamento; e
- A definição das características que transformam as informações 2D em vetores separados por região da face.

Após a execução dessas etapas, para finalizar, é feito o agrupamento de amostras onde, através de classificadores não supervisionados, são encontradas similaridades entre

as amostras, os *clusters*. Essa fase gerou 3 modelos de classificadores não supervisionados. Isso foi necessário para o processo de rotulação das características das imagens.

No processo de rotulação das características das imagens, foram submetidas as amostras de *blendshapes* aos classificadores não supervisionados para a identificação do *cluster*, a qual a mesma pertence, permitindo a rotulação das amostras da base de dados de imagens em função das características das amostras de *blendshapes*. Após esse procedimento, foi gerada uma base de dados de exemplares de imagens rotuladas com os nomes dos *blendshapes*, o qual, foi utilizado na fase do treinamento do classificador supervisionado.

Na etapa do classificador supervisionado, foram treinados modelos usando a base de dados de amostras rotuladas gerando 3 modelos, um para cada área da face. Esses modelos são utilizados na fase *online* para a identificação dos *blendshapes* relevantes para cada área da face, quando são submetidos à amostra da face capturada.

Na fase *online*, a face do ator é capturada por uma câmera RGB e, cada um dos quadros do fluxo de vídeo, passa pelas seguintes etapas: pré-processamento; extração de marcadores e normalização e extração de características, similares às etapas descritas no processo *offline*. Após, a extração das características, as mesmas são submetidas aos modelos supervisionados treinados na etapa *offline*, a fim de se identificar os *blendshapes* relevantes para cada área da face, necessários para o processamento no módulo de *retarget*. Ao final da fase *online*, tem-se dois tipos de saídas: um arquivo com o *blendshape* final ou, então, a exibição do *blendshape* em um ambiente virtual criado no motor de jogos *Unity 3D*.

5.2 Visão detalhada do método

É apresentado nessa seção, o detalhamento de cada uma das fases e *pipelines* decorrentes do método desenvolvido nesta tese.

5.2.1 Criação da base de dados de marcadores faciais

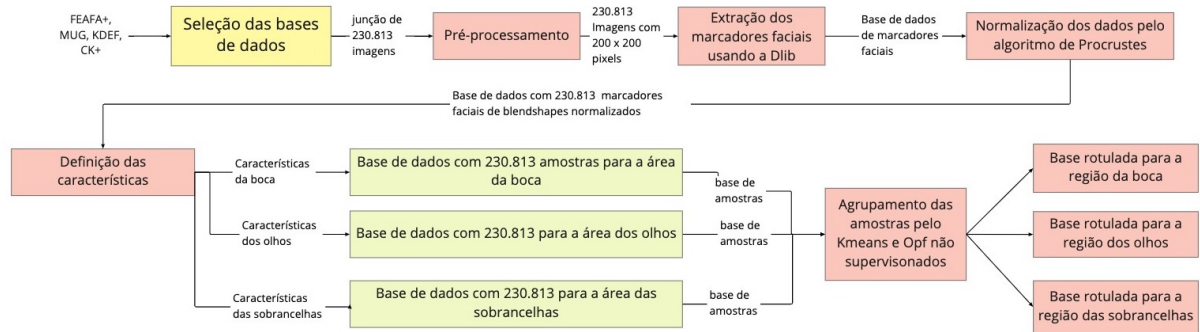
A criação da base de dados foi dividida em cinco etapas: seleção das bases de dados de faces, pré-processamento, extração dos marcadores faciais, normalização dos dados e geração da base de dados de características das imagens (Figura 5.2).

5.2.1.1 Seleção das bases de dados de faces

Para a criação da base de expressões faciais foram utilizadas 4 bases de dados: FEAFa+, MUG, KDEF e CK+.

A FEAFa+ (*An Extended Well-Annotated Dataset for Facial Expression Analysis and 3D Facial Animation*) consiste em dois subconjuntos de dados: a FEAFa, desenvolvida

Figura 5.2 – Etapas para a criação da base de dados.



Fonte: elaborada pelo autor.

por Yan et al. (2019) e a DISFA, desenvolvida por Mavadati et al. (2013). A FEAFA disponibiliza 127 vídeos, que somados, possuem 99.356 quadros. Participaram 122 pessoas, incluindo crianças, adultos jovens e idosos, que foram registrados em condições reais. Na DISFA, existem 27 vídeos de atores em expressões espontâneas, somando no total, 130.828 quadros. Os participantes eram principalmente adultos jovens de várias raças (Figura 5.3).

Figura 5.3 – Exemplos de algumas imagens coletadas de vídeos da base de dados FEAFA+.



Fonte: Mavadati et al. (2013).

A *MUG (Facial Expression Database)*, desenvolvida por Aifanti, Papachristou e Delopoulos (2010) consiste em sequências de vídeos curtos de 86 pessoas realizando expressões faciais. Os vídeos dos atores foram gravados a uma taxa de 19 quadros por segundo. Todos os atores estavam sentados e à frente de um fundo azul (vide Figura 5.4). Cada imagem foi salva em formato jpg com uma resolução de 896 x 896 *pixels*. Nos vídeos participaram 35 mulheres e 51 homens, todos de origem caucasiana, entre 20 e 35 anos de idade. Não há oclusões nas faces, exceto por alguns cabelos caindo no rosto. A criação da base de dados foi dividida em duas partes. Na primeira parte, foi solicitado aos atores que realizassem as seis expressões básicas, que são: raiva, nojo, medo, alegria, tristeza e surpresa. A segunda parte contém emoções induzidas em laboratório.

Para cada uma das seis expressões, foram salvos vídeos com, aproximadamente, de 50 a 160 quadros. Os vídeos são disponibilizados em formato AVI totalizando 1462 arquivos.

Na segunda parte do banco de dados, os atores foram filmados enquanto assistiam a um vídeo que foi criado para induzir emoções. O objetivo desta parte foi obter expressões autênticas no ambiente de laboratório, em contraste com as posadas capturadas na primeira parte. Foram salvos vídeos capturados a 19 quadros por segundo e com uma resolução de 896 x 896 pixels.

Figura 5.4 – Exemplos de algumas imagens coletadas de vídeos da base de dados MUG.



Fonte: Aifanti, Papachristou e Delopoulos (2010).

A *KDEF* (*Karolinska Directed Emotional Faces*), desenvolvida por Goeleven et al. (2008), é um conjunto de 4900 imagens de expressões faciais humanas. O conjunto das mesmas, contém 70 indivíduos exibindo 7 expressões emocionais diferentes (neutro, feliz, zangado, com medo, desgostoso, triste, surpreso), conforme mostrado na Figura 5.6. Cada expressão é vista de 5 ângulos diferentes (perfil esquerdo completo, perfil meio esquerdo, reto, perfil meio direito, perfil direito completo). Participaram da criação dessa base de dados 70 atores amadores, sendo 35 mulheres e 35 homens, com idade entre 20 e 30 anos sem barba, bigode, brincos ou óculos. As imagens possuem resolução de 562 x 762 *pixels*.

A *Extended Cohn-Kanade (CK+)* desenvolvida por Lucey et al. (2010) foi criada para classificação de expressões faciais. É uma base de dados codificada por FACS e, os rótulos de emoção, foram revisados e validados. Contém 593 sequências de vídeos de um total de 123 pessoas, entre 18 a 50 anos de idade, com uma variedade de gêneros. Cada vídeo mostra uma mudança facial da expressão neutra, para uma expressões de raiva, desprezo, nojo, medo, felicidade, tristeza e surpresa, gravada a 30 quadros por segundo com resolução de 640 x 490 ou 640 x 480 *pixels*.

Figura 5.5 – Exemplos de algumas imagens coletadas de vídeos da base de dados KDEF.



Fonte: Goeleven et al. (2008).

Figura 5.6 – Exemplos de algumas imagens coletadas de vídeos da base de dados CK+.



Fonte: Lucey et al. (2010).

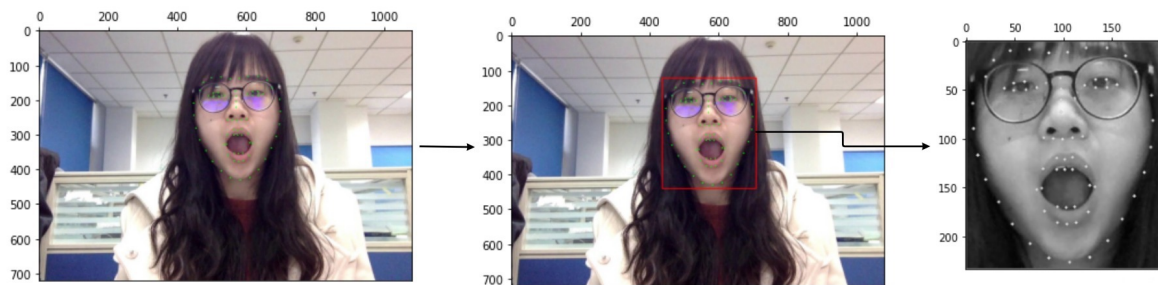
De maneira a compatibilizar e padronizar as bases de dados houve a necessidade de um pré-processamento nas imagens. Para isso, nas bases de dados que disponibilizaram apenas vídeos, foram extraídas as imagens quadro a quadro. Por padrão, as imagens ficaram com 200 x 200 *pixels*. O recorte abrangeu apenas a região da face e, o restante da imagem, foi descartada (Figura 5.7)

Para o desenvolvimento, foi utilizada a biblioteca *Dlib*, apresentada por King (2009), que é um conjunto de aplicativos e, bibliotecas de código aberto escrito em C++. O *Dlib* oferece uma ampla gama de funcionalidades em vários setores de aprendizado de máquina, incluindo classificação e regressão, algoritmos numéricos, como solucionadores de programas quadráticos, uma variedade de ferramentas de processamento de imagem e diversas funcionalidades de rede.

O processo teve início com a detecção da face na imagem e, caso não fosse encontrada,

a imagem foi descartada. Após isso, foram encontrados os marcadores que delimitaram o contorno da face, utilizado como referência para o recorte das imagens, mantendo na foto toda a face e descartando o restante. Além disso, as imagens recortadas foram convertidas para tons de cinza.

Figura 5.7 – Pré-processamento da base de dado de imagens.



Fonte: elaborada pelo autor.

5.2.1.2 Extração dos marcadores faciais

Para o uso da base de dados de imagens no treinamento dos classificadores, foi necessária a extração dos marcadores faciais utilizados como referência para a extração das características aproveitadas no treinamento dos modelos. Eles são pontos virtuais identificados no rosto que delimitam a boca, os olhos, as sobrancelhas e o contorno da face.

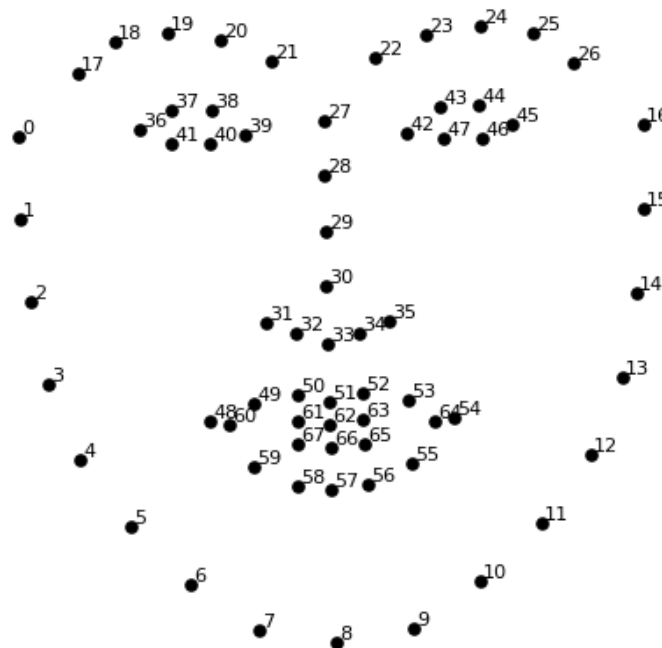
Wu e Ji (2018) citam vários tipos de métodos para a extração de marcadores faciais. Para essa pesquisa, foi utilizado um modelo de extração de marcadores faciais, que se encontra na biblioteca *Dlib* (KING, 2009) na sua versão em Python, que é usado para estimar a localização de 68 pontos da face em coordenadas cartesianas (x, y). Esses pontos são identificados a partir de um modelo pré-treinado que usa a base de dados iBUG 300-W (Sagonas et al., 2013) como referência.

Foram selecionados, do total de 68 marcadores faciais, apenas 42 que correspondem aos pontos internos do rosto que abrangem as áreas da boca, dos olhos e das sobrancelhas. Os 26 pontos desprezados correspondem ao contorno da face e a região do nariz, que não são relevantes, uma vez que as características utilizadas nos classificadores foram extraídas apenas das regiões internas mencionadas.

Para a extração dos marcadores faciais, com o uso da biblioteca *Dlib*, o processo inicia com a detecção das faces na imagem e, depois, com a extração dos marcadores, retornando um vetor de coordenadas cartesianas, identificando a posição de cada ponto na face no total de 68 (Figura 5.8).

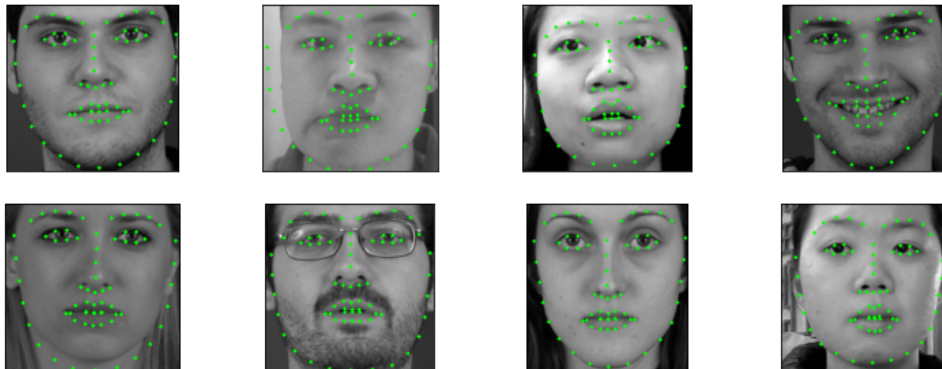
Foram extraídos marcadores faciais de todas as fotos pré-processadas na etapa anterior. Na Figura 5.9 pode ser visto um exemplo.

Figura 5.8 – Exemplo dos pontos detectados pela biblioteca *Dlib* na extração dos marcadores faciais.



Fonte: elaborada pelo autor.

Figura 5.9 – Exemplo da extração dos marcadores faciais nas imagens das bases de dados pré-processadas.



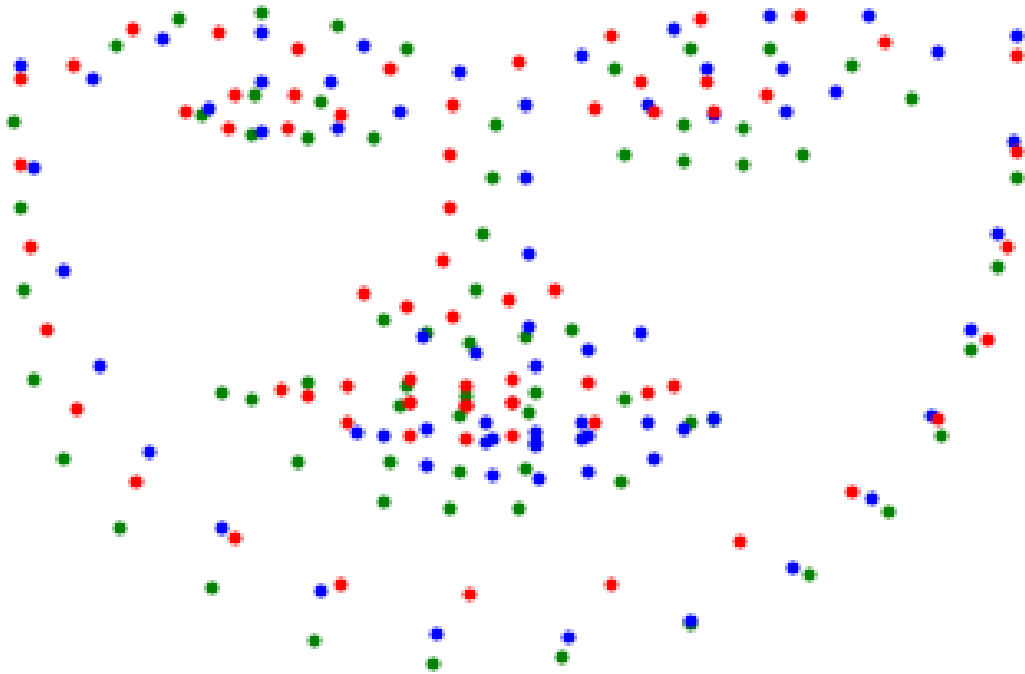
Fonte: elaborada pelo autor.

5.2.1.3 Normalização dos dados

A normalização geométrica consiste na eliminação da variação da posição, rotação e escala entre as imagens das faces. Nessa etapa, fez-se necessária, pelo fato das imagens serem provenientes de pessoas diferentes, bem como, de bases diferentes. Antes da geração do conjunto de dados de treinamento, foi preciso o ajuste dos marcadores faciais extraídos das imagens, dado à sua rotação e translação, pois vinham de bases de dados diferentes. Um exemplo dessas diferenças pode ser visto na Figura 5.10, onde se percebe que existem

diferenças de alinhamento entre alguns marcadores, principalmente, na região dos olhos. Para isso, utilizou-se o algoritmo de Procrustes (Anexo A.1).

Figura 5.10 – Exemplo de desalinhamento das faces na base de imagens pré processadas.



Fonte: elaborada pelo autor.

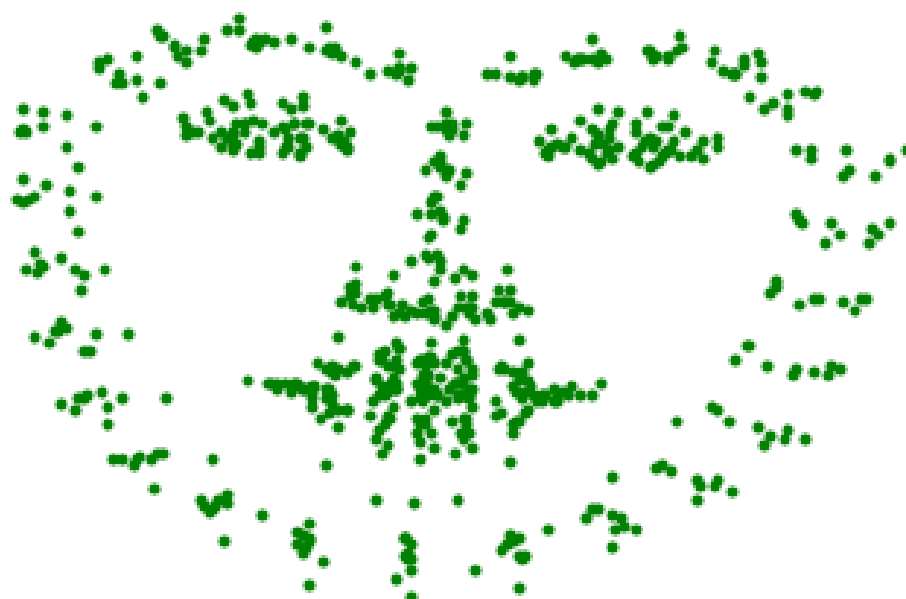
Para utilização do algoritmo de Procrustes, segundo Panceri (2017), é necessária a escolha de uma referência, para que o algoritmo faça os ajustes de translação, rotação e escala. Para isso, foi identificado, de maneira aleatória, um conjunto de marcadores faciais extraído de uma foto (Figura 5.11). Esse conjunto inicial foi transladado para a origem (ponto 0, 0), usando-se como referência, o ponto 33 dos marcadores faciais, que corresponde a ponta do nariz, conforme mostra a Figura 5.8. A seguir, o algoritmo de Procrustes ajustou o alinhamento, a escala e fez a translação de todos os marcadores faciais extraídos da base de dados de imagens.

Após essa ação foi calculada uma referência média, levando-se em conta todos os marcadores faciais extraídos da base de dados de imagens que foram normalizados. Essa referência foi utilizada nas etapas de captura que necessitaram de normalização dos dados. Ao final do processo, uma base de dados de marcadores faciais foi gerada (Figura 5.12).

5.2.1.4 Definição das características

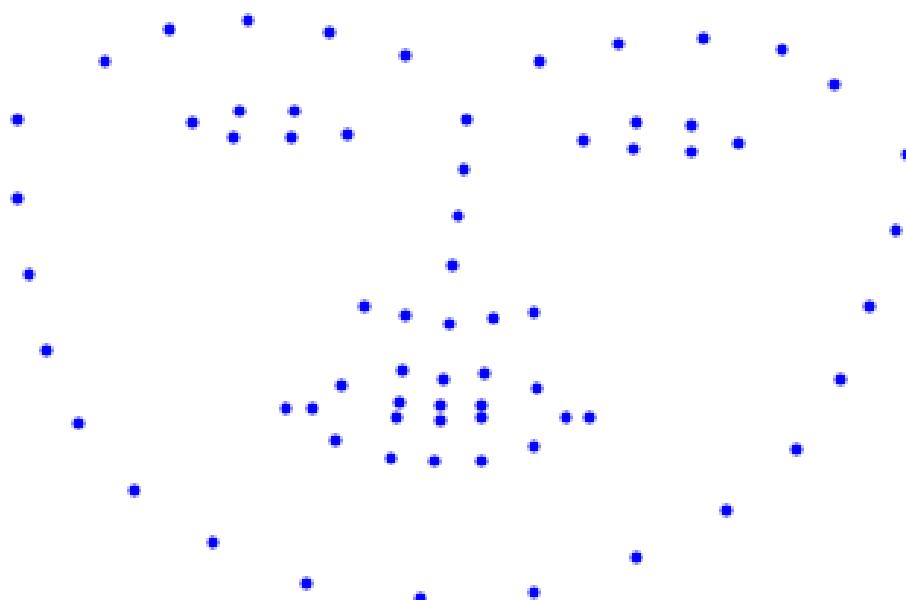
A utilização de marcadores faciais 2D, para o treinamento dos classificadores, permitiu que câmeras RGB convencionais tipo *webcams* fossem utilizadas, não tendo dependência com equipamentos específicos de alto custo.

Figura 5.11 – Translação, alinhamento e escala do conjunto de dados usando como referência uma das amostras.



Fonte: elaborada pelo autor.

Figura 5.12 – Posição média dos marcadores faciais gerados pelo processo de normalização utilizada pelo algoritmo de Procrustes.



Fonte: elaborada pelo autor.

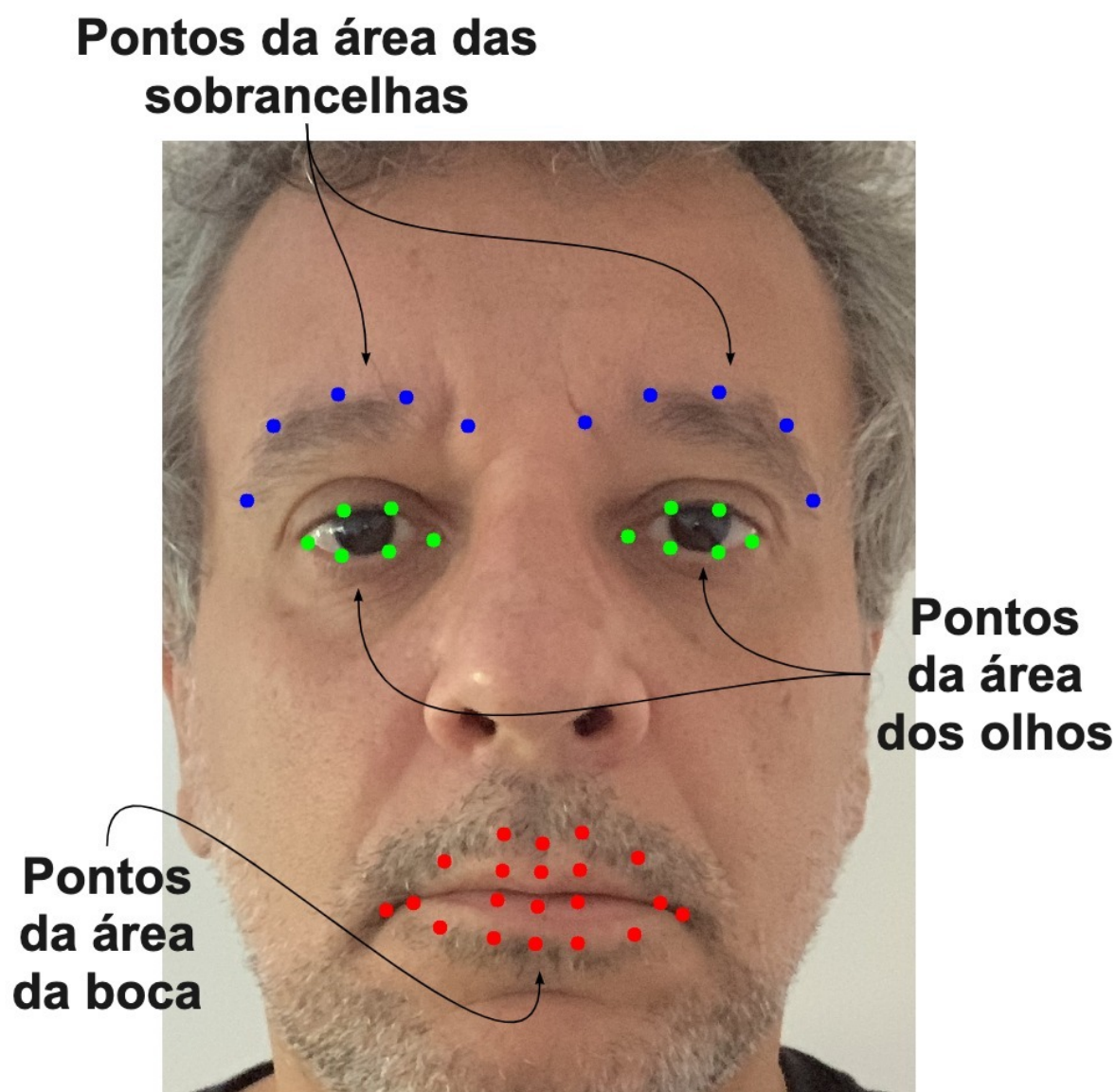
Alguns métodos desenvolvidos para o redirecionamento facial, como os de Li et al. (2013b), Weise et al. (2011b), Behrens et al. (2013) também utilizaram informações 2D e 3D diretamente como características, mas se utilizam de abordagens que exigem que a face capturada (origem) e a face virtual destino (*blendshape* final) sejam estruturalmente

semelhantes ou seja, possui a mesma aparência e a geometria.

Nos casos, onde não temos essa correspondência direta, da origem com o destino, as características para o redirecionamento facial, devem englobar medidas que sejam mais relevantes para a face, refletindo as mudanças das expressões de forma independente, não sendo viável o uso direto do posicionamento dos marcadores 2D.

Para isso, como a face possui movimentos complexos e independentes, iniciou-se a extração das características dividindo-se a face em três áreas: boca, olhos e sobrancelhas, com base na divisão proposta por Costigan, Prasad e McDonnell (2014), conforme ilustrado na Figura 5.13.

Figura 5.13 – As três regiões da face e os pontos escolhidos em cada região.



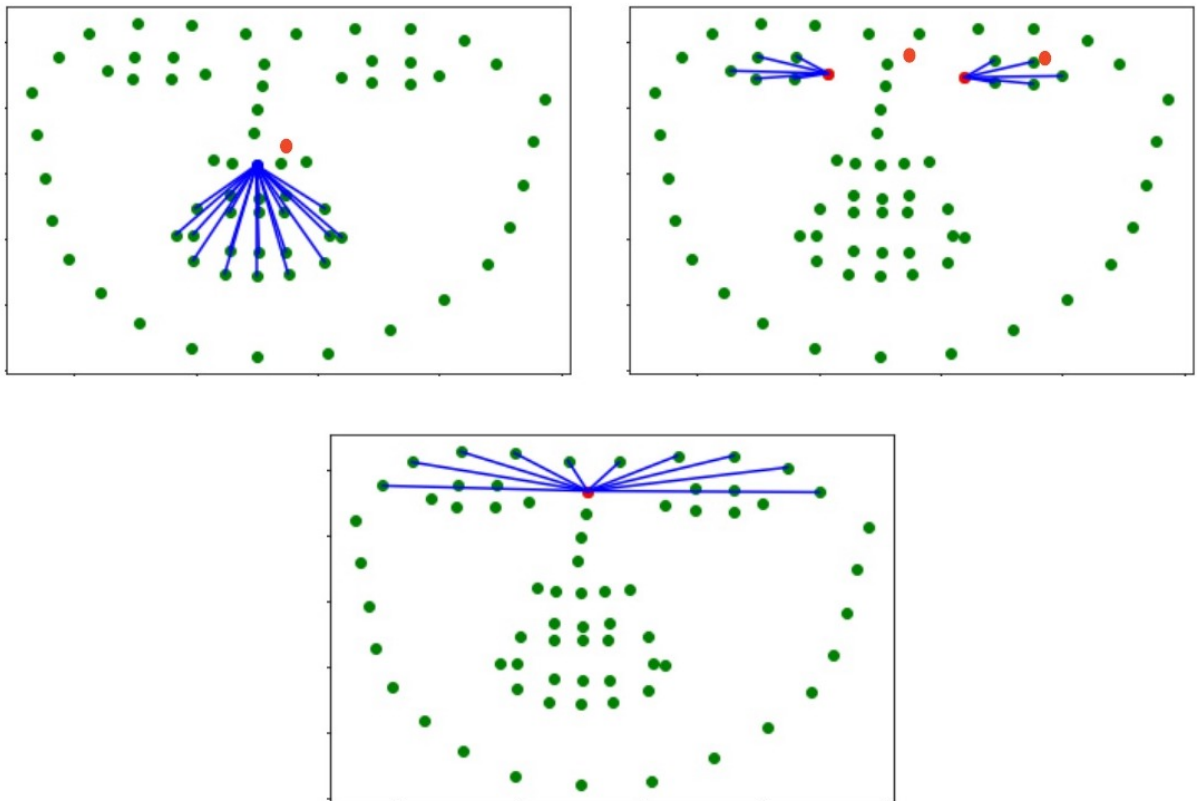
Fonte: elaborada pelo autor.

Com essa divisão, foram utilizados 10 marcadores faciais para a região das sobrancelhas, 12 marcadores para a região dos olhos e 20 marcadores para a região da boca.

A extração das características, para cada área da face, seguiu o método indicado em Ekmen e Ekenel (2018) com algumas alterações, pois o mesmo, calcula a diferença das distâncias baseado em uma face neutra e, como as bases de dados das imagens são diferentes, optou-se por outra abordagem, calculando-se a distância entre um ponto fixo e os outros pontos nas áreas propostas. Isso trouxe mais relevância às características, pois refletem com mais acurácia os movimentos internos das áreas analisadas, ao invés, da posição geométrica 2D de cada marcador facial.

Nessa abordagem, o ponto fixo usado na área da boca foi a ponta do nariz. Para a área dos olhos foi usado os pontos do canto direito e canto esquerdo. Já, na região da sobrancelha, foi utilizado um ponto no meio da testa. Todos esses pontos foram escolhidos, porque não apresentam grandes variações quando é executado algum movimento com a face (Figura 5.14).

Figura 5.14 – Características definidas através do cálculo da distância entre um ponto fixo e os demais pontos das áreas da face.



Fonte: elaborada pelo autor.

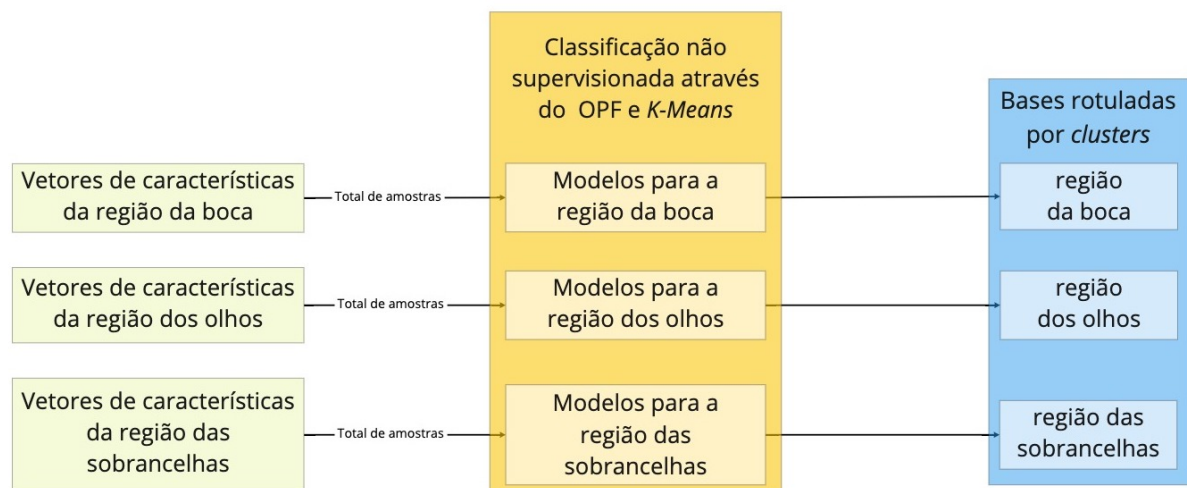
No cálculo da distância entre os pontos definidos foi utilizada a distância euclidiana

(Anexo A.2), criando assim, um vetor de características para cada área da face. Para a região da boca foram extraídas 20 características. Na dos olhos foram extraídas 12 e, para a região da sobrancelha, foram extraídas 10 características. No final, foram geradas 3 bases de dados de amostras, uma para cada região da face.

5.2.1.5 Agrupamento das amostras

A ideia central, na etapa de agrupamento das amostras, foi encontrar similaridades nas características extraídas para a criação de classes. Para isso, foi utilizado o aprendizado não supervisionado, que é o método de aprendizado de máquina que não precisa de dados pré-rotulados. No aprendizado não supervisionado existe uma técnica chamada *clustering* (ou Aglomeração) que serve, exatamente, para agrupar dados que tenham algumas características parecidas. O procedimento pode ser visto na Figura 5.15.

Figura 5.15 – Geração dos modelos não supervisionados das características das imagens. Foram gerados 3 modelos, um para cada região da face.



Fonte: elaborada pelo autor.

Foram utilizados dois tipos de classificadores não supervisionados, o *k-Means*, (MACQUEEN, 1967), que é um classificador citado em vários trabalhos como Kanungo et al. (2002), Zheng e An (2015), Huang, Wang e Shao (2008), Zhang et al. (2022) e Ahmed, Seraj e Islam (2020) e o OPF, Papa e Falcão (2010) na sua versão não supervisionada.

Fazendo-se uso, das 3 bases de dados de amostras geradas no processo de extração de características (subseção 5.2.1.4), o classificador *k-Means* gerou 3 modelos não supervisionados com 230.813 amostras e 10.000 *clusters*, sendo que, as amostras foram rotuladas de acordo com o *cluster*, a qual foram incluídas recebendo um rótulo numérico.

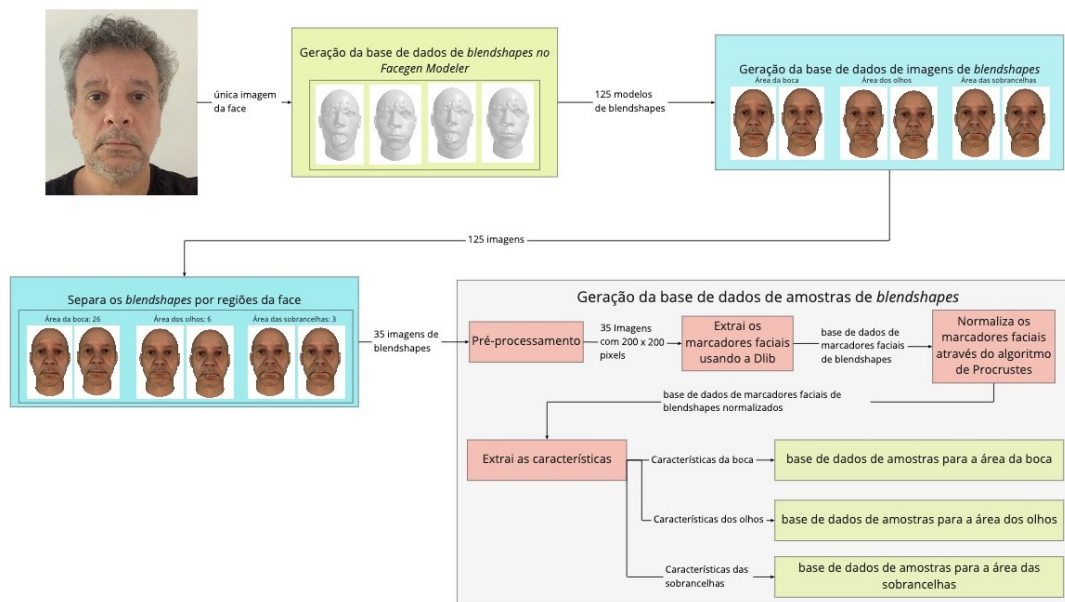
Já, o classificador OPF, também gerou os modelos com as mesmas quantidades de amostras geradas pelo *k-Means* que foram 230.813 com, aproximadamente, 10.000 *clusters*.

5.3 Criação da base de dados de *Blendshapes*

Os *blendshapes*, como apresentado na seção 2.1, são definidos como modelos faciais em que vetores representam expressões individuais e, através, de uma simples soma de vetores, é possível a criação de poses variadas. Além disso, a sua geração pode ser feita de maneira automatizada, como o que é mostrado no trabalho de Sumner e Popović (2004), onde os *blendshapes* são gerados por meio da deformação de malhas.

O redirecionamento do movimento facial, nesta tese, é baseado nos *blendshapes*, tendo sido necessária a criação de uma base de dados nesse formato, para extração das características e a geração das amostras, que foram utilizadas no treinamento dos classificadores. A Figura 5.16 ilustra as etapas do processo.

Figura 5.16 – Etapas para a geração da base de dados características de *blendshapes*.



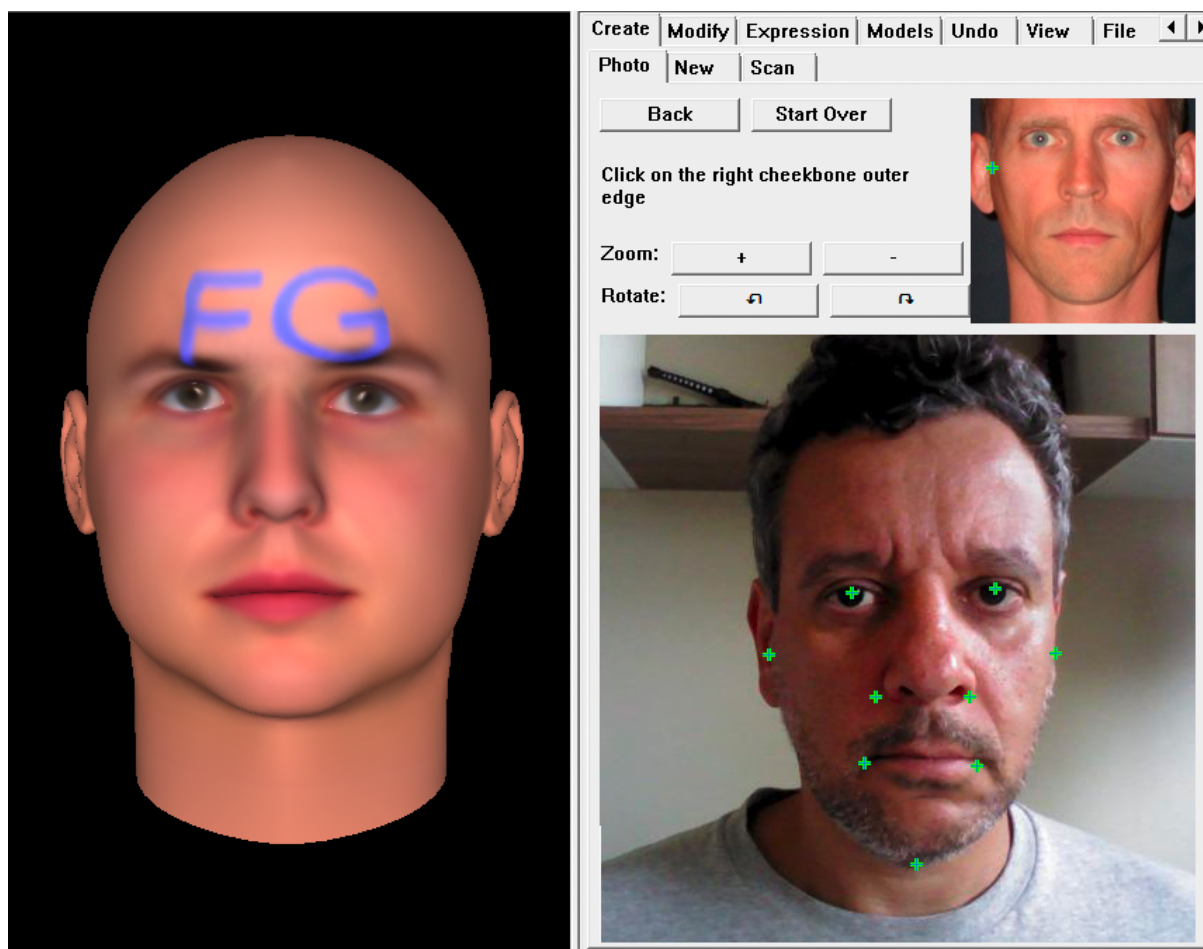
Fonte: elaborada pelo autor.

5.3.1 Geração da base de dados de *blendshapes*

Através do *FaceGen Modeller 3.3* (Figura 5.17), uma ferramenta comercial projetada para a criação de faces 3D de forma realista, gerou-se um conjunto de modelos em 3D customizado, por meio, de fotografias (Figura 5.17) em expressão neutra na posição frontal. Ao exportar o objeto criado, é gerado um conjunto de dados de expressões baseadas nas FACS, que seguem a classificação proposta por Ekman e Rosenberg (1997), sendo uma delas a expressão neutra. O *software* foi utilizado na sua versão gratuita.

No total foram geradas 125 expressões, sendo 71 baseadas em AUS, que é um modelo detalhado na seção 2.2, 24 expressões geradas através da combinação de AUS,

Figura 5.17 – Software FaceGen Modeller 3.3.



Fonte: elaborada pelo autor.

além de 30 expressões dos movimentos da face, como olho direito, olho esquerdo piscando, movimento da cabeça para os lados e expressões, por meio, de uso de fonemas como as letras a, u. Exemplos podem ser vistos na Figura 5.18.

Figura 5.18 – Base de dados de *blendshapes*. Algumas expressões são baseadas em AUS, enquanto outras, são expressões básicas ou combinações de AUS.



Fonte: elaborada pelo autor.

Os *blendshapes* possuem uma textura que é extraída da imagem utilizada para a criação da base de dados como pode ser visto na Figura 5.19

Figura 5.19 – Face 3D customizada gerada a partir de uma única foto.



Fonte: elaborada pelo autor.

5.3.2 Geração da base de dados de imagens de *blendshapes*

Os *blendshapes* são malhas 3D que possuem informações dos 3 eixos ordenados (X, Y, Z), mas, as características que foram usadas para a geração das amostras são os marcadores faciais, que possuem apenas informações 2D. Nessa etapa, foi realizada a extração da imagem 2D de cada personagem, gerando-se, portanto, 125 imagens. Essas imagens usam a textura da figura utilizada na geração da base de dados. Isso é necessário para a etapa de extração dos marcadores faciais, pois a região da face apresentada nessa textura é lida pela biblioteca que extrai os pontos. Caso contrário, isso não seria possível. Para a extração dessa imagem 2D foi utilizado o *MeshLab* que é um sistema de código aberto para processamento e edição de malhas 3D (CIGNONI et al., 2008). Na Figura 5.20 resultado pode ser visto.

5.3.3 Separação das imagens dos *blendshapes* por região da face

Os *blendshapes* gerados pelo *FaceGen Modeller 3.3* utilizam expressões básicas, AUS e as combinadas. Isso faz com que, em alguns casos, já exista uma mistura de movimentos entre as áreas da face. Como esses modelos de *blendshapes* servirão para gerar a face final no redirecionamento facial, os modelos que apresentam esse comportamento foram descartados, pois não seria possível utilizá-los para a soma final dos vetores, em que, essa mistura irá acontecer através da escolha dos classificadores. Com isso, foram selecionados, do total da base de imagens de *blendshapes*, 35 imagens, sendo 26 que representam apenas os movimentos da boca, 3 representam os movimentos da sobrancelha e 6, os movimentos dos olhos. Essas imagens selecionadas possuem os movimentos nas

Figura 5.20 – Imagens exportadas dos *blendshapes* usando a textura da imagem de entrada.



Fonte: elaborada pelo autor.

regiões específicas e, nas outras duas áreas da face restantes, uma representação neutra. Exemplo dessa seleção pode ser visto na Figura 5.21.

Figura 5.21 – Imagens selecionadas por região da face. (a) Imagem que corresponde a região da boca; (b) Imagem que corresponde a região dos olhos; (c) Imagem que corresponde a região das sobrancelhas.



Fonte: elaborada pelo autor.

A Tabela 2 mostra como ficou a divisão das imagens dos *blendshapes*. A separação seguiu, como referência, a nomenclatura das AUS definidas por Ekman e Rosenberg (1997) e foi utilizada para rotular as imagens dos *blendshapes*.

Tabela 2 – Separação dos *Blendshapes* por região usando a referência das AUS como rótulos.

Áreas da face	AUS
Boca	AU08_AU26, AU08, AU10, AU12, AU13, AU14_Left, AU14_Right, AU14_Dimpler, AU15, AU16, AU17, AU18, AU20, AU22, AU23, AU24, AU25, AU26, AU27, AU28, AU29, AU30_Left, AU30_Right, AU3, AU34, AU35
Olhos	AU5, AU6, AU7, AU41, AU42, AU43
Sobrancelha	AU1, AU2, AU4

Fonte: elaborada pelo autor.

5.3.4 Geração da base de dados de amostras de *blendshapes*

A geração da base de dados de amostras de *blendshapes* possui etapas semelhantes ao apresentado na subseção 5.2.1. Inicialmente, na etapa de pré-processamento, é feito um ajuste no tamanho da imagem para 200 x 200 *pixels*. Em seguida, é feita a extração dos marcadores faciais, por meio, da biblioteca *Dlib*, seguindo as mesmas definições apresentadas na subseção 5.2.1.2. Posteriormente, os marcadores faciais foram normalizados, seguindo, como referência, a expressão média extraída na etapa de normalização e, no final, é feita a extração das características seguindo os mesmos passos apresentados na subseção 5.2.1.4. São geradas 3 bases de dados de amostras do *blendshapes* separados por região da face e rotulados, seguindo a nomenclatura apresentada na Tabela 2.

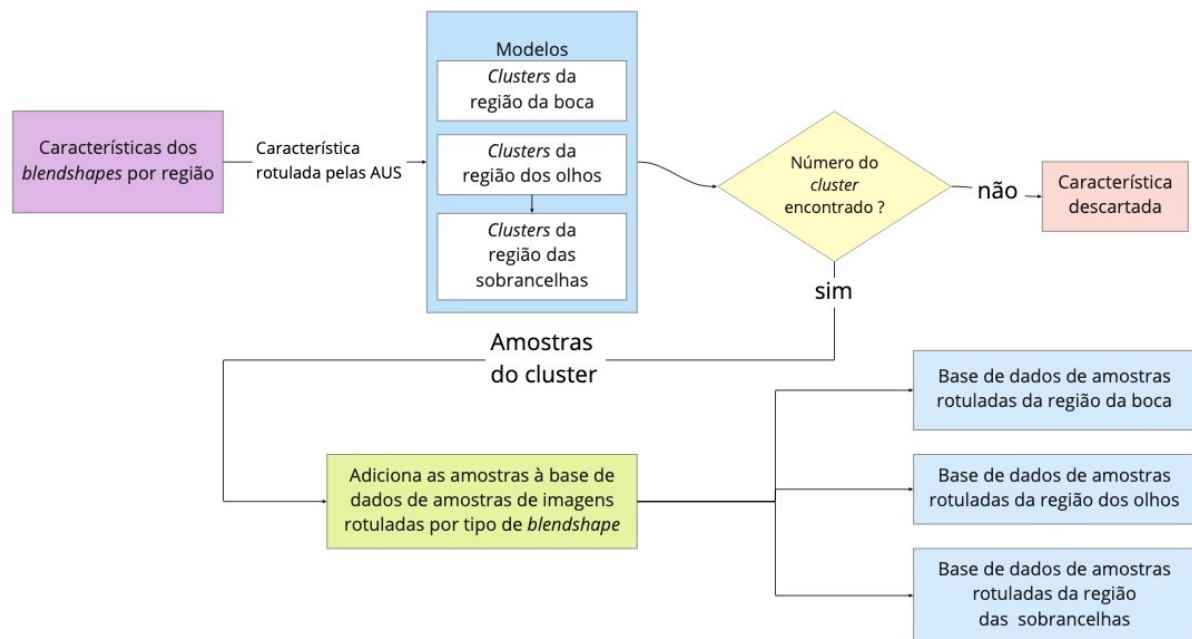
5.4 Rotulação das características das imagens

Para o treinamento supervisionado foi necessária a identificação das amostras semelhantes às características dos *blendshapes*. Para isso, valeu-se dos modelos criados na subseção 5.2.1.5 que possuíam a referência dos agrupamentos. Assim, cada *blendshape*, através das suas características, foi submetido aos classificadores que informaram qual seria o agrupamento mais próximo e quais amostras poderiam representá-lo. As amostras, depois de identificadas, foram rotuladas com a identificação do *blendshape*, criando assim, uma base de amostras de imagens rotuladas. As amostras de *blendshapes* que não tiveram um *cluster* identificado, foram descartadas. No final, foram geradas 3 novas bases de dados de amostras rotuladas para cada área da face. O processo é ilustrado pela Figura 5.22.

O processo foi feito com os dois classificadores. A base de dados rotulada pelo classificador OPF, foi a que teve um melhor desempenho em função de encontrar amostras por *cluster*.

No total foram selecionados, utilizando os modelos classificados pelo OPF, 16 *blendshapes*, sendo 8 da região da boca, 5 da região dos olhos e 3 da região das sobrancelhas. Essas amostras criaram uma base de faces rotuladas com o total de 7.145 amostras. A divisão pode ser vista na Tabela 3.

Figura 5.22 – Processo de rotulação da base de dados de amostras de imagens baseado nos rótulos dos *blendshapes*.



Fonte: elaborada pelo autor.

Tabela 3 – Quantidade de amostras rotuladas por região.

Região	Rótulos	Quantidade de amostras
Boca	AU14	347
Boca	AU16	92
Boca	AU24	1969
Boca	AU25	574
Boca	AU26	46
Boca	AU27	30
Boca	AU29	18
Boca	AU33	506
Olhos	AU05	116
Olhos	AU06	74
Olhos	AU07	420
Olhos	AU41	254
Olhos	AU42	116
Sobrancelhas	AU01	936
Sobrancelhas	AU02	1049
Sobrancelhas	AU04	598

Fonte: elaborada pelo autor.

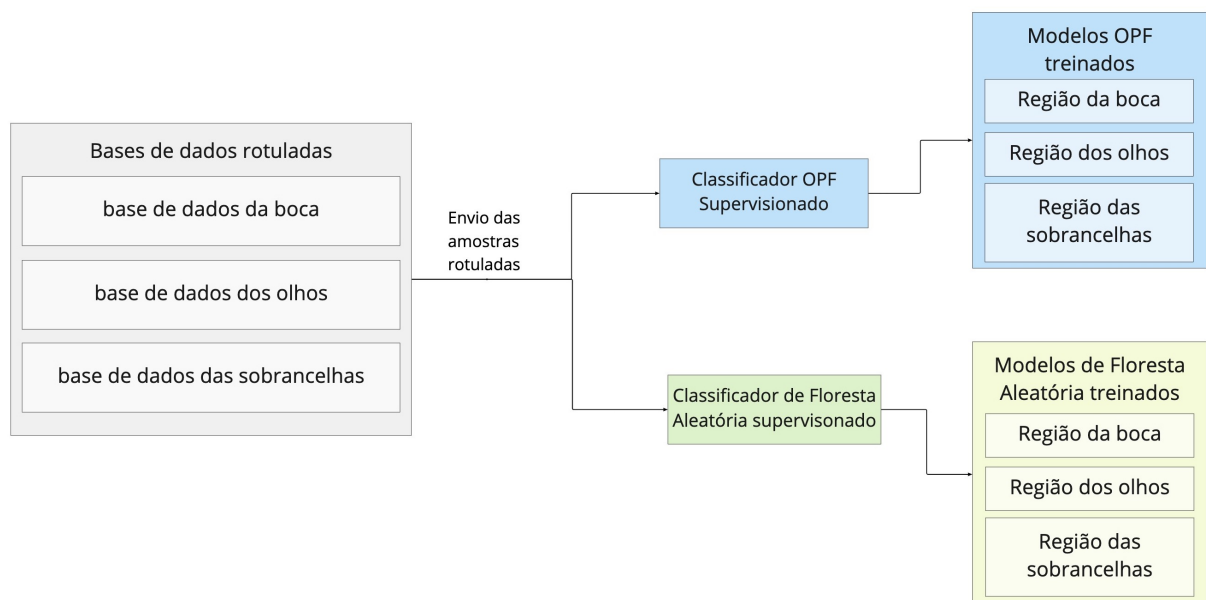
5.5 Treinamento dos modelos supervisionados

A base de dados de amostras rotuladas foi treinada usando os classificadores OPF (seção 3.2) e o classificador de Florestas Aleatórias (*Random Forest*) (subseção 3.4.1). O classificador de Floresta Aleatória foi escolhido por ser disponibilizado em pesquisas na mesma área, utilizando as mesmas características, que são os marcadores faciais, como encontradas nos trabalhos de Munasinghe (2018) e Tiwari e Patel (2020).

Conforme ilustrado na Figura 5.23, foram gerados seis modelos treinados, sendo três para cada classificador, um para cada área da face.

O processo de geração iniciou-se com o envio da base de dados rotulada de amostras de imagens aos classificadores os quais geraram os modelos supervisionados para as três áreas da face.

Figura 5.23 – Etapas para a criação da base de dados supervisionada.



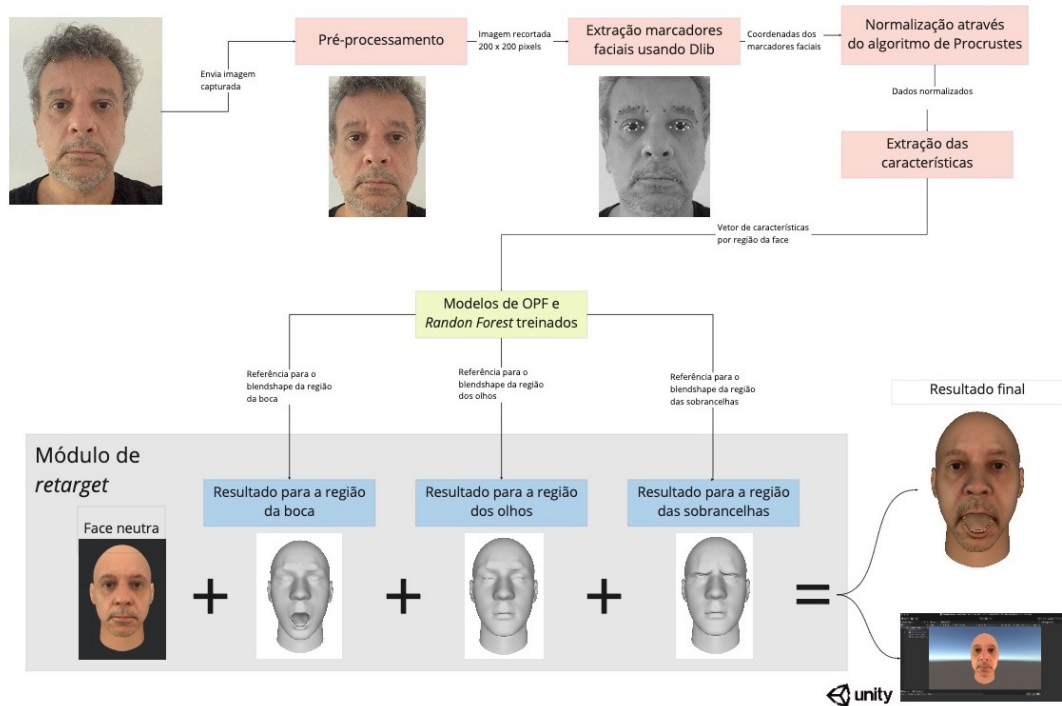
Fonte: elaborada pelo autor.

5.6 Utilizando o modelo supervisionado

Finalizado o processo da criação do modelo supervisionado, foi criado um *pipeline* para testes, que pode ser visto na Figura 5.24.

O *pipeline* inicia com a entrada de um vídeo ou uma imagem. Caso seja vídeo, serão extraídas as imagens das faces, quadro a quadro. Na etapa de pré-processamento, a imagem será redimensionada para a resolução de 200 x 200 *pixels* e sua cor convertida para escalas de cinza. Ao final, essa imagem é enviada para a etapa de extração dos marcadores faciais. Usando o mesmo processo descrito na subseção 5.2.1.2, são extraídos

Figura 5.24 – Aplicando o modelo treinado à captura em tempo real.



Fonte: elaborada pelo autor.

os marcadores faciais e suas referências são enviadas para a etapa de normalização, que segue o mesmo processo descrito na subseção 5.2.1.3, em que, os dados são ajustados, por meio, do algoritmo de Procrustes, utilizando como referência a amostra média gerada na fase de processamento das imagens da base de dados. Finalizada essa etapa, são extraídas as características dos marcadores faciais seguindo o mesmo processo da subseção 5.2.1.4.

Depois da extração das características, os vetores de cada área da face são submetidos ao classificador correspondente, o qual escolherá o *blendshape*, a partir de seu rótulo. No final, o módulo de *retarget* fará a combinação dos *blendshapes* para compor a face 3D. Essa combinação acontece com um *blendshape* em posição neutra e a soma das malhas de cada *blendshape* selecionado. Detalhes sobre o procedimento de criação da face utilizando o *blendshape* são apresentados no subseção 2.1.2.

No *pipeline* proposto foram implementados dois tipos de saídas. Uma é o *blendshape* final, obtido por meio da soma dos malhas e a outra é para o motor de jogos *Unity 3D*, onde ele envia a identificação dos *blendshapes* selecionados, que é aplicado a uma face virtual previamente carregada no cenário também virtual, fazendo com que ela se movimente de acordo com a captura.

6 Experimentos e análises dos resultados

Este capítulo apresenta a análise dos resultados obtidos a partir de testes realizados utilizando-se o sistema de animação facial proposto.

6.1 Ambiente experimental

O ambiente experimental utilizado neste trabalho é descrito a seguir.

6.1.1 Hardware

Para desenvolver os experimentos, utilizou-se um notebook com as seguintes configurações:

- *Notebook MacBook Pro* com sistema Operacional de 64 *bits*, processador com base em x64, Mac os BigSur;
- Placa gráfica AMD Radeon Pro 5300M 4 GB;
- Processador: Intel Core(TM) i7 CPU (2.9GHz);
- Memória RAM: 16 GB; e
- Disco rígido SSD: 500 GB;

Recorreu-se também a um servidor linux com as seguintes configurações:

- Sistema operacional linux Ubuntu 16.04.7;
- Memória RAM: 32 GB;
- Processador Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60GHz; e
- Disco rígido SSD: 170 GB;

6.1.2 Software

Nesta tese também foram utilizados os seguintes componentes de software:

- *FaceGen Modeler*, versão 2022;
- Biblioteca *Dlib*;

- Biblioteca *Opfython*, versão 1.0.12;
- Linguagem de desenvolvimento Python, versão 3.9; Biblioteca *Scikit-Learn*, versão 1.1.2; Biblioteca *OpenCv*, versão 4.0.6;
- *MeshLab*, 64 bits, versão 2021.10;
- *MeshMixer*, 64 bits, versão 3.5.474;
- Plataforma de desenvolvimento *PyCharm*, versão CE 2021.2; e
- *Unity Pro*, versão 2022.1.

6.2 Métrica de validação

No redirecionamento dos movimentos da face do ator para uma face virtual, espera-se uma alta acurácia. A determinação desta acurácia se dará pelo cálculo da diferença entre, o posicionamento dos marcadores faciais capturados no módulo que captura a face do ator (considerados como referência ou *ground truth*) e o posicionamento dos marcadores obtidos, após a aplicação dos *blendshapes* gerados através dos classificadores, no módulo de processamento na fase *online*.

O cálculo se deu por meio do erro médio normalizado (*NME* - *Normalised mean error*), que é uma métrica de validação de detecção de pontos de referências faciais, utilizados em trabalhos como Lai et al. (2019), Huang et al. (2021), Chen et al. (2020) e Pourramezan Fard e Mahoor (2022). Ele calcula a distância, entre os marcadores faciais do *ground truth*, com os marcadores faciais extraídos da face gerada pelo *pipeline*.

O NME é definido como:

$$NME(x, y) = \frac{1}{Nx} \sum_{i=1}^{Nx} \frac{\|xi - yi\|^2}{d} \quad (6.1)$$

onde x e y são, respectivamente, as coordenadas geradas e as de referência, o *ground truth*. Nx é o numero total de marcadores e d é a distância de referência para normalizar os erros absolutos. Normalmente, d pode ser a distância entre os cantos internos dos olhos ou a distância entre os centros da pupila. Para este trabalho, foram utilizados os marcadores faciais que correspondem à distância entre os cantos internos dos olhos, para a normalização.

6.3 Análise dos classificadores de Floresta Aleatória e OPF

Neste trabalho foram utilizados os classificadores OPF na sua versão em *Python* (ROSA; PAPA, 2021) e Florestas Aleatórias, os quais geraram, no total, seis modelos

treinados de forma supervisionada. Para a validação dos mesmos, foram utilizadas as métricas descritas na seção 3.5, a saber: a precisão, a especificidade (ou Recall) e o F1-score.

Para analisar o desempenho dos classificadores, as amostras da base de dados de imagens foram separadas em dois grupos, sendo o primeiro com 80% do total da base usado para treino e, o segundo com 20% da base, usado para testes.

Foram geradas matrizes de confusão, para cada modelo, mostrando a frequência de classificação para cada classe na sua diagonal principal. A matriz usa tons e cores destacando a quantidade de amostras classificadas como correta.

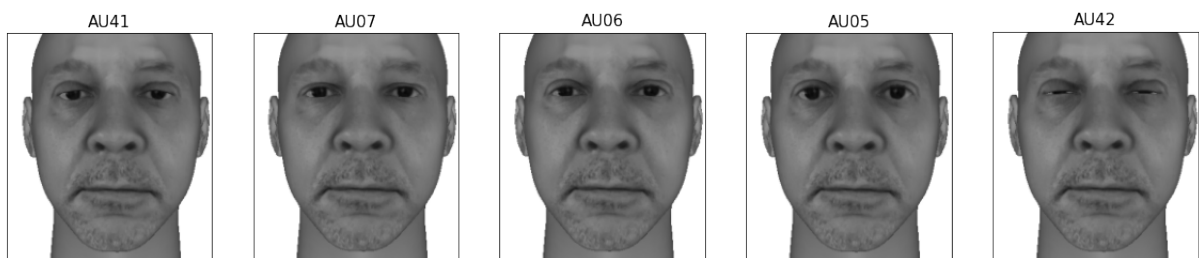
Os *blendshapes*, classificados pelos modelos para a região da boca, podem ser vistos na Figura 6.1; para a região dos olhos, são apresentados na Figura 6.2; e, para a região das sobrancelhas, encontram-se ilustrados pela Figura 6.3.

Figura 6.1 – *Blendshapes* classificados para região da boca.



Fonte: elaborada pelo autor.

Figura 6.2 – *Blendshapes* classificados para região dos olhos.



Fonte: elaborada pelo autor.

Figura 6.3 – *Blendshapes* classificados para região das sobrancelhas.

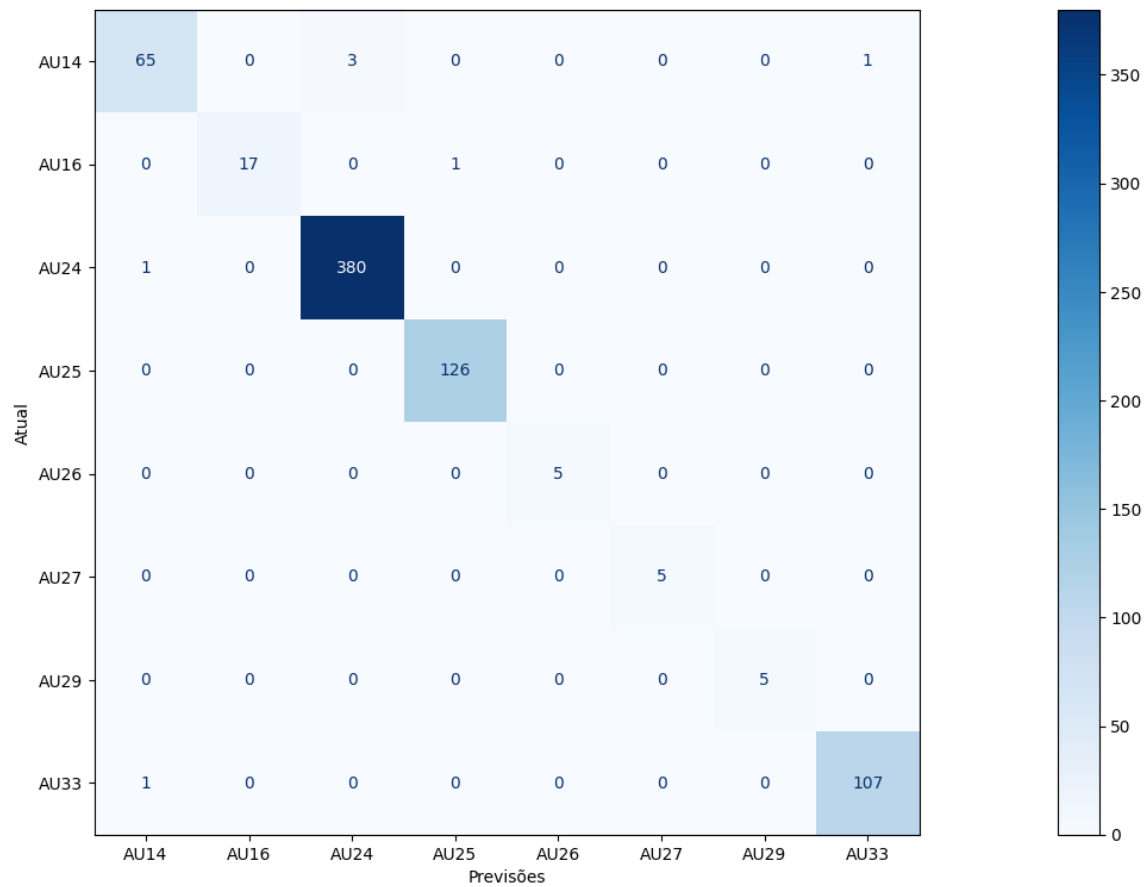


Fonte: elaborada pelo autor.

6.3.1 Análise do classificador de Florestas Aleatórias supervisionado

O desempenho do classificador de Florestas Aleatórias para a região da boca, pode ser observado na matriz de confusão ilustrada pela Figura 6.4.

Figura 6.4 – Matriz de confusão do modelo da região da boca do classificador de *Floresta Aleatória*.



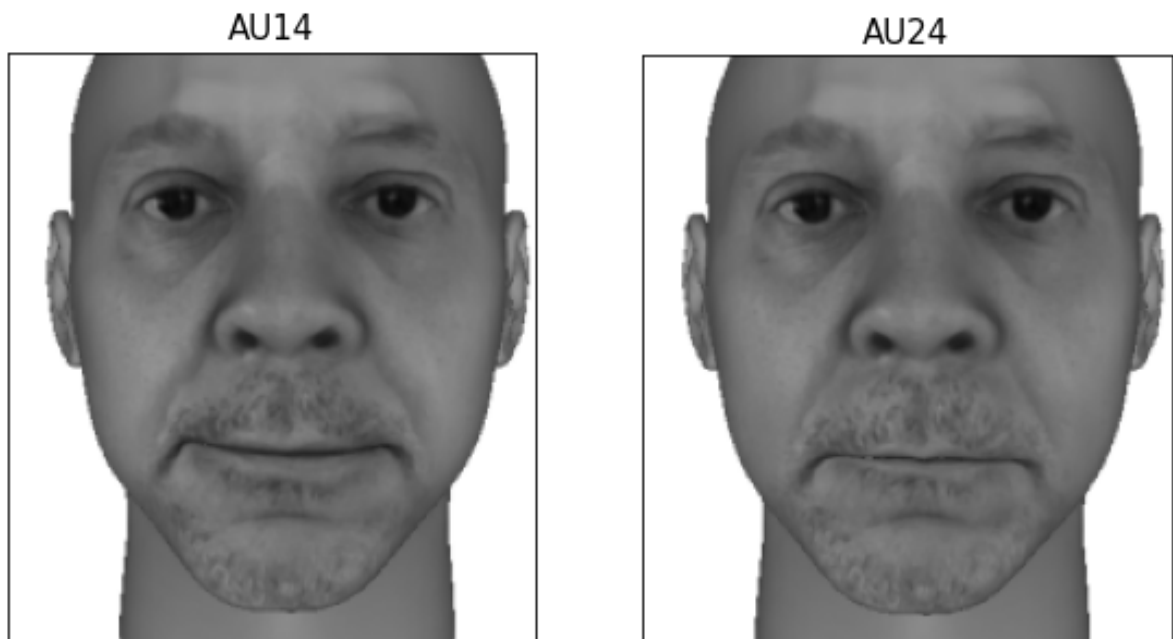
Fonte: elaborada pelo autor.

A matriz de confusão, para a região da boca, mostra que o modelo classificou 8 classes chegando a uma acurácia de 99%.

É possível observar que, o classificador para o rótulo AU14 conseguiu classificar, corretamente, 65 amostras, mas, 3 foram classificadas, erroneamente, como pertencendo à classe AU24. Isso pode ter ocorrido, pois ambas as amostras, possuem características próximas, como pode ser visto na Figura 6.5, indicando que as características utilizadas para o treinamento do classificador não foram suficientes, necessitando de mais informações e, não somente, as características 2D extraídas dos marcadores faciais.

A classe AU24, também teve uma amostra prevista pelo classificador de maneira errada, como sendo da classe AU14, reforçando a questão da proximidade entre as características.

Figura 6.5 – Proximidade entre características das classes AU14 e AU24.

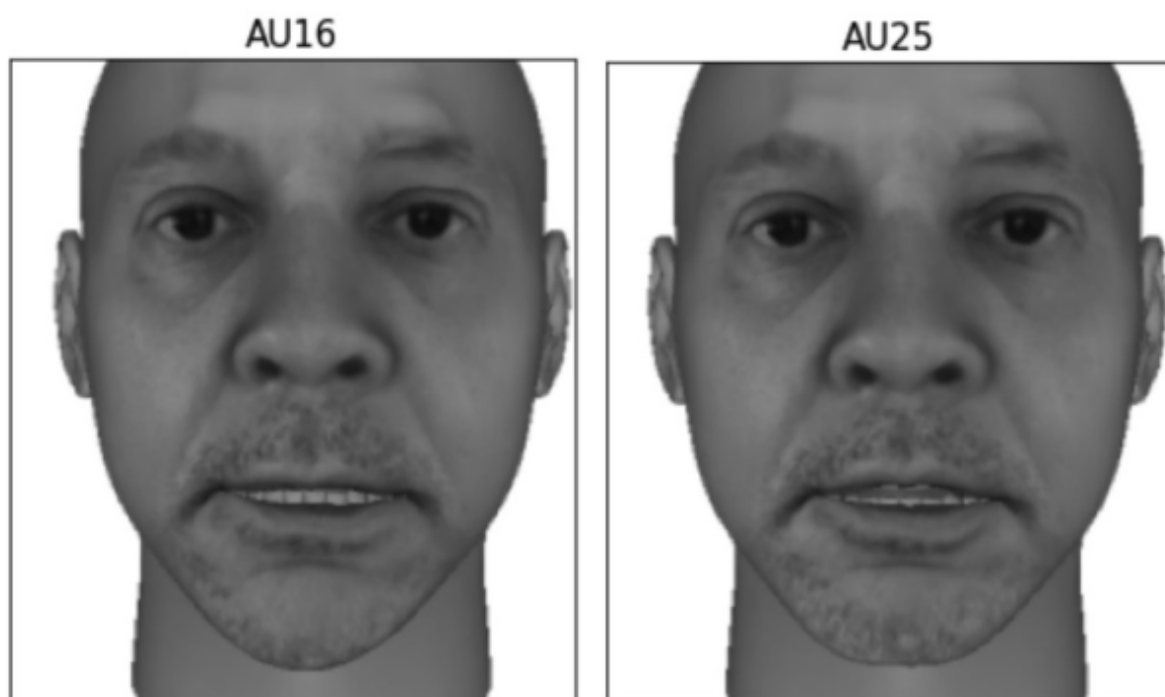


Fonte: elaborada pelo autor.

O mesmo ocorre com a classe AU16, onde o classificador previu uma classe como AU25 de maneira errada. Percebe-se na Figura 6.6, que a região da boca, é muito semelhante para essas duas classes.

Os valores das três métricas, para cada uma das 8 classes de blendshapes para a região da boca, são apresentados na Tabela 4. O *recall*, que é a quantidade de observações classificadas como positiva, que realmente são positivas, apresentou um bom resultado para todas as classes, ou seja, valores entre 94% e 100%.

Figura 6.6 – Semelhanças entre as classes AU16 e AU25.



Fonte: elaborada pelo autor.

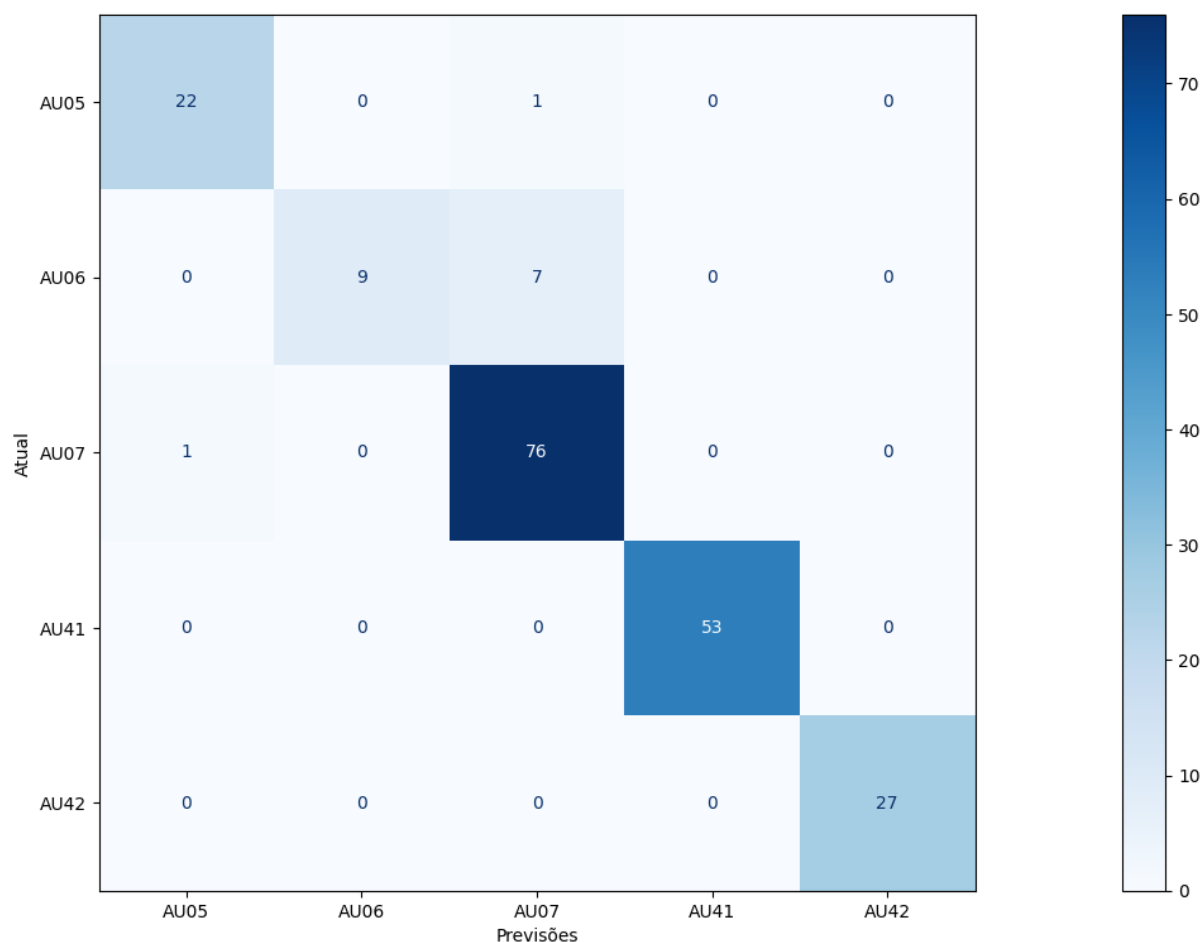
Tabela 4 – Métricas do modelo para região da boca do classificador de Floresta Aleatória.

Classes	Precisão	Recall	F1-Score
AU14	0,97	0,94	0,96
AU16	1,00	0,94	0,97
AU24	0,99	1,00	0,99
AU25	0,99	1,00	1,00
AU26	1,00	1,00	1,00
AU27	1,00	1,00	1,00
AU29	1,00	1,00	1,00
AU33	0,99	0,99	0,99

Fonte: elaborada pelo autor.

A matriz de confusão para a região dos olhos, é mostrada na Figura 6.7 e indica que o modelo classificou 5 classes com uma acurácia de 95%, sendo que, a classe AU06, não conseguiu ser predita corretamente.

Figura 6.7 – Matriz de confusão do modelo da região dos olhos do classificador de Floresta Aleatória.



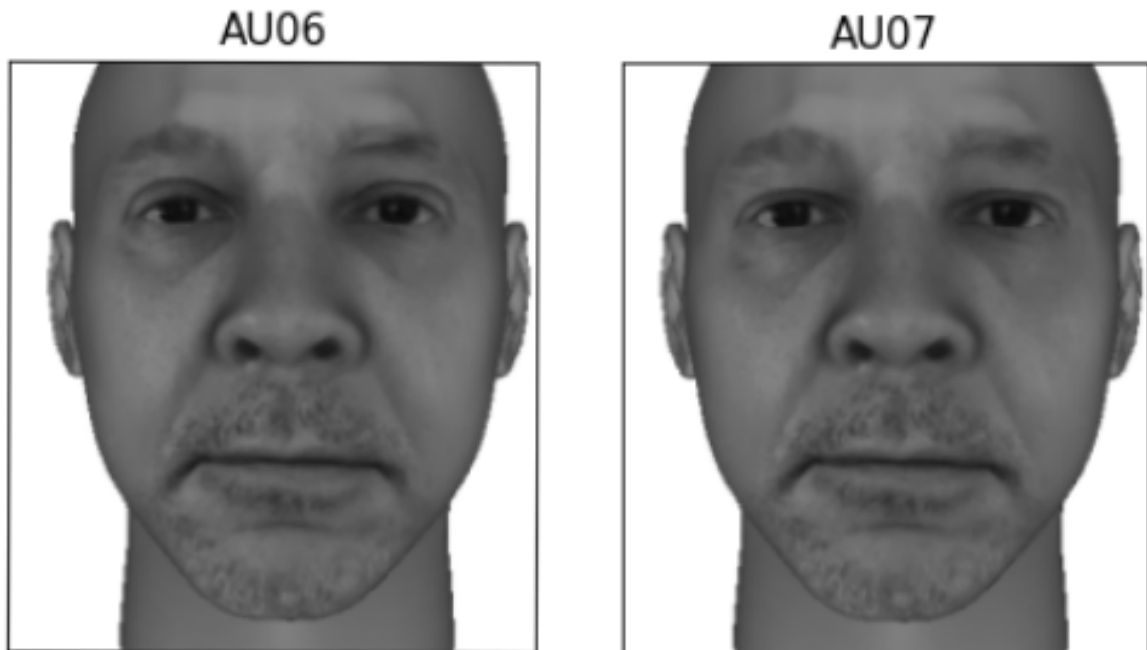
Fonte: elaborada pelo autor.

Das 16 amostras da classe AU06 submetidas ao classificador, 7 foram classificadas como sendo da classe AU07. Este erro de classificação provavelmente aconteceu devido às similaridades entre as características destas duas classes, conforme exemplificado na Figura 6.8.

A classe AU07, no entanto, apesar de ter uma semelhança com a AU06, foi classificada corretamente com apenas uma predição errada. Isso aconteceu, porque ela é diferente da classe AU06 e possui mais amostras e, possivelmente, o treinamento do classificador foi melhor.

A Tabela 5 apresenta os valores das métricas para este modelo. Percebe-se que a métrica de *recall* para a classe AU06 apresentou o menor valor (56%) se comparado às demais métricas, refletindo corretamente o que foi apresentado na matriz de confusão.

Figura 6.8 – Semelhanças entre as classes AU06 e AU07.



Fonte: elaborada pelo autor.

Tabela 5 – Métricas do modelo da região da olhos do classificador de Floresta Aleatória.

Classes	Precisão	Recall	F1-score
AU05	0,96	0,96	0,96
AU06	1,00	0,56	0,72
AU07	0,90	0,99	0,94
AU41	1,00	1,00	1,00
AU42	1,00	1,00	1,00

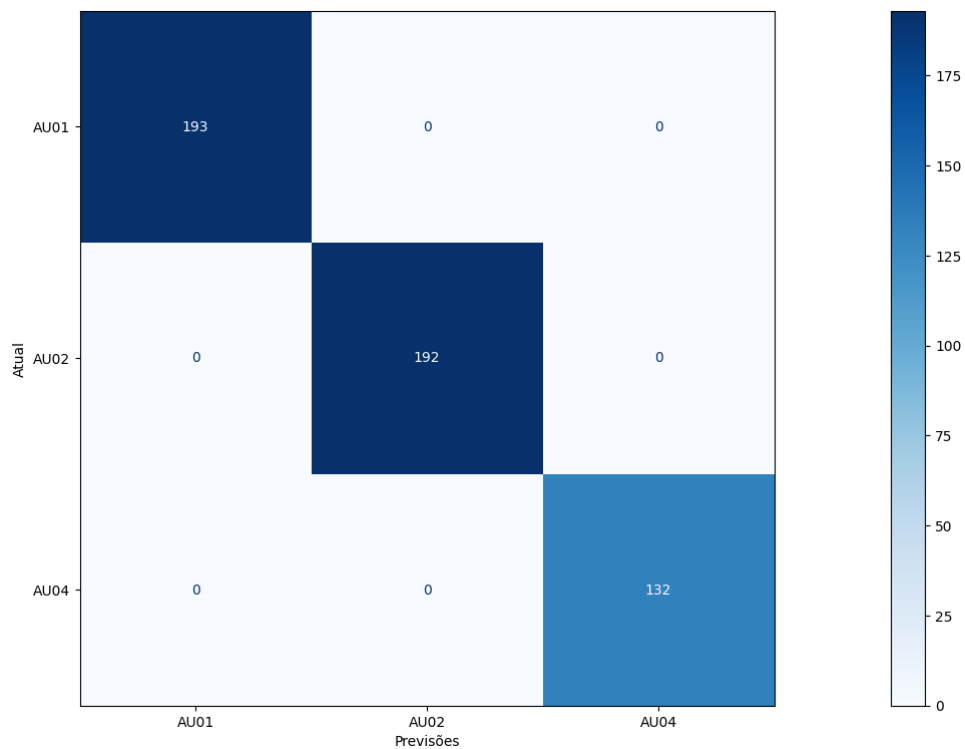
Fonte: elaborada pelo autor.

Para a região das sobrancelhas, a matriz de confusão correspondente (Figura 6.9) indica que todas as amostras foram classificadas corretamente, com relação às 3 classes desta área, obtendo uma acurácia de 100%.

Na Tabela 6 são vistas as métricas que refletem o que foi mostrado na matriz de confusão, onde todas elas apresentam um valor ótimo.

Isso indica que, o classificador conseguiu identificar corretamente porque, provavelmente, as amostras dessa área não possuem características muito semelhantes, sendo ambas, diferentes umas das outras como pode ser visto na Figura 6.3.

Figura 6.9 – Matriz de confusão do modelo da região das sobrancelhas do classificador de Floresta Aleatória.



Fonte: elaborada pelo autor.

Tabela 6 – Métricas do modelo da região das sobrancelhas do classificador de Floresta Aleatória.

Classes	Precisão	Recall	F1-score
AU01	1,00	1,00	1,00
AU02	1,00	1,00	1,00
AU04	1,00	1,00	1,00

Fonte: elaborada pelo autor.

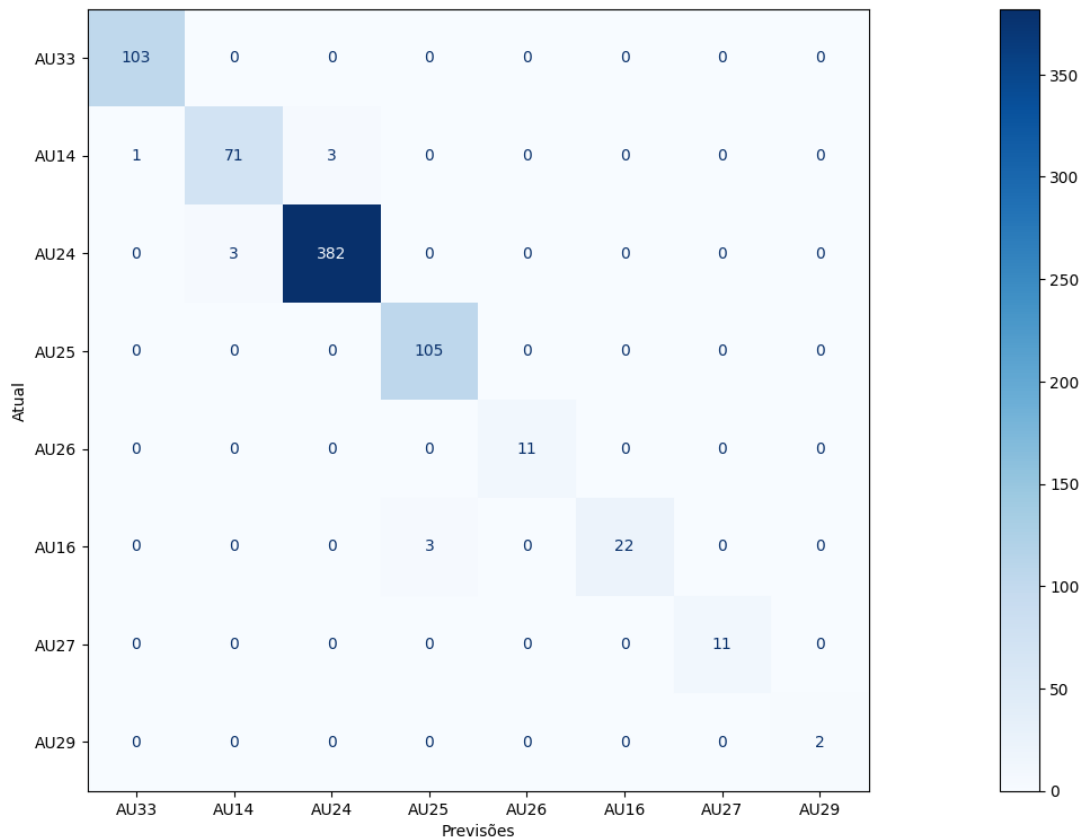
6.3.2 Análise do classificador *OPF* supervisionado

Os resultados do classificador OPF, para a região da boca, podem ser vistos na matriz de confusão mostrada na Figura 6.10.

Para essa região, o modelo teve uma acurácia de 99%, classificando 8 classes. É possível perceber que o OPF teve um bom resultado, mesmo em casos onde algumas classes foram identificadas de maneira errada, como nas classes AU14 e AU24.

Observa-se que nas classes onde o algoritmo de Floresta Aleatória teve maior

Figura 6.10 – Matriz de confusão do modelo da região da boca do classificador OPF.



Fonte: elaborada pelo autor.

difficuldade de classificação (classes AU14, AU16, AU24 e AU25), o classificador OPF obteve um melhor desempenho.

Em função do bom desempenho do classificador, pode-se ver na Tabela 7 que, a maioria das métricas extraídas, trazem um bom resultado, exceto, a métrica de *recall* para a classe AU16, que ficou abaixo de 90%, porque teve 3 amostras classificadas de maneira incorreta. Isso ocorreu também na classe AU24, mas, como a quantidade de amostras foi maior, não teve tanto impacto no resultado.

A matriz de confusão que mostra os resultados da região dos olhos, apresentada na Figura 6.11, indica que foram classificadas 5 classes com alguns erros, como na classificação das amostras da classe AU06. O classificador OPF identificou de forma errada as amostras como sendo da classe AU07. Provavelmente, como observado anteriormente, isso ocorreu porque as características da região dos olhos são bem próximas (Figura 6.8).

As métricas para a região dos olhos podem ser vistas na Tabela 8. Elas mostram que o classificador OPF teve um bom desempenho, especialmente se observarmos a métrica *recall*. Portanto, comparando-se as métricas obtidas para os dois classificadores, o algoritmo OPF conseguiu um melhor desempenho.

Tabela 7 – Métricas da região da boca do classificador *OPF*.

Classes	Precisão	Recall	F1-Score
AU33	0,99	1,00	1,00
AU14	0,96	0,95	0,95
AU24	0,99	0,99	0,99
AU25	0,99	1,00	0,99
AU26	1,00	1,00	1,00
AU16	1,00	0,88	0,94
AU27	1,00	1,00	1,00
AU29	1,00	1,00	1,00

Fonte: elaborada pelo autor.

Tabela 8 – Métricas da região da olhos do classificador *OPF*.

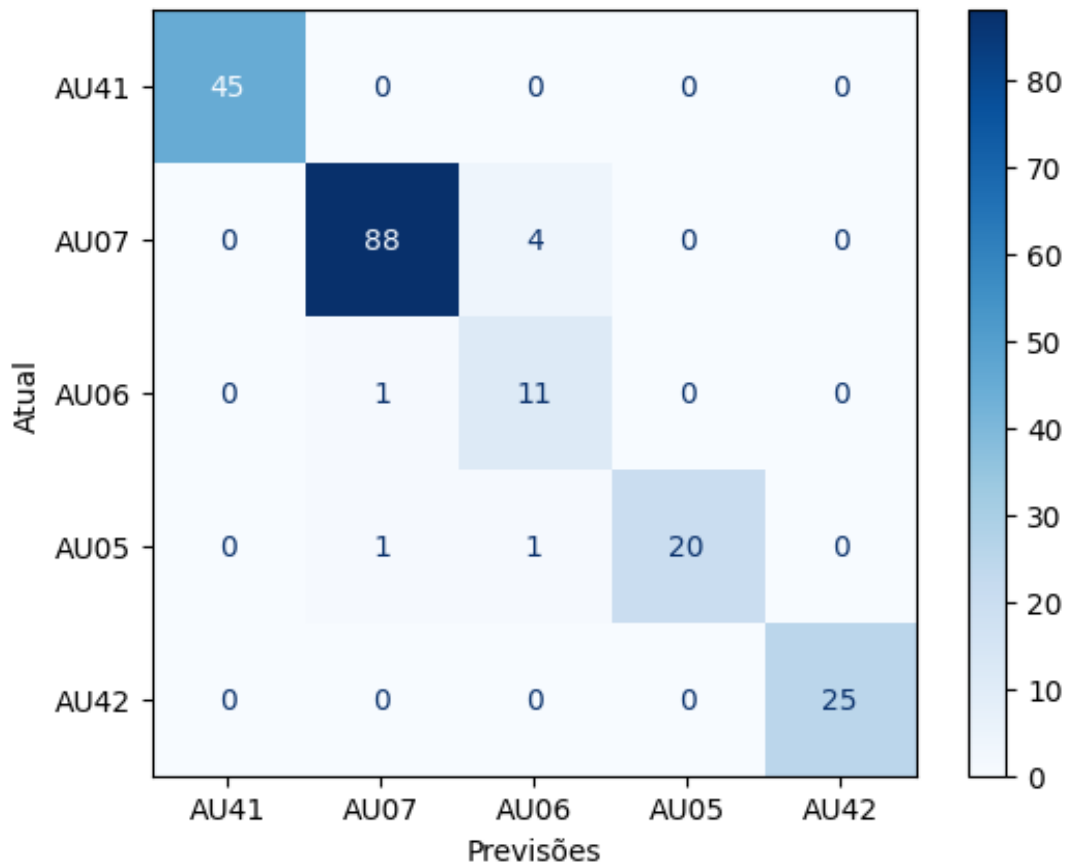
Classes	Precisão	Recall	F1-score
AU41	1,00	1,00	1,00
AU07	0,98	0,96	0,97
AU06	0,69	0,92	0,79
AU05	1,00	0,91	0,95
AU42	1,00	1,00	1,00

Fonte: elaborada pelo autor.

A Figura 6.12 exibe a matriz de confusão para a região das sobrancelhas. Analisando-se a matriz, verifica-se que todas as amostras foram classificadas corretamente.

A Tabela 9 mostra os valores de cada métrica. Como todas as amostras foram identificadas corretamente, as métricas trazem um cenário onde todas estão muito boas. Isso pode indicar que, nas amostras submetidas à análise, não existe uma grande variedade de tipos, podendo ser um problema no treinamento do modelo pois isso pode indicar que o modelo se adaptou muito as amostras. Esse comportamento também pode ser observado com o classificador de Florestas Aleatórias. Para melhorar este treinamento, uma base de amostras mais heterogênea poderia ajudar no treinamento e deixaria o classificador com uma maior capacidade de generalização (GHASEMIAN; HOSSEINMARDI; CLAUSET,

Figura 6.11 – Matriz de confusão do modelo da região dos olhos do classificador OPF.



Fonte: elaborada pelo autor.

2019).

Tabela 9 – Métricas da região das sobrancelhas do classificador *OPF*.

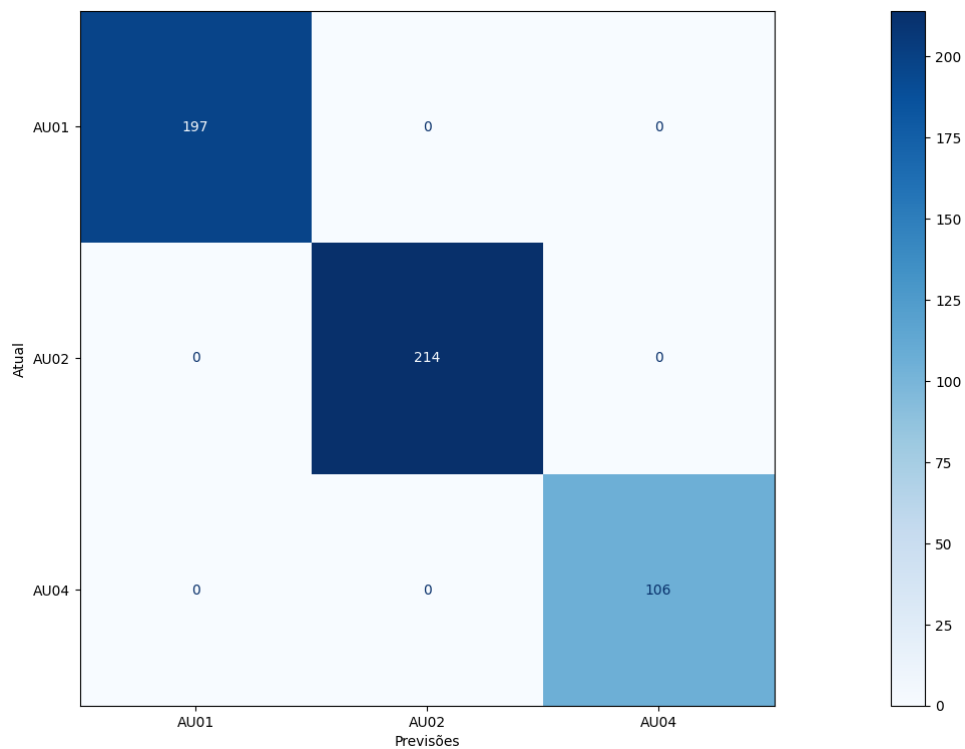
Classes	Precisão	Recall	F1-score
AU01	1,00	1,00	1,00
AU02	1,00	1,00	1,00
AU04	1,00	1,00	1,00

Fonte: elaborada pelo autor.

6.3.3 Comparação entre os classificadores de Floresta Aleatória e OPF supervisionados

Diante dos cenários apresentados, entre os modelos gerados pelos classificadores, pode-se observar que ambos obtiveram um desempenho próximo, mas, em casos onde as

Figura 6.12 – Matriz de confusão do modelo da região das sobrancelhas do classificador OPF.



Fonte: elaborada pelo autor.

classes tinham características muito semelhantes, como é o caso da região da boca e dos olhos, o classificador OPF conseguiu um melhor resultado. Na tabela Tabela 12 pode ser vista a comparação da métrica de *recall*, que representa a capacidade que o modelo teve em prever a classe correta.

Tabela 10 – Comparação da métrica de *recall* entre os classificadores para a região da boca.

Classes	<i>Recall</i> Floresta Aleatória	<i>Recall</i> OPF
AU14	0,90	0,95
AU16	0,69	0,88
AU24	0,99	0,99
AU25	1,00	1,00
AU26	1,00	1,00
AU27	1,00	1,00
AU29	1,00	1,00
AU33	1,00	1,00

Fonte: elaborada pelo autor.

Tabela 11 – Comparação da métrica de *recall* entre os classificadores para a região dos olhos.

Classes	<i>Recall</i> Floresta Aleatória	<i>Recall</i> OPF
AU05	0,96	0,91
AU06	0,56	0,92
AU07	0,99	0,96
AU41	1,00	1,00
AU42	1,00	1,00

Fonte: elaborada pelo autor.

Tabela 12 – Comparação da métrica de *recall* entre os classificadores para a região das sobrancelhas.

Classes	<i>Recall</i> Floresta Aleatória	<i>Recall</i> OPF
AU01	1,00	1,00
AU02	1,00	1,00
AU04	1,00	1,00

Fonte: elaborada pelo autor.

Pode-se destacar como evidência, que o classificador OPF foi melhor, olhando para as métricas de *recall* da classe AU16, onde o classificador de Floresta Aleatória identificou apenas 69% das amostras de forma correta, enquanto, o OPF identificou 88% das amostras corretas. Isso ocorreu também com a classe AU06, onde o classificador de Floresta Aleatória classificou 56 % das amostras dessa classe corretamente e o OPF 92%.

6.4 Experimentos

Nos experimentos e na validação dos resultados foram utilizados o *pipeline* desenvolvido, a base de dados de *blendshapes* e a de amostras extraídas das imagens. Recorreu-se, também, aos vídeos curtos com aproximadamente 60 quadros. Cada vídeo contém cenas de um ator fazendo um tipo de expressão. Os vídeos foram extraídos da base de dados *MUG* (AIFANTI; PAPACHRISTOU; DELOPOULOS, 2010).

Todos os vídeos têm início com o ator em expressão neutra, seguindo para a expressão determinada e retornando à expressão neutra. Além dos vídeos, também foram disponibilizados modelos 3D da face, com expressões de raiva, desgosto, medo, felicidade, tristeza e surpresa. No total, foram 6 vídeos e 6 faces 3D.

Para a criação dos modelos 3D da face, que serviram de *ground truth* nos experi-

mentos, utilizou-se o mesmo *software* que gerou a base de dados de *blendshapes*, o *FaceGen Modeller 3.3*. Ele exporta as faces em modelos 3D de forma automática ou customizada, por meio da sua ferramenta de ajuste.

Esse dispositivo possui dois tipos de abordagem: a primeira permite customizar a face 3D levando-se em conta as AUS e, a segunda, as expressões como referência. O ajuste, por meio das AUS, possibilita maior liberdade, permitindo ajustes mais pontuais em áreas diferentes da face e de forma independente. Já, o ajuste através das expressões, permite mais abrangência em toda a área da face.

Para se verificar a qualidade da animação facial gerada pelo *pipeline*, procedeu-se o cálculo da NME entre as coordenadas 3D dos modelos gerados pelo *pipeline*, com as coordenadas 3D dos modelos produzidos como referência, ou seja, os *ground truths* (Figura 6.13).

Figura 6.13 – Exemplo de comparação entre a face 3D de referência (*ground truth*) e a face 3D gerada pelo *pipeline* desenvolvido.



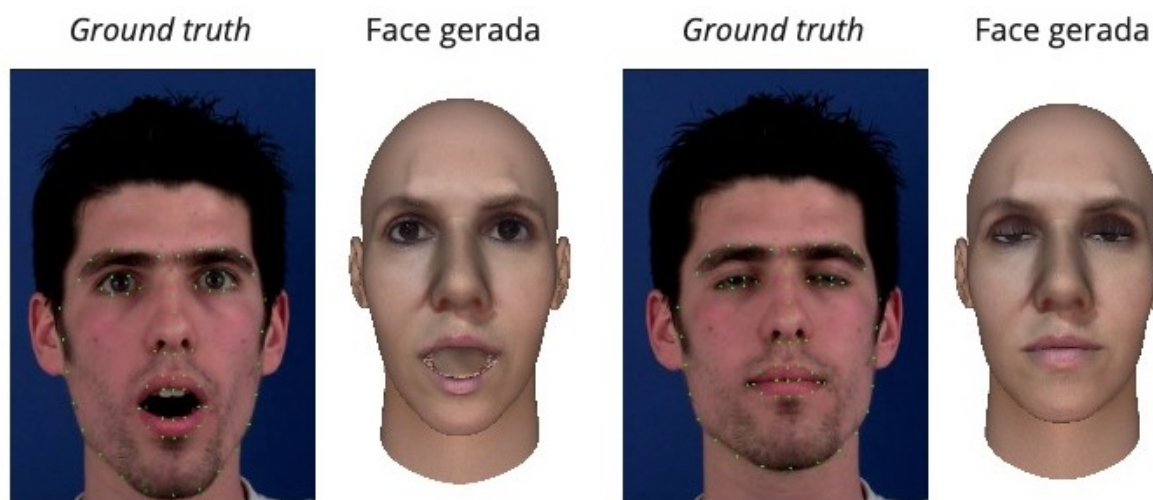
Fonte: elaborada pelo autor.

Para os vídeos, a acurácia foi determinada pelo cálculo do NME entre as coordenadas 2D dos marcadores faciais extraídos dos vídeos originais, quadro a quadro, com os marcadores faciais extraídos das faces geradas pelo *pipeline* desenvolvido. Um exemplo pode ser visto na Figura 6.14.

6.4.1 Desempenho dos classificadores

Para avaliar o desempenho dos classificadores, foram realizadas comparações em faces 3D e vídeos curtos de expressões.

Figura 6.14 – Exemplo de comparação entre a imagem da face do ator (*ground truth*) e a imagem do *blendshape* gerado pelo *pipeline*.



Fonte: elaborada pelo autor.

6.4.1.1 Comparação de desempenho através de expressões 3D

Na Figura 6.15 é mostrado o desempenho dos classificadores na comparação de uma face 3D, através do cálculo do NME, entre as expressões de referência, com as expressões 3D geradas pelo *pipeline*. A métrica utilizada mostra que, quanto menor o valor do NME, melhor é o desempenho do classificador.

Percebe-se que, as expressões desgosto e tristeza tiveram um desempenho semelhante entre os classificadores analisados. No entanto, para as expressões de surpresa e alegria, o classificador de Floresta Aleatória, não obteve um bom resultado sendo que o valor do NME para a expressão de surpresa foi 0,07 e para alegria foi 0,05, comparado com o classificador OPF, que para surpresa o valor foi 0,009 e, para alegria, foi 0,02. Nas expressões de medo e raiva, os classificadores obtiveram um desempenho semelhante sendo que, para expressão de raiva o classificador de Floresta Aleatória obteve o valor do NME de 0,017 e o OPF 0,014.

Ao observar a expressão de surpresa, nota-se que o classificador OPF obteve um melhor resultado na região da boca. Já, o classificador de Floresta Aleatória nessa região, não conseguiu identificar corretamente o *blendshape* mais relevante.

Na expressão de alegria o OPF também teve um resultado melhor, sendo o seu NME de 0,024 e o do classificador de Floresta Aleatória 0,055.

Figura 6.15 – Desempenhos dos classificadores por expressões.



Fonte: elaborada pelo autor.

6.4.1.2 Comparação de desempenho através de vídeos 2D quadro a quadro

Nos vídeos, o cálculo do NME foi aplicado quadro a quadro. Essa comparação, deu-se, por meio dos marcadores faciais extraídos da face no momento da captura, com a imagem 2D renderizada do *blendshape* gerado na saída do *pipeline*, no final do processo.

Os resultados mostram o desempenho dos classificadores durante a análise dos quadros por expressão, através do valor do NME normalizado.

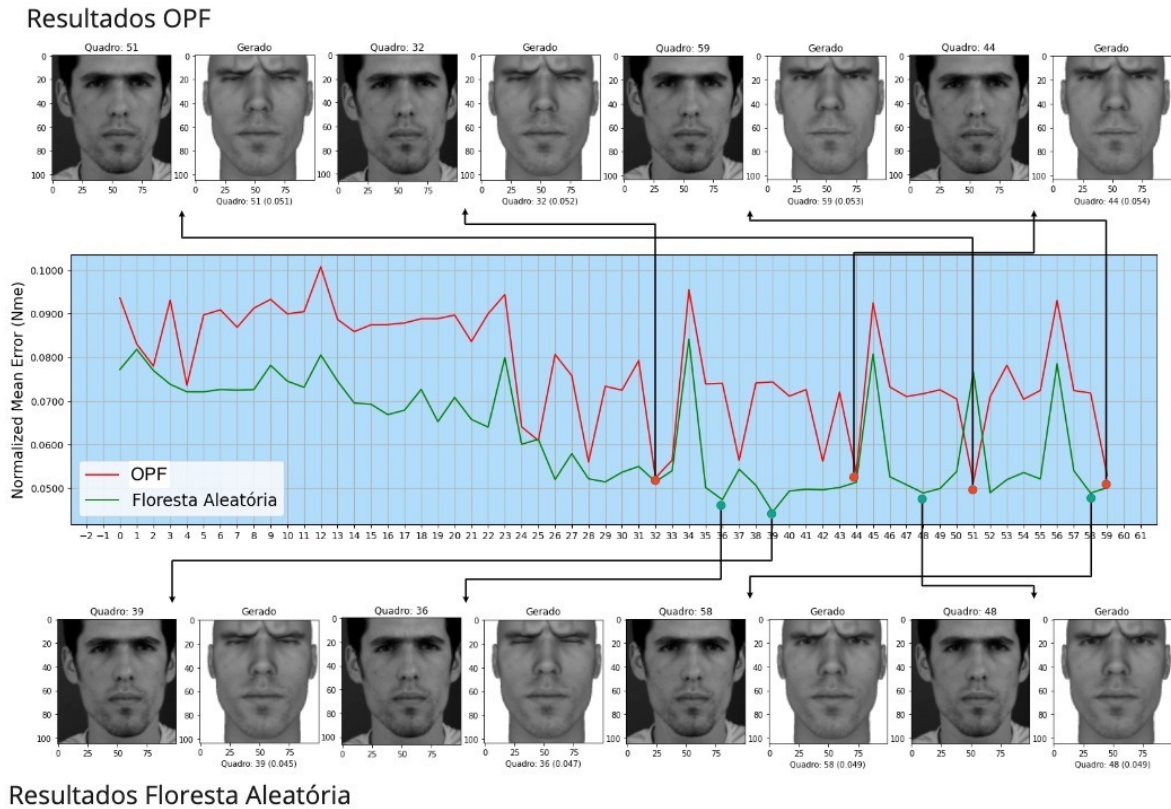
Além da análise de desempenho é mostrado, para cada expressão, os quadros que o classificador OPF e de Floresta Aleatória tiveram a melhor acurácia. Na comparação é mostrada a captura original e a face gerada.

As análises indicam os quadros de vídeos de atores em expressão neutra, mudando gradualmente, até atingir a expressão esperada nos quadros centrais, voltando, gradualmente, para a expressão neutra na parte final.

Na Figura 6.16 pode ser visto que, para a expressão de raiva, o melhor desempenho é visto para o classificador de Floresta Aleatória. Na maioria dos quadros, é apresentado um valor de NME melhor para esse classificador, por exemplo, no quadro 39, o valor do NME do classificador de Floresta Aleatória ficou abaixo de 0,05, enquanto que o OPF ficou acima de 0,07. No quadro 36 também acontece a mesma coisa, o NME do classificador de

Floresta Aleatória ficou abaixo de 0,05 e o OPF ficou acima de 0,07, com uma diferença grande entre eles.

Figura 6.16 – Desempenhos dos classificadores pela expressão de raiva.



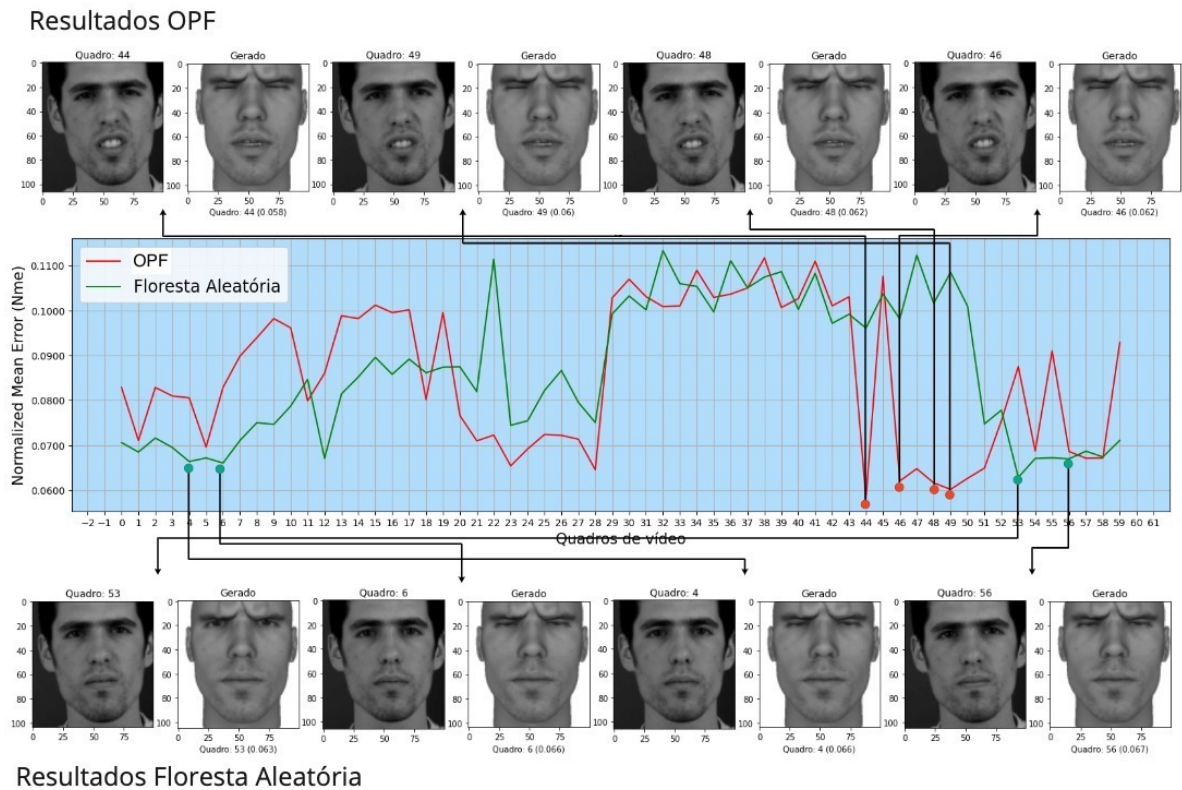
Fonte: elaborada pelo autor.

No gráfico que mostra o resultado para a face de desgosto, visto na Figura 6.17, percebe-se que o classificador OPF apresenta um resultado semelhante ao classificador de Floresta Aleatória, mas, em alguns pontos, ele apresenta um resultado melhor. Por exemplo, no quadro 53 o OPF ficou com o valor abaixo de 0,07 enquanto que, o classificador de Floresta Aleatória, ficou acima desse valor com 0,09 de NME.

O resultado apresentado para a expressão medo, observado na Figura 6.18, pode ser contemplada uma pequena variação entre os classificadores em função de desempenho, mas, com o OPF melhor do que o classificador de Floresta Aleatória. Por exemplo, nos quadros 23, o NME do OPF ficou abaixo de 0,09, enquanto que, do classificador de Floresta Aleatória ficou próximo de 0,10 e, no quadro 56, onde o OPF obteve um valor de NME abaixo de 0,09 e o classificador de Floresta Aleatória acima de 0,10.

Quanto à expressão facial de felicidade, apresentada na Figura 6.19, repete-se o mesmo cenário que na expressão de medo, em que, os classificadores OPF e o classificador de Floresta Aleatória tiveram um desempenho semelhante, mas, com uma pequena diferença

Figura 6.17 – Desempenhos dos classificadores pela expressão de desgosto.



Fonte: elaborada pelo autor.

para o classificador de Floresta Aleatória que, em alguns quadros foi melhor, como no 43 e 46 que ficou com o NME abaixo de 0,07 e o OPF ficou próximo de 0,08.

Na expressão de tristeza, pontuada na Figura 6.20, o classificador de Floresta Aleatória, na maioria dos quadros, teve desempenho melhor que o OPF. Por exemplo, nos quadros 22 e 24 ele ficou com o NME abaixo de 0,08 e o OPF ficou próximo de 0,9. Nos outros quadros existem algumas oscilações de valores, mas, com predominância de um melhor desempenho para o classificador de Floresta Aleatória.

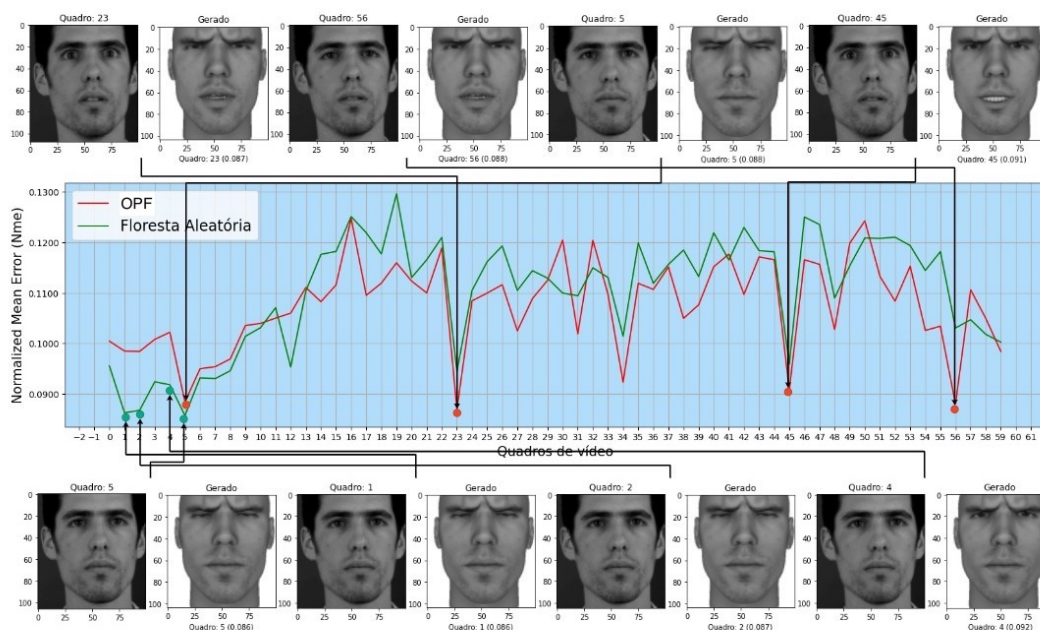
No resultado da análise da expressão facial de surpresa, mostrada na Figura 6.21, os classificadores OPF e Floresta Aleatória apresentaram um resultado semelhante, porém, com o classificador OPF melhor, principalmente, nos quadros centrais, entre os quadros 13 a 44, em que, as expressões de fato acontecem e os movimentos faciais são mais acentuados.

Um resumo do desempenho mostrado nos gráficos por expressão pode ser visto na Tabela 13. É feita uma amostragem de comparação da atuação dos classificadores, através da análise da porcentagem da quantidade de quadros que obtiveram o melhor valor de NME.

Na expressão de raiva pode ser visto que, o classificador de Floresta Aleatória, teve 96,67 % dos quadros com valores de NME, melhores que os quadros do classificador OPF.

Figura 6.18 – Desempenhos dos classificadores pela expressão de medo.

Resultados OPF

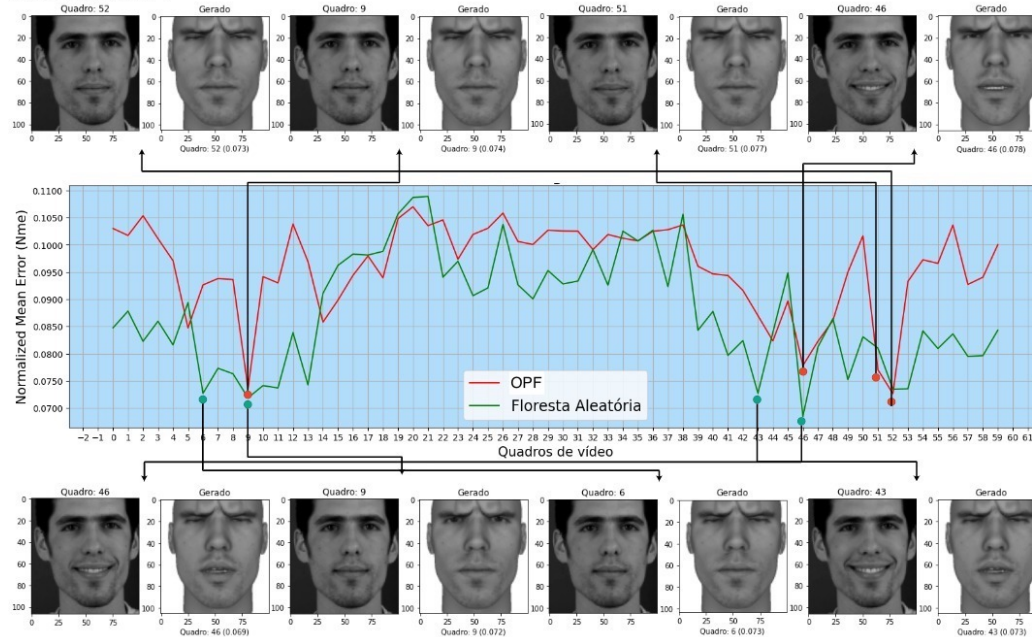


Resultados Floresta Aleatória

Fonte: elaborada pelo autor.

Figura 6.19 – Desempenhos dos classificadores pela expressão de felicidade.

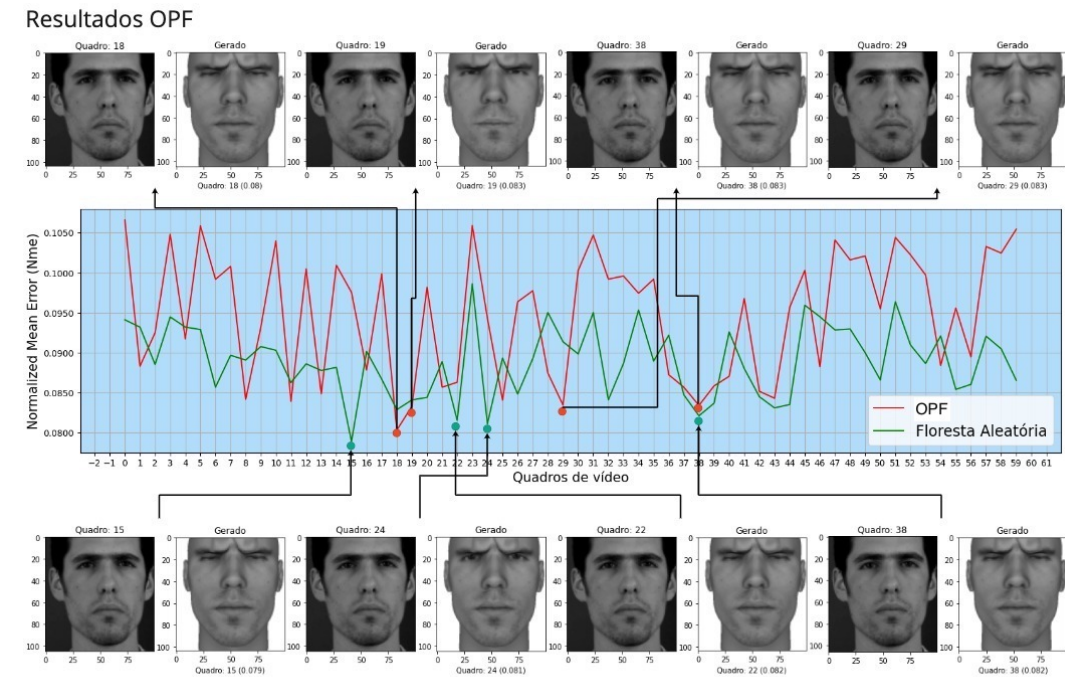
Resultados OPF



Resultados Floresta Aleatória

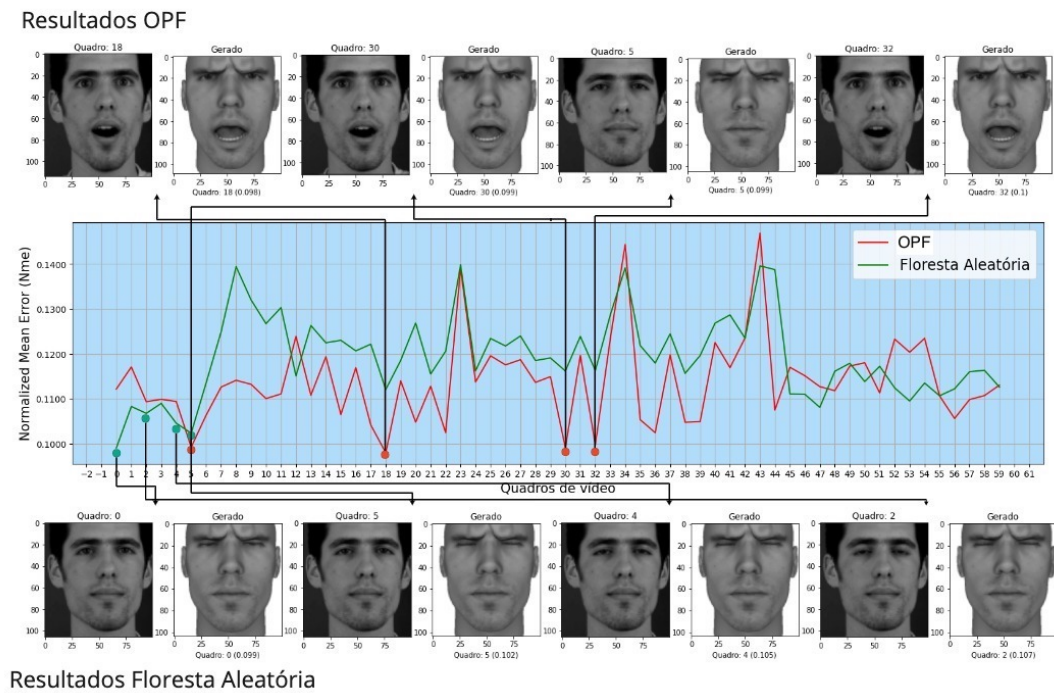
Fonte: elaborada pelo autor.

Figura 6.20 – Desempenhos dos classificadores pela expressão de tristeza.



Fonte: elaborada pelo autor.

Figura 6.21 – Desempenhos dos classificadores pela expressão de surpresa.



Fonte: elaborada pelo autor.

Tabela 13 – Comparação do desempenho dos classificadores através da análise da porcentagem dos quadros melhores.

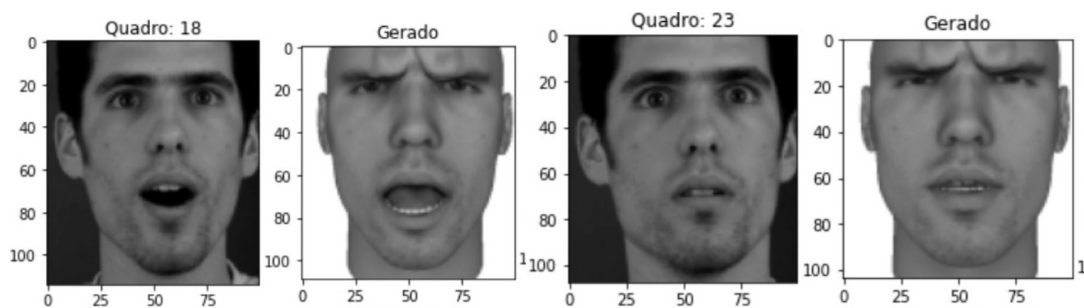
Expressão	OPF	Floresta Aleatória
Raiva	3,33%	96,67%
Desgosto	43,33%	56,67%
Medo	66,67%	33,33%
Felicidade	30,0%	70,0%
Tristeza	26,67%	73,33%
Surpresa	71,67%	28,33%

Já, nas expressões de surpresa e medo, o classificador OPF conseguiu um desempenho melhor na maioria dos quadros e nas expressões de raiva, felicidade e tristeza, o classificador de Floresta Aleatória obteve um desempenho melhor.

Por fim, na expressão de desgosto, ambos os classificadores tiveram um resultado semelhante, com uma pequena vantagem para o classificador de Floresta Aleatória.

Percebe-se que, nas expressões de medo e surpresa, as faces capturadas têm mais alterações, evidenciando que o classificador OPF consegue classificar melhor este tipo de captura (Figura 6.22).

Figura 6.22 – Capturas dos quadros dos vídeos de medo e surpresa.

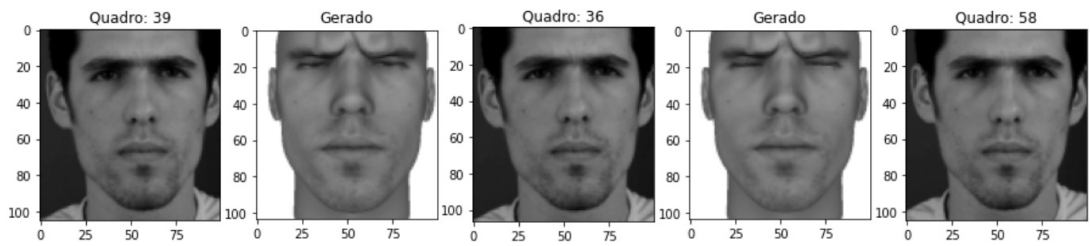


Fonte: elaborada pelo autor.

Na expressão de raiva, por exemplo, onde o classificador de Floresta Aleatória teve um desempenho muito acima do OPF, percebe-se que as expressões não se alteram muito, ficando próxima da expressão neutra (Figura 6.23).

Os vídeos analisados começam sempre com o ator exibindo uma expressão neutra, a qual vai sendo gradativamente alterada até atingir a expressão desejada (nos quadros centrais do vídeo). Depois, o ator retorna à expressão neutra. Quando se faz a análise, mostrada na Tabela 14, apenas nos quadros centrais, a OPF consegue uma melhor qualidade com relação às expressões de desgosto, medo e surpresa que, possuem mais alterações, como mostram alguns exemplos nas Figura 6.24. Isso mostra, mais uma vez, que o classificador

Figura 6.23 – Capturas dos quadros dos vídeos da expressão raiva.



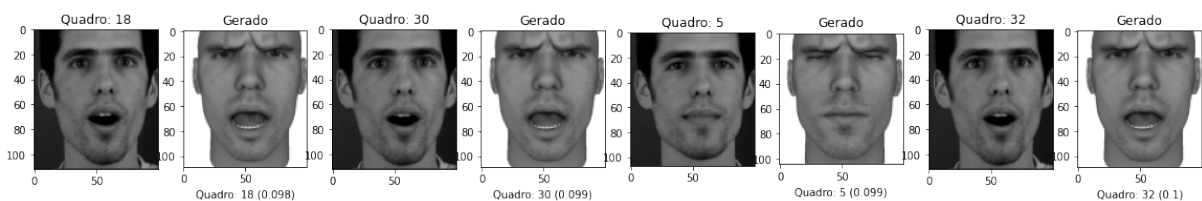
Fonte: elaborada pelo autor.

OPF consegue classificar melhor as faces que possuem expressões mais acentuadas, em relação a uma face neutra.

Tabela 14 – Análise de desempenho dos classificadores dos quadros do centro do vídeo.

Expressão	OPF	Floresta Aleatória
Raiva	5,0 %	95,5 %
Desgosto	65,0 %	35,0%
Medo	90,0 %	10,0 %
Felicidade	30,0 %	70,0 %
Tristeza	30,0 %	70,00 %
Surpresa	95,0 %	5,0 %

Figura 6.24 – Exemplo de desempenhos do classificador OPF na região da boca.



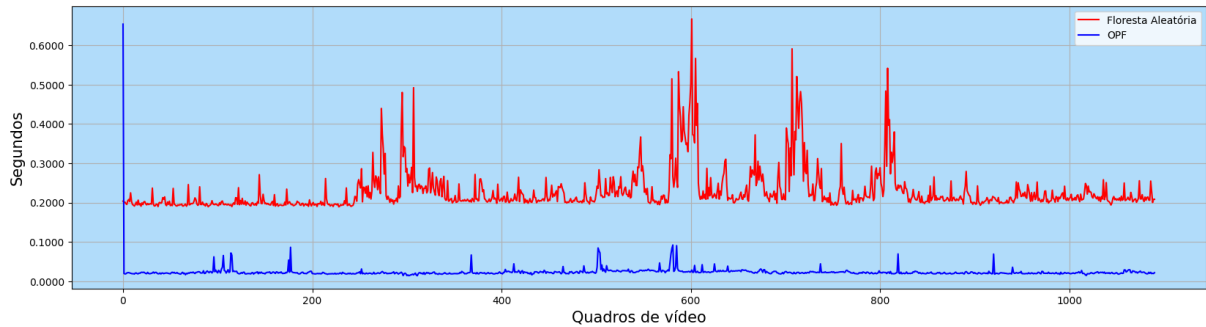
Fonte: elaborada pelo autor.

6.4.1.3 Comparando os classificadores em função do tempo

Além da análise dos resultados, comparando o *ground truth* com a face gerada, a atuação do classificador, foi calculada em função do tempo de processamento para a classificação das amostras. Esse tempo, foi medido apenas na etapa de classificação, não levando em conta o processo de captura e construção do *blendshape* final. A aferição do tempo foi realizada de 4 maneiras diferentes, porque no processo de redirecionamento facial, o *pipeline* desenvolvido analisa áreas da face separadamente e, no final, elas são combinadas criando o *blendshape*. Os gráficos são apresentados em função do tempo para cada quadro dos vídeos analisados, que foram os mesmos utilizados na análise das expressões.

Na Figura 6.25, a ação dos classificadores é mostrada, após a análise das 3 áreas da face, com isso, ela traz o somatório, em segundos, das 3 etapas executadas que é a pesquisa da área da boca, dos olhos e das sobrancelhas.

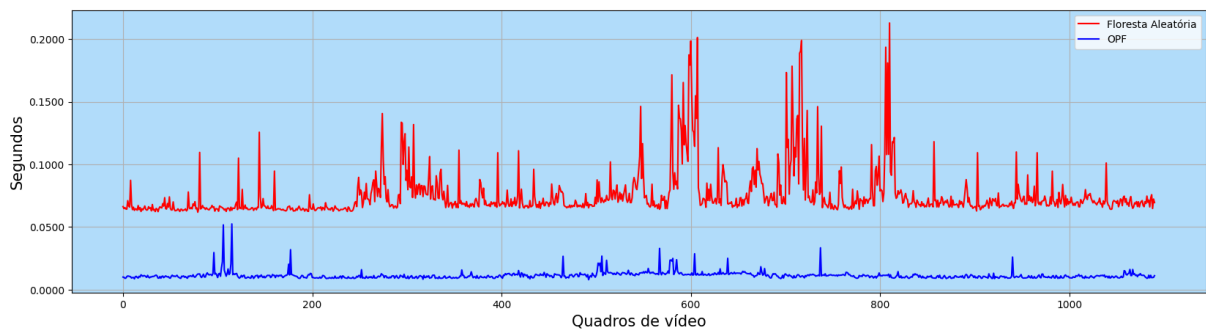
Figura 6.25 – Comparação do tempo de processamento entre os classificadores de Floresta Aleatória e OPF.



Fonte: elaborada pelo autor.

Nas figuras 6.26, 6.27 e 6.28, o estudo do desempenho é feito para cada área da face.

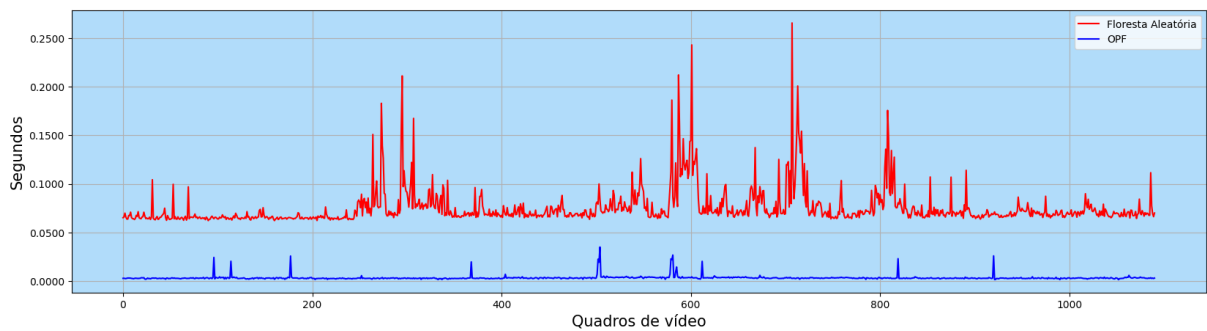
Figura 6.26 – Comparação do tempo de processamento entre os classificadores de Floresta Aleatória e OPF para a região da boca.



Fonte: elaborada pelo autor.

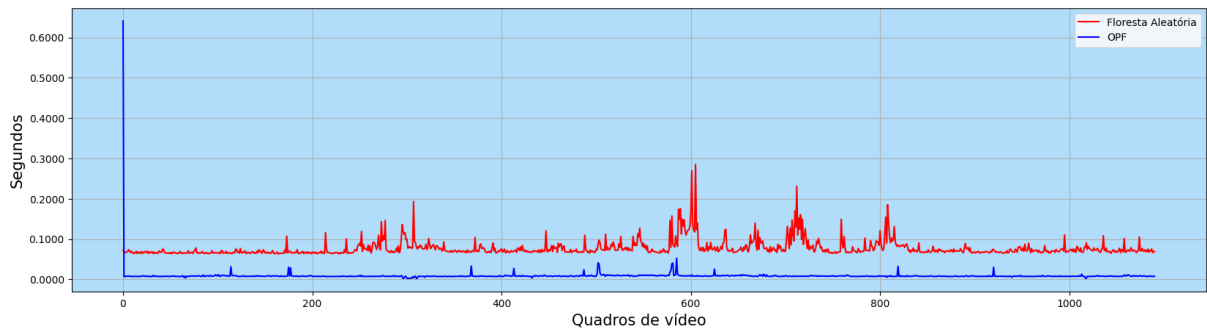
Segundo Microsoft (2018), o mínimo de quadros necessários para se obter tempo real é de 30 quadros por segundo. Com base nos valores obtidos, pode-se afirmar que o classificador OPF apresentou sucesso no processamento do quadro, menor que 0.3 segundos e, enquadra-se no tempo necessário, para obter o tempo real. Isso, pode ser observado tanto no gráfico onde está o processamento total (Figura 6.25) quanto nos gráficos que demonstram o tempo de processamento por áreas da face. Entretanto, o classificador de Floresta Aleatória, só obteve o desempenho necessário para o tempo real na classificação da região das sobrancelhas (6.28). Em nenhum outro caso, realizou um tempo abaixo de 30 quadros por segundo, ou seja, abaixo de 0,03 segundos.

Figura 6.27 – Comparação do tempo de processamento entre os classificadores de Floresta Aleatória e OPF para a região dos olhos.



Fonte: elaborada pelo autor.

Figura 6.28 – Comparação do tempo de processamento entre os classificadores de Floresta Aleatória e OPF para a região dos sobrelhos.



Fonte: elaborada pelo autor.

6.4.1.4 Comparação entre a face da entrada e um *blendshape* diferente na saída

O sistema de animação facial é utilizado para transferir uma captura de entrada para um objeto 3D na saída. Existem casos onde, a entrada e a saída não são iguais, por exemplo, em jogos e animações, em que, os personagens podem ser animais ou até objetos cartunescos. Em alguns trabalhos, como o de Aneja et al. (2016), foi construído um *pipeline* que transfere uma captura humana para animações, totalmente diferentes da entrada.

No *pipeline* do sistema de animação facial desenvolvido, também mostrou-se possível este tipo de redirecionamento, pois ele consegue identificar as semelhanças mesmo que a face de entrada não seja a mesma face de saída. A limitação é que sejam faces com características humanas na captura, caso contrário, a extração dos marcadores faciais fica comprometida, não permitindo que sejam extraídas as características corretamente.

Para comparação de desempenho, nesse tipo de abordagem, alguns trabalhos usam pesquisas, através de formulários como o de Oosterom (2018), outros fazem avaliações visuais com participação de outras pessoas, como é o caso de Ekmen e Ekenel (2018). A

comparação de desempenho utilizada aqui foi a inspeção visual, pois, não foi possível o uso das métricas aplicadas nos outros experimentos, uma vez que as entradas e saídas são muito discrepantes em termos de anatomia e estrutura geométrica.

Como exemplo dessa possibilidade, pode ser visto na Figura 6.29 uma captura da face de homem diferente do modelo 3D da face gerada. Apesar de não serem semelhantes, a face final gerada possui semelhanças com a imagem do ator, principalmente na região das sobrancelhas e boca.

Figura 6.29 – Exemplo de captura e saídas diferentes utilizando apenas um *blendshape* por área da face.

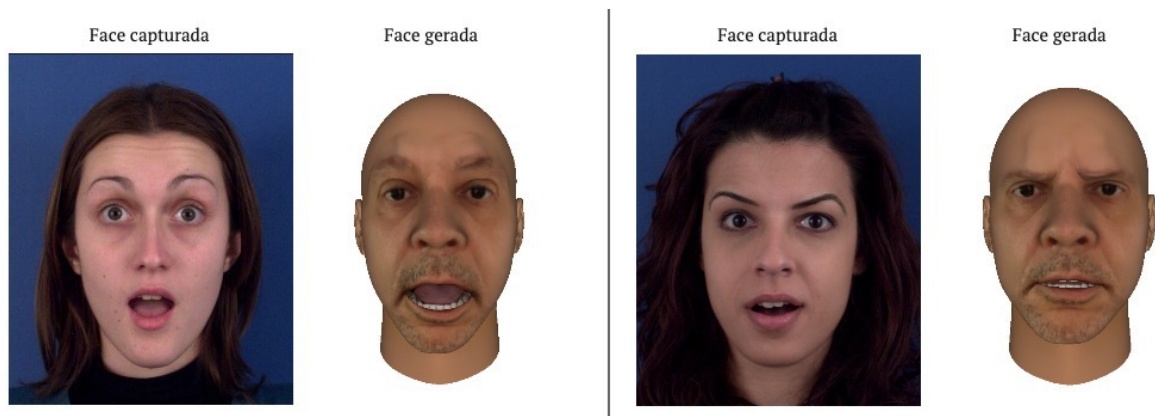


Fonte: elaborada pelo autor.

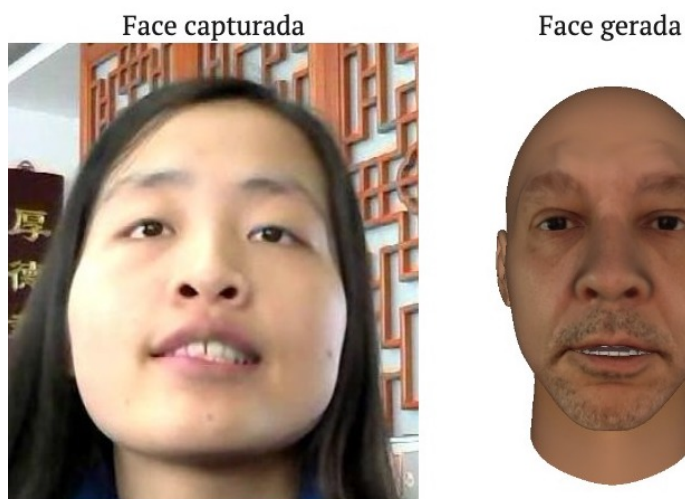
Outro exemplo pode ser visto na Figura 6.30, onde as entradas são faces femininas e o *blendshape* utilizado na saída é de uma face masculina. Percebe-se, visualmente, que os classificadores conseguiram identificar algumas características e gerar o *blendshape* com semelhanças.

Pode-se perceber que o *blendshape* final, em ambos os casos, conseguiu trazer algumas características da face, como a região das sobrancelhas, que tem um movimento mais acentuado. A região da boca, não está tão semelhante, mas, apresenta um movimento próximo ao da face de entrada, mostrando que o classificador identificou que a captura estava fazendo movimentos nessa região.

Na Figura 6.31 foi feito um teste com uma captura de uma mulher oriental, para um *blendshape* masculino que tem o rosto, principalmente a região dos olhos, de pessoas ocidentais. Percebe-se, no primeiro caso, que há semelhanças entre as regiões da boca e dos olhos, mostrando que os classificadores conseguiram identificar os movimentos.

Figura 6.30 – Exemplos de capturas de uma faces feminina para um *blendshape* masculino.

Fonte: elaborada pelo autor.

Figura 6.31 – Exemplo de uma captura de uma face feminina oriental para um *blendshape* masculino ocidental.

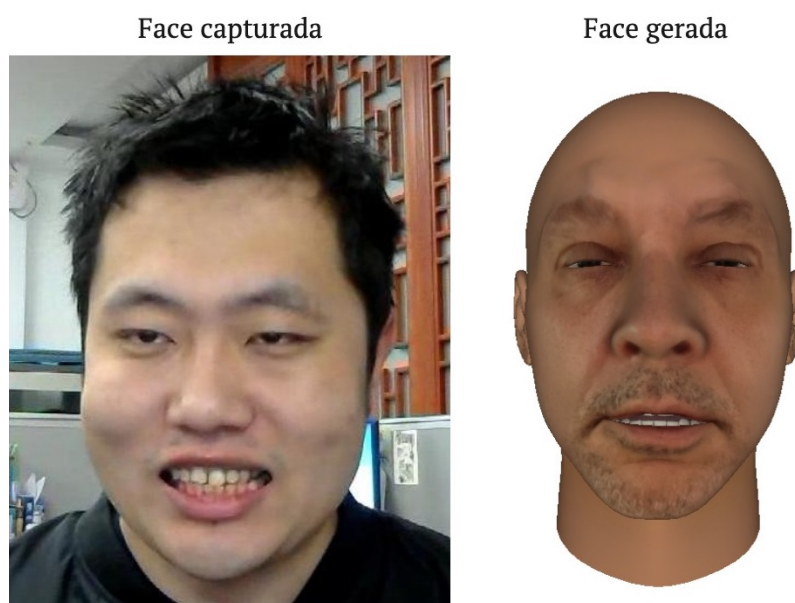
Fonte: elaborada pelo autor.

E no exemplo mostrado na Figura 6.32, percebe-se, claramente, que o classificador tentou aplicar o *blendshape* na região dos olhos e obteve sucesso, mas como ele não foi treinado para distinguir tipos de olhos com características ocidentais ou orientais, a expressão gerada parece errada, porém, isso mostra umas da limitações do processo.

Com base nos testes apresentados é possível afirmar que o sistema desenvolvido consegue fazer o redirecionamento de uma captura para um *blendshape* diferente. Isso é possível, porque os modelos treinados utilizaram uma base de dados com amostras de diferentes tipos de faces e gêneros. Isso fez, com que a sua classificação, não fosse vinculada a um tipo de rosto ou geometria específica.

Além disso, a rotulação das classes utilizou as referências propostas por Ekman e Rosenberg (1997), que dividiram a face em diferentes grupos chamados unidades de ação,

Figura 6.32 – Exemplo de uma captura de uma face masculina oriental para um *blendshape* masculino ocidental.



Fonte: elaborada pelo autor.

as AUS. Isso fez com que, os modelos treinados pudessem fazer o *retarget* de uma face capturada para um *blendshape* totalmente diferente, desde que, o *blendshape* de saída, seja rotulado seguindo as AUS estabelecidas deixando o sistema mais flexível.

7 Conclusões

Diante das análises e dos experimentos executados, neste trabalho, pode-se afirmar que foi implementado e validado um sistema de animação facial. Através do *pipeline* proposto foi possível, utilizando modelos treinados, identificar as características da face capturada e, redirecioná-la de forma automática, para um modelo 3D. O *pipeline* demonstrou a possibilidade da utilização de modelos diferentes de classificadores como o OPF e Floresta Aleatória para gerar a face final no seu módulo de *retarget*.

Para o sistema de animação facial foi gerada uma base de dados de *blendshapes* e uma base de dados de amostras rotulados seguindo as AUS, permitindo o treinamento de classificadores de forma supervisionada.

O treinamento realizado com os classificadores OPF e de Floresta Aleatória foi capaz de identificar na face capturada, as diferentes movimentações por região e, transportar isso, para uma face virtual.

Apesar de usar apenas informações 2D, foi possível identificar características relevantes para o treinamento dos classificadores, mas a pesquisa mostrou que se faz necessária a inclusão de mais informações para que os modelos sejam mais precisos. Isso pode ser demonstrado na análise dos classificadores quando, por exemplo, as amostras eram muito próximas e o modelo não conseguia prever corretamente as suas classes.

Mesmo não sendo realizado o somatório dos pesos e, tendo feito uso apenas de um *blendshape* para cada região da face, a saída final em algumas capturas, ficaram próximas, mostrando ser possível o uso de um pequeno número de *blendshapes*. Isso pode ser visto, por exemplo, na subseção 6.4.1.4. Mas, como a face gerada final depende dos modelos treinados, essa saída ficou limitada pela quantidade de classes, sendo necessário, para ter faces com maior acurácia, um maior número de classes de *blendshapes* para cada região.

Ao treinar os classificadores com bases rotuladas pelas AUS, foi possível desvincular a necessidade do modelo 3D utilizado para gerar a face final, de ser próxima ou semelhante à face capturada. Experimentos feitos com capturas diferentes da saída como, a de mulheres e pessoas orientais, tiveram, visualmente, resultados próximos ao da captura quanto à movimentação da face.

Casos onde, as faces capturadas tinham expressões mais distantes da expressão neutra, o classificador OPF teve um melhor desempenho do que o classificador de Floresta Aleatória. Isso pode ser visto nas análises dos vídeos por expressão. No caso das expressões de desgosto, medo e surpresa, na análise dos quadros centrais, onde a expressão está mais evidente, o OPF apresentou melhores resultados mostrados na Tabela 14.

Para a execução em tempo real, os experimentos mostraram que, o classificador OPF teve um desempenho compatível com os 30 quadros por segundos em todos os casos analisados, inclusive, quando somados os tempos de previsão das 3 regiões da face. Já, o classificador de Floresta Aleatória não, ficando muito acima desse tempo.

Como trabalhos futuros, vislumbra-se que os seguintes desdobramentos são possíveis:

- Aperfeiçoamento da extração das características, para deixar os modelos mais sensíveis, em função das pequenas diferenças entre as classes;
- Aumentar a quantidade da base de *blendshapes*, para abranger uma quantidade maior de movimentos faciais;
- Treinar outras redes neurais ou classificadores, para aprimorar os modelos;
- Tratamento das características da face humana que não foram consideradas neste trabalho, tais como: globo ocular, dentes, língua e cabelo;
- O sistema de animação facial desenvolvido possibilita o uso de classificadores diferentes, sendo possível a troca de algoritmos para futuros testes. Em trabalhos futuros, a utilização de diferentes classificadores, uma para cada região da face, pode ser uma aplicação válida, pois permitiria uma maior flexibilidade de escolhas entre diferentes classificadores; e
- Melhorar, no *pipeline* desenvolvido, os processos de captura e previsão, incluindo processamento paralelo para aumentar velocidade de processamento final.

Referências

- ABDELAZIZ, A. H.; THEOBALD, B.-J.; DIXON, P.; KNOTHE, R.; APOSTOLOFF, N.; KAJAREKER, S. Modality dropout for improved performance-driven talking faces. In: _____. *Proceedings of the 2020 International Conference on Multimodal Interaction*. New York, NY, USA: Association for Computing Machinery, 2020. p. 378–386. ISBN 9781450375818. Disponível em: <<https://doi.org/10.1145/3382507.3418840>>.
- ACADEMY, D. S. *Arquitetura de Redes Neurais Long Short Term Memory (LSTM)*. 2019. Disponível em: <<http://deeplearningbook.com.br/arquitetura-de-redes-neurais-long-short-term-memory>>.
- AGARWAL, P. K.; PROCOPIUC, C. M. Exact and approximation algorithms for clustering. *Algorithmica*, Springer, v. 33, n. 2, p. 201–226, 2002.
- AHMED, M.; SERAJ, R.; ISLAM, S. M. S. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, MDPI, v. 9, n. 8, p. 1295, 2020.
- AIFANTI, N.; PAPACHRISTOU, C.; DELOPOULOS, A. The mug facial expression database. In: *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*. [S.l.: s.n.], 2010. p. 1–4.
- ANEJA, D.; CHAUDHURI, B.; COLBURN, A.; FAIGIN, G.; SHAPIRO, L.; MONES, B. Learning to generate 3d stylized character expressions from humans. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. [S.l.: s.n.], 2018. p. 160–169.
- ANEJA, D.; COLBURN, A.; FAIGIN, G.; SHAPIRO, L.; MONES, B. Modeling stylized character expressions via deep learning. In: SPRINGER. *Asian Conference on Computer Vision*. [S.l.], 2016. p. 136–153.
- ARORA, S.; RAGHAVAN, P.; RAO, S. Approximation schemes for euclidean k-medians and related problems. In: *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. [S.l.: s.n.], 1998. p. 106–113.
- BALTRUSAITIS, T.; ZADEH, A.; LIM, Y. C.; MORENCY, L.-P. Openface 2.0: Facial behavior analysis toolkit. In: *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. [S.l.: s.n.], 2018. p. 59–66.
- BEHRENS, S.; AL-HAMADI, A.; REDWEIK, E.; NIESE, R. Automatic realtime user performance-driven avatar animation. In: IEEE. *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. [S.l.], 2013. p. 2694–2699.
- BILAKHIA, S.; PETRIDIS, S.; NIJHOLT, A.; PANTIC, M. The mahnob mimicry database: A database of naturalistic human interactions. *Pattern Recognition Letters*, v. 66, p. 52–61, 2015. ISSN 0167-8655. Pattern Recognition in Human Computer Interaction. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167865515000768>>.
- BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: springer, 2006.

- BLANZ, V.; VETTER, T. A morphable model for the synthesis of 3d faces. In: ACM PRESS/ADDISON-WESLEY PUBLISHING CO. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. [S.l.], 1999. p. 187–194.
- BRAUN, A. Aprendizado e utilização do estilo de movimento facial na animação de avatares. Pontifícia Universidade Católica do Rio Grande do Sul, 2014.
- CAO, C.; HOU, Q.; ZHOU, K. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graph.*, Association for Computing Machinery, New York, NY, USA, v. 33, n. 4, jul 2014. ISSN 0730-0301. Disponível em: <<https://doi.org/10.1145/2601097.2601204>>.
- CAO, C.; WENG, Y.; ZHOU, S.; TONG, Y.; ZHOU, K. Facewarehouse: A 3d facial expression database for visual computing. *Visualization and Computer Graphics, IEEE Transactions on*, v. 20, n. 3, p. 413–425, March 2014. ISSN 1077-2626.
- CHAUDHURI, B.; VESDAPUNT, N.; WANG, B. Joint face detection and facial motion retargeting for multiple faces. In: . [S.l.: s.n.], 2019. p. 9711–9720.
- CHAUHAN, N. *Model Evaluation Metrics in Machine Learning*. 2020. Disponível em: <<https://www.kdnuggets.com/2020/05/model-evaluation-metrics-machine-learning.html>>.
- CHEN, H.; ZHANG, X.; MA, T.; YUE, H.; WANG, X.; ZHANG, B. Efficient facial landmark localization based on binarized neural networks. *Electronics*, v. 9, n. 8, 2020. ISSN 2079-9292. Disponível em: <<https://www.mdpi.com/2079-9292/9/8/1236>>.
- CHEN, Z.; WU, J.; GAN, W.; QI, Z. *Metaverse Security and Privacy: An Overview*. arXiv, 2022. Disponível em: <<https://arxiv.org/abs/2211.14948>>.
- CHO, K.; MERRIËNBOER, B. V.; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- CHOI, H.; PARK, S. A survey of machine learning-based system performance optimization techniques. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 11, n. 7, p. 3235, 2021.
- CHOLLET, F. et al. *Keras*. GitHub, 2015. Disponível em: <<https://github.com/fchollet/keras>>.
- CIGNONI, P.; CALLIERI, M.; CORSINI, M.; DELLEPIANE, M.; GANOVELLI, F.; RANZUGLIA, G. MeshLab: an Open-Source Mesh Processing Tool. In: SCARANO, V.; CHIARA, R. D.; ERRA, U. (Ed.). *Eurographics Italian Chapter Conference*. [S.l.]: The Eurographics Association, 2008. ISBN 978-3-905673-68-5.
- COMMUNITY, B. O. *Blender - a 3D modelling and rendering package*. Stichting Blender Foundation, Amsterdam, 2018. Disponível em: <<http://www.blender.org>>.
- COSTIGAN, T.; GERDELAN, A.; CARRIGAN, E.; MCDONNELL, R. Improving blendshape performance for crowds with gpu and gpgpu techniques. In: *Proceedings of the 9th International Conference on Motion in Games*. New York, NY, USA: Association for Computing Machinery, 2016. (MIG '16), p. 73–78. ISBN 9781450345927. Disponível em: <<https://doi.org/10.1145/2994258.2994275>>.

- COSTIGAN, T.; PRASAD, M.; MCDONNELL, R. Facial retargeting using neural networks. In: *Proceedings of the Seventh International Conference on Motion in Games*. New York, NY, USA: Association for Computing Machinery, 2014. p. 31–38. ISBN 9781450326230. Disponível em: <<https://doi.org/10.1145/2668064.2668099>>.
- CUBAS, C.; SEMENTILLE, A. The development of a facial animation system based on performance and the use of an rgb-d camera. In: *2017 19th Symposium on Virtual and Augmented Reality (SVR)*. [S.l.: s.n.], 2017. p. 78–86. ISSN null.
- DENG, Z.; NEUMANN, U. Data-driven 3d facial animation. Springer, 2008.
- EKMAN, P.; ROSENBERG, E. L. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. [S.l.]: Oxford University Press, 1997.
- EKMEN, B.; EKENEL, H. From 2d to 3d real-time expression transfer for facial animation. *Multimedia Tools and Applications*, 10 2018.
- FALCÃO, A. X.; STOLFI, J.; LOTUFO, R. A. The image foresting transform theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 1, p. 19–29, 2004.
- FRATARCANGELI, M. Computational models for animating 3d virtual faces. Linköping University Electronic Press, 2013.
- GAWLIKOWSKI, J.; TASSI, C. R. N.; ALI, M.; LEE, J.; HUMT, M.; FENG, J.; KRUSPE, A.; TRIEBEL, R.; JUNG, P.; ROSCHER, R. et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021.
- GELMAN, S. *First Concept Art from Lord of the Rings: Gollum Game Surfaces*. 2020. <<https://www.cbr.com/concept-art-from-lord-of-the-rings-gollum-game/>>. Accessed: 2022-09-26.
- GHASEMIAN, A.; HOSSEINMARDI, H.; CLAUSET, A. Evaluating overfit and underfit in models of network community structure. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 32, n. 9, p. 1722–1735, 2019.
- GILBERT, M.; DEMARCHI, S.; URDAPILLETA, I. Facshuman a software to create experimental material by modeling 3d facial expression. In: *Proceedings of the 18th International Conference on Intelligent Virtual Agents*. New York, NY, USA: Association for Computing Machinery, 2018. (IVA '18), p. 333–334. ISBN 9781450360135. Disponível em: <<https://doi.org/10.1145/3267851.3267865>>.
- GOELEN, E.; RAEDT, R. D.; LEYMAN, L.; VERSCHUERE, B. The karolinska directed emotional faces: a validation study. *Cognition and emotion*, Taylor & Francis, v. 22, n. 6, p. 1094–1118, 2008.
- GUO, Y.; ZHANG, J.; CAI, J.; JIANG, B.; ZHENG, J. Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 41, p. 1294–1307, 2017.
- HAAS, J. K. A history of the unity game engine. Worcester Polytechnic Institute, 2014.

- HO, T. K. Random decision forests. In: *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*. M: IEEE Computer Society, 1995. (ICDAR '95), p. 278–282. ISBN 0-8186-7128-9.
- HUANG, P.; WANG, Y.; SHAO, M. A new method for multi-view face clustering in video sequence. In: *2008 IEEE International Conference on Data Mining Workshops*. [S.l.: s.n.], 2008. p. 869–873.
- HUANG, Y.; YANG, H.; LI, C.; KIM, J.; WEI, F. *ADNet: Leveraging Error-Bias Towards Normal Direction in Face Alignment*. arXiv, 2021. Disponível em: <<https://arxiv.org/abs/2109.05721>>.
- INDOLIA, S.; GOSWAMI, A.; MISHRA, S.; ASOPA, P. Conceptual understanding of convolutional neural network- a deep learning approach. *Procedia Computer Science*, v. 132, p. 679–688, 01 2018.
- KANUNGO, T.; MOUNT, D.; NETANYAHU, N.; PIATKO, C.; SILVERMAN, R.; WU, A. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 7, p. 881–892, 2002.
- KING, D. E. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, v. 10, p. 1755–1758, 2009.
- LAI, S.; CHAI, Z.; LI, S.; MENG, H.; YANG, M.; WEI, X. Enhanced normalized mean error loss for robust facial landmark detection. In: *BMVC*. [S.l.: s.n.], 2019.
- LEE, L.-H.; BRAUD, T.; ZHOU, P.; WANG, L.; XU, D.; LIN, Z.; KUMAR, A.; BERMEJO, C.; HUI, P. *All One Needs to Know about Metaverse: A Complete Survey on Technological Singularity, Virtual Ecosystem, and Research Agenda*. arXiv, 2021. Disponível em: <<https://arxiv.org/abs/2110.05352>>.
- LEWIS, J. P.; ANJYO, K.; RHEE, T.; ZHANG, M.; PIGHIN, F. H.; DENG, Z. Practice and theory of blendshape facial models. In: CITESEER. *Eurographics (State of the Art Reports)*. [S.l.], 2014. p. 199–218.
- LI, D.; SUN, C.; HU, F.; ZANG, D.; WANG, L.; ZHANG, M. Real-time performance-driven facial animation with 3ds max and kinect. In: *Consumer Electronics, Communications and Networks (CECNet), 2013 3rd International Conference on*. [S.l.: s.n.], 2013. p. 473–476.
- LI, H.; YU, J.; YE, Y.; BREGLER, C. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 32, n. 4, p. 42:1–42:10, jul. 2013. ISSN 0730-0301. Disponível em: <<http://doi.acm.org/10.1145/2461912.2462019>>.
- LORENZETT, C. dal C.; TELOKEN, A. V. Estudo comparativo entre os algoritmos de mineração de dados random forest e j48 na tomada de decisão. In: . [S.l.: s.n.], 2016.
- LUCEY, P.; COHN, J. F.; KANADE, T.; SARAGIH, J.; AMBADAR, Z.; MATTHEWS, I. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. [S.l.: s.n.], 2010. p. 94–101.

- MACQUEEN, J. B. Some methods for classification and analysis of multivariate observations. In: CAM, L. M. L.; NEYMAN, J. (Ed.). *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*. [S.l.]: University of California Press, 1967. v. 1, p. 281–297.
- MARAR, J. F.; JUNIOR, H. F.; HOLDSCHIP, R.; SEMENTILLE, A. C.; BUSATO, S. L. Emoção & design: Quando a face revela o sucesso do produto. In: CORES, E. das Letras e (Ed.). *Metodologia em design : inter-relacoes*. [S.l.], 2011. p. 162–183.
- MAVADATI, S. M.; MAHOOR, M. H.; BARTLETT, K.; TRINH, P.; COHN, J. F. Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, v. 4, n. 2, p. 151–160, 2013.
- Microsoft. *Understanding Frames Per Second (FPS)*. 2018. <<https://support.microsoft.com/pt-br/help/269068/understanding-frames-per-second-fps>>. Accessed: 2020-02-22.
- MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: *Sistemas Inteligentes Fundamentos e Aplicações*. 1. ed. Barueri-SP: Manole Ltda, 2003. p. 89–114. ISBN 85-204-168. Disponível em: <<http://dcm.ffclrp.usp.br/~augusto/publications/2003-sistemas-inteligentes-cap4.pdf>>.
- MUNASINGHE, M. I. N. P. Facial expression recognition using facial landmarks and random forest classifier. In: *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*. [S.l.: s.n.], 2018. p. 423–427.
- NAVLANI, A. *Understanding Random Forests Classifiers in Python*. 2018. Disponível em: <<https://www.datacamp.com/community/tutorials/random-forests-classifier-python#how>>.
- NENDYA, M. B.; YUNIARNO, E. M.; GANDANG, S. Facial rigging for 3d character. *Int. J. Comput. Graph. Animat*, v. 4, n. 3, p. 21–29, 2014.
- NISHIMURA, R.; SAKATA, N.; TOMINAGA, T.; HIJIKATA, Y.; HARADA, K.; KIYOKAWA, K. Speech-driven facial animation by lstm-rnn for communication use. *2019 12th Asia Pacific Workshop on Mixed and Augmented Reality (APMAR)*, p. 1–8, 2019.
- OOSTEROM, T. van. *Facial Animation Retargeting using Recurrent Neural Networks*. Dissertação (Mestrado), 2018.
- PANCERI, J. A. *Reconhecimento de Expressões Faciais Baseado em Active Appearance Model*. 2017.
- PANDZIC, I. S.; FORCHHEIMER, R. Mpeg-4 facial animation. *The standard, implementation and applications*. Chichester, England: John Wiley & Sons, Wiley Online Library, 2002.
- PAPA, J.; FALCÃO, A.; SUZUKI, C. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, John Wiley & Sons, Inc., New York, NY, USA, v. 19, n. 2, p. 120–131, 2009.
- PAPA, J.; FALCÃO, A. Optimum-path forest: A novel and powerful framework for supervised graph-based pattern recognition techniques. 01 2010.

- PAPA, J. P.; FALCÃO, A. X.; FREITAS, G. M. de; ÁVILA, A. M. H. de. Robust pruning of training patterns for optimum-path forest classification applied to satellite-based rainfall occurrence estimation. *IEEE Geoscience and Remote Sensing Letters*, v. 7, n. 2, p. 396–400, 2010.
- PAPA, J. P.; FALCAO, A. X.; SUZUKI, C. T. N. Supervised pattern classification based on optimum-path forest. *Int. J. Imaging Syst. Technol.*, USA, v. 19, n. 2, p. 120–131, jun. 2009. ISSN 0899-9457.
- PAPA, J. P.; FERNANDES, S. E. N.; FALCAO, A. X. Optimum-path forest based on k-connectivity. *Pattern Recogn. Lett.*, Elsevier Science Inc., USA, v. 87, n. C, p. 117–126, fev. 2017. ISSN 0167-8655. Disponível em: <<https://doi.org/10.1016/j.patrec.2016.07.026>>.
- PAULY, M. Realtime performance-based facial avatars for immersive gameplay. In: *ACM. Proceedings of Motion on Games*. [S.l.], 2013. p. 23–28.
- PENNINGTON, A. *What Do Motion Capture Actors Actually Do?* 2018. <<https://www.backstage.com/uk/magazine/article/trends-and-intelligence-motion-capture-performance-66001/>>. Accessed: 2022-09-26.
- PEREIRA, C. *Detecção de Intrusão em Redes de Computadores Utilizando Floresta de Caminhos Ótimos*. 100 p. Dissertação (Mestrado) — Unesp, Universidade Estadual Paulista Júlio de Mesquita, 08 2012.
- PEREIRA, C. R.; NAKAMURA, R.; PAPA, J. P.; COSTA, K. Intrusion detection system using optimum-path forest. In: *LCN'11*. [S.l.]: IEEE, 2011. p. 183–186.
- Pham, H. X.; Pavlovic, V.; Jianfei Cai; Tat-jen Cham. Robust real-time performance-driven 3d face tracking. In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. [S.l.: s.n.], 2016. p. 1851–1856.
- PISANI, R.; RIEDEL, P.; FERREIRA, M.; MARQUES, M.; MIZOBE, R.; PAPA, J. Land use image classification through optimum-path forest clustering. In: *IGARSS'11*. [S.l.]: IEEE, 2011. p. 826–829.
- PORTO, E. *Webcam in Dicionário infopédia da Língua Portuguesa*. 2003. Disponível em: <<https://www.infopedia.pt/dicionarios/lingua-portuguesa/webcam>>.
- Pourramezan Fard, A.; MAHOOR, M. H. Facial landmark points detection using knowledge distillation-based neural networks. *Computer Vision and Image Understanding*, v. 215, p. 103316, 2022. ISSN 1077-3142. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1077314221001582>>.
- RITCHIE, K.; CALLERY, J.; BIRI, K. *The Art of Rigging: A definitive guide to character technical direction with Alias Maya*. [S.l.]: CG Toolkit, 2005.
- ROBISON, S. *Geometria Plana*. 2014. <<http://www.infoescola.com/geometria-plana/>>. Accessed: 2016-12-05.
- ROSA, G. H.; PAPA, J. P. Opfython: A python implementation for optimum-path forest. *Software Impacts*, p. 100113, 2021. ISSN 2665-9638.

- RUSINKIEWICZ, S.; LEVOY, M. Efficient variants of the icp algorithm. In: IEEE. *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. [S.l.], 2001. p. 145–152.
- Sagonas, C.; Tzimiropoulos, G.; Zafeiriou, S.; Pantic, M. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: *2013 IEEE International Conference on Computer Vision Workshops*. [S.l.: s.n.], 2013. p. 397–403. ISSN null.
- SAITO, S.; LI, T.; LI, H. Real-time facial segmentation and performance capture from rgb input. *ArXiv*, abs/1604.02647, 2016.
- SEOL, Y.; MA, W.-C.; LEWIS, J. P. Creating an actor-specific facial rig from performance capture. In: *Proceedings of the 2016 Symposium on Digital Production*. New York, NY, USA: Association for Computing Machinery, 2016. (DigiPro '16), p. 13–17. ISBN 9781450344296. Disponível em: <<https://doi.org/10.1145/2947688.2947693>>.
- SEOL, Y.; SEO, J.; KIM, P. H.; LEWIS, J. P.; NOH, J. Artist friendly facial animation retargeting. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 30, n. 6, p. 162:1–162:10, dez. 2011. ISSN 0730-0301. Disponível em: <<http://doi.acm.org/10.1145/2070781.2024196>>.
- SHAKIR, S. D.; AL-AZZA, A. A. Facial modelling and animation: An overview of the state-of-the art. *Iraqi Journal for Electrical & Electronic Engineering*, v. 18, n. 1, 2022.
- SILVEIRA, F. *O Que sao keyframes e Como usa-los Corretamen*. 2017. Disponível em: <<https://medium.com/uxmotiondesign/o-que-sa%CC%83o-keyframes-e-como-usa%CC%81-los-corretamente-4d108fd6aacc>>.
- SOUZA, E. *Entendendo o que é Matriz de Confusão com Python*. 2020. Disponível em: <<https://medium.com/data-hackers/entendendo-o-que-%C3%A9-matriz-de-confus%C3%A3o-com-python-114e683ec509>>.
- STATISTICS, L. B.; BREIMAN, L. Random forests. In: *Machine Learning*. [S.l.: s.n.], 2001. p. 5–32.
- STRUIJK, S. van der; HUANG, H.-H.; MIRZAEI, M. S.; NISHIDA, T. Facsvatar: An open source modular framework for real-time face based facial animation. In: *Proceedings of the 18th International Conference on Intelligent Virtual Agents*. New York, NY, USA: Association for Computing Machinery, 2018. (IVA '18), p. 159–164. ISBN 9781450360135. Disponível em: <<https://doi.org/10.1145/3267851.3267918>>.
- SUMNER, R. W.; POPOVIĆ, J. Deformation transfer for triangle meshes. *ACM Transactions on Graphics (TOG)*, ACM, v. 23, n. 3, p. 399–405, 2004.
- TIPPING, M. E.; BISHOP, C. M. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Wiley Online Library, v. 61, n. 3, p. 611–622, 1999.
- TIWARI, K.; PATEL, M. Facial expression recognition using random forest classifier. In: MATHUR, G.; SHARMA, H.; BUNDELE, M.; DEY, N.; PAPRZYCKI, M. (Ed.). *International Conference on Artificial Intelligence: Advances and Applications 2019*. Singapore: Springer Singapore, 2020. p. 121–130. ISBN 978-981-15-1059-5.

- TREVINO, A. *Introduction to K-means Clustering*. 2016. Disponível em: <<https://blogs.oracle.com/ai-and-datascience/post/introduction-to-k-means-clustering>>.
- VELUSAMY, S.; KANNAN, H.; ANAND, B.; SHARMA, A.; NAVATHE, B. A method to infer emotions from facial action units. In: *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. [S.l.: s.n.], 2011. p. 2028–2031. ISSN 1520-6149.
- WEISE, T.; BOUAZIZ, S.; LI, H.; PAULY, M. Kinect-based facial animation. In: *SIGGRAPH Asia 2011 Emerging Technologies*. New York, NY, USA: ACM, 2011. (SA '11), p. 1:1–1:1. ISBN 978-1-4503-1136-6. Disponível em: <<http://doi.acm.org/10.1145/2073370.2073371>>.
- WEISE, T.; BOUAZIZ, S.; LI, H.; PAULY, M. Realtime performance-based facial animation. In: ACM. *ACM Transactions on Graphics (TOG)*. [S.l.], 2011. v. 30, n. 4, p. 77.
- WU, C.; BRADLEY, D.; GROSS, M.; BEELE, T. An anatomically-constrained local deformation model for monocular face capture. *ACM Trans. Graph.*, Association for Computing Machinery, New York, NY, USA, v. 35, n. 4, jul 2016. ISSN 0730-0301. Disponível em: <<https://doi.org/10.1145/2897824.2925882>>.
- WU, Y.; JI, Q. Facial landmark detection: A literature survey. *International Journal of Computer Vision*, v. 127, p. 115–142, 2018.
- Yan, Y.; Lu, K.; Xue, J.; Gao, P.; Lyu, J. Feafa: A well-annotated dataset for facial expression analysis and 3d facial animation. In: *2019 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*. [S.l.: s.n.], 2019. p. 96–101. ISSN null.
- YANG, Y.; WU, X.-J.; KITTLER, J. Landmark weighting for 3dmm shape fitting. *ArXiv*, abs/1808.05399, 2018.
- ZHANG, J.; ZHOU, K.; LUXIMON, Y.; LI, P.; IFTIKHAR, H. 3d-guided facial shape clustering and analysis. *Multimedia Tools and Applications*, Springer, v. 81, n. 6, p. 8785–8806, 2022.
- ZHANG, Z. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, IEEE Computer Society Press, Washington, DC, USA, v. 19, n. 2, p. 4–10, abr. 2012. ISSN 1070-986X. Disponível em: <<https://doi.org/10.1109/MMUL.2012.24>>.
- ZHENG, Y.; AN, G. A novel facial expression recognition approach based on mosift. In: *6th International Conference on Wireless, Mobile and Multi-Media (ICWMMN 2015)*. [S.l.: s.n.], 2015. p. 223–227.
- ZHOU, M.; LIANG, L.; SUN, J.; WANG, Y. Aam based face tracking with temporal matching and face segmentation. In: IEEE. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. [S.l.], 2010. p. 701–708.

A Apêndice

A.1 Amostra da base de dados de *blendshapes*.









A Anexos

A.1 Procrustes

O método de Procrustes ¹ tem por finalidade minimizar as transformações de similaridade, através da translação, rotação e/ou escalonamento (PANCERI, 2017).

A primeira etapa é o deslocamento da amostra, de modo que, o centroide fique localizado na origem. Matematicamente, dados k pontos de 2 dimensões,

$$[(x1, y1), (x2, y2), ..., (xk, yk)] \quad (\text{A.1})$$

o centroide $(\bar{x}, \bar{y})$ pode ser obtido da seguinte forma:

$$\bar{x} = \frac{x1 + x2 + ... + xk}{k} \quad (\text{A.2})$$

$$\bar{y} = \frac{y1 + y2 + ... + yk}{k} \quad (\text{A.3})$$

A translação de cada ponto se dá através da Equação A.4,

$$[(x1, \bar{x}), (y1, \bar{y}), (x2, \bar{x}), (y2, \bar{y}), ..., (xk, \bar{x}), (yk, \bar{y})] \quad (\text{A.4})$$

Da mesma forma, a componente escalar é removida dividindo cada ponto pelo fator,

$$s = \sqrt{\frac{(x1-\bar{x})^2 + (y1-\bar{y})^2 + ... + (xk-\bar{x})^2 + (yk-\bar{y})^2}{k}} \quad (\text{A.5})$$

de modo que, o valor da raiz quadrada da média das distâncias, em relação a origem seja 1. Assim, geram-se os pontos:

$$[((x1 - \bar{x}) / s), (y1 - \bar{y}) / s], ((x2 - \bar{x}) / s), (y2 - \bar{y}) / s), ..., ((xk - \bar{x}) / s), (yk - \bar{y}) / s)] \quad (\text{A.6})$$

A remoção da componente rotacional é mais complexa, uma vez que a orientação de referência padrão nem sempre está disponível. Todavia, devido ao fato do objeto em

¹ O nome Procrustes refere-se a um criminoso da mitologia grega, que fez suas vítimas caberem em sua cama, esticando seus membros ou cortando-os.

questão estar, em geral, alinhado verticalmente, o objeto de referência pode ser definido pela média de todas as máscaras de pontos.

Uma vez fixado o objeto de referência, todas as instâncias são rotacionadas em relação à origem, de forma que seja encontrado um ângulo que minimize o quadrado da distância entre os pontos correlatos.

A.2 Distância Euclidiana

A geometria euclidiana se ocupa do estudo das formas e das ligações algébricas conectadas a elas. A geometria euclidiana (plana) se fundamenta na ideia intuitiva de ponto, sendo que, a partir dele, formam-se as ideias de retas e dos planos. As retas e os planos, nada mais são do que um conjunto de pontos, que não se limitam a um fim, ou seja, são infinitos em ambas as direções (ROBISON, 2014).

A distância euclidiana (DE) é a medida que calcula a distância entre dois pontos em um espaço vetorial, o qual, pode-se deduzir a partir do teorema de Pitágoras. Aplicando essa distância, este espaço se torna um espaço métrico.

Se $u = (x_1, y_1)$ e $v = (x_2, y_2)$ são dois pontos em um plano, a distância euclidiana é dada por:

$$d_2(x, y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (\text{A.7})$$

Geometricamente, é a distância do segmento entre u e v .

A distância euclidiana entre os pontos $P = (p_1, p_2, \dots, p_n)$ e $Q = (q_1, q_2, \dots, q_n)$ de um espaço euclidiano n -dimensional é definido como:

$$(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{n=1}^N (p_i - q_i)^2}. \quad (\text{A.8})$$

Para calcular a distância euclidiana entre dois vetores de n -dimensões x e y é definida como o escalar:

$$d_e(x, y) = \|x - y\| = \|y - x\| = \left[(x_1 - y_1)^2 + \dots + (x_n - y_n)^2 \right]^{1/2} \quad (\text{A.9})$$

.

A Equação A.9 é a Norma da diferença entre os vetores.