



UNIVERSIDADE D  
COIMBRA

**Viviana Rubina Gonçalves Abreu**

**Reconhecimento Facial - Comparação do Uso de  
Descritores Geométricos Heurísticos e Aprendizagem  
Profunda**

VOLUME 1

Dissertação no âmbito do Mestrado Integrado em Engenharia Electrotécnica e de Computadores do Ramo de Computadores orientada pelo Professor Doutor Nuno Miguel Mendonça da Silva Gonçalves e apresentada ao Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Fevereiro de 2021

# **Reconhecimento Facial - Comparação do Uso de Descritores Geométricos Heurísticos e Aprendizagem Profunda**

**Viviana Rubina Gonçalves Abreu**

VOLUME 1

Dissertação no âmbito do Mestrado Integrado em Engenharia Electrotécnica e de Computadores do Ramo de Computadores orientada pelo Professor Doutor Nuno Miguel Mendonça da Silva Gonçalves e apresentada ao Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Fevereiro de 2021

# Abstract

The human face is one of the most used biometric characteristics as a way of identification because it composes a convenient passive and non-intrusive system for verifying personal identity. From the oldest identification documents to the most recent applications applied to mobile devices, its use has become a habit in the daily lives of millions of people, being therefore an area of great interest and the subject of constant evolution. A system based on the biometry of the face refers to the techniques used for extraction, selection and classification of characteristics through human face images. That said, this dissertation aims to study and compare two different techniques, one dedicated to a more traditional approach and the other to a modern perspective linked to pure deep learning, both applied to the context of facial recognition.

The first method emerged from the adaptation of an authentication system, created as part of a project developed between the Institute of Systems and Robotics (ISR) of the University of Coimbra with the Portuguese Mint and Official Printing Office, called UniqueMark, where it was developed a mechanism capable of representing a set of points based on individual characteristics or relationships between them, based on global geometric characteristics which, due to the good results obtained, made it interesting to carry out its analysis in other contexts, highlighting the problems of biometrics. This algorithm then consists of a system that receives as input a set of keypoints belonging to certain structured dispersions obtained by detecting the facial landmarks of frontal images of faces, which through the study of heuristic geometric relationships between the detected points returns a histogram with information analyzed between the different points belonging to the same pattern. This histogram is later used in the validation process using the KNN and SVM classifiers.

Following the current trends, the second method under study focuses on an approach associated with deep learning, which uses artificial neural networks, more specifically convolutional networks, in which the deep layers of the complex networks function as extractors of sets of characteristics, therefore having the ability to learn directly from the input images. Motivated by low computational complexity, were made attempts to optimize the network architecture chosen for this purpose, MobieleNetV2, through analysis of the trade-off between the number of features and the architecture width, as well as different loss functions (Softmax, CosFace, Sphereface and ArcFace).

Therefore, the objective of this dissertation is to understand the drawbacks of the most classic methods in facial recognition, moving to a more recent approach that includes the testing and optimization of models, with special attention to the consumption of computational resources.

**Keywords:** Biometric system, Facial recognition, Machine Learning, Facial landmarks, Point descriptor, Deep Learning, Convolutional networks, Optimization.

# Resumo

A face humana é uma das características biométricas mais utilizadas como forma de identificação por constituir um sistema passivo e não intrusivo, conveniente para verificação de identidade pessoal. Desde os mais antigos documentos de identificação às mais recentes aplicações interligadas com dispositivos móveis, a sua utilização tornou-se um hábito no quotidiano de milhões de pessoas, sendo por isso uma área de grande interesse e alvo de constante evolução. Um sistema baseado na biometria da face refere-se a técnicas utilizadas para extracção, selecção e classificação de características obtidas através de imagens de faces humanas. Posto isto, esta dissertação tem como objectivo o estudo e comparação de duas técnicas distintas, sendo uma dedicada a uma abordagem mais tradicional, e outra a uma perspectiva mais moderna ligada à aprendizagem profunda, ambas aplicadas ao contexto de reconhecimento facial.

O primeiro método surgiu da adaptação de um sistema de autenticação, criado no âmbito de um projecto desenvolvido pelo Instituto de Sistemas e Robótica (ISR) da Universidade de Coimbra juntamente com a Imprensa Nacional-Casa da Moeda (INCM), o UniqueMark, onde foi desenvolvido um mecanismo capaz de representar um conjunto de pontos a partir de características individuais ou relações entre eles, baseado em características geométricas globais que devido aos bons resultados obtidos tornou interessante fazer a sua análise noutros contextos, destacando-se os problemas de biometria. Este algoritmo consiste num sistema que recebe como entrada os pontos pertencentes a dispersões estruturadas obtidas por meio da detecção dos pontos salientes da face de imagens frontais, que através do estudo das relações geométricas heurísticas entre os pontos detectados devolve um histograma com as informações resultantes da análise dos diversos pontos pertencentes ao mesmo padrão. Este histograma é posteriormente utilizado no processo de validação com recurso aos classificadores KNN e SVM.

Seguindo as tendências actuais, o segundo método sob estudo foca-se numa abordagem ligada à aprendizagem profunda que recorre à utilização de redes neuronais artificiais, mais especificamente redes convolucionais, em que as camadas profundas das redes complexas funcionam como extractores de conjuntos de características, possuindo por isso a capacidade de aprender directamente a partir das imagens de entrada. Tendo como motivação o baixo consumo computacional foram feitas tentativas de optimizações da arquitectura de rede escolhida para o efeito, MobileNetV2, através da análise da relação de compromisso entre o número de *features* e a arquitectura, assim como de diferentes funções de perda (Softmax, CosFace, Sphreface e ArcFace).

Assim sendo, o propósito desta dissertação visa compreender as inconvenientes dos métodos mais clássicos em reconhecimento facial, passando para uma abordagem mais recente que inclui a testagem e optimização de modelos, com especial atenção ao consumo de recursos computacionais.

**Keywords:** Sistema biométrico, Reconhecimento facial, Aprendizagem de máquina, Pontos salientes da face, Descritor de pontos, Aprendizagem profunda, Redes convolucionais, Optimização

# Agradecimentos

Primeiramente, deixo um agradecimento especial ao meu orientador pelo incentivo e conhecimento que me transmitiu ao longo de todo o projecto e por todo o tempo que disponibilizou, ajudando-me assim a concluir da melhor forma esta fase do meu percurso académico. Queria agradecer também a toda a equipa do Visteam do ISR que estiveram presentes durante toda a concepção desta dissertação, deixando um particular obrigada ao Iurii Medvedev que se mostrou disponível para me esclarecer e auxiliar sempre que lhe era possível.

Um muito obrigada a toda a minha família, por me proporcionar todas as oportunidades que me trouxeram até aqui, podendo eu sempre contar com o seu apoio incondicional.

Não podia deixar de mencionar os meus amigos, desde os mais antigos aos mais recentes, que me acompanharam nesta jornada e me ajudaram a ultrapassar todos os obstáculos que foram surgindo, e com os quais espero poder continuar a contar para as oportunidades que se avizinham.

Por fim, queria deixar o meu reconhecimento a todos os professores, colegas, conhecidos e a todos os que de uma maneira ou de outra moldaram o meu percurso e me ajudaram a atingir este grande objectivo.

Viviana Abreu

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização . . . . .	1
1.2	Motivação . . . . .	2
1.3	Objectivo . . . . .	2
1.4	Contribuições . . . . .	3
1.5	Organização do Documento . . . . .	3
<b>2</b>	<b>Fundamentos de processamento de imagem e machine learning</b>	<b>4</b>
2.1	Características das imagens . . . . .	4
2.1.1	Descritores geométricos de pontos . . . . .	5
2.1.2	Histogramas . . . . .	5
2.2	Classificadores de machine learning . . . . .	6
2.2.1	K-Nearest Neighbors . . . . .	6
2.2.2	Support Vector Machine . . . . .	6
2.3	Redes neuronais artificiais . . . . .	6
2.4	Redes perceptron multi-camadas . . . . .	8
2.5	Redes neuronais convolucionais . . . . .	8
<b>3</b>	<b>Estruturas de Dados para Reconhecimento de Padrões</b>	<b>11</b>
3.1	Funções Físicas não clonáveis . . . . .	11
3.2	Autenticação de padrões aleatórios . . . . .	12
3.2.1	1º Nível de autenticação . . . . .	12
3.2.2	2ª Nível de autenticação . . . . .	15
3.2.3	3º Nível de autenticação . . . . .	15
3.3	mIDoT . . . . .	16
3.4	UniqueMark . . . . .	16
<b>4</b>	<b>Reconhecimento Facial</b>	<b>18</b>
4.1	Introdução ao reconhecimento facial . . . . .	18
4.2	<i>Deep Learning</i> aplicado a Reconhecimento facial . . . . .	19
<b>5</b>	<b>1ª Parte - Método clássico de Machine Learning</b>	<b>22</b>
5.1	Metodologia . . . . .	22
5.1.1	Estrutura de avaliação . . . . .	22
5.1.2	Extracção de pontos . . . . .	23
5.1.3	Alinhamento . . . . .	24
5.1.4	Descritor geométrico de pontos - UniqueMark . . . . .	25
5.1.5	Aplicação do descritor em padrões de diferentes naturezas . . . . .	26

5.1.6	Implementação de um algoritmo com a mesma metodologia para verificação e comparação de resultados . . . . .	28
5.1.7	Adição de um sistema de <i>scrambling</i> . . . . .	28
5.2	Trabalho Experimental . . . . .	29
5.2.1	Dados . . . . .	29
5.2.2	Métricas . . . . .	30
5.2.3	Descrição Experimental . . . . .	31
5.2.4	Detalhes da implementação . . . . .	33
5.2.5	Resultados e sua análise . . . . .	33
<b>6</b>	<b>2ª Parte - Método <i>Pure Deep Learning</i></b>	<b>38</b>
6.1	Metodologia . . . . .	38
6.1.1	Estrutura de avaliação . . . . .	38
6.1.2	Filtragem e pre-processamento das imagens . . . . .	38
6.1.3	MobileNetV2 . . . . .	39
6.1.4	Optimização de parâmetros . . . . .	41
6.1.5	Treino . . . . .	43
6.1.6	Teste . . . . .	43
6.2	Trabalho Experimental . . . . .	44
6.2.1	Dados . . . . .	44
6.2.2	Métricas . . . . .	45
6.2.3	Detalhes da implementação . . . . .	45
6.3	Descrição Experimental . . . . .	46
6.3.1	Resultados e sua análise . . . . .	46
6.4	Comparação entre o método clássico de <i>machine learning</i> e o método de <i>pure deep learning</i> . . . . .	51
<b>7</b>	<b>Conclusões e Futuro Trabalho</b>	<b>53</b>
7.1	Considerações finais . . . . .	53
7.2	Trabalho futuro . . . . .	54
	<b>Referências</b>	<b>55</b>

# Lista de Figuras

2.1	Exemplo de imagens cuja cena contém dispersões de pontos que lhe são característicos. . . . .	5
2.2	Modelo não linear de um neurónio. . . . .	7
2.3	Rede perceptron multi-camadas com duas camadas ocultas. . . . .	9
3.1	Adaptação do exemplo do processo de produção de uma chave PUF presente em [4].	12
3.2	Padrão de fibras aleatório usado em autenticação de documentos, exemplo real (a), imagem sintética (b) [76]. . . . .	13
3.3	Ilustração da correspondência do primeiro ponto [9]. . . . .	14
3.4	Ilustração da correspondência do segundo ponto [9]. . . . .	14
3.5	Ilustração da correspondência do terceiro ponto [9]. . . . .	15
3.6	Adaptação do esboço da correspondência entre imagens com os algoritmos ORB e RANSAC presente em [41]. . . . .	17
3.7	Adaptação da figura referente aos processos de validação e registo do Unique-Mark, presente em [18]. . . . .	17
5.1	Exemplo da divisão dos dados para a validação cruzada entre três subconjuntos e iterações. . . . .	23
5.2	Imagem exemplo da extração das <i>facial landmarks</i> . . . . .	23
5.3	Exemplo ilustrativo do processo de alinhamento. . . . .	24
5.4	Características geométricas globais. . . . .	26
5.5	Características geométricas locais. . . . .	26
5.6	Divisão das características geométricas por quadrantes. . . . .	27
5.7	Sistema de avaliação da performance do descritor geométrico de pontos quando aplicado em padrões aleatórios sintéticos. . . . .	27
5.8	Sistema de avaliação da performance do descritor geométrico de pontos quando aplicado em imagens reais de faces. . . . .	28
5.9	Figura exemplificativa do pipeline do sistema de <i>scrambling</i> . . . . .	29
5.10	Conjunto exemplificativo do tipo de imagens encontradas no primeiro conjunto [17].	30
5.11	Conjunto exemplificativo do tipo de imagens encontradas no segundo conjunto [79].	30
5.12	Exemplo do resultado das transformações geométricas aplicadas a um padrão aleatório sintético. . . . .	34
6.1	Exemplo de imagem resultante do processo de filtragem. O antes e depois do pré-processamento. . . . .	39
6.2	Blocos de construção característicos da arquitectura MobileNetV2 [68]. . . . .	40
6.3	Arquitectura da rede utilizada. . . . .	42

6.4	Amostra de imagens presentes no <i>dataset</i> VGGFace2 [10], utilizado para treino e testagem das redes neuronais artificiais após a aplicação do processo de filtragem descrito na secção 6.1.2. . . . .	45
6.5	Curvas ROC referentes a alpha igual a 0.35. . . . .	46
6.6	Curvas ROC referentes a alpha igual a 1.0. . . . .	47
6.7	Curvas ROC referentes a alpha igual a 1.4. . . . .	47
6.8	Curvas ROC referentes ao número de <i>features</i> igual a 16. . . . .	47
6.9	Curvas ROC referentes ao número de <i>features</i> igual a 32. . . . .	48
6.10	Curvas ROC referentes ao número de <i>features</i> igual a 64. . . . .	48
6.11	Curvas ROC referentes ao número de <i>features</i> igual a 128. . . . .	48
6.12	Curvas ROC referentes ao número de <i>features</i> igual a 256. . . . .	49
6.13	Curvas ROC referentes ao número de <i>features</i> igual a 512. . . . .	49
6.14	Curvas ROC resultantes das aplicações de diferentes funções de perda. . . . .	50
6.15	Curva ROC referente à rede otimizada (alpha = 0.35, <i>features</i> = 32, loss function = SphereFace), segundo o protocolo 1. . . . .	52

# Lista de Tabelas

5.1	Exemplo do funcionamento do sistema <i>scrambler</i> desenvolvido. . . . .	29
5.2	Parâmetros avaliados na etapa de otimização dos classificadores. . . . .	33
5.3	Resultados dos testes descritor geométrico de pontos com padrões sintéticos de dispersões aleatórias. . . . .	34
5.4	Resultados dos testes do descritor geométrico de pontos aplicado a dispersões estruturados obtidas das imagens de faces em que o <i>dataset</i> de validação é constituído por 240 imagens, com 8 imagens distribuídas de forma uniforme por 30 identidades, pertencentes à 1ª Base de dados ( <i>2D face sets Aberdeen</i> ). . . . .	34
5.5	Resultados dos testes do descritor geométrico de pontos aplicado a dispersões estruturados obtidas das imagens de faces em que o <i>Dataset</i> de avaliação é constituído por 342 imagens, com 19 imagens distribuídas de forma uniforme por 18 identidades, pertencentes à 2ª Base de dados ( <i>Frontal face dataset coletado por Markus Weber</i> ). . . . .	35
5.6	Resultados do teste do método de Brzakovic utilizando dispersões aleatórias obtidas de padrões gerados sinteticamente. Foram considerados 30 padrões gerados aleatoriamente, cada qual com 8 correspondências geradas através de transformações geométricas dos padrões originais, perfazendo um total de 240 padrões. . . . .	36
5.7	Resultados do teste do método de Brzakovic utilizando dispersões estruturados obtidas pelo <i>dataset</i> constituído por 240, com 8 imagens distribuídas de forma uniforme por 30 identidades, pertencentes à 1ª Base de dados ( <i>2D face sets Aberdeen</i> ). . . . .	36
5.8	Resultados do teste do método de Brzakovic utilizando dispersões estruturados obtidas pelo <i>dataset</i> constituído por 342, com 19 imagens distribuídas de forma uniforme por 18 identidades, pertencentes à 2ª Base de dados ( <i>Frontal face dataset coletado por Markus Weber</i> ). . . . .	36
5.9	Resultados dos testes do sistema de <i>scrambling</i> aplicado em dispersões estruturados resultantes do <i>Dataset</i> de teste constituído por 270 com 19 imagens distribuídas de forma uniforme por 18 identidades, pertencentes à 2ª Base de dados ( <i>2D face sets Aberdeen</i> ) . . . . .	37
6.1	<i>Bottleneck residual block</i> - Considerando $h \times w \times k$ como as dimensões de entrada de cada camada, é possível observar a estrutura específica de um bloco de <i>bottleneck</i> residual, que transforma $k$ para $k'$ canais, com <i>stride</i> , $s$ , e factor de expansão, $t$ [68]. . . . .	41
6.2	Arquitetura geral da MobileNet [68]. . . . .	41
6.3	Parâmetros de treino comuns a todas as redes. . . . .	43
6.4	Resultados dos testes da rede otimizada utilizando os datasets seleccionados para o primeiro método 5.2.1 . . . . .	51

# Acrónimos

AAML	Additive Angular Margin Los
AMM	Active Appearance Model
ANN	Artificial Neural Network
API	Application Programming Interface
CMOS	Complementary Metal–Oxide–Semiconductor
CNN	Convolutional Neural Network
DHE	Deep Hypersphere Embeddin
EQM	Erro Quadrático Médio
FBNN	Feed Backward Neural Network
FFNN	Feed Forward Neural Network
HOG	Histogram of Oriented Gradients
ICP	Iterative Closest Point
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
INCM	Imprensa Nacional-Casa da Moeda
IoT	Internet of Things
ISR	Instituto de Sistemas e Robótica
KNN	K-Nearest Neighbor
LBP	Local Binary Pattern
LMCL	Large Margin Cosine Loss
LMCL	Large Margin Cosine Loss
ORB	Oriented FAST and Rotated BRIEF
PCA	Análise de Componentes Principais
PUF	Physical Unclonable Functions
RANSAC	RANdom SAmples Consensus
ReLU	Rectified Linear Unit
RFID	Radio-Frequency IDentification
ROC	Receiver Operating Characteristic
SIFT	Scale-Invariant Feature Transform
SRAM	Static Random-Access Memory
SURF	Speeded Up Robust Feature
SVM	Support Vector Machin

# Capítulo 1

## Introdução

### 1.1 Contextualização

Sistemas de computação visual são muito requisitados para a detecção e reconhecimento de padrões e objectos. Estes temas têm levantado interesse por parte da comunidade científica devido ao aumento da procura de sistemas de autenticação, reconhecimento de padrões, até mesmo sistemas de anti-contrafacção [16]. No âmbito da crescente necessidade de garantir a segurança de artefactos valiosos, o Instituto de Sistemas e Robótica (ISR) da Universidade de Coimbra juntamente com a Imprensa Nacional-Casa da Moeda (INCM), desenvolveram um projecto apelidado de UniqueMark [18]. Este projecto consistiu na criação de um sistema capaz de adicionar a um metal precioso uma marca única, não clonável e irreproduzível. Para gerar esta marca era criada uma dispersão de partículas de pó de diamante que por puncinamento era incrustada no artefacto, produzindo assim um padrão aleatório de partículas em que a sua verificação se baseava na análise das posições dos pontos que a constituíam. Posto isto, e de modo a garantir a autenticação da marca gerada, foi desenvolvido um descritor de pontos, mecanismo capaz de representar um conjunto de pontos a partir de características individuais ou relações entre eles, baseado em características geométricas globais que, por apresentar bons resultados neste âmbito da autenticação de artefactos de metais preciosos, tornou interessante fazer a sua análise noutros contextos, destacando-se os problemas de biometria.

Biometria é um método muito utilizado em reconhecimento de identidades humanas baseado na morfologia, comportamento e características orgânicas como impressões digitais, íris, faces, retinas, palmas e geometria da mão, entre outras [60]. Em particular, as faces tornaram-se numa área de investigação de relevo devido à sua ampla gama de aplicações nos mais variados contextos, que vão desde o controlo de acessos, aos relógios de ponto, até aos mais diversos dispositivos, entre os quais os móveis. Reconhecimento facial tem a vantagem de ser um sistema passivo, não intrusivo para verificação de identidade pessoal [50]. Este refere-se a técnicas utilizadas para extracção, selecção e classificação de características obtidas através imagens de faces humanas [60] e dividem-se de uma forma geral em duas abordagens: as técnicas clássicas e as técnicas mais modernas.

Os métodos clássicos normalmente baseiam-se na análise das relações geométricas presentes em características manualmente desenhadas, as *handcrafted features*, que posteriormente são avaliadas por um algoritmo de classificação. O sistema de autenticação acima descrito é um exemplo deste tipo de abordagem. Em contrapartida, os sistemas baseados em métodos modernos, relacionados com *deep learning*, muitas vezes associados a redes neuronais convolucionais, usam as camadas profundas das redes complexas como extractores de conjuntos de características [59], tendo por isso a capacidade de aprender directamente a partir das imagens de entrada, obtendo assim uma boa performance quando aplicado em reconhecimento de objectos.

## 1.2 Motivação

O uso de traços biométricos, tais como impressões digitais, íris e faces deixaram de ser específicos de ambientes de alta segurança e passaram a fazer parte da vida quotidiana de grande parte da população, estando presentes em sistemas de controlo de acessos, diferentes processos de identificação, até mesmo em mecanismos de desbloqueio de simples dispositivos como é o caso dos telemóveis [43]. Um sistema biométrico é essencialmente um sistema de reconhecimento de padrões que opera através da obtenção de informações biométricas de um indivíduo, ao extrair um conjunto de características dos dados adquiridos, e ao compará-lo com o conjunto correspondente presente na base de dados [42]. Em particular, a biometria da face suscitou desde cedo interesse por parte da comunidade científica pela sua vasta gama de aplicações, sendo que actualmente o reconhecimento da face encontra-se entre os problemas mais amplamente estudados em visão por computadores [60]. Ao longo dos anos foram desenvolvidas as mais diversas técnicas para solucionar o problema de reconhecimento facial, que se distribuem entre soluções baseadas em métodos tradicionais que incluem diversas técnicas de *machine learning* que se aproveitam de conjuntos de características de imagens em duas dimensões, até às mais actuais, normalmente relacionadas com redes neuronais artificiais, adoptadas na tentativa de aumentar a robustez dos sistemas de reconhecimento.

Apesar das vantagens individuais de cada uma das abordagens nenhuma é isenta de inconvenientes. Porém, as abordagens mais modernas têm conseguido obter melhores resultados, demonstrados pelas altas precisões obtidas numa maior gama de ambientes e contextos, o que vem a dificultar a competição para as técnicas mais clássicas. Contudo, a dependência em grande poder computacional e a necessidade de uma grande quantidade de parâmetros de modelos [11], abrem espaço para que ambas as perspectivas possam crescer e encontrar soluções capazes de ultrapassar os obstáculos conhecidos, contribuindo individualmente para o crescimento do conhecimento científico nesta área.

## 1.3 Objectivo

O principal objectivo desta dissertação passa por fazer um estudo no campo *machine learning* a respeito de duas técnicas distintas, sendo uma dedicada a uma abordagem mais tradicional e

outra a uma perspectiva mais moderna ligada ao *pure deep learning*, ambas aplicadas em contexto de reconhecimento facial.

A primeira parte tem como propósito a aplicação e aprimoramento do algoritmo desenvolvido no projecto UniqueMark [18], tendo por base dispersões estruturadas obtidas por meio da detecção dos *facial landmarks* de imagens frontais de faces, através do estudo das relações geométricas heurísticas entre os pontos detectados. Em contraste, a segunda parte foca-se na utilização de redes neuronais artificiais, no mesmo contexto, com a intenção de estudo sobre relação de compromisso entre o número de *features* e a arquitectura, assim como, optimização de outros parâmetros. Por conseguinte, o propósito desta dissertação visa compreender os inconvenientes dos métodos mais clássicos em reconhecimento facial, passando para uma abordagem mais recente que inclui a testagem e optimização de modelos, com especial atenção ao consumo de recursos computacionais.

## 1.4 Contribuições

Das diferentes metodologias desenvolvidas ao longo desta dissertação, surgiram algumas contribuições consideráveis, entre as quais:

- Análise e adaptação de um método clássico de classificação de objectos (padrões) para contexto de biometria, reconhecimento das faces.
- Comparação entre um método tradicional de *machine learning* e um método mais moderno de *pure deep learning*, ambos aplicados em reconhecimento facial.
- Estudo acerca da arquitectura da rede convolucional MobileNetV2 [68] relativamente à relação de compromisso entre a largura da arquitectura e número de *features*.
- Optimização de uma rede segundo os parâmetros anteriores com a incorporação de um estudo sobre três funções de perda distintas, tendo em consideração o objecto de estudo (as faces) e a aplicabilidade em dispositivos de menor capacidade computacional.

## 1.5 Organização do Documento

Este primeiro capítulo condensa a contextualização e motivação desta dissertação, seguida dos objectivos, contribuições e presente secção que descreve a estrutura do documento. O capítulo seguinte denota os fundamentos de processamento de imagem e *machine learning* essenciais para a compreensão do trabalho experimental desenvolvido ao longo do documento (Capítulo 2). O capítulo que se segue apresenta algumas das estruturas de dados para reconhecimento de padrões relevantes para esta dissertação (Capítulo 3), seguido por um capítulo que faz uma introdução ao reconhecimento facial (Capítulo 4). Seguidamente, a dissertação divide-se em duas partes, a primeira que diz respeito a um método mais clássico aplicado em contexto de reconhecimento facial, e uma segunda parte dedicada a um método moderno aplicado também no mesmo contexto, e onde são explicadas as metodologias relacionadas com os trabalhos experimentais, assim como, os resultados obtidos respectivos a cada uma das partes, (Capítulo 5 e Capítulo 6). Por fim, o documento termina com as conclusões e propostas de trabalho a desenvolver no futuro (Capítulo 7).

## Capítulo 2

# Fundamentos de processamento de imagem e machine learning

### 2.1 Características das imagens

Muitos métodos de classificação, especialmente os mais convencionais, não utilizam diretamente imagens como entrada, mas sim uma representação das mesmas [63]. Estes métodos necessitam de um passo adicional capaz de extrair e representar as características específicas de cada uma das imagens, dando destaque aos atributos/propriedades que irão ser estudadas.

Assim sendo, grande parte da literatura de computação visual foca-se em descobrir, perceber, caracterizar e melhorar as características que podem ser extraídas das imagens. Existem duas abordagens muito utilizadas. A primeira diz respeito a características que são manualmente desenhadas, em inglês utiliza-se a expressão "*handcrafted*". Estas têm o objectivo de ultrapassar problemas específicos como oclusão e variações na escala de iluminação. Normalmente o desenho deste tipo de características baseia-se na relação de compromisso entre a precisão e a eficiência computacional, sendo um dos exemplos mais conhecidos, o SIFT e o SURF [59], brevemente introduzidos na secção 2.1.1.1 e 2.1.1.2, respectivamente. Em segundo lugar, sendo esta uma abordagem mais moderna, temos as características obtidas através da utilização de redes neuronais, onde as camadas profundas deste tipo de redes complexas agem como extractores de conjuntos de características que normalmente são consideradas genéricas (até certo ponto) e independentes de qualquer tarefa de classificação específica [59].

Na primeira parte desta dissertação, dedicada a um método clássico, o interesse é por imagens cuja cena contenha uma dispersão de pontos característica, particularmente em dispersões de pontos que podem ser de natureza caótica, como é o caso de dispersões formadas por punção de partículas de diamante em metais preciosos, dispersões formadas por impressões com tintas com glitter, ou estruturada, como é o caso das imagens de faces, ou até mesmo de íris e impressões digitais. Através da utilização de um descritor geométrico de pontos obtêm-se as informações acerca das relações geométricas dos pontos que as constituem, que depois de processadas são armazenadas em histogramas para serem posteriormente utilizados pelos métodos de classificação.

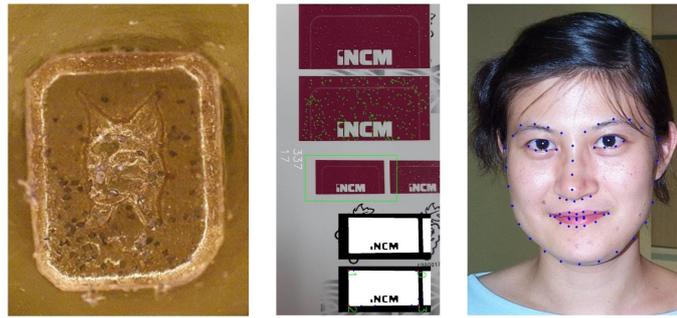


Figura 2.1: Exemplo de imagens cuja cena contém dispersões de pontos que lhe são característicos.

### 2.1.1 Descritores geométricos de pontos

Um descritor é um conjunto de dados estruturados, tipicamente unidimensional, que contém uma descrição de características chave de um elemento a ser descrito, sendo que no contexto desta dissertação esse elemento consiste numa imagem ou em apenas uma área de uma imagem.

Nesta dissertação irá ser explorado um descritor geométrico de pontos. Este descritor resume-se a um mecanismo de representação de um conjunto de pontos a partir das características *hand-crafted* desse mesmo conjunto, que podem ser globais ou locais.

Os descritores mais amplamente conhecidos no estado da arte são:

#### 2.1.1.1 SIFT

SIFT [55], Scale Invariant Feature Transform, é um descritor que usa uma classe de características de imagem locais. Estas características são invariantes à escala da imagem, translação, rotação e parcialmente invariantes a mudanças de iluminação.

#### 2.1.1.2 SURF

SURF [5], sigla de Speeded Up Robust Feature, é um detector e descritor invariante em escala e rotação que é obtido através da utilização de imagens integrais na convolução da imagem. SURF é baseado na simplificação dos melhores descritores e detectores precedentes a este, por isso apresenta uma melhor performance em relação a aspectos como a repetibilidade, distinção, robustez e rapidez no que diz respeito ao cálculo e comparação.

### 2.1.2 Histogramas

No contexto em estudo, os histogramas serão utilizados como formas de representar a informação obtida pelo descritor geométrico de pontos segundo as relações geométricas entre os pontos das dispersões consideradas para o efeito. Estes histogramas serão depois utilizados pelos classificadores que consoante a sua avaliação, procederão à sua classificação.

## 2.2 Classificadores de machine learning

Classificação de objectos é uma área de investigação relevante e de inúmeras aplicações práticas numa variedade de campos, incluindo reconhecimento de padrões e inteligência artificial, estatística, psicologia cognitiva, análise de visão e medicina [46]. Esta procura faz com que haja um maior cuidado no desenvolvimento/selecção dos classificadores tendo em conta as necessidades individuais e características do objecto sob estudo. Posto isto, os algoritmos seleccionados para o processo de classificação do método clássico foram:

### 2.2.1 K-Nearest Neighbors

K-Nearest Neighbors (KNN)[23] é um classificador não paramétrico bem conhecido, amplamente usado em reconhecimento de padrões [31]. Este algoritmo fornece um procedimento não paramétrico simples, utilizado na atribuição de uma classe a um padrão de entrada com base nas classes pertencentes aos  $K$ -vizinhos mais próximos desse vector [46].

A ideia é que sejam usados grandes quantidades de dados durante o treino, onde cada ponto de dados é caracterizado por um conjunto de variáveis. Conceptualmente, cada ponto é desenhado num espaço de alta dimensão, onde cada eixo no espaço corresponde a uma variável individual. Assim, quando temos um novo ponto de dados, queremos descobrir os  $K$  vizinhos que se situam mais perto [58], e deste modo proceder à sua respectiva classificação.

### 2.2.2 Support Vector Machine

Support Vector Machine [15] é uma técnica clássica de *machine learning* que continua a ser muito utilizada em problemas de classificação [72]. É utilizada com sucesso em várias aplicações, seja na separação entre duas classes ou em problemas multi-classe [32].

O principal objectivo ao utilizar esta técnica é encontrar um critério de classificação, por exemplo uma função de activação, capaz de separar devidamente dados nunca antes vistos com uma boa habilidade de generalização na fase de testes [28].

Este método consiste em mapear os pontos de entrada para um espaço de características de dimensão superior, para posteriormente encontrar um hiperplano que maximiza a margem entre as duas classes e as separa neste espaço [53].

## 2.3 Redes neuronais artificiais

Redes neuronais artificiais referem-se a sistemas computacionais cujo tema central é emprestado da analogia das redes neuronais biológicas [57]. Tal como o seu nome indica, são redes computacionais que tentam simular, de um grosso modo, o processo de decisão presente nas redes de células nervosas do sistema nervoso central biológico [33], sendo estas consideradas um método útil e inovador quando aplicado à resolução de problemas de classificação [2].

O modelo *perceptron* que é considerado um dos modelos mais simples, como também possivelmente uma das primeiras redes neuronais a ser alguma vez introduzidas. Esta é composta por apenas um neurónio ou uma única camada de neurónios. Um neurónio é composto por três elementos básicos como demonstra a figura 2.2, e pode ser representado pela equação matemática 2.1.

$$y = f\left(\sum_{i=1}^n w_i x_i + \theta\right) \quad (2.1)$$

É a unidade de processamento de informação fundamental para as operações das redes neuronais e é constituído por [34]:

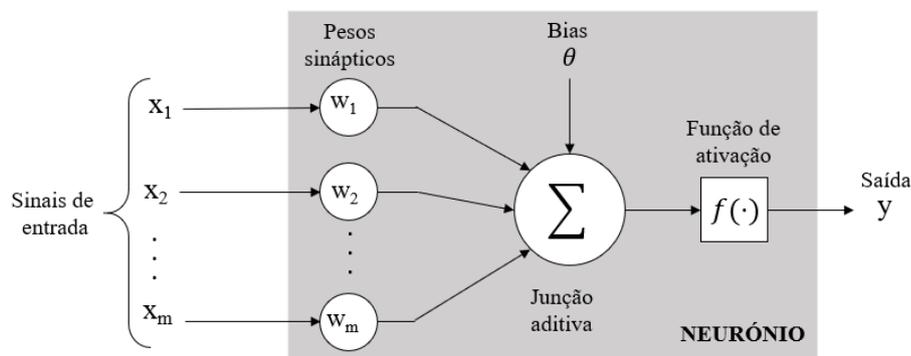


Figura 2.2: Modelo não linear de um neurónio.

1. **Conjunto de sinapses ou elos de conexão** - Cada sinapse é caracterizada individualmente por um peso. Assim, cada sinal de entrada ( $x_i$ ) é multiplicado pelo peso sináptico ( $w_i$ ) da sinapse correspondente.
2. **Somador** - Responsável por somar todos os sinais de entrada ponderados pelas suas respectivas sinapses.
3. **Função de activação** - Esta fica encarregue de restringir a amplitude da saída do neurónio. Tipicamente, o intervalo normalizado da amplitude da saída de um neurónio é escrito como o intervalo unitário fechado  $[0, 1]$ , ou alternativamente  $[-1, 1]$ . Existem vários tipos de funções de activação, sendo as mais utilizadas:
  - **Função sigmóide** - Garante que a saída se situe dentro do intervalo  $[0,1]$  e foi durante muito tempo umas das funções mais utilizadas. No entanto, estas estão ligadas a problemas relacionados com o desvanecimento do gradiente. Isto ocorre porque a sua derivada satura a partir de certo valor, o que faz com valores superiores a esse limite tendam sempre para zero, o que tem consequências negativas no processo de treino da rede.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$

- **Tangente hiperbólica** - Tal como a função descrita anteriormente, também esta na origem de problemas relacionados com o desvanecimento do gradiente. Porém, para valores de derivada superiores em comparação aos verificados na função anterior, sendo por isso uma alternativa mais atractiva.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.3)$$

- **ReLU (Rectified Linear Unit)** - Esta função de activação tem em conta os problemas de não-linearidade característicos das funções sigmoidais, sendo por isso um tipo de activação mais eficiente. Esta função é activada quando o sinal de entrada é positivo e quando a derivada parcial é igual a 1. No entanto, esta alternativa tem alguns problemas relacionados com facto do gradiente ser sempre zero quando o sinal de entrada não é positivo. Isto faz com que alguns neurónios nunca venham a ser activados, o que provoca problemas similares ao desvanecimento do gradiente.

$$f(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (2.4)$$

- **Leaky ReLU** - Esta função apresenta-se como uma solução ao problema anterior sendo uma variação da mesma, onde contrariamente à anterior, um sinal de entrada não positivo resulta num gradiente diferente de zero, sendo  $\alpha$  a inclinação negativa da função, geralmente uma constante pequena.

$$f(x) = \begin{cases} x & x > 0 \\ \alpha x & x \leq 0 \end{cases} \quad (2.5)$$

Este modelo também inclui um parâmetro chamado *bias*, que tem como efeito o aumento ou diminuição da entrada líquida da função de activação, dependendo se este é positivo ou negativo.

## 2.4 Redes perceptron multi-camadas

Mais tarde surgiram as redes perceptron multi-camadas, que contrariamente à rede descrita anteriormente, possui neurónios ou camadas de neurónios que não estão directamente conectadas com a saída, conhecidas como camadas ocultas (figura 2.3), sendo esta a primeira abordagem para resolver problemas de separação não-linear, capaz de atingir um alto nível de complexidade [2].

## 2.5 Redes neuronais convolucionais

Redes neuronais convolucionais, também conhecidas como CNNs, do inglês *Convolutional Neural Networks*, são uma classe especial das redes neuronais utilizadas no processamento de dados que possuem uma tipologia do tipo grelha (*grid-like topology*), o que as torna apropriadas

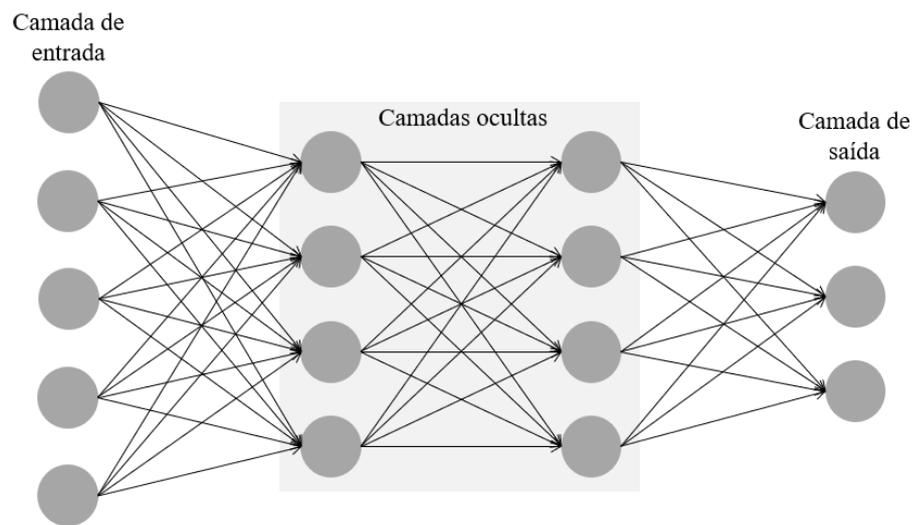


Figura 2.3: Rede perceptron multi-camadas com duas camadas ocultas.

para o processamento de imagens uma vez que recebem matrizes como parâmetro de entrada, contrariamente aos métodos mais clássicos de *machine learning* que recebem vetores. Estas redes têm sido extremamente bem-sucedidas em aplicações práticas e tal como o nome indica são redes que empregam a operação matemática de convolução, que é um caso especial das operações lineares [2]. Tal como na secção anterior (secção 2.4) estas são compostas por camadas de entrada e saída, assim como, camadas ocultas que se dividem em:

- **Camada de convolução** - esta camada é composta por um grupo de filtros, também conhecido como *kernels*, que são convolucionados com os dados de entrada, e que retornam um mapa de características. É esta a camada que concede às redes artificiais a capacidade de aprendizagem acerca dos padrões locais e a capacidade de reconhecer as características anteriormente aprendidas.
- **Camada de agrupamento (*pooling*)** - esta camada é responsável por substituir a saída num determinado local da rede, normalmente a saída resultante da camada convolucional, por estatísticas resumidas dos valores presentes em regiões igualmente distribuídas com intenção de reduzir o espaço das representações, tornando-as mais compactas e invariantes a modificações. Utiliza para isso funções do tipo *pooling*, como é o caso da operação de *pooling* máximo que guarda os valores máximos de uma determinada vizinhança rectangular. Outras funções de agrupamento habituais incluem a média de uma vizinhança rectangular, a norma  $L_2$  de uma vizinhança rectangular, ou alternativamente, a média ponderada baseada na distância do pixel central [30].
- **Camada totalmente conectada** - tal como o próprio nome indica, interconecta todos os neurónios de uma determinada camada com todos os neurónios da camada seguinte.

Existem infinitas possibilidades no que diz respeito à organização e agrupamento destes diferentes tipos de camadas e, é esta disposição que define e caracteriza as redes conferindo-lhe as

propriedades desejadas. No entanto, regra geral as camadas de convolução ficam responsáveis pela etapa de extracção de características das imagens, e as camadas totalmente conectadas ficam encarregues da classificação das mesmas. Também é prática comum que cada camada seja seguida de uma função de activação não-linear, como acontece nas redes artificiais mais clássicas.

Apesar deste tipo de redes apresentar excelentes resultados em áreas como classificação de imagens, detecção de objectos e reconhecimento facial, estas não estão livres de inconvenientes como é o caso do *overfitting*. *Overfitting* ocorre quando uma rede tem um bom funcionamento durante o processo de treino, mas tem uma baixa performance durante o teste, frequentemente causado pelo excesso de dependências presentes em características específicas do conjunto de dados designado para treino [74]. Afim de abordar este problema, como também outros inconvenientes actualmente enfrentados pelas CNNs, muitos investigadores propuseram soluções eficazes como é o caso das técnicas de aumento de dados, regularização, eliminação, normalização de lote e funções de activação inovadoras [84]. Em pesquisas mais recentes a preocupação tem vindo a estender-se para o design de funções de perda de modo a aumentar o poder discriminativo, ou seja, em conseguir uma maior compactação das intra-classes e a maior separação das inter-classes, o que consequentemente resulta numa melhoria da exactidão das classificações obtidas [84].

De modo geral, as redes neuronais convolucionais tornaram-se num ponto crucial dos algoritmos de *deep learning* em visão por computador [48]. Isto deve-se, em grande parte, à disponibilidade de volumes massivos de dados, à crescente capacidade do poder computacional e à introdução de novos algoritmos que tornam o treino e a dedução muito eficientes [47]. Outra característica que está relacionada com o seu sucesso deve-se à sua flexibilidade, pois consoante o objectivo estas permitem a utilização de diferentes camadas e operações, que tornam possível o aumento da sua complexidade e consequente crescimento da capacidade de aprendizagem.

## Capítulo 3

# Estruturas de Dados para Reconhecimento de Padrões

No contexto de contrafação de bens preciosos, âmbito no qual surgiu o projecto em que se insere (em parte) esta dissertação, surgiram sistemas relacionados com marcações baseadas em processos estocásticos, como é o exemplo das *key PUFs* (*physical unclonable functions*), e respectivos sistemas de autenticação, alguns dos quais introduzidos ao longo deste capítulo. As marcas produzidas por este tipo de sistemas são praticamente irreprodutíveis por serem resultado de um processo caótico impossível de recriar, o que permite colmatar as necessidades que os métodos de marcações tradicionais com origem em métodos determinísticos, não conseguem dar resposta.

### 3.1 Funções Físicas não clonáveis

Funções físicas não clonáveis (PUFs), também conhecidas como funções físicas unilaterais (em inglês *physical one-way functions*), constituem uma componente importante em segurança de objectos, assim como, em abordagens de autenticação. Existem inúmeras variações e aplicações das PUFs [27], tais como: PUFs ópticas que exploram as desordens físicas não clonáveis utilizadas por exemplo, em selos e marcações que normalmente resultam de processos químicos ou mecânicos; PUFs arbitrárias que utilizam a variabilidade resultante da fabricação no que diz respeito às interconexões aleatórias e atrasos de tempo das portas dos transístores, frequentemente aplicadas na área de IoT em pastilhas CMOS; PUFs intrínsecas baseadas em memória que utilizam o estado de energização favorecido das células SRAM como resposta, e o endereço das células como desafio, aplicadas em autenticação de sistemas embebidos, também no âmbito de IoT.

Um exemplo notoriamente conhecido da sua aplicabilidade são as *key PUFs*, etiquetas (*tags*) físicas e únicas, geradas através de processos estocásticos com recurso a métodos químicos como está ilustrado na figura 3.1. Assim, quando uma *PUF tag* é lida, ou seja, tem a sua informação caracterizada e, posteriormente armazenada numa base de dados através de vectores de minúcias correspondentes, a aleatoriedade gerada pelo processo não determinístico torna impossível a sua

falsificação, mesmo através de tentativas manuais [4], sendo por isso muito empregue em métodos destinados ao combate da contrafacção devido à sua utilidade na encriptação, como também, na autenticação de produtos genuínos.

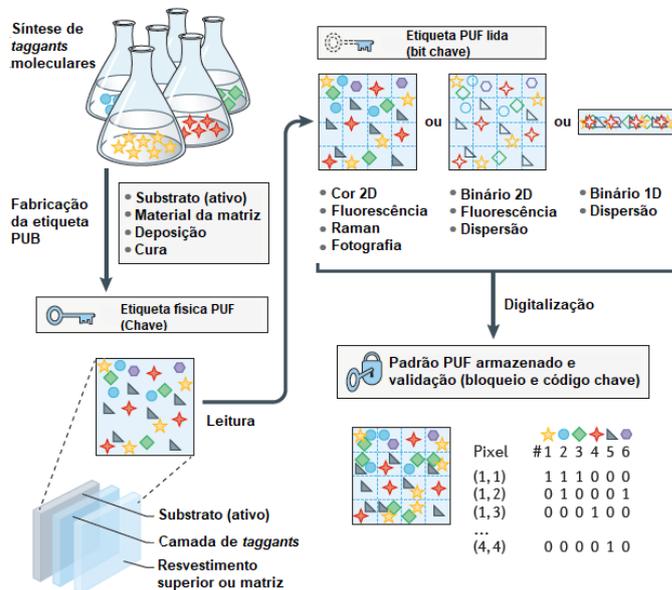


Figura 3.1: Adaptação do exemplo do processo de produção de uma chave PUF presente em [4].

## 3.2 Autenticação de padrões aleatórios

D. Brzakovic e N. Vujovic desenvolveram um algoritmo de autenticação [9] [77] capaz de estabelecer correspondências entre padrões gerados através da incorporação aleatória de pequenas fibras no meio de documentos durante o seu processo de fabrico (figura 3.2), com imagens armazenadas numa grande base de dados.

De modo a reduzir a quantidade de informação a ser recuperada e processada, o algoritmo é hierarquicamente organizado em três níveis, e a transição entre cada nível apenas ocorre caso seja estabelecida com sucesso a correspondência no nível anterior. Fazendo uma análise mais abstracta deste método, é possível verificar que se trata de uma espécie de triangulação, uma vez que são utilizadas diferentes técnicas de forma a colectar, comparar, e, interpretar os dados. Posto isto, em seguida será feita uma breve revisão deste mesmo método presente no artigo [9].

### 3.2.1 1º Nível de autenticação

No primeiro nível de autenticação são utilizados os pontos de extremidade das fibras, numa tentativa de determinar a existência de quaisquer três pontos em algum documento presente na base de dados que obtenham correspondência com quaisquer três pontos do documento teste. O objectivo deste nível é então determinar correspondências candidatas, que depois venham a ser verificadas pelos níveis seguintes. Neste nível, um documento é representado por  $M$  pontos de

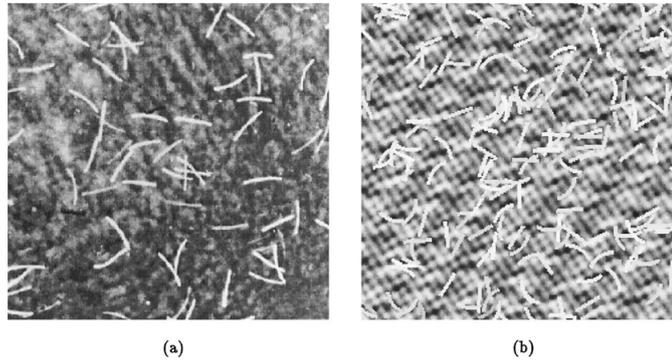


Figura 3.2: Padrão de fibras aleatório usado em autenticação de documentos, exemplo real (a), imagem sintética (b) [76].

extremidade das fibras, e cada ponto,  $P'(x', y')$ , é representado pelas suas coordenadas,  $(x', y')$ , como também por uma sequência ascendente de números,  $\delta' = (d'_1, d'_2, \dots, d'_k)$ , em que  $d'_{i-1} \leq d'_i$  para  $i=2, \dots, k$ , e onde  $d'_i$  representa a distância entre o ponto  $P'$  e os  $i^{os}$  pontos mais próximos. A sequência regista todas as distâncias entre o ponto  $P'$  e os pontos que se encontram num círculo de raio  $R$  centrado em  $P'$ .

As imagens dos documentos são divididas em quatro quadrantes e a correspondência entre pontos é feita entre os quadrantes correspondentes. Na base de dados, cada quadrante é representado por  $J$  pontos ( $J = M/4$ ) e para um  $N_{max}$  de fibras em falta arbitrariamente seleccionadas no documento, é necessário guardar  $J \geq (N_{max} + 1)/2$  pontos de extremidade por quadrante de modo a garantir o estabelecimento das correspondências. O processo de correspondência do primeiro nível divide-se em três passos:

1. O primeiro passo consiste na determinação da correspondência do primeiro ponto. Para estabelecer esta correspondência, é feita uma tentativa de correspondência de qualquer ponto pertencente ao primeiro quadrante entre os documentos. Se não for encontrada nenhuma correspondência neste quadrante, o processo repete-se no segundo quadrante, e assim sucessivamente. Em adição, a sequência de distâncias do ponto  $P_1$ ,  $\delta_1 = (d_1, d_2, \dots, d_n)$ , deve corresponder com a sequência de distâncias  $\delta'_1 = (d'_1, d'_2, \dots, d'_m)$  do ponto  $P'_1$ . Tendo em conta as fibras em falta, a correspondência entre as sequências existe se  $N \leq M$  e todas as distâncias de  $P_1$  estiverem dentro de um valor limite,  $T_d$ , em relação às distâncias correspondente de  $P'_1$ . A ideia base por detrás do estabelecimento da correspondência do primeiro ponto é ilustrada na figura 3.3.
2. Dado que foi estabelecida a correspondência do ponto  $P_1(x_1, y_1)$  do documento teste com o ponto  $P'_1(x'_1, y'_1)$  no  $K^o$  documento da base de dados, a correspondência do segundo ponto é procurada no quadrante seguinte. O ponto  $P'_2(x'_2, y'_2)$  do documento da base de dados tem uma potencial correspondência com  $P_2(x_2, y_2)$  presente no documento teste, se  $P_2$  se encontrar no círculo definido por:

$$(x - x_1)^2 + (y - y_1)^2 = D^2 \quad (3.1)$$

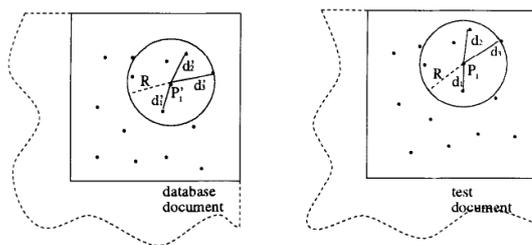


Figura 3.3: Ilustração da correspondência do primeiro ponto [9].

Na prática, devido a efeitos de digitalização, o ponto correspondente situa-se num anel definido por:

$$(D - |T_d|)^2 \leq (x - x_1)^2 + (y - y_1)^2 \leq (D + |T_d|)^2 \quad (3.2)$$

onde  $T_d$ , tal como anteriormente, é o parâmetro que modela os efeitos da digitalização. Uma potencial correspondência é considerada como verdadeira se a sequência do ponto  $P_2(x_2, y_2)$  tiver correspondência com a sequência de  $P'_2(x'_2, y'_2)$  no mesmo sentido em que foi descrito no primeiro passo. Se a correspondência do segundo ponto não for estabelecida no  $K^o$  documento da base de dados, a procura continua, o algoritmo volta ao passo anterior, e tenta novamente estabelecer correspondência do primeiro ponto ainda na mesma imagem. Caso isso não se verifique a procura recomeça na imagem seguinte da base de dados.

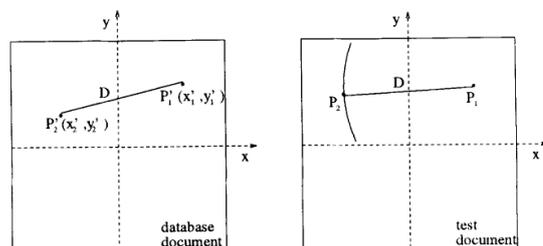


Figura 3.4: Ilustração da correspondência do segundo ponto [9].

3. Quando é estabelecido o primeiro e o segundo ponto, uma tentativa de correspondência de um terceiro ponto é feita no quadrante seguinte ao qual se verificou correspondência no segundo nível. Um ponto  $P_3(x_3, y_3)$  na imagem teste diz-se corresponde com o ponto  $P'_3(x'_3, y'_3)$  no  $K^o$  documento da base de dados se se verificar as seguintes condições:

$$\begin{aligned} & (\sqrt{(x'_1 - x'_3)^2 + (y'_1 - y'_3)^2} - |T_d|)^2 \leq (x_1 - x_3)^2 + (y_1 - y_3)^2 \\ & \leq (\sqrt{(x'_1 - x'_3)^2 + (y'_1 - y'_3)^2} + |T_d|)^2 \end{aligned} \quad (3.3)$$

e

$$\begin{aligned} & (\sqrt{(x'_3 - x'_2)^2 + (y'_3 - y'_2)^2} - |T_d|)^2 \leq (x_3 - x_2)^2 + (y_3 - y_2)^2 \\ & \leq (\sqrt{(x'_3 - x'_2)^2 + (y'_3 - y'_2)^2} + |T_d|)^2 \end{aligned} \quad (3.4)$$

Tal como anteriormente, é necessário que se verifique a correspondência entre as seqüências de distâncias entre o par de pontos correspondente. Se não for possível estabelecer uma correspondência para o terceiro ponto no  $K^o$  documento da base de dados, então o processo de procura considera uma correspondência alternativa para a correspondência do segundo ponto. Caso isto não seja possível, a procura recomeça na imagem seguinte  $(K + 1)^o$  da base de dados.

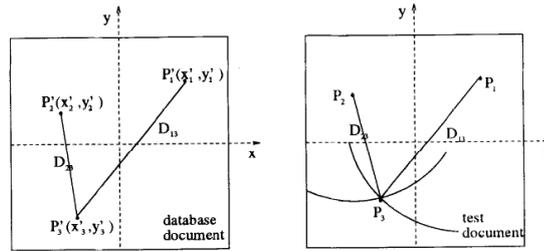


Figura 3.5: Ilustração da correspondência do terceiro ponto [9].

### 3.2.2 2ª Nível de autenticação

No segundo nível de autenticação, depois do algoritmo validar com sucesso a correspondência do primeiro nível, este fica encarregue de verificar a correspondência entre todos os pontos de extremidade das fibras, começando para isso, por determinar os parâmetros de translação,  $x_0$  e  $y_0$ , e rotação,  $\theta$ , entre os dois documentos. Para este propósito são utilizados quaisquer dois pares de pontos anteriormente determinados como correspondentes.

Para um par de pontos correspondente  $P_k(x_k, y_k)$ ,  $P'_k(x'_k, y'_k)$  e  $P_j(x_j, y_j)$ ,  $P'_j(x'_j, y'_j)$ , os parâmetros por descobrir são obtidos com recurso a três das equações que se seguem:

$$x'_p = x_p \cos \theta - y_p \sin \theta + x_0, p = j, k, \quad (3.5)$$

e

$$y'_p = x_p \sin \theta - y_p \cos \theta + y_0, p = j, k, \quad (3.6)$$

Dois documentos dizem-se correspondentes no segundo nível se  $2n_1$  pontos de extremidade do documento teste obtiverem correspondência com os pontos do documento presente na base de dados. O documento da base de dados contém  $2n$  pontos de extremidade e  $n - n_1 \leq N_{max}$ , onde  $N_{max}$  corresponde ao número máximo de fibras em falta.

### 3.2.3 3º Nível de autenticação

Se todos os pontos de extremidade obtiverem correspondência no segundo nível, o terceiro passo no processo de autenticação passa por fazer correspondência das fibras completas, com recurso a correlação. O terceiro nível de autenticação apenas é utilizado quando é solicitado um alto nível de singularidade, pois o grau de confiança proporcionado pelo segundo nível é normalmente considerado suficiente.

Segundo os testes executados pelos autores em [9], é possível afirmar que fibras embebidas em documentos podem ser utilizadas como uma assinatura dos mesmos, sendo possível autenticá-los através da identificação da sua correspondência numa base de dados.

O algoritmo de autenticação descrito tem vantagens que vão desde a oferta de um algoritmo com diferentes níveis de autenticação de documentos, até à necessidade de pouco armazenamento, uma vez que a representação de documentos é feita apenas por um conjunto de pontos e uma sequência ordenada de números inteiros, tornando-se também computacionalmente mais eficiente que outros métodos que envolvem obrigatoriamente a correlação.

### 3.3 mIDoT

Em 2016 foi desenvolvido um método chamado mIDoT [41]. Este método consiste na adição de num ponto de tinta metálica ou glitter, que possui na sua constituição micro partículas, a um objecto. Este ponto, apelidado de mIDoT, é então identificado por correspondência entre imagens do padrão microscópico da superfície irregular formada pelo ponto com a imagem correspondente armazenada na base de dados. O padrão gerado é de natureza aleatória e única, o que permite uma identificação exacta [49].

O algoritmo de correspondência entre imagens está dividido em três etapas que são: a extração e correspondência de pontos; a verificação geométrica de pares de pontos correspondentes; e por fim, o cálculo da pontuação de correspondência.

A técnica usada durante a correspondência tem por base características locais de modo a estimar transformações geométricas entre a imagem consultada e a imagem adquirida. É utilizado o algoritmo ORB [66] e a posterior verificação geométrica dos pares de pontos correspondentes é feita através do algoritmo RANSAC [26]. O cálculo da pontuação é o resultado do número de correspondências identificadas pelo algoritmo RANSAC pelo número de pares correspondentes identificados pelo ORB. Este valor quantifica a semelhança entre a imagem armazenada na base de dados e a imagem que está sob análise (figura 3.6).

Quando comparado com outros métodos, como por exemplo, o código de barras ou etiquetas RFID, o mIDoT tem capacidade de ultrapassar limitações existentes devido ao seu pequeno tamanho, ao seu baixo custo e ao seu eficiente método de marcação de objectos. No entanto, tem como principal desvantagem o facto de requerer microscopia no processo de autenticação/validação, o que é um factor limitativo do ponto de vista de aplicação prática [41].

### 3.4 UniqueMark

O projecto UniqueMark tem como objectivo a criação de um sistema que fornece a um metal precioso uma marca única, não clonável e irreprodutível, como também, a concepção de um sistema para a validação e autenticação da mesma [18].

A marca pode ser obtida de duas maneiras distintas. A primeira consiste na punção de uma dispersão de partículas de diamante na superfície do metal, sendo esta marca não clonável por

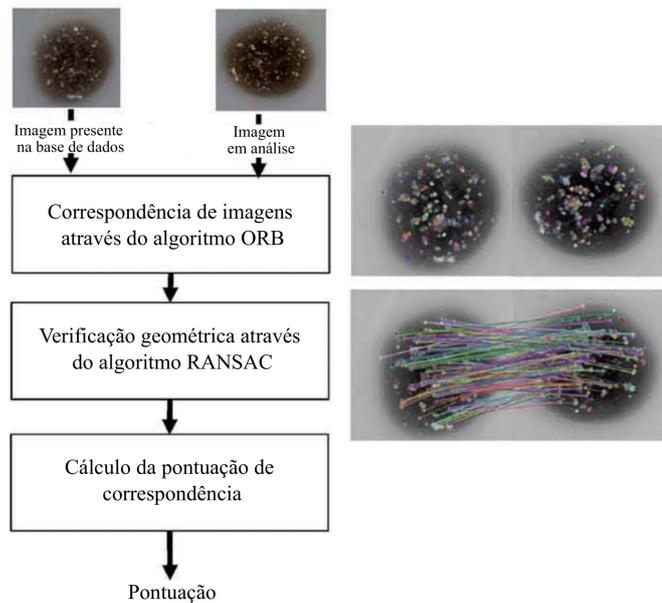


Figura 3.6: Adaptação do esboço da correspondência entre imagens com os algoritmos ORB e RANSAC presente em [41].

ser resultado de um processo aleatório e caótico. O segundo tipo de marca é obtido através da marcação a laser segundo um caminho determinístico descrito por uma função matemática em que o efeito no metal é único, dada a escala do feixe de laser e da dimensão das marcas, tornando possível a distinção entre a marca original e outras marcas [18].

Como é possível observar na figura 3.7, o sistema de verificação e autenticação de uma marca gerada por este sistema possui seis etapas. A primeira etapa consiste na aquisição da imagem através de um microscópio ou pela câmara de um telemóvel. Segue-se a deteção da marca (A) e ajuste da imagem (B) de modo a permitir a deteção das partículas de diamante (C), e procede-se ao cálculo do descritor em relação à dispersão (D). Por fim, e dependendo do pretendido, a marca é adicionada à base de dados para o registo da mesma (E), ou é validada através da busca na base de dados usando o modelo kd-tree (F).

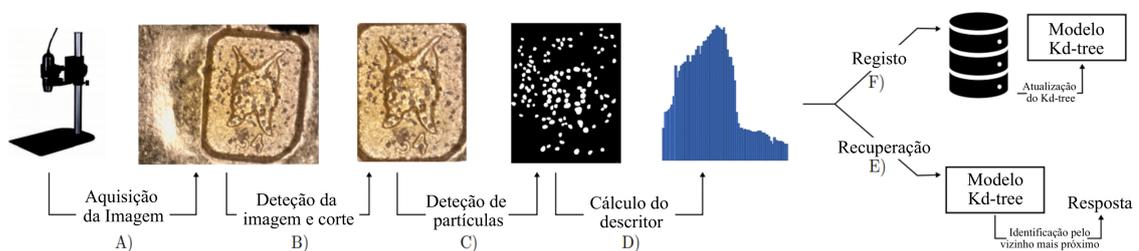


Figura 3.7: Adaptação da figura referente aos processos de validação e registo do UniqueMark, presente em [18].

## Capítulo 4

# Reconhecimento Facial

Como referido anteriormente, esta tese tem como principal objectivo o estudo de diferentes métodos aplicadas em reconhecimento facial. Assim sendo, será feito uma breve introdução a uma aplicação relevante da análise de imagens de faces.

### 4.1 Introdução ao reconhecimento facial

Os primeiros passos na área de reconhecimento facial iniciaram-se nos anos sessenta. Pela primeira vez em 1964 e 1965, em estudos conduzidos por Woody Bledsoe, Helen Chan Wolf e Charles Bisson [8] [7], foram utilizados computadores no reconhecimento de faces humanas. Este método consistia na extracção manual das coordenadas de vários pontos estratégicos das faces presentes em fotografias, onde as distâncias resultantes entre estes pontos eram posteriormente calculadas e comparadas automaticamente com a intenção de determinar a correta identidade.

Na década de setenta, Goldstein, Harmon e Lesk [29] continuaram o trabalho desenvolvido anteriormente ao adicionarem mais 21 marcadores específicos, que incluíam por exemplo, a cor do cabelo e a espessura dos lábios. Apesar das melhorias a nível da precisão, as medições e extracção de pontos continuavam a depender do trabalho manual.

Só no final da década de oitenta é que houve um progresso mais significativo. Sirovich e Kirby [71] introduziram um sistema, apelidado de Eigenface, que consistia num algoritmo baseado no uso de características estatísticas fundamentais da imagem e na matriz de variância para extrair o descritor da face no domínio linear de alta dimensão. De modo a reduzir dimensionalidade, era aplicada álgebra linear e a análise de componentes principais (PCA), mostrando assim, que eram necessários menos de uma centena de valores para codificar uma imagem da face, adequadamente normalizada e alinhada, com exactidão. Mais tarde, no ano de 1991, Turk e Pentland [75] deram continuidade a este trabalho, ao descobrir que o erro residual poderia ser utilizado no desenvolvimento de sistemas de reconhecimento em tempo real, o que eventualmente originou o primeiro sistema de reconhecimento facial totalmente automatizado.

Nos anos que se seguiram foi possível obter melhorias significativas através de métodos dedicados ao reconhecimento facial onde foram aplicadas diversas técnicas de detecção automática das *facial landmarks*, normalmente associados a classificadores treinados. Para a extracção de características da face surgiram diversas abordagens que se dividiram pelo uso do método AMM

(*Active Appearance Model*) [1], pelas distâncias geométricas faciais [61], até mesmo pela invariância geométrica das faces [44]. Uma das técnicas que se tornou mais popular está relacionada com a análise de características locais da textura das faces, onde são consideradas tanto a forma como a textura das faces, representadas através de histogramas de *Local Binary Pattern* (LBP), presente por exemplo em [3].

Outro método que obteve muito sucesso, envolve o uso de *Histogram of Oriented Gradients* (HOG). Este ao contar a ocorrência de um certa orientação do gradiente permite a extracção dos vectores de características discriminativos das imagens sob estudo. Está por isso presente em muitos métodos, entre os quais, [69] e [22].

Técnicas que envolvem a aplicação de filtros Gabor, usualmente utilizados em análise de texturas, também foram bem sucedidos em tarefas de reconhecimento facial. Com este método é possível extrair a direcção e frequência do conteúdo de uma região localizada em torno do ponto ou região sob análise. É possível constatar o sucesso da aplicação desta técnica em [52].

Para concluir, é importante mencionar que apesar de todos os progressos alcançados com este tipo de técnicas mais tradicionais de *machine learning*, estudos actuais afirmam que estas encontram-se em evidente desvantagem quando comparadas a modelos mais recentes ligados ao *deep learning*.

## 4.2 *Deep Learning* aplicado a Reconhecimento facial

O actual *pipeline* em reconhecimento facial convencional consiste em quatro etapas: detecção da face, alinhamento da face, extracção de características e classificação, sendo a terceira etapa uma das mais importantes. Em cenários com constrangimentos, em que por exemplo, há controlo da iluminação e escolha da posição da face, os métodos tradicionais, alguns deles descritos na secção anterior (secção 4.1), alcançaram um desempenho em reconhecimento facial notável. Porém, quando aplicados a ambientes sem constrangimentos a performance obtida através das características extraídas degrada-se drasticamente, principalmente quando as imagens utilizadas possuem variações intra-pessoais complexas no que concerne às variações da pose, à iluminação, às expressões faciais e a oclusões parciais [38].

De entre as várias técnicas desenhadas para extrair as características de imagens, a rede neuronal convolucional foi uma das que mais se destacou pela positiva, isto porque, ao contrário das características *handcrafted*, as características aprendidas por este tipo de rede são mais robustas às referidas variações intra-pessoais complexas.

Um dos primeiros trabalhos a ser publicado sobre este tipo de rede no contexto de reconhecimento facial foi apresentado em [50], onde as amostras de entrada eram primeiramente processadas por uma rede neuronal de mapas auto-organizáveis de Kohonen, com o intuito de reduzir dimensionalidade, para depois serem processadas por uma arquitectura de rede sequencial apelidada de LeNet, mais conhecida pela sua aplicação em reconhecimento de caracteres manuais [51].

De modo geral, nas tentativas que se seguiram, o objectivo concentrou-se em encontrar a melhor arquitectura para as redes convolucionais, o que eventualmente resultou no aumento da

complexidade das mesmas e, conseqüentemente levou a problemas relacionados com o treino, como também, a dificuldades no processo de optimização. Um dos eventos que contribuiu para a evolução desta área foi o Desafio de Reconhecimento Visual em Grande Escala (*ILSVRC*). Com este desafio várias arquitecturas foram desenvolvidas, sendo as mais conhecidas: ZefNet [82], VGG [70], GoogLeNet [73], ResNet [35], DenseNet [39], MobileNet (com as suas várias versões [37, 68, 36]).

Actualmente os estudos que incluem redes artificiais convolucionais utilizam este tipo de arquitecturas dando preferência às mais conhecidas e detentoras dos melhores resultados, tendo em conta o contexto em que estão inseridas. Tal como descrito na secção 2.5, estas redes possuem alguns inconvenientes, havendo necessidade de aumentar o poder discriminativo, especialmente em contexto de reconhecimento facial, o que pode ser alcançado através de determinadas técnicas (algumas delas brevemente mencionadas na secção 2.5), com foco especial na modificação da função de perda, que é o método utilizado para calcular o erro da previsão da rede neuronal. *Softmax*, também conhecida como *Categorical Cross-Entropy loss* é a função de perda mais utilizada em classificação de multi-classes, e é definida pela seguinte equação [21]:

$$L_s = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T x_i + b_j}} \quad (4.1)$$

onde  $x_i \in \mathbb{R}^d$  denota a característica profunda da  $i^a$  amostra, pertencente à  $y_i^a$  classe, em que  $d$  representa a dimensão das características incorporadas.  $W_j \in \mathbb{R}^d$  representa a  $j^a$  coluna dos pesos  $W \in \mathbb{R}^{d \times n}$  e  $b_j \in \mathbb{R}^n$  refere-se ao termo *bias*. Por fim, o tamanho do lote e o número de classes são definidos por  $N$  e  $n$ , respectivamente.

Apesar de esta ser a função de perda mais conhecida e conseqüentemente a mais utilizada, falta-lhe poder discriminativo, sendo este um dos motivos que levou à introdução de funções de perda alternativas [78]. No âmbito de reconhecimento facial foram propostas algumas variantes *margin-based* com a intenção de aumentar a discriminação das características do tipo de imagens analisadas. No contexto desta dissertação, para além da função de perda softmax, irá ser estudado o impacto de outras funções de perda recentemente desenvolvidas, nomeadamente:

- **CosFace** - *Large Margin Cosine Loss (LMCL)* - Reformula a função de perda softmax para a função de perda cosseno, através da normalização  $L_2$ , tanto nas características como nos vectores de peso, de modo a remover variações radiais tendo por base o termo introduzido pela margem do cosseno, de forma a maximizar a margem de decisão no espaço angular [78]. Formalmente, LMCL é definida como:

$$L_{lmc} = \frac{1}{N} \sum_i -\log \frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{j \neq y_i} e^{s \cos(\theta_{j,i})}} \quad (4.2)$$

sujeito a

$$W = \frac{W^*}{\|W^*\|}, \quad x = \frac{x^*}{\|x^*\|}, \quad \cos(\theta_j, i) = W_j^T x_i \quad (4.3)$$

onde  $N$  corresponde ao número de amostras treinadas,  $x_i$  o vector de características correspondente à verdadeira classe de  $y_i$ .  $W_j$  representa o vector de pesos da classe  $j^o$ ,  $\theta_j$  o ângulo entre  $W_j$  e  $x_i$ , e,  $s$  a escala de características.

- **SphereFace - Deep Hypersphere Embedding (DHE)** - Também conhecida como A-Softmax, é uma modificação da função de perda *Softmax* original (equação 4.1). Esta possui uma interpretação geométrica clara, pois quando utilizada, as características aprendidas constroem uma métrica discriminativa de distância angular que equivale à distância geodésica numa hipersfera manifold (*hypersphere manifold*). Por este motivo, esta função de perda pode ser interpretada como o restringir das características aprendidas a serem discriminadas numa hipersfera manifold, o que as torna mais eficazes em reconhecimento facial, sendo esta a razão pela qual ficou conhecida como *SphereFace* [54].

$$L_{ang} = \frac{1}{N} \sum_i -\log\left(\frac{e^{\|x_i\| \psi(\theta_{y_i}, i)}}{e^{\|x_i\| \psi(\theta_{y_i}, i)} + \sum_{j \neq y_i} e^{\|x_i\| \cos(\theta_j, i)}}\right) \quad (4.4)$$

onde  $\psi(\theta_{y_i}, i) = (-1)^k \cos(m\theta_{y_i}, i) - 2k$ ,  $\theta_{y_i, i} \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}]$ ,  $k \in [0, m-1]$  e,  $m \geq 1$  é um inteiro que controla o tamanho da margem angular.

- **Arcface - Additive Angular Margin Loss (AAML)** - é uma função de perda que otimiza directamente a distância geodésica em virtude da correspondência exacta entre o ângulo e o arco na hipersfera normalizada. Para isso, é utilizada a função arco-cosseno para calcular o ângulo entre a característica actual e o peso alvo. Posteriormente, é adicionada uma margem angular aditiva ao ângulo alvo e é obtido o *logit*, novamente pela função cosseno. Em seguida, todos os *logits* são redimensionados por uma norma de características fixa, e onde os passos subsequentes são idênticos aos presentes na função *Softmax* [21].

$$L_{aam} = -\frac{1}{N} \sum_i \log \frac{e^{s(\cos(\theta_{y_i} + m))}}{e^{s(\cos(\theta_{y_i} + m))} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}} \quad (4.5)$$

a simbologia segue as mesmas directrizes que as enunciadas anteriormente.

Uma das grandes vantagens das CNNs é o facto de que dado um amplo conjunto de treino, os sistemas de reconhecimento de padrões em grande escala poderem ser optimizados de uma ponta a outra, sendo capazes de desenvolver recursos que amplificam o sinal de identidade, ao mesmo tempo em que são robustas a variações de pose, iluminação e expressão, os maiores desafios de reconhecimento automático até aos dias de hoje [56].

## Capítulo 5

# 1ª Parte - Método clássico de Machine Learning

### 5.1 Metodologia

Esta dissertação divide-se em duas partes. A primeira parte foca-se num método clássico de *machine learning*, onde o principal objectivo consiste na aplicação do algoritmo desenvolvido no projecto UniqueMark, descrito na secção 3.4, em reconhecimento facial, mais precisamente em dispersões estruturadas obtidas pela dectecção dos *facial landmarks* de imagens de faces frontais, com recurso a diferentes classificadores, o KNN e o SVM. Na segunda parte, o foco está em *pure deep learning*, ou seja, utilizam-se redes neuronais artificiais no contexto de reconhecimento facial com a intenção de estudar a relação de compromisso (*trade-off*) entre o número de *features* e a arquitectura, como também, optimização da rede através do estudo do impacto de diferentes funções de perda, tendo como motivação a aplicabilidade em dispositivos com menor capacidade computacional. Por serem partes relativamente distintas, estas serão divididas por diferentes capítulos na tentativa de melhorar a sua apresentação e compreensão.

#### 5.1.1 Estrutura de avaliação

Na avaliação do método implementado, é providenciado ao algoritmo um conjunto de imagens sob a forma de coordenadas dos pontos salientes que as constituem, juntamente com respectivas etiquetas. Estas depois de processadas por um determinado descritor de pontos são alimentadas aos classificadores para que seja criado o modelo de predição adequado. Devido às limitações numéricas das bases de dados disponíveis, as imagens foram divididas apenas em dois subconjuntos destinados para treino e para teste segundo o algoritmo de *N-fold cross validation*, utilizado para estimar a habilidade de cada modelo, onde os dados disponíveis foram divididos por um número predefinido que muda de forma iterativa pelos dois processos, como mostra a figura 5.1.

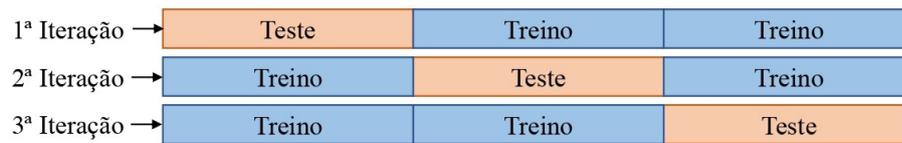


Figura 5.1: Exemplo da divisão dos dados para a validação cruzada entre três subconjuntos e iterações.

## 5.1.2 Extração de pontos

Tendo em conta, que o objecto de estudo nesta primeira parte são padrões formados por dispersões de pontos salientes, uma parte fundamental do processo é efectivamente a extração dos pontos que as constituem, pois tal como descrito anteriormente, o descriptor é responsável por representar, por meio de histogramas, as relações geométricas entre estes. Quando a intenção é aplicar este tipo de método ao reconhecimento facial, é necessário assegurar uma extração adequada dos pontos, de modo a proporcionar uma representação apropriada e de assegurar a correta distinção entre diferentes identidades.

### 5.1.2.1 Detecção das *facial landmarks*

Uma forma eficaz de representar faces contidas em imagens consiste na localizar/extrair um conjunto de pontos de interesse, que contêm informações e características acerca da forma das faces sob análise. Este conjunto de pontos, a que é atribuído o nome *facial landmarks*, utiliza 68 pontos para representar as faces, ao contornar sistematicamente regiões salientes da face que se resumem aos olhos, sobrancelhas, nariz, boca, e mandíbula, como é demonstrado pela figura 5.2.

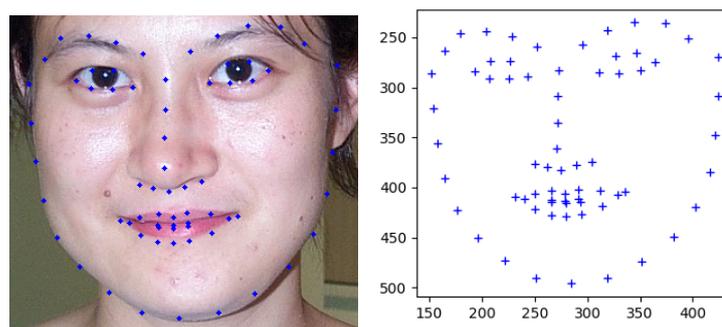


Figura 5.2: Imagem exemplo da extração das *facial landmarks*.

O método utilizado para esta operação, descrito em [45], é capaz de encontrar o alinhamento facial em milissegundos com uma elevada precisão. Este método utiliza funções de regressão em cascata, ou seja, cada função de regressão do modelo calcula eficazmente a forma da face através de uma estimativa inicial e das intensidades de um conjunto disperso de pixels indexados a essa estimativa inicial.

### 5.1.3 Alinhamento

No processo de autenticação de padrões constituídos por dispersões de pontos salientes há um procedimento padrão que permite assegurar correspondências entre imagens pertencentes à mesma identidade, mas com representações distintas. Este processo é necessário uma vez que após o processo de obtenção dos pontos que formam a dispersão sob análise, é essencial que haja algum tipo de normalização devido à arbitrariedade da localização e tamanho dos mesmos.

Uma forma de garantir que estas diferenças não afectam negativamente o processo de correspondência consiste em fazer um alinhamento prévio das dispersões em análise, sendo este particularmente importante para as comparações entre correspondências que têm por base as características locais. Por este motivo, foi implementado o algoritmo ICP [6] para imagens 2D. *Iterative closest point* (ICP) é um algoritmo utilizado para minimizar as diferenças entre duas nuvens de pontos, sendo este frequentemente utilizado em reconstrução de superfícies 2D e 3D. Dados dois conjuntos de pontos,  $P$  e  $M$ , o objectivo do algoritmo é alinhar o conjunto de pontos  $P$  da melhor forma possível com  $M$ . O ICP baseia-se em três passos básicos [12]: (1) corresponder cada ponto de  $P$  com o ponto mais próximo de  $M$ ; (2) calcular a transformação que minimiza o erro quadrático médio (EQM) entre o par de pontos; (3) aplicar a transformação óptima ao conjunto de pontos  $P$  e actualizar o EQM. Estes passos são iterativos, isto quer dizer que se repetem até que o mínimo local da função objectivo do erro quadrático médio seja encontrado [6]. Este algoritmo fica então responsável por encontrar uma matriz que descreva a transformação a que a imagem foi sujeita e, deste modo, reverter o processo.

No que diz respeito à autenticação de faces com auxílio do descritor geométrico de pontos foi também necessário adicionar um outro procedimento, o escalamento. Este consiste em escalar as coordenadas dos pontos das *facial landmarks* de modo a permitir que as representações da mesma identidade possuam proporções semelhantes. Assim, cada uma das coordenadas dos pontos é multiplicada pelo cálculo da razão entre o perímetro de uma face pertencente a uma determinada identidade, tida como referência, sobre o perímetro da face em estudo também pertencente a essa identidade.

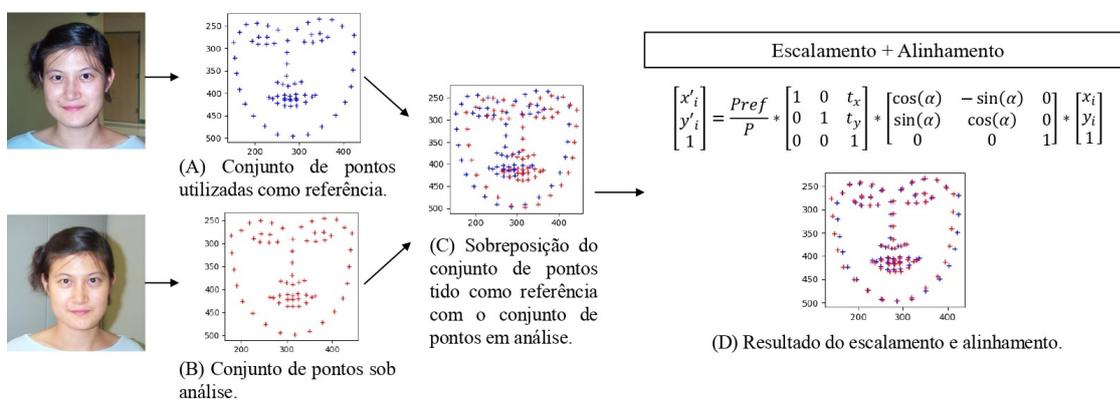


Figura 5.3: Exemplo ilustrativo do processo de alinhamento.

#### 5.1.4 Descritor geométrico de pontos - UniqueMark

O descritor geométrico de pontos utilizado nesta secção é uma adaptação do descritor descrito no projecto UniqueMark [18]. Resumidamente, este descritor consiste na criação de histogramas que contêm informação sobre as características geométricas, previamente escolhidas, da dispersão de pontos sob estudo. Estas características dividem-se em globais e locais. As características globais, como o nome indica, descrevem a imagem como um todo e vão desde todas as combinações possíveis de distâncias entre pontos, ângulos formados por trios de pontos, até ângulos entre combinações formadas por uma extremidade e dois pontos aleatórios da dispersão. No que diz respeito às características locais, estas analisam a imagem por partes e podem ser o número de vizinhos de um dado ponto tendo em consideração um raio previamente determinado, a distância entre um ponto e os seus vizinhos, até mesmo o número de pontos que existe por coluna ou linha.

De uma forma resumida, o algoritmo utilizado consiste num sistema que recebe como entrada um padrão de pontos armazenado num ficheiro que integra as suas coordenadas, que depois da análise dos dados fornecidos devolve um histograma com informações sobre as relações geométricas entre os diversos pontos pertencentes ao mesmo padrão. Este histograma é posteriormente utilizado no processo de validação com recurso aos classificadores KNN e SVM.

As características geométricas globais utilizadas nesta secção são:

1. Número de distâncias individuais entre combinações de cinco pontos e um ponto tido como referência, exemplo na figura 5.4a. Uma alternativa baseada nesta característica, consiste em atribuir pesos às distâncias, ou seja, em vez de simplesmente contabilizar os números de distâncias que vão sendo calculadas entre os pontos, são definidos intervalos em que consoante o valor da distância calculada, é atribuída uma maior ou menor importância à mesma através da adição de mais ou menos valores à contagem final.
2. Número de ângulos obtidos por um triângulo formado por um trio de pontos, figura 5.4b.
3. Número de ângulos obtidos utilizando dois pontos aleatórios e uma das extremidades, figura 5.4c.

No que diz respeito às características locais utilizadas, estas dividem-se em:

1. Número de pontos encontrados num determinado intervalo do eixo x ou do eixo y, em que o intervalo é definido através da divisão equalitária da largura ou comprimento, respectivamente, por um número definido de vezes, figura 5.5a.
2. Número de vizinhos de ponto, definido por um raio  $R$ , para cada um dos pontos pertencente à dispersão, figura 5.5b.
3. Número de distâncias individuais encontradas entre um ponto tido como referência e os pontos pertencente à sua vizinhança, figura 5.5c.

Uma técnica habitualmente utilizada para representar características geométricas passa por dividir a imagem em quadrantes, ou seja, em dividir as dimensões previamente conhecidas, neste

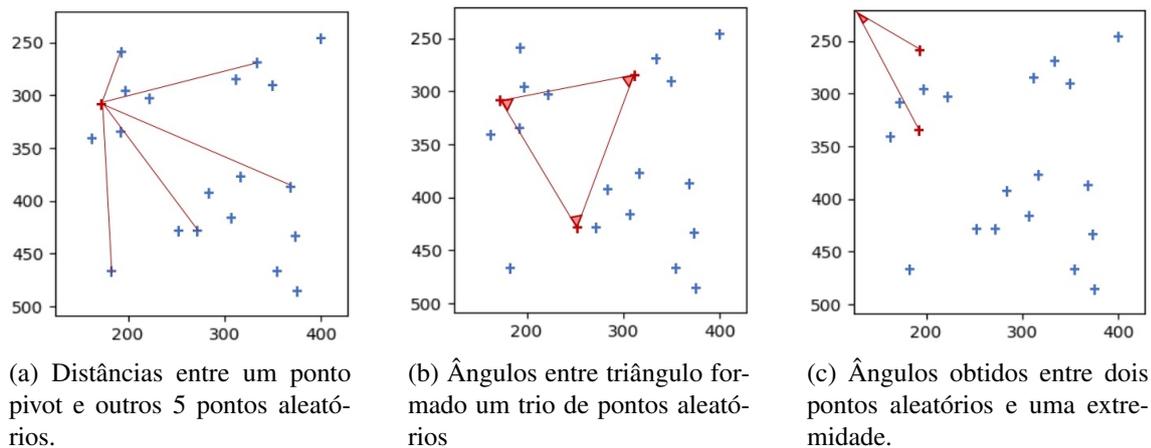


Figura 5.4: Características geométricas globais.

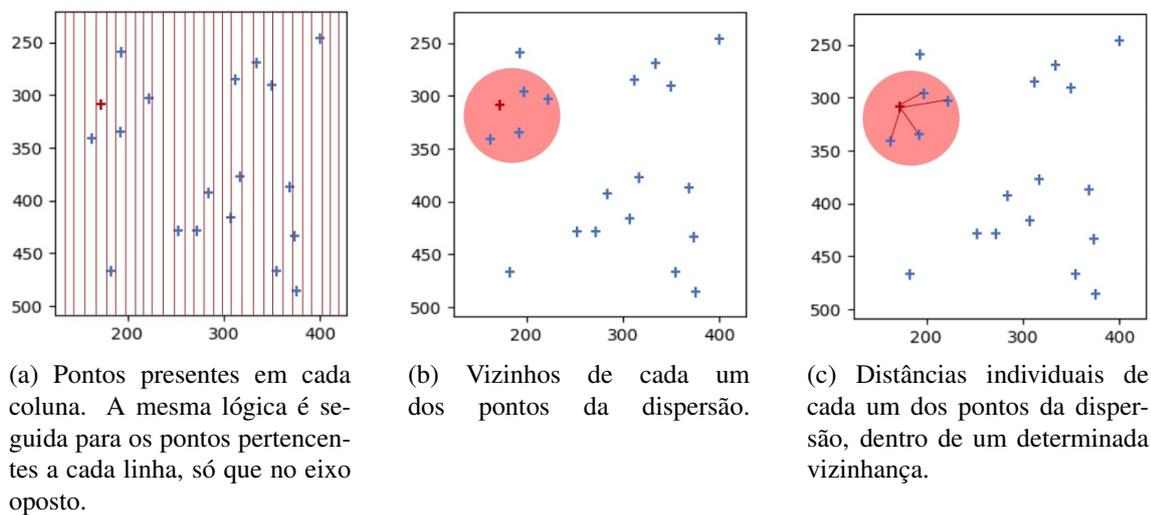


Figura 5.5: Características geométricas locais.

caso das dispersões de pontos salientes, em quatro partes iguais (figura 5.6). Assim, as características entre pontos descritas anteriormente, tanto globais como locais, passam a ser exclusivas entre pontos pertencentes ao mesmo quadrante, o que conseqüentemente resulta numa diminuição do número de combinações, tendo por isso um impacto positivo no tempo de execução do processo.

### 5.1.5 Aplicação do descritor em padrões de diferentes naturezas

Tal como detalhado no artigo UniqueMark [18], o sistema de validação que envolve a utilização de histogramas com origem em relações geométricas entre pontos pertencentes a padrões aleatórios resultantes de punções de pó de diamante em metais preciosos, revelou-se promissor; tão promissor que levantou interesse no que diz respeito à sua aplicabilidade em outras áreas de relevo como é o caso da biometria, mais especificamente em reconhecimento facial. Para chegar a este objectivo realizaram-se uma série de testes e tentativas de optimização do algoritmo com a intenção de melhorar os resultados que serão descritas nas secções seguintes.

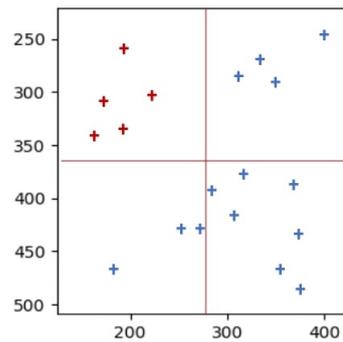


Figura 5.6: Divisão das características geométricas por quadrantes.

### 5.1.5.1 Padrões de dispersões aleatórias gerados sinteticamente

De modo a avaliar correctamente o funcionamento do processo que envolve a utilização do descritor, assim como, obter valores de referência para posteriores comparações relacionadas com dispersões estruturadas, foram efectuados alguns testes com dispersões aleatórias que consistiram nos seguintes passos:

1. **Geração de padrões aleatórios** - Neste passo, para cada identidade é gerado um padrão que fica definido como o original. Seguidamente, são gerados os padrões que lhe são correspondentes através da adição de transformações geométricas graduais, nomeadamente rotações e a translações, em todos os pontos ou em apenas uma porção deles.
2. **Cálculo dos diferentes histogramas** - São gerados os histogramas segundo as características geométricas, já mencionadas na secção 5.1.4, para cada um dos padrões gerados anteriormente.
3. **Análise da taxa de exactidão** - Utilização dos classificadores KNN e SVM, onde os padrões das diferentes identidades são distribuídos por conjuntos destinados para treino e para teste. A taxa de acerto do algoritmo é medida através da métrica da exactidão.

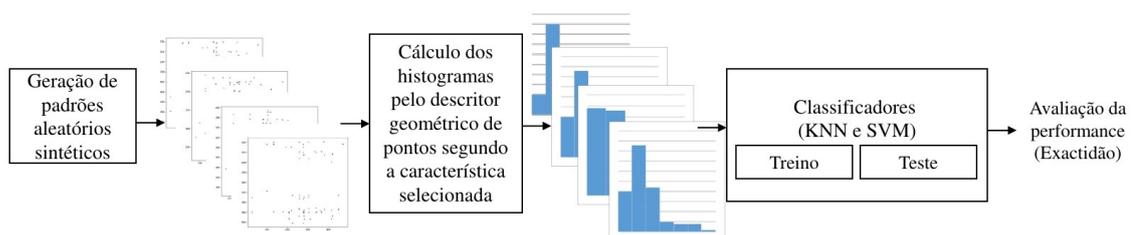


Figura 5.7: Sistema de avaliação da performance do descritor geométrico de pontos quando aplicado em padrões aleatórios sintéticos.

### 5.1.5.2 Padrões de dispersões estruturadas, reais e sintéticos

Com o intuito de avaliar a performance do descritor geométrico de pontos em dispersões estruturadas, foram efectuados testes muito semelhantes aos detalhados na secção anterior, com a

natural diferença no que diz respeito à obtenção/geração dos padrões de pontos para análise. Foram feitos dois tipos de teste: um que envolvia padrões gerados sinteticamente com o propósito de aprimorar o sistema e verificar o seu correto funcionamento e nos quais foram simuladas as características dos padrões reais, e outro com recurso a imagens reais. No que diz respeito à utilização de imagens reais, foram utilizadas as imagens dos conjuntos de dados devidamente referenciados na secção 5.2.1, onde todas as imagens seleccionadas foram submetidas ao processo de detecção de *facial landmarks* descrito na secção 5.1.2.1. A cada *landmark* foi atribuída uma coordenada  $x$  e  $y$ , que depois de devidamente identificadas foram armazenadas num ficheiro, que posteriormente será utilizado pelo descriptor geométrico de pontos.

Para os padrões obtidos do modo anterior foram repetidos os passos 2 e 3 descritos na secção 5.1.5.1.

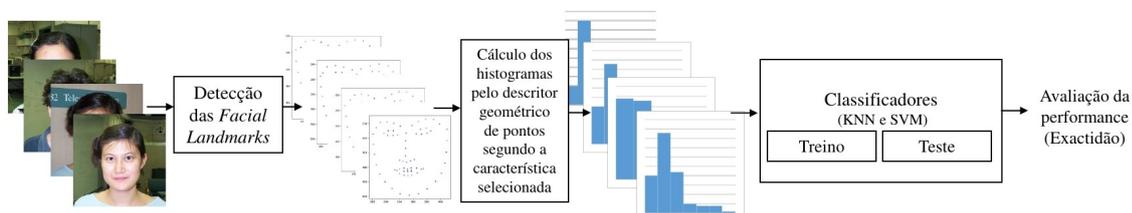


Figura 5.8: Sistema de avaliação da performance do descriptor geométrico de pontos quando aplicado em imagens reais de faces.

### 5.1.6 Implementação de um algoritmo com a mesma metodologia para verificação e comparação de resultados

De modo a ter um meio de comparação, e também ser possível verificar a veracidade dos resultados obtidos nos testes que envolveram o descriptor geométrico de pontos, relativamente à utilização de padrões de diferentes naturezas, foi implementado o algoritmo de Brzakovic [9], que é dos poucos algoritmos semelhantes ao método descrito anteriormente, encontrados na literatura, que também utiliza informação geométrica para o reconhecimento.

A implementação do algoritmo seguiu os primeiros dois níveis e respectivos passos detalhadamente descritos na secção 3.2.

### 5.1.7 Adição de um sistema de *scrambling*

No decorrer do estudo do descriptor geométrico de pontos verificou-se que este obtém uma melhor performance quando aplicado a dispersões aleatórias do que em dispersões estruturadas (como é o caso das faces), sendo esta diferença muito significativa. Assim, pretendeu-se testar a tese de que se aumentarmos o grau de aleatoriedade deste último tipo de dispersões, por meio de sistema de *lifting* ou *scrambling*, a performance aumentaria também. Deste modo, surgiu ideia de utilizar funções de sentido único (*one-way functions*) na base do algoritmo destinado ao *scrambler*. Tal como introduzido na secção 3.1, ainda que aplicadas noutra contexto, estas são funções utilizadas na criptografia que calculam facilmente os valores de entrada segundo as operações

que as constituem, contudo, são praticamente impossíveis de inverter sem informações adicionais [62]. Foram testadas as funções candidatas RSA[65], Rabin[80] e Exponencial Modular[14], que no entanto, não corresponderam às expectativas. Isto porque, as pequenas diferenças entre as *facial landmarks* da mesma identidade resultantes do processo de obtenção e extração, quando submetidas ao sistema de *scrambler* baseado em funções de sentido único tornaram-se mais evidentes e exageradas, o que se traduziu numa redução drástica da performance.

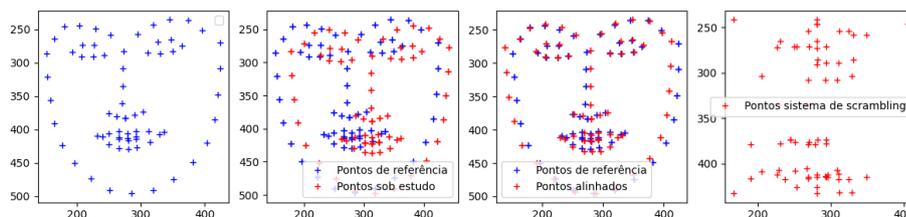


Figura 5.9: Figura exemplificativa do pipeline do sistema de *scrambling*.

Como alternativa, apostou-se num sistema que ao invés de introduzir alterações nos valores das coordenadas, apenas baralhava os índices das mesmas, tanto da coordenada  $x$  como da coordenada  $y$ , independentemente uma da outra, através de uma função que atribuía de forma aleatória os índices das 68 *facial landmarks*. Isto permitiu conservar os seus valores, visto que apenas altera a sua disposição, tal como é possível observar pelo exemplo elucidativo presente na tabela 5.1. Assim, as coordenadas agora atribuídas de forma desordenada formam um padrão com propriedades aleatórias, exemplificado no último *plot* da figura 5.9.

Pos inicial	0	1	2	3	4	5	6	7	8	9	10	...	67
X	152	154	158	165	177	196	222	251	285	319	351	...	266
Y	286	321	356	391	423	450	473	491	496	491	474	...	412
Pos X	67	3	5	8	1	7	0	6	2	9	4	...	10
Novo X	266	165	196	285	154	251	152	222	158	319	177	...	351
Pos Y	10	67	3	7	0	1	5	2	9	8	4	...	6
Novo Y	474	412	391	491	286	321	450	356	491	486	423	...	473

Tabela 5.1: Exemplo do funcionamento do sistema *scrambler* desenvolvido.

## 5.2 Trabalho Experimental

### 5.2.1 Dados

2D face sets Aberdeen [17] e frontal face *dataset* de Markus Weber[79] foram os conjuntos de dados escolhidos para os procedimentos experimentais relacionados com o descritor de pontos. Os critérios por detrás da escolha destes conjuntos de dados foram definidos com base na necessidade de imagens frontais, ou seja, imagens em que o indivíduo estivesse de frente para a câmara, como

também na imposição de que teria de existir um número mínimo de imagens pertencentes a cada um dos indivíduos.

- **2D face sets Aberdeen** consiste em 687 imagens de faces a cores e foi criado por Ian Craw em Aberdeen. Possui entre 1 a 18 imagens pertencentes a 90 indivíduos diferentes, onde existem algumas variações de luz, como também, variações da pose em 8 identidades. A resolução varia entre 336x480 e 624x544.

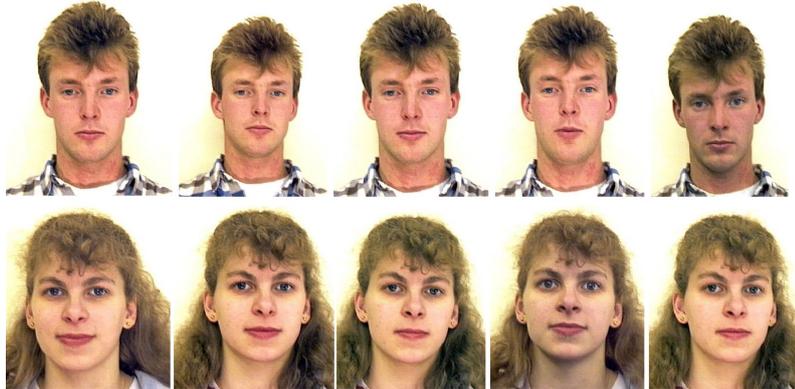


Figura 5.10: Conjunto exemplificativo do tipo de imagens encontradas no primeiro conjunto [17].

- **Frontal face dataset** coletado por Markus Weber no Instituto de Tecnologia da Califórnia. Possui 450 imagens de faces com dimensões iguais a 896x592 em formato JPEG, representativas de 27 indivíduos diferentes, em que estão presentes algumas variações de luz, expressões faciais e fundos.



Figura 5.11: Conjunto exemplificativo do tipo de imagens encontradas no segundo conjunto [79].

## 5.2.2 Métricas

### 5.2.2.1 Exactidão

Com o objectivo de avaliar a performance dos diferentes procedimentos e classificadores, e deste modo avaliar a taxa de acerto, recorreu-se à métrica da exactidão. Esta corresponde à relação de resultados verdadeiros, mais especificamente à relação da soma entre os verdadeiros positivos e os verdadeiros negativos com o número total de casos examinados, que corresponde à soma entre

os verdadeiros positivos ( $vp$ ) com os verdadeiros negativos ( $vn$ ) com os falsos positivos ( $fp$ ) e com os falsos negativos ( $fn$ ) (equação 5.1).

$$accuracy = (vp + vn) / (vp + vn + fp + fn) \quad (5.1)$$

### 5.2.2.2 Precisão

A precisão, é uma métrica que representa a habilidade do classificador em não considerar uma amostra positiva quando na verdade é negativa [64]. Diz respeito à relação entre o número de verdadeiros positivos ( $vp$ ) e a soma entre o número de verdadeiros positivos e os falsos positivos ( $fp$ ) (equação 5.2).

$$Precisão = vp / (vp + fp) \quad (5.2)$$

### 5.2.3 Descrição Experimental

Para avaliar a performance do descritor geométrico de pontos aplicado em reconhecimento facial, foram feitos diferentes testes experimentais que envolveram dispersões de diferentes naturezas, diferentes tipos de classificadores, um algoritmo com a mesma lógica para verificação de resultados, assim como, diferentes abordagens na tentativa de melhorar os resultados, como é o caso do sistema de *scrambling*. Deste modo, para tornar a representação dos dados mais acessível foi criado um sistema de identificação, onde as características utilizadas, enunciadas na secção 5.1.4, ficam definidas por um número segundo a seguinte enumeração:

- **Característica 1** - Distâncias entre um ponto *pivot* e cinco pontos aleatórios;
- **Característica 2** - Número de vizinhos de um ponto;
- **Característica 3** - Ângulos formados por 3 pontos aleatórios;
- **Característica 4** - Ângulo formado por dois pontos e um vértice;
- **Característica 5** - Distâncias entre um ponto *pivot* e pontos aleatórios pertencentes a uma determinada vizinhança;
- **Característica 6** - Números de pontos por linha e coluna;
- **Característica 7** - Distâncias com peso por intervalo de distância.

Todas as características em cima enumeradas foram individualmente aplicadas nas dispersões divididas por quadrantes, isto quer dizer, que em vez de as relações geométricas incluírem todos os pontos pertencentes à dispersão, apenas são consideradas para o efeito as relações entre pontos do mesmo quadrante. Assim, e para continuar a mesma lógica de etiquetagem, foram atribuídas as seguintes legendas:

- **Característica 1Q** - Distâncias entre um ponto *pivot* e cinco pontos aleatórios, com dispersão dividida por quadrantes;

- **Característica 2Q** - Número de vizinhos de um ponto, com a dispersão dividida por quadrantes;
- **Característica 3Q** - Ângulos formados por 3 pontos aleatórios, com a dispersão dividida por quadrantes;
- **Característica 4Q** - Ângulo formado por dois pontos e um vértice, com a dispersão dividida por quadrantes;
- **Característica 5Q** - Distâncias entre um ponto *pivot* e pontos aleatórios pertencentes a uma determinada vizinhança, com a dispersão dividida por quadrantes;
- **Característica 6Q** - Números de pontos por linha e coluna, com a dispersão dividida por quadrantes;
- **Característica 7Q** - Distâncias com peso por intervalo de distância, com a dispersão dividida por quadrantes.

Como já foi mencionado anteriormente foram utilizadas dois conjuntos de dados, devidamente identificados na secção 5.2.1. O conjunto de dados de Aberdeen [17] possui menos variações tanto ao nível de posição quanto ao nível de expressões faciais, mas em contrapartida têm uma menor qualidade de imagem e o número de imagens por cada identidade é limitado. O segundo conjunto de dados [79] é mais desafiante, pois possui imagens com mudanças mais bruscas de iluminação, movimentos e expressões faciais, contudo, possui um número superior de imagens por identidade em relação ao *dataset* anterior. As diferenças entre os dois *datasets* tornam possível ter uma noção mais generalizada acerca do desempenho do descritor de pontos quando aplicado em imagens de faces capturadas em diferentes contextos.

Tal como referido na secção 5.1.1, é necessário que haja divisão do conjunto de imagens em dois grupos, em que uma parte fica destinada à fase de treino e outra para a fase de teste. Esta distribuição fica ao encargo do modelo de avaliação *K-fold Cross Validation*, em que *K* ficou definido pelo número 3. Desta forma, por causa destas restrições e também por algumas dependências do algoritmo em si, apenas foi possível aproveitar uma parte do número total de imagens presente nas respectivas bases de dados.

- Relativamente à primeira base de dados [17], foram utilizadas 30 identidades diferentes com oito imagens por identidade que perfazem um total de 240 imagens. Este número é dividido em 3 subconjuntos equilibrados (80 imagens por cada grupo), que de forma rotativa são utilizados para teste enquanto, que as restantes compõem o treino.
- No que diz respeito à segunda base de dados [79], foram consideradas para o estudo 18 identidades com 19 imagens por identidade, o que resulta num total de 342 imagens. A divisão segue a mesma lógica que no ponto anterior, sendo os grupos compostos por 114 imagens.

Tendo em consideração que o objecto de estudo são imagens de faces, em que os pontos salientes compõem dispersões estruturadas, torna-se importante ter uma comparação directa com

as dispersões aleatórias, que já é conhecido à priori que atingem bons resultados, tendo como referencia o que foi mencionado no documento [18]. Assim, os testes que recriaram as dispersões de natureza aleatória seguiram as mesmas características numéricas dos testes com dispersões estruturadas, já que estas se encontram limitadas pelo número de imagens presentes nas base de dados analisadas.

#### 5.2.4 Detalhes da implementação

Relativamente aos detalhes técnicos da implementação da primeira parte, todo o programa foi desenvolvido em python 3.8 em ambiente Ubuntu 20.04.

Para o processo de detecção e extracção das *facial landmarks*, que foi disponibilizado pela equipa de pesquisa e desenvolvimento VIS Team do Instituto de Sistemas e Robótica da Universidade de Coimbra, este utiliza a biblioteca multiplataforma OpenCV que possui funções específicas para leitura e tratamento de imagens. O próprio descritor geométrico de pontos original foi concedido pela mesma equipa, responsável pela sua criação no projecto UniqueMark [18]. Todas as operações básicas sobre matrizes foram possíveis através do pacote *Numpy*, e os gráficos das dispersões de pontos com o módulo *Matplotlib*. A etapa de classificação iniciou-se com a optimização dos classificadores, KNN e SVM, através da alteração de alguns dos seus parâmetros (tabela 5.2), que foram seleccionados tendo em conta os valores mais altos de exactidão possivelmente alcançados quando aplicados aos conjuntos de dados sob estudo, tendo sido estes implementados através da biblioteca *scikit-learn* (pelas funções *KNeighborsClassifier*, *SVC*). Como mencionado na secção 5.1.1, foi empregue a validação cruzada, implementada pela função *cross\_val\_score*, que também pertence à mesma biblioteca, *scikit-learn*.

Classificador	Parâmetro	Valores
KNN	Weights	"uniform", " <b>distance</b> "
	K	1-10 ( <b>7</b> )
SVM	Kernel	"linear", "poly", " <b>rbf</b> ", "sigmoid"
	Regularização do parâmetro C	1-10 ( <b>7</b> )

Tabela 5.2: Parâmetros avaliados na etapa de optimização dos classificadores.

#### 5.2.5 Resultados e sua análise

##### 5.2.5.1 Descritor geométrico de pontos

###### Dispersões aleatórias sintéticas:

De modo a conferir às amostras sintéticas características reais, foram adicionadas transformações geométricas às mesmas. Nos resultados abaixo apresentados, apenas vinte por cento dos pontos foram translacionados por um raio máximo de 10 pixels, isto quer dizer que dos 68 pontos que constituem a dispersão, 14 pontos sofreram translação e obtiveram novas coordenadas que varia entre 0 a 10 pixels de deslocamento. De uma perspectiva de deslocamento relativo, e tendo

em conta que nas imagens utilizadas a dimensão das faces ronda os  $300 \times 300$  (comprovado pela figura 5.2), isto equivale a um deslocamento relativo de 3%.

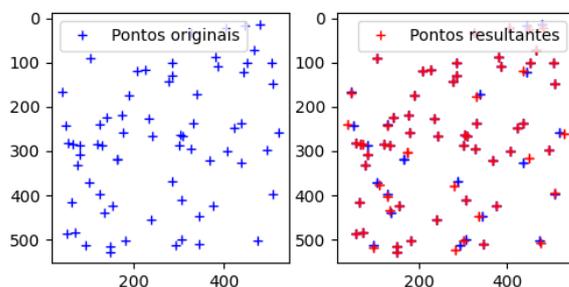


Figura 5.12: Exemplo do resultado das transformações geométricas aplicadas a um padrão aleatório sintético.

(a) Classificador KNN				(b) Classificador SVM			
Feat	Exac	Feat	Exac	Feat	Exac	Feat	Exac
1	0.987	1Q	<b>1.000</b>	1	0.975	1Q	<b>1.000</b>
2	0.721	2Q	0.996	2	0.708	2Q	0.996
3	<b>1.000</b>	3Q	0.996	3	<b>1.000</b>	3Q	<b>1.000</b>
4	<b>1.000</b>	4Q	<b>1.000</b>	4	<b>1.000</b>	4Q	<b>1.000</b>
5	0.987	5Q	<b>1.000</b>	5	0.983	5Q	<b>1.000</b>
6	<b>1.000</b>	6Q	<b>1.000</b>	6	0.996	6Q	<b>1.000</b>
7	0.987	7Q	<b>1.000</b>	7	0.983	7Q	<b>1.000</b>

Tabela 5.3: Resultados dos testes descritor geométrico de pontos com padrões sintéticos de dispersões aleatórias.

#### Padrões de dispersões estruturadas (faces):

(a) Classificador KNN				(b) Classificador SVM			
Feat	Exac	Feat	Exac	Feat	Exac	Feat	Exac
1	0.579	1Q	0.696	1	0.667	1Q	0.799
2	0.563	2Q	0.696	2	0.675	2Q	0.800
3	0.379	3Q	0.700	3	0.158	3Q	0.542
4	0.533	4Q	0.667	4	0.588	4Q	0.796
5	0.579	5Q	0.692	5	0.667	5Q	0.799
6	0.799	6Q	<b>0.871</b>	6	0.908	6Q	<b>0.929</b>
7	0.579	7Q	0.696	7	0.667	7Q	0.799

Tabela 5.4: Resultados dos testes do descritor geométrico de pontos aplicado a dispersões estruturadas obtidas das imagens de faces em que o *dataset* de validação é constituído por 240 imagens, com 8 imagens distribuídas de forma uniforme por 30 identidades, pertencentes à 1ª Base de dados (2D face sets Aberdeen).

(a) Classificador KNN				(b) Classificador SVM			
Feat	Exac	Feat	Exac	Feat	Exac	Feat	Exac
1	0.781	1Q	0.742	1	0.883	1Q	0.863
2	0.781	2Q	0.693	2	0.819	2Q	0.825
3	0.596	3Q	0.637	3	0.404	3Q	0.643
4	0.678	4Q	0.746	4	0.749	4Q	0.830
5	0.781	5Q	0.739	5	0.883	5Q	0.863
6	<b>0.886</b>	6Q	0.871	6	<b>0.929</b>	6Q	0.906
7	0.781	7Q	0.743	7	0.883	7Q	0.863

Tabela 5.5: Resultados dos testes do descritor geométrico de pontos aplicado a dispersões estruturadas obtidas das imagens de faces em que o *Dataset* de avaliação é constituído por 342 imagens, com 19 imagens distribuídas de forma uniforme por 18 identidades, pertencentes à 2ª Base de dados (*Frontal face dataset coletado por Markus Weber*).

- O classificador SVM demonstrou uma certa vantagem em relação ao classificador KNN. Apesar dos valores de exactidão em algumas características serem semelhantes, de uma forma geral, a maioria dos descritores têm um melhor desempenho com o classificador SVM.
- Como podemos ver pelos resultados, estes são consideravelmente melhores quando os descritores são aplicados a dispersões aleatórias do que em dispersões estruturadas. Enquanto que os resultados referentes ao primeiro caso estão próximos do valor óptimo, o mesmo não se verifica com as dispersões estruturadas, em que os valores ficam aquém das expectativas.
- Entre os resultados obtidos pelas dispersões estruturadas, referentes aos diferentes *datasets*, existe uma discrepância entre os dois, sendo que os resultados que dizem respeito ao segundo conjunto de dados mostraram-se superiores. Acredita-se que isto se deva ao facto de o segundo *dataset* possuir um maior número de imagens por cada identidade, o que se sobrepõe ao facto deste *dataset* apresentar maiores variações no que diz respeito a expressões faciais, fundos e iluminações comparativamente ao primeiro *dataset*.
- Atendendo aos resultados obtidos, surgiu a tese de que o grau de aleatoriedade terá influência na performance do método tal como ele é. Isto quer dizer que se for possível aumentar este parâmetro nas dispersões estruturadas, na expectativa de ampliar as diferenças inter-classes (assumindo que este efeito é superior ao efeito de separação intra-classes), deste modo, serão obtidos os resultados desejados.

### 5.2.5.2 Algoritmo de Brzakovic

- Para a avaliação deste método fez sentido utilizar uma outra métrica, a precisão. No método anterior apenas se optou por uma das métricas porque em testes preliminares, tanto precisão como exactidão originavam valores semelhantes. Através da observação dos resultados obtidos nas dispersões estruturadas é possível concluir que a baixa precisão resulta da

Precisão	Exactidão
1.00	1.00

Tabela 5.6: Resultados do teste do método de Brzakovic utilizando dispersões aleatórias obtidas de padrões gerados sinteticamente. Foram considerados 30 padrões gerados aleatoriamente, cada qual com 8 correspondências geradas através de transformações geométricas dos padrões originais, perfazendo um total de 240 padrões.

Precisão	Exactidão
0.39	0.94

Tabela 5.7: Resultados do teste do método de Brzakovic utilizando dispersões estruturadas obtidas pelo *dataset* constituído por 240, com 8 imagens distribuídas de forma uniforme por 30 identidades, pertencentes à 1ª Base de dados (*2D face sets Aberdeen*).

Precisão	Exactidão
0.43	0.96

Tabela 5.8: Resultados do teste do método de Brzakovic utilizando dispersões estruturadas obtidas pelo *dataset* constituído por 342, com 19 imagens distribuídas de forma uniforme por 18 identidades, pertencentes à 2ª Base de dados (*Frontal face dataset coletado por Markus Weber*).

incorrecta classificação de imagens que o algoritmo deveria identificar como sendo correspondentes, o que resulta em muitos falsos positivos. Em contrapartida, consegue identificar correctamente a vasta maioria das não correspondências, os verdadeiros negativos, o que justifica os altos valores de exactidão.

- Assim, constatamos que o problema do reconhecimento de dispersões aleatórios continua a ser mais bem-sucedido do que em dispersões estruturadas. Portanto, temos mais uma evidência de que a tese descrita anteriormente, que refere que o grau de aleatoriedade das dispersões sob estudo pode efectivamente influenciar o sucesso de métodos que têm por base as relações geométricas entre os pontos que a constituem, poderá ser verdadeira.

### 5.2.5.3 Adição de um sistema de *scrambling*

- É relevante dizer que a exactidão das relações geométricas calculadas pelos descritores, como é o caso das distâncias e dos ângulos, é directamente afectada pela qualidade das características, que podem ser degradadas pelas naturais variações das representações dos pontos e presença de ruído, assim como, falsas correlações presumidas entre classificações.
- Quando se verificou que de facto os descritores iniciais não eram adequados a este tipo de dispersões, surgiu a necessidade de alterar de algum modo as estruturas das mesmas, fornecendo-lhe propriedades que à partida resultariam numa melhoria da resposta dos descritores. Deste modo, surgiu o interesse em aumentar a aleatoriedade dos conjuntos de

(a) Classificador KNN				(b) Classificador SVM			
Feat	Exac	Feat	Exac	Feat	Exac	Feat	Exac
1	0.760	Q1	0.754	1	0.836	1Q	0.845
2	0.649	Q2	0.622	2	0.728	2Q	0.766
3	0.643	Q3	0.643	3	0.363	3Q	0.669
4	0.836	Q4	0.784	4	0.865	4Q	0.854
5	0.760	Q5	0.754	5	0.836	5Q	0.845
6	0.804	Q6	<b>0.847</b>	6	0.865	6Q	<b>0.897</b>
7	0.760	Q7	0.754	7	0.836	7Q	0.845

Tabela 5.9: Resultados dos testes do sistema de *scrambling* aplicado em dispersões estruturadas resultantes do *Dataset* de teste constituído por 270 com 19 imagens distribuídas de forma uniforme por 18 identidades, pertencentes à 2ª Base de dados (*2D face sets Aberdeen*)

pontos utilizados e, como é possível comprovar pelos resultados representados, as expectativas não foram correspondidas. Apesar da maioria dos descritores geométricos classificados segundo o KNN terem obtido ligeiros aumentos de performance com a implementação do sistema de *scrambling*, os valores adquiridos sem estes sistemas através do classificador SVM, mostraram-se superiores.

- No contexto de biometria, mais precisamente em reconhecimento facial, é extremamente difícil encontrar duas amostras de pontos salientes que produzam exactamente o mesmo conjunto de características. Isto deve-se às diferenças na aparência facial geradas por movimentos, obtenção das imagens das faces, ambiente envolvente (condições de iluminação), entre outras. Neste cenário de classificação de multi-classes, o objectivo dos métodos de classificação é sempre aumentar a distância inter-classes e reduzir a das intra-classes. O que se comprovou é que tanto os descritores utilizados como o sistema de *scrambling*, fazem o oposto do pretendido, ou seja, são responsáveis por aumentar as dispersões intra-classes, o que justifica a baixa performance dos métodos testados.
- Tendo isto em conta, e considerando o objectivo desta dissertação em assingelar um dos descritores e atendendo aos resultados obtidos, o descritor que apresentou a melhor performance, conseguindo ultrapassar a marca dos 0.90 quando aplicado às faces do segundo *dataset* sem o uso de sistema de *scrambling*, foi o descritor representado pelo número 6, em que a característica geométrica é do tipo local e se baseia na contagem do número de pontos por linha e no número de pontos por coluna, sendo que o histograma final resulta da soma destes dois.

## Capítulo 6

# 2ª Parte - Método *Pure Deep Learning*

Depois de testar os descritores baseados em relações geométricas entre pontos resultantes de características manuais, em conjunto com classificadores convencionais de *machine learning*, chegou a vez de avaliar o potencial de redes artificiais neuronais no mesmo contexto, passando assim para uma abordagem de *pure deep learning*. De um modo geral, neste capítulo pretende-se:

- Afinação da dimensão da largura da rede convolucional definida para este estudo.
- Afinação da dimensão das *features* responsáveis pelo processo de classificação.
- Análise do impacto na performance de diferentes funções de perda na rede resultante do processo de optimização dos pontos anteriores.
- Comparação com o método clássico estudado no capítulo anterior (capítulo 5).

### 6.1 Metodologia

#### 6.1.1 Estrutura de avaliação

Tal como no capítulo anterior, será utilizado um conjunto de imagens da base de dados para treino e um conjunto mais pequeno para teste. Ao conjunto definido para treino da rede neuronal artificial são seleccionadas aleatoriamente algumas imagens, de modo a criar um subconjunto destinado para a validação. Esta divisão é essencial para a regularização deste tipo de métodos, sendo uma prática comum para evitar problemas como o *overfitting*.

#### 6.1.2 Filtragem e pre-processamento das imagens

A base de dados seleccionada (VGGFace2 [10], detalhada mais à frente na secção 6.2.1) para a avaliação deste método é dedicada ao objecto de estudo, no entanto, é conhecida por conter todo o tipo de imagens de faces, desde imagens que contêm mais do que um rosto por imagem, a capturas dos mais variados ângulos da face (que incluem perfis e faces parcialmente ocultas), até as mais diversas variações de iluminação e fundos. Para evitar o declínio da performance provocada pelas

variações das imagens descritas anteriormente, como também, reproduzir as condições semelhantes ao método estudado na primeira parte da dissertação, todas as imagens contidas nesta base de dados passaram por um processo de filtragem, onde apenas as imagens com faces frontais foram seleccionadas para o próximo passo.

Neste processo todas as imagens são analisadas, com o intuito de detectar o alinhamento da face principal, considerando a hipótese de que uma imagem pode conter mais do que uma face. Caso se verifique os requisitos necessários para ser considerada uma imagem facial frontal, esta é recortada e guardada, de modo a fazer parte do dataset final, composto apenas por imagens com as características desejadas. Este alinhamento é conseguido por meio da biblioteca MTCNN (*Multi-Task Cascaded Convolutional Neural Network*) [83], responsável pela detecção das faces e consequente localização de pontos referentes aos olhos, nariz e cantos da boca, que por meio das relações geométricas entre estes, determina se se trata de uma imagem de uma face frontal ou não.

O pré-processamento de uma imagem é um passo importante que antecede a extracção das características. Desta forma, é possível realçar os atributos e transformar as imagens que posteriormente serão disponibilizadas à rede, ajudando assim no processo de aprendizagem dos modelos empregues. Com o objectivo de normalizar os dados de entrada, neste caso as imagens, e a fim de reduzir a redundância e melhorar a integridade dos mesmos é aplicado o filtro da subtracção da média, em que a média é referente aos valores RGB de todas as imagens que constituem o *dataset* final. Para além disto, as imagens foram novamente redimensionadas de modo a incluir apenas a região de maior interesse.

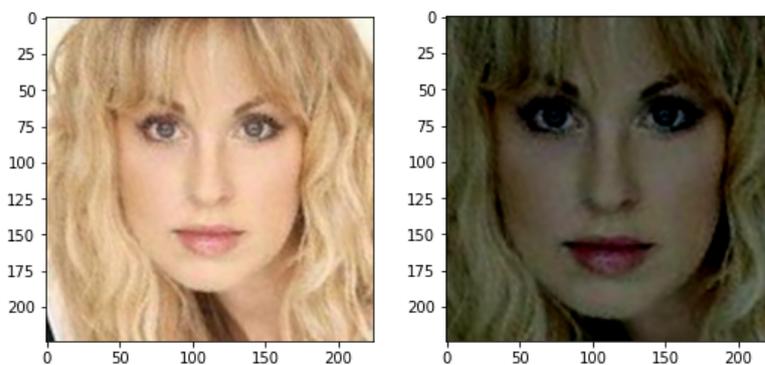


Figura 6.1: Exemplo de imagem resultante do processo de filtragem. O antes e depois do pré-processamento.

### 6.1.3 MobileNetV2

Como já foi mencionado no capítulo 4, secção 4.1, onde foi feita uma breve introdução ao reconhecimento facial, houve um evento que promoveu um impulsionamento da investigação nesta área, o desafio *ImageNet ILSVRC*. Esta competição tem como objectivo a avaliação de algoritmos para a detecção de objectos e classificação de imagens em grande escala, permitindo que investigadores comparem os progressos desenvolvidos em detecção e reconhecimento, numa ampla

variedade de objectos [67]. Deste modo, foram estudadas uma vasta gama de redes notórias pelos seus bons resultados. Um bom exemplo desse tipo de rede é a MobileNet [37], um modelo popularmente utilizado, que mostrou ser capaz de obter o mesmo nível de precisão que a rede VGG-16, quando aplicada a ImageNet (grande base de dados projectada para ser utilizada em investigação de software para reconhecimento de objectos) com apenas 1/30 do custo computacional [40], sendo umas das redes mais pequenas actualmente conhecidas, com um tamanho de apenas 16 MB. Isto é possível porque os blocos que a constituem são blocos de convolução separáveis em profundidade, isto significa que a convolução padrão é factorizada numa convolução em profundidade que realiza uma filtragem leve e onde aplica um único filtro convolucional por canal de entrada, seguida de uma convolução  $1 \times 1$ , chamada de *pointwise convolution*, que fica responsável por construir novas características por meio do cálculo de combinações lineares dos canais de entrada [68].

No seguimento deste modelo, surgiu a MobileNetv2 [68]. A MobileNetv2 segue a mesma ideologia que a MobileNetV1 [37] ao utilizar também a convolução em profundidade como blocos de construção (ou blocos de convolução). No entanto, esta versão introduziu algumas diferenças relacionadas com inserção de *bottlenecks* entre camadas e estrutura residual invertida (também conhecidos como atalhos entre os *bottlenecks*). A MobileNetV2 combina convolução *pointwise* com *bottlenecks*, e usa *pointwise* para construir o *bottleneck* propriamente dito. É empregue activação linear ao *bottleneck* de modo a evitar que a informação seja perdida devido à aplicação da função ReLU aos dados resultantes da redução de dimensionalidade. Em relação à inversão de resíduos, é adicionada uma camada de expansão ao início do bloco, com o intuito de adicionar alguma não linearidade ao modelo, sendo que um atalho nesta posição significa que é usado o resumo da adição do input e *output* como saída de todo o bloco com a intenção de melhorar a propagação dos gradientes. Posto isto, nesta rede existem dois tipos de blocos, representados na figura 6.2, em que um é conhecido como bloco residual identificado com *stride* 1, e outro, habitualmente utilizado para a redução de dados, definido com *stride* 2.

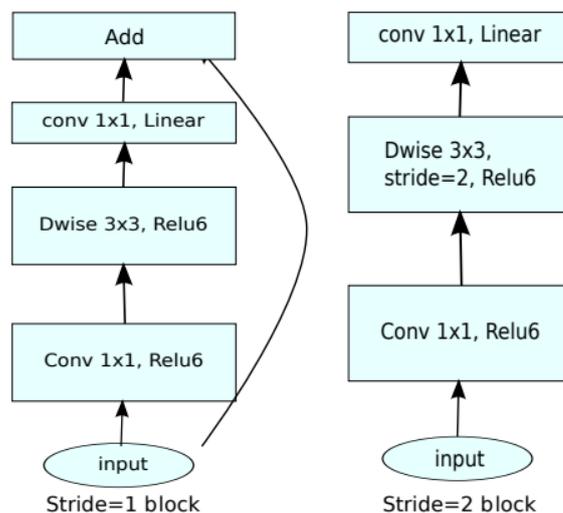


Figura 6.2: Blocos de construção característicos da arquitetura MobileNetV2 [68].

Input	Operador	Ouput
$h \times w \times k$	$1 \times 1$ conv2d, ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	$3 \times 3$ dwise $s=s$ , RELU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear $1 \times 1$ conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Tabela 6.1: *Bottleneck residual block* - Considerando  $h \times w \times k$  como as dimensões de entrada de cada camada, é possível observar a estrutura específica de um bloco de *bottleneck* residual, que transforma  $k$  para  $k'$  canais, com *stride*,  $s$ , e factor de expansão,  $t$  [68].

Pelos motivos enunciados e por todas as vantagens que a utilização do modelo MobileNetV2 oferece, sem desprezar o facto de que esta rede possui um tamanho de apenas 14 MB, ficou definido que este seria o modelo a ser explorado nesta segunda parte da dissertação. A sua arquitectura geral tem as seguintes propriedades:

Input	Operador	$t$	$c$	$n$	$s$
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d $1 \times 1$	-	1280	1	1
$7^2 \times 1280$	avgpool $7 \times 7$	-	-	1	-
$1 \times 1 \times 1280$	conv2d $1 \times 1$	-	$k$	-	-

Tabela 6.2: Arquitectura geral da MobileNet [68].

onde  $t$  representa o factor de expansão,  $c$  o número de canais de saída, e  $n$  o número de repetições. A primeira camada de cada sequência tem um *stride*,  $s$ , de 2 enquanto que as restantes usam *stride* igual a 1. Todas as convoluções espaciais usam *kernels*  $3 \times 3$ .

#### 6.1.4 Optimização de parâmetros

Com a intenção de otimizar a performance e, consequentemente definir os melhores parâmetros para uma rede onde temos como motivação o baixo consumo computacional, sem desprezar o rigor do método sob estudo, foram testadas diferentes redes produzidas através da alteração de alguns dos parâmetros do modelo MobileNetV2 (representada pela figura 6.3).

Inicialmente o estudo incidiu-se entre a relação de compromisso entre largura da rede (que influencia directamente a arquitectura da rede) e o número de características (*features number*) a utilizar durante processo de classificação. Uma das novidades que advém da criação do modelo MobileNetV1 e consequentemente MobileNetv2 é a introdução de dois hiper-parâmetros globais: o parâmetro da largura e o parâmetro da resolução, que equilibram eficientemente a eficácia e a

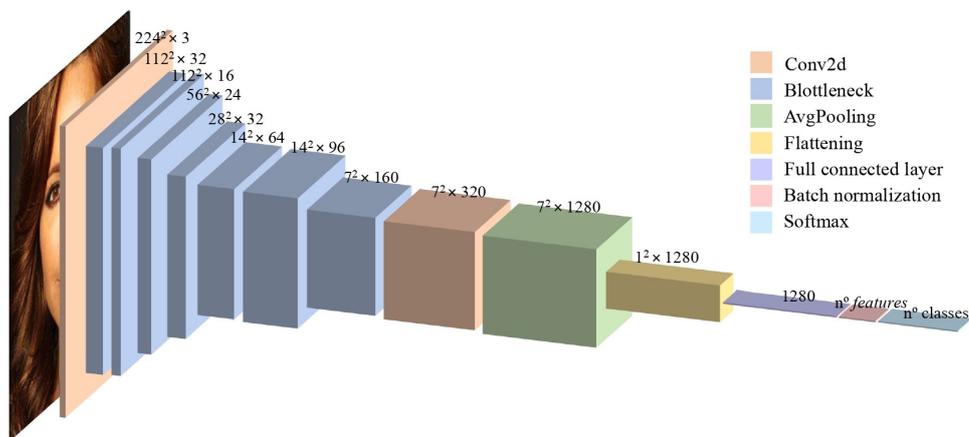


Figura 6.3: Arquitetura da rede utilizada.

precisão, permitindo escolher o modelo correto para cada aplicação baseada em problemas reais [25]. Ao definir o parâmetro da largura, que na função disponibilizada pela *Keras Applications* é simbolizado por  $\alpha$ , é alterado o número de filtros por cada camada e, por conseguinte, a largura da rede. Redes mais largas tendem a ser capazes de capturar características mais refinadas (fine-grained) sendo mais fáceis de treinar. No entanto, redes muito largas mas superficiais tendem a ter dificuldades em capturar características de níveis superiores [81]. Segundo [13], um valor de  $\alpha < 1.0$ , faz decrescer proporcionalmente o número de filtros em cada camada, em contrapartida,  $\alpha > 1.0$  aumenta proporcionalmente o número de filtros por cada camada. Caso se mantenha o valor por defeito ( $\alpha = 1.0$ ), é usado o número de filtros definido do documento [68].

Considerando agora o número de *features*, e atendendo que de um ponto de vista abstracto uma rede neuronal é considerada um sistema de caixa preta, no sentido em que não é possível conhecer quais as características utilizadas durante o processo de classificação, muito menos ter opinião acerca da selecção das mesmas. Contudo, é possível definir o número deste parâmetro, o que corresponde à acção de delimitar o número de nós (neurónios) que constituirão a penúltima camada da rede (camada que antecede à classificação propriamente dita, que inicialmente será definida pela função de perda *Softmax Cross Entropy*). Para proceder a esta escolha, basta alterar os parâmetros de entrada da camada *Batch normalization* representada pela cor rosa na figura 6.3. Naturalmente, esta decisão terá grandes implicações no processo de previsão da respectiva classe, o que afectará a precisão do sistema, o tamanho da rede e consequentemente o esforço computacional.

Após serem encontrados os parâmetros referentes à largura da arquitectura e ao número de características que melhor satisfazem as necessidades do nosso objecto de estudo e respectivas motivações, a intenção continua a focar-se no aprimoramento do modelo. Para otimizar mais a rede resultante do processo anterior, surge a necessidade de alterar a função de perda (definida anteriormente pela função padrão *Softmax*), através da implementação de alternativas mais recentes do estado da arte, dando preferência àquelas que são dedicadas ao problema do reconhecimento

facial. Foram escolhidas para este efeito, funções que têm por base a função *Softmax* com a adição de diferentes tipos de penalidade de margem, nomeadamente: margem angular multiplicativa, margem angular aditiva e margem de cosseno aditiva, também conhecidas como *SphereFace* [54], *ArcFace* [21], *CosFace* [78], respectivamente, e onde os fundamentos teóricos foram abordados na secção 4.2. Nos modelos definidos para trabalho experimental foram empregues os valores das margens definidos por defeito para cada uma das funções.

### 6.1.5 Treino

O processo de treino fez uso da técnica de *fine tuning* que é baseada no conceito de *transfer learning*, que consiste em transferir informações de uma arquitectura previamente treinada. A *keras applications* (API utilizada) disponibiliza esta opção para o modelo seleccionado (a *MobileNetV2*). Assim, em vez dos pesos serem aleatoriamente inicializados, são utilizados os valores dos pesos pré-treinados referentes a base de dados *ImageNet*. No que diz respeito aos parâmetros de treino da rede, para além dos já descritos na secção anterior, e que se mantiveram idênticos em todos os modelos estudados, foram definidos os seguintes critérios:

Parâmetro	Valor
Número de classes	5095
Número de <i>epochs</i>	20
<i>Batch size</i>	16
Optimizador	SGD
<i>Learning rate</i>	0.01
<i>Weight decay</i>	$1^{-05}$
<i>Momentum</i>	0.5

Tabela 6.3: Parâmetros de treino comuns a todas as redes.

É também importante mencionar que de forma a garantir um treino mais minucioso, apenas foram utilizadas identidades com mais de 100 imagens.

### 6.1.6 Teste

Depois de treinar uma rede, existe a necessidade de testá-la de forma a avaliar a sua performance num contexto real e verificar a existência de erros. Tal como previsto, foi utilizado um lote mais pequeno de imagens da base de dados definida para o efeito, que também foi submetido ao processo de filtragem e pré-processamento, descritos na secção 6.1.2. Deste modo, foram desenhados três protocolos diferentes com o objectivo de colmatar diferentes problemas:

- **Protocolo 1** - Utiliza 50 identidades cuidadosamente seleccionadas de modo a tornar o *dataset* final para teste equilibrado tanto em género, como a nível racial. São usadas um total de 5210 imagens distribuídas de forma não uniforme pelas diferentes identidades.

- **Protocolo 2** - Na validação de um método de reconhecimento de objectos, e atendendo que métrica definida para avaliação deste método foi a curva ROC (introduzida mais à frente na secção 6.2.2.1), durante o processo de testagem, o algoritmo ao analisar uma determinada classe, percorre todas as imagens do dataset, estipulando se cada uma delas é correspondente à classe sob estudo, ou não. Sabendo que apenas uma pequena porção da base de dados pertence de facto à mesma identidade, isto faz com que o número de correspondências estabelecidas como verdadeiras seja muito inferior ao número de correspondências estabelecidas como falsas, resultando assim num desnível entre estas duas componentes. Com a intenção de corrigir este desequilíbrio, o segundo protocolo utiliza o mesmo *dataset* definido anteriormente, com a diferença de que em vez de considerar o número total de correspondências identificadas como não correspondentes, apenas considera uma parte desse número na construção da curva ROC, ficando assim estipulado que para cada uma das identidades apenas é considerado o equivalente a 10% do número total de imagens.
- **Protocolo 3** - É o protocolo de maiores dimensões. Utiliza um total de 492 identidades, o que perfaz um total de 28.205.989 imagens distribuídas de forma não uniforme pelas diferentes identidades. Tal como no protocolo anterior, este procura equilibrar a relação entre correspondências e não correspondências estabelecidas para as diferentes classes sob análise. Para isso, utiliza um rácio de 1 para 2, de correspondências estabelecidas como verdadeiras correspondências para as correspondências estabelecidas como falsas, isto significa que consoante o número de correspondências estabelecidas como verdadeiras são consideradas na avaliação do modelo (Curva ROC neste caso) duas vezes mais correspondências estabelecidas como não verdadeiras, seleccionadas de forma aleatória de modo a evitar qualquer tipo de enviesamento de dados.

## 6.2 Trabalho Experimental

### 6.2.1 Dados

Neste tipo de abordagem, que são as redes neuronais artificiais, é necessário garantir que esta seja treinada por um grande número de imagens, caso contrário a rede irá aprender dependências e relações inadequadas, exclusivas do conjunto de imagens utilizadas no treino. Por este motivo não foi possível reutilizar os *datasets* utilizados durante o estudo do método clássico (secção 5.2.1), tendo sido necessário encontrar bases de dados de maiores dimensões. Posto isto, foi seleccionada para este estudo a base de dados VGGFace2 [10] que inclui mais de nove mil identidades com cerca de 80 a 800 imagens, ultrapassando um total de 3 milhões de imagens.

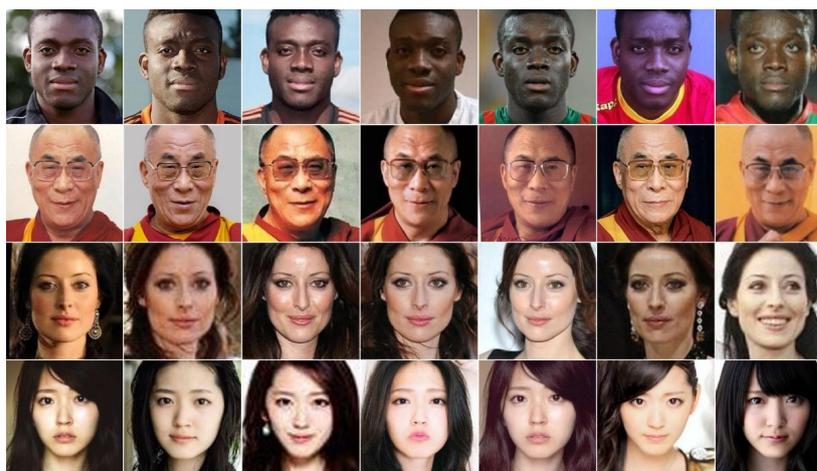


Figura 6.4: Amostra de imagens presentes no *dataset* VGGFace2 [10], utilizado para treino e testagem das redes neuronais artificiais após a aplicação do processo de filtragem descrito na secção 6.1.2.

## 6.2.2 Métricas

### 6.2.2.1 Curva ROC

A métrica utilizada para avaliação da performance das redes neuronais convolucionais é a Curva ROC. Estas curvas são representações gráficas frequentemente utilizadas na ilustração do desempenho de sistemas dedicados a problemas de decisão binária e mostram como o número de exemplos classificados correctamente como positivos (verdadeiros positivos -  $vp$ ) variam com o número de exemplos classificados incorrectamente como negativos (falsos negativos -  $fn$ ) [19]. Este tipo de curva representa graficamente a razão entre a *sensibilidade* (equação 6.1) e  $(1 - \textit{especificidade})$  (equação 6.2)).

$$\textit{Sensibilidade} = vp / (vp + fn) \quad (6.1)$$

$$\textit{Especificidade} = vn / (vn + fp) \quad (6.2)$$

Idealmente, a melhor relação de compromisso entre o valor de verdadeiros positivos e verdadeiros negativos e, por conseguinte, o valor que fornece a maior *sensibilidade* e maior *especificidade*, é obtido na representação gráfica deste tipo de curvas e corresponde ao ponto mais alto do eixo vertical e o mais distante à esquerda no eixo horizontal [24]. A área da curva representada é reconhecida como a medida do poder discriminatório do problema sob análise, e por isso uma boa ferramenta para medir o desempenho de um sistema.

### 6.2.3 Detalhes da implementação

De um modo geral, todos os testes experimentais foram desenvolvidos em Python 3.7 no ambiente de desenvolvimento integrado Anaconda Spyder. As diferentes redes artificiais foram

implementadas através da Keras API, mais precisamente *Keras Applications* que disponibiliza diferentes modelos de redes juntamente com pesos pré-treinados. Devido ao grande volume de imagens necessárias, o treino só foi possível por meio de um computador com maior capacidade computacional, em que os programas foram lançados via GPU (*Graphics Processing Units*).

Em relação ao pré-processamento, mais precisamente ao processo de detecção e alinhamento das faces, este foi disponibilizado pela equipa VIS Team, no qual foi implementado o algoritmo MTCNN da autoria de Iván de Paz Centeno [20]. No que diz respeito ao manuseamento e transformação das imagens estas foram da responsabilidade da biblioteca *scikit-image*.

Por fim, todas as operações básicas relacionadas com *arrays* tornaram-se possíveis com os pacotes *Numpy* e a biblioteca *panda*, sendo todos os gráficos gerados através da biblioteca *Matplotlib*.

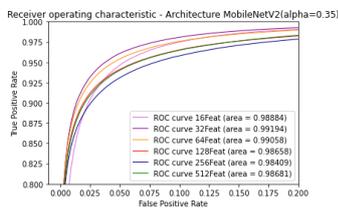
## 6.3 Descrição Experimental

### 6.3.1 Resultados e sua análise

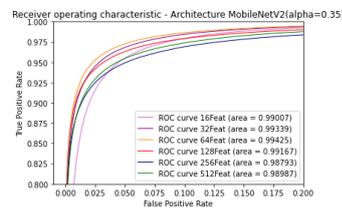
#### 6.3.1.1 Trade-off entre largura da arquitetura e diferentes números de características

##### Largura da rede vs diferentes números de características:

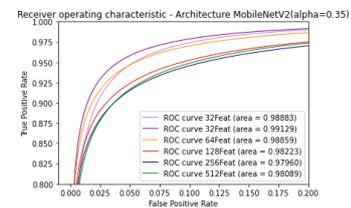
Os resultados serão apresentados sob a forma de gráficos, mais precisamente as curvas ROC (descrita na secção 6.2.2.1) obtidas através de diferentes combinações e segundo os diferentes protocolos. Primeiramente, cada gráfico demonstrado conterá as diferentes curvas ROC de cada número de features testado (16, 32, 64, 128, 256, 516) segundo um determinado alpha (0.35, 1.0, 1.4). Posteriormente os resultados serão apresentados de forma inversa, em que as curvas ROC resultantes correspondem aos diferentes números de alpha (0.35, 1.0, 1.4) segundo um determinado número de *features* (16, 32, 64, 128, 256, 516). Isto permite fazer observações, e retirar as respectivas conclusões de forma mais fundamentada e completa.



(a) Curvas ROC referentes às combinações resultantes entre arquitetura alpha igual a 0.35 e respectivos números de *features* testados, referentes ao protocolo 1.

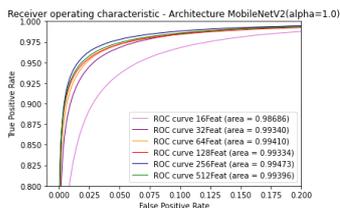


(b) Curvas ROC referentes às combinações resultantes entre arquitetura alpha = 0.35 e respectivos números de *features* testados, referentes ao protocolo 2.

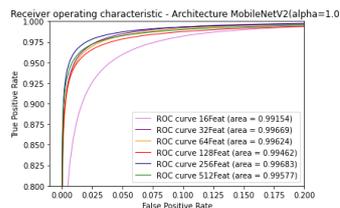


(c) Curvas ROC referentes às combinações resultantes entre arquitetura alpha igual a 0.35 e respectivos números de *features* testados, referentes ao protocolo 3.

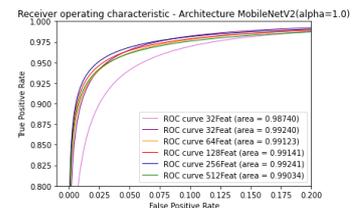
Figura 6.5: Curvas ROC referentes a alpha igual a 0.35.



(a) Curvas ROC referentes às combinações resultantes entre arquitetura alpha igual a 0.35 e respectivos números de *features* testados, referentes ao protocolo 3.

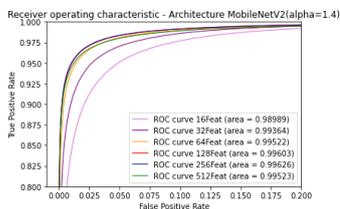


(b) Curvas ROC referentes às combinações resultantes entre arquitetura alpha igual a 1.0 e respectivos números de *features* testados, referentes ao protocolo 2.

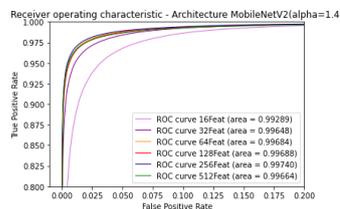


(c) Curvas ROC referentes às combinações resultantes entre arquitetura alpha igual a 1.0 e respectivos números de *features* testados, referentes ao protocolo 3.

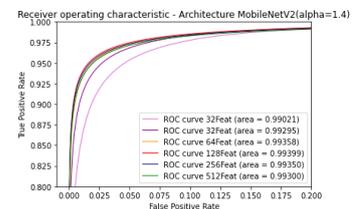
Figura 6.6: Curvas ROC referentes a alpha igual a 1.0.



(a) Curvas ROC referentes às combinações resultantes entre arquitetura alpha igual a 1.4 e respectivos números de *features* testados, referentes ao protocolo 1.



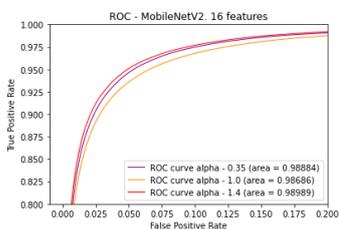
(b) Curvas ROC referentes às combinações resultantes entre arquitetura alpha igual a 1.4 e respectivos números de *features* testados, referentes ao protocolo 2.



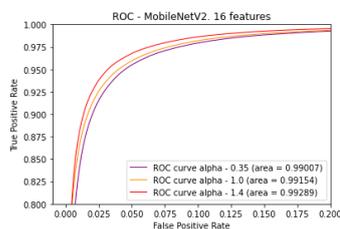
(c) Curvas ROC referentes às combinações resultantes entre arquitetura alpha igual a 1.4 e respectivos números de *features* testados, referentes ao protocolo 3.

Figura 6.7: Curvas ROC referentes a alpha igual a 1.4.

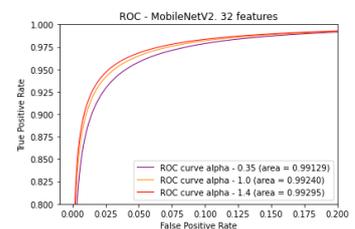
**Número de *features* vs diferentes larguras de rede:**



(a) Curvas ROC referentes às combinações resultantes o número de *features* igual a 16 e respectivas larguras entre de rede, referentes ao protocolo 1.



(b) Curvas ROC referentes às combinações resultantes o número de *features* igual a 16 e respectivas larguras entre de rede, referentes ao protocolo 2.



(c) Curvas ROC referentes às combinações resultantes o número de *features* igual a 16 e respectivas larguras entre de rede, referentes ao protocolo 3.

Figura 6.8: Curvas ROC referentes ao número de *features* igual a 16.

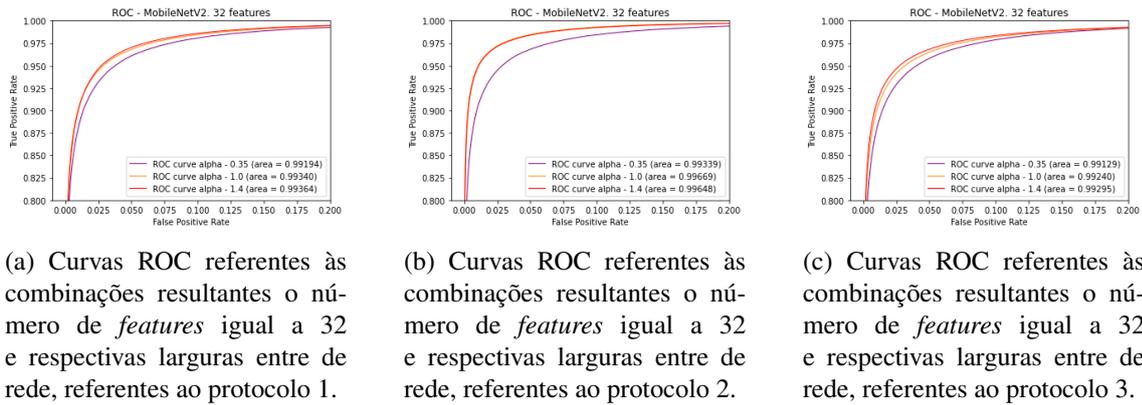


Figura 6.9: Curvas ROC referentes ao número de *features* igual a 32.

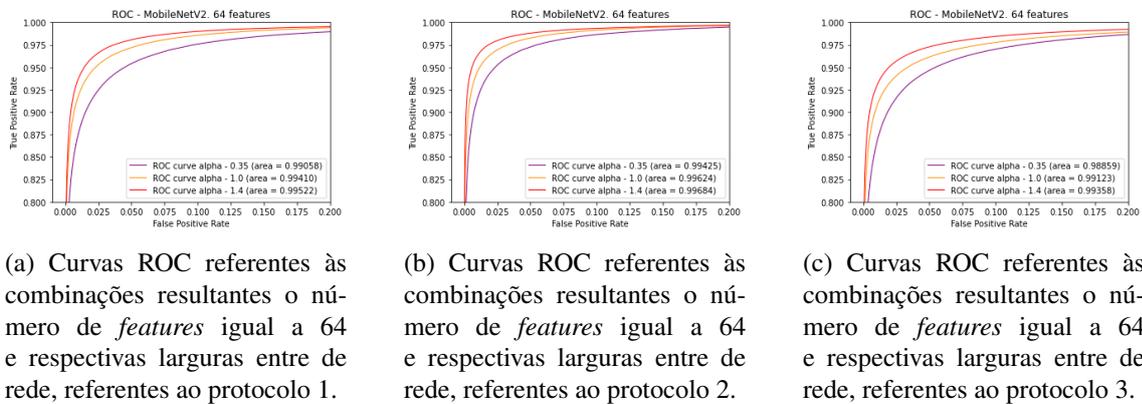


Figura 6.10: Curvas ROC referentes ao número de *features* igual a 64.

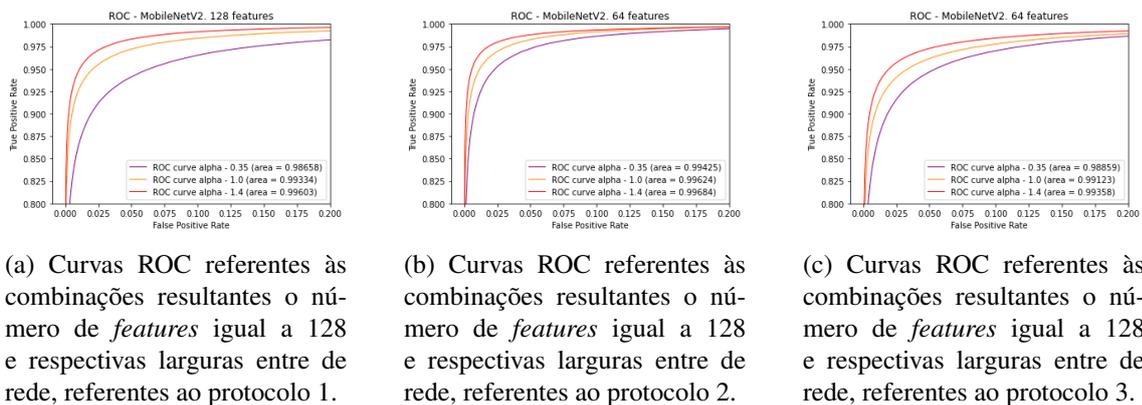


Figura 6.11: Curvas ROC referentes ao número de *features* igual a 128.

- É possível observar que é atingido o limite do poder discriminatório no que diz respeito ao valor de  $\alpha$ . Isto quer dizer que mesmo que se aumente este valor, e consequentemente se amplifique a largura da arquitectura, os resultados continuarão semelhantes aos obtidos para a gama de *features* testada. Isto é justificado pelas curvas ROC correspondentes às maiores arquitecturas, em que  $\alpha = 1.0$  e  $\alpha = 1.4$ , onde é possível observar que os resultados

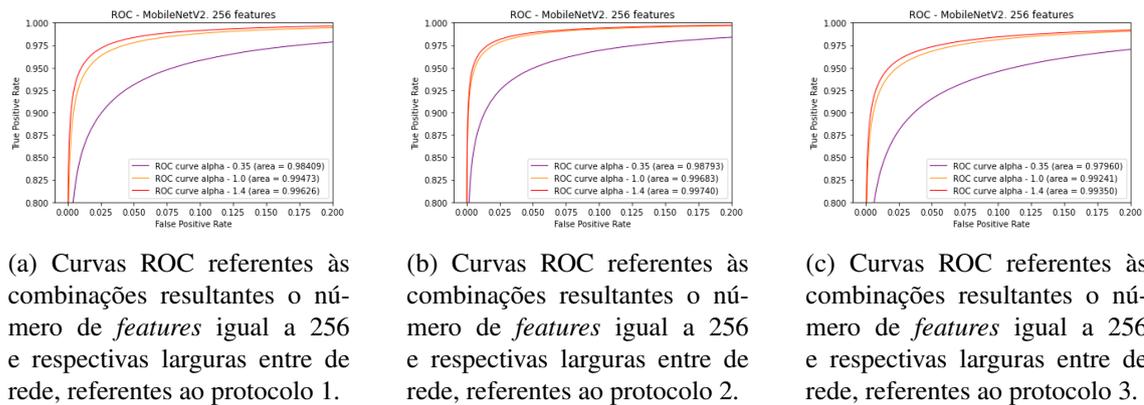


Figura 6.12: Curvas ROC referentes ao número de *features* igual a 256.

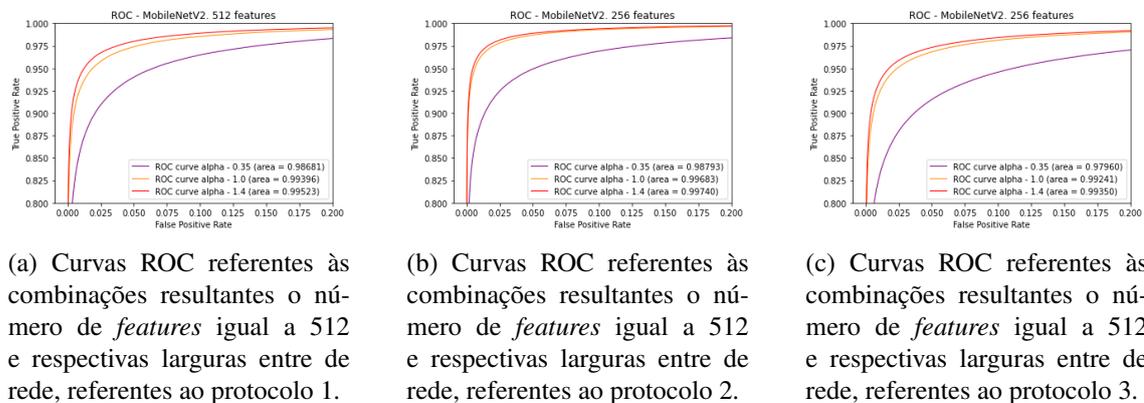


Figura 6.13: Curvas ROC referentes ao número de *features* igual a 512.

obtidos tornaram-se muito próximos uns dos outros.

- Através da análise destes dados é também possível verificar que foi atingido outro limite, desta vez relacionado com o número de *features*, atendendo a que a partir de determinado valor, nomeadamente de 128 a 512, as curvas ROC obtidas pelas diferentes arquiteturas tornam-se, tal como anteriormente, muito próximas umas das outras.
- Tendo em conta a capacidade computacional requerida e o objectivo do sistema, verifica-se que segundo os resultados obtidos a melhor forma de optimização da rede passa por utilizar *alpha* igual a 0.35 com número de *features* definido pelo valor 32. Apesar desta combinação obter valores mais baixos comparativamente a redes maiores e com números de *features* superiores, as curvas ROC obtidas pela rede mencionada continuam a ser bastante satisfatórias com a vantagem de se tratar de uma rede consideravelmente mais pequena. Deste modo, é possível afirmar que foi encontrada a melhor relação de compromisso entre estes dois parâmetros, que permitem alcançar uma boa performance sem comprometer os recursos computacionais dos equipamentos alvo.

### 6.3.1.2 Utilização de diferentes tipos de função de perda

O protocolo a utilizar no processo de teste está directamente dependente dos recursos computacionais disponíveis. Caso se esteja perante um cenário em que a capacidade computacional é limitada, é recomendado o primeiro protocolo tendo em conta que é aquele que apresenta os resultados mais próximo dos obtidos pelo protocolo número 3, que é claramente o mais completo.

Tendo em conta que no contexto desta dissertação não temos essas limitações e o objectivo é tirar conclusões baseadas no máximo de informação possível, os próximos testes serão feitos segundo o protocolo número 3.

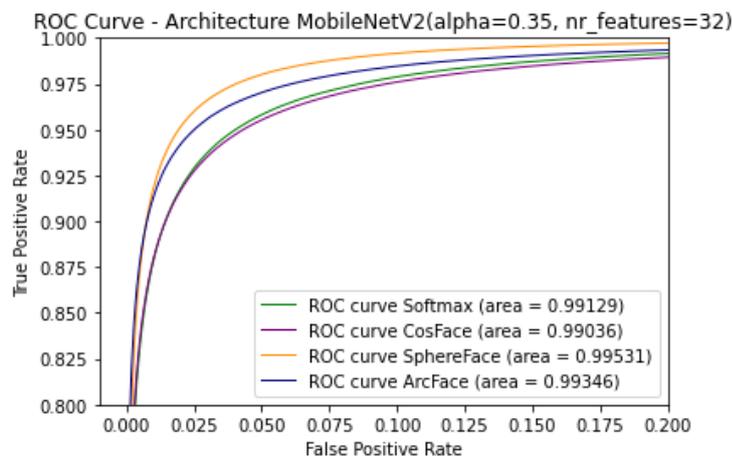


Figura 6.14: Curvas ROC resultantes das aplicações de diferentes funções de perda.

- Tendo em consideração que anteriormente foram definidos os valores dos parâmetros  $\alpha$  e número de  $features$  que melhor satisfazem as necessidades do sistema de reconhecimento analisado, levantou-se a questão de que outras formas contribuiriam para uma melhor optimização da rede. Assim sendo, surgiu a hipótese de testar diferentes funções de perda, com o intuito de melhorar a performance e consequentemente alcançar uma optimização da rede sob estudo superior. A partir dos resultados obtidos, em que foram aplicadas três novas funções de perda (CosFace, Spherefaced e Arcface) é desde já observável, que a função de perda Softmax, tal como previsto, não é a melhor escolha, tendo apenas alcançado um desempenho (ligeiramente) superior do que a rede em que foi aplicada a função de perda CosFace baseada em margem de cosseno aditiva.
- É importante mencionar que estes resultados já se encontram delimitados pela escolha dos parâmetros anteriores, mas atendendo às configurações utilizadas, a rede que obteve a melhor performance, comprovada pela curva ROC obtida, foi a função de perda SphereFace, que é baseada em margem angular multiplicativa. Esta apenas mostrou (até certo ponto) um pequeno atraso em relação à função de perda Arcface no que diz respeito à taxa de falsos positivos.

- Posto isto, podemos afirmar que a melhor forma de optimização da rede, e tendo por base a motivação por detrás da escolha da arquitectura e respectivos parâmetros, é obtida através da utilização da função perda SphereFace combinada com 32 *features* e  $\alpha = 0.35$ .

## 6.4 Comparação entre o método clássico de *machine learning* e o método de *pure deep learning*

Ao longo desta dissertação foram estudados dois métodos distintos dedicados ao reconhecimento facial, um mais clássico baseado em características *handcrafted* e outro mais moderno, com recurso a redes neuronais artificiais em que o processo de extracção das características não depende de nenhuma configuração manual, ficando exclusivamente ao critério da rede.

Atendendo aos resultados obtidos e respectivas análises, descritas na secção 5.2.5 e 6.3.1, é evidente que o segundo método destaca-se com grande vantagem no que diz respeito à performance obtida, o que era expectável tendo em conta os actuais documentos que constituem a literatura nesta área, perdendo apenas para o primeiro método no que concerne aos custos computacionais requeridos, em que neste tipo de abordagem continuam a ser muito superiores ao primeiro. Deste modo, e com o intuito de tentar equilibrar da melhor forma possível a comparação entre estes dois métodos, foi feita uma tentativa de optimização de uma rede convolucional, em que os resultados foram analisados na secção anterior, justificando as escolhas da arquitectura e respectivos parâmetros na aplicabilidade em dispositivos de menor capacidade computacional.

Apesar desta comparação continuar a ser de certo modo injusta, um último teste foi efectuado com o intuito de obter uma analogia mais directa entre os desempenhos dos diferentes métodos. Este consistiu em testar a rede neuronal artificial resultante da optimização, descrita na secção anterior (secção 6.3.1.2), com as bases de dados aplicadas no primeiro método (utilizadas em conjunto, onde foram 410 imagens utilizadas ao todo), sendo avaliada segundo o cálculo (tal como no primeiro método) da exactidão obtida, segundo o primeiro protocolo (descrito na secção 6.1.6). Tal como anteriormente foi calculada a curva ROC resultante do processo de teste que permitiu encontrar o valor de *threshold* óptimo para classificação binária permitindo assim calcular correctamente a métrica pretendida. Assim sendo, os resultados obtidos foram os seguintes:

Exactidão
0.999094

Tabela 6.4: Resultados dos testes da rede optimizada utilizando os datasets seleccionados para o primeiro método 5.2.1

- Como podemos observar pelos resultados obtidos, o método da segunda parte da dissertação não deixa margens para dúvidas quando à sua superioridade em relação ao primeiro método baseado em técnicas clássicas de *machine learning*. A rede convolucional optimizada conseguiu obter uma exactidão de quase 100% quando aplicada aos mesmos datasets, enquanto

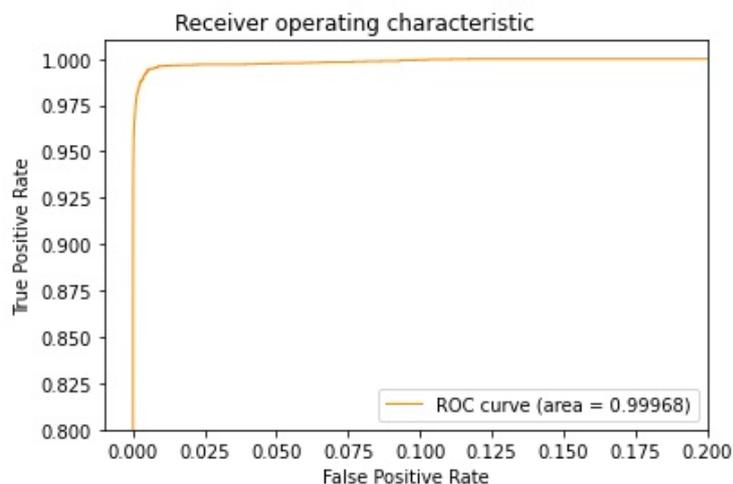


Figura 6.15: Curva ROC referente à rede otimizada ( $\alpha = 0.35$ ,  $features = 32$ , loss function = SphereFace), segundo o protocolo 1.

que o primeiro método apenas conseguiu uma exactidão máxima de 0.929 obtida com o descritor de pontos baseado na contagem do número de pontos por linha e por coluna.

- Este resultado veio confirmar o que já se vem a verificar em inúmeros documentos da actual literatura que, afirmam que as CNN's realmente são a escolha mais confiável para a tarefa de reconhecimento de objectos.

## Capítulo 7

# Conclusões e Futuro Trabalho

Desde o surgimento de sistemas baseados em características biométricas, a biometria da face ganhou particular relevo por ser um método prático e não intrusivo. Hoje em dia a procura de técnicas mais eficazes têm vindo a aumentar devido à sua aplicabilidade em sistemas que vêm a ganhar destaque associados a segurança e identificação. Apesar disto, alguns dos problemas que prejudicavam esta área no início dos seus passos, continuam a não ter soluções definitivas. Problemas relacionados com a pose, iluminação, oclusão parcial da face, até problemas mais actuais relacionados com fraudes e adulteração de imagens, fazem com que esta área continue em constante evolução.

Tendo isto em conta, esta dissertação teve como objectivo o estudo de dois métodos distintos em contexto de reconhecimento facial, um método clássico e outro mais inovador relacionado com *pure deep learning*, em que ambos foram submetidos a tentativas de aprimoramento e optimização de processos, sendo este o foco do todo o trabalho experimental desenvolvido ao longo desta dissertação.

### 7.1 Considerações finais

Assim como já foi mencionado, o trabalho experimental desta dissertação pode ser representado pela jornada entre o estudo de dois métodos com origens distintas, aplicados no contexto de reconhecimento facial.

A primeira abordagem, que surgiu devido à obtenção de bons resultados de um sistema de autenticação de objectos preciosos, baseado em descritores de pontos aplicados a dispersões aleatórias, que originou uma tentativa de aplicação na biometria da face com o objectivo de assingelar um ou mais descritores, em contraste com as mais recente abordagens de *deep learning* em que o descritor é optimizado pela rede. Tal como a maioria dos classificadores tradicionais de padrões, estes trabalham sob a ideia de calcular as relações geométricas entre características globais ou locais, tais como distâncias e ângulos calculados entre os vectores de características seleccionadas, seguido pela classificação com a intenção de determinar as características que melhor representam as classes dos padrões sob análise. Apesar de já ser conhecido à priori que não seria fácil atingir o objectivo, por se tratar de dispersões que não seguem uma dispersão nem uniforme, nem gaussiana, surgiu a tese de que o grau de aleatoriedade poderia afectar a performance dos descritores.

Assim sendo, foi implementado um sistema de *scrambling* com o objectivo de aumentar a aleatoriedade das dispersões estruturadas. Apesar dos esforços, o desempenho do sistema em geral continuou abaixo do esperado, justificado pelo facto de não ter sido possível alcançar o aumento de distâncias inter-classes e redução das intra-classes, atendendo que se trata de um método de classificação de multi-classes.

Posto isto, o seguimento desta dissertação seguiu o curso mais natural, levando em consideração as actuais tendências que envolvem técnicas de *deep learning*, onde foi definido o plano de estudo de uma rede convolucional, aplicada também em contexto de reconhecimento facial, tendo como motivação a aplicabilidade em dispositivos de menor capacidade computacional. Assim sendo, depois de seleccionada uma rede que cumprira os requisitos, a rede MobileNetV2, foi analisada a relação de compromisso entre a largura da rede e o número de *features*, ficando definido que a opção que melhor contribuiu para a optimização da rede no âmbito estudado foi  $\alpha=0.35$  e número de *features* = 32. Afim de optimizar mais os resultados, foram analisadas mais três funções de perda, ficando delineado que a melhor opção está associada à utilização da função de perda *SphereFace*. Com isto foi possível fazer uma comparação directa entre os resultados obtidos entre o primeiro método e a rede optimizada. Ficou claro que apesar de continuar a necessitar de mais recursos computacionais, a CNN ficou em vantagem, mostrando assim, que de facto as abordagens de *pure deep learning* representam uma escolha mais segura para a tarefa de reconhecimento facial.

## 7.2 Trabalho futuro

Atendendo que o trabalho desenvolvido nesta dissertação se dividiu em duas partes, o trabalho futuro também se decompõem por cada um dos métodos. No que diz respeito à primeira parte, a atenção focar-se-á em utilizar bases de dados maiores. Devido às limitações geradas pela natureza de métodos baseados em relações geométricas, apenas foram estudadas bases de dados com faces frontais, o que na altura se apresentou como um obstáculo, tendo em conta a escassez deste tipo de bases de dados na literatura. Também é preciso reforçar a ideia de que o objectivo final para este método não foi alcançado na totalidade, isto é, como ficou comprovado pelos resultados obtidos a adaptação do método de autenticação desenvolvido no projecto Uniquemark [18], não apresentou os resultados desejados, sendo por isso inevitável que sejam testadas abordagens mais inovadoras, que poderão passar pela testagem de diferentes classificadores, como também, pela utilização de outros descritores baseados em diferentes características *handcrafted*.

Relativamente à segunda parte, o processo de optimização de um rede, é um processo longo e demorado, pois não existem parâmetros conhecidos como ideais. A escolha dos mesmo é feita muitas vezes através de abordagens ligadas a tentativa e erro, estando intrinsecamente interligadas com o objecto de estudo. Dito isto, para além dos parâmetros elegidos ao longo do trabalho experimental, existem outros que devem ser analisadas tais como: número de *epochs*, *batch size*, optimizadores, *learning rate*, *weight decay* e *momentum*, de modo a obter a melhor optimização da rede.

# Referências

- [1] M. Abdulameer, S. Abdullah, e Z. Othman. Face recognition technique based on active appearance model. *International Review on Computers and Software*, 8:2733–2739, 2013.
- [2] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nacchaat AbdElatif Mohamed, e Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.
- [3] Hadid A Pietikäinen M Ahonen T. Face recognition with local binary patterns. Em *Computer Vision, ECCV 2004 Proceedings, Lecture Notes in Computer Science*, volume 3021, páginas 469–481. Springer, 2004.
- [4] R. Arppe e T. J. Sørensen. Physical unclonable functions generated through chemical methods for anti-counterfeiting. *Nature Reviews Chemistry*, 1:31, 0031, 2017. <https://doi.org/10.1038/s41570-017-0031>.
- [5] Herbert Bay, Tinne Tuytelaars, e Luc Van Gool. Surf: Speeded up robust features. Em Aleš Leonardis, Horst Bischof, e Axel Pinz, organizadores, *Computer Vision – ECCV 2006*, páginas 404–417, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [6] P. J. Besl e N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [7] W. W Bledsoe. Man-Machine Facial Recognition: Report on a Large-Scale Experiment, Technical Report PRI 22, Panoramic Research. Inc., Palo Alto, 1966.
- [8] W. W Bledsoe. Semiautomatic Facial Recognition, Technical Report SRI Project 6693. Stanford Research Institute, Menlo Park, California, 1968.
- [9] D. Brzakovic e N. Vujovic. Authentication of random patterns by finding a match in an image database. *Image and Vision Computing*, 14(7):485–499, 1996. <http://www.sciencedirect.com/science/article/pii/S026288569501067X>.
- [10] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, e Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age, 2018.
- [11] H. Chen e C. Su. An enhanced hybrid mobilenet. Em *2018 9th International Conference on Awareness Science and Technology (iCAST)*, páginas 308–312, 2018.
- [12] Dmitry Chetverikov, Dmitry Stepanov, e Pavel Krsek. Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(3):299 – 309, 2005. <http://www.sciencedirect.com/science/article/pii/S0262885604001179>.

- [13] François Chollet et al. Keras. <https://keras.io>, 2015.
- [14] P. G. Comba. Exponentiation cryptosystems on the ibm pc. *IBM Systems Journal*, 29(4):526–538, 1990.
- [15] Corinna Cortes e Vladimir Vapnik. Support-vector networks. página 273–297, 1995. <https://doi.org/10.1007/BF00994018>.
- [16] Valter Costa. *Vision Methods to Find Uniqueness Within a Class of Objects*. Tese de doutoramento, Faculdade de Engenharia da Universidade do Porto, 2019.
- [17] Ian Craw. 2d face sets. [http://pics.stir.ac.uk/2D\\_face\\_sets.htm](http://pics.stir.ac.uk/2D_face_sets.htm).
- [18] N. Gonçalves N. L. Cruz. Uniquemark - a method to create and authenticate a unique mark in precious metal artefacts. Em *Jewellery Materials Congress*, London, England, 2019.
- [19] Jesse Davis e Mark Goadrich. The relationship between precision-recall and roc curves. Em *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, página 233–240, New York, NY, USA, 2006. Association for Computing Machinery. <https://doi.org/10.1145/1143844.1143874>.
- [20] Iván de Paz Centeno. Mtcnn face detection implementation for tensorflow, as a pip package. <https://github.com/ipazc/mtcnn>.
- [21] Jiankang Deng, Jia Guo, e Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *CoRR*, abs/1801.07698, 2018. <http://arxiv.org/abs/1801.07698>.
- [22] O. Deniz, G. Bueno, J. Salido, e F. De la Torre. Face recognition using Histograms of Oriented Gradients. *Pattern Recognition Letters*, 32:1598–1603, 09 2011.
- [23] R. O. Duda e P. E. Hart. *Transactions on Systems, Man, and Cybernetics*, 1973.
- [24] Jerome Fan, Suneel Upadhye, e Andrew Worster. Understanding receiver operating characteristic (roc) curves. *Canadian Journal of Emergency Medicine*, 8(1):19–20, 2006.
- [25] Xiaoqin Feng, Rong Xie, Junyang Sheng, e Shuo Zhang. Population statistics algorithm based on MobileNet. *Journal of Physics: Conference Series*, 1237:022045, jun 2019. <https://doi.org/10.1088/1742-6596/1237/2/022045>.
- [26] M. Fishier e R. Boles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. 24(6):381–395, 1981.
- [27] Yansong Gao, Said F. Al-Sarawi, e Derek Abbott. Physical unclonable functions. *Nature Electronics*, 3(2):81–91, February 2020. <https://doi.org/10.1038/s41928-020-0372-5>.
- [28] Raoof Gholami e Nikoo Fakhari. *Chapter 27 - Support Vector Machine: Principles, Parameters, and Applications*. Handbook of Neural Computation. Academic Press, 2017.
- [29] A. J. Goldstein, L. D. Harmon, e A. B. Lesk. Identification of human faces. *Proceedings of the IEEE*, 59(5):748–760, 1971.
- [30] Ian Goodfellow, Yoshua Bengio, e Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.

- [31] Jianping Gou, Hongxing Ma, Weihua Ou, Shaoning Zeng, Yunbo Rao, e Hebiao Yang. A generalized mean distance-based k-nearest neighbor classifier. *Expert Systems with Applications*, 115:356 – 372, 2019. <http://www.sciencedirect.com/science/article/pii/S0957417418305293>.
- [32] Robert Gove e Jorge Faytong. Chapter 4 - machine learning and event-based software testing: Classifiers for identifying infeasible gui event sequences. volume 86 de *Advances in Computers*, páginas 109 – 135. Elsevier, 2012. <http://www.sciencedirect.com/science/article/pii/B9780123965356000041>.
- [33] Daniel Graupe. *Principles of artificial neural networks*, volume 3. World Scientific, 1 edição, 1997.
- [34] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2 edição.
- [35] K. He, X. Zhang, S. Ren, e J. Sun. Deep residual learning for image recognition. Em *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 770–778, Las Vegas, NV, USA, June 2016. IEEE.
- [36] A. Howard, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, e Q. Le. Searching for MobileNetV3. Em *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, páginas 1314–1324, 2019.
- [37] Andrew Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, e Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv*, 04 2017.
- [38] Guosheng Hu, Yongxin Yang, Dong Yi, Josef Kittler, William Christmas, Stan Z. Li, e Timothy Hospedales. When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition. Em *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.
- [39] G. Huang, Z. Liu, L. Van Der Maaten, e K. Q. Weinberger. Densely Connected Convolutional Networks. Em *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 2261–2269, 2017.
- [40] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, e Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [41] R. Ishiyama, Y. Kudo, e T. Takahashi. midot: micro identifier dot on things -a tiny, efficient alternative to barcodes, tags, or marking for industrial parts traceability. Em *Proceeding of the 17th IEEE International Conference on Industrial Technology (ICIT'16)*, páginas 781–786, Taipei, Taiwan, 2016. IEEE.
- [42] A. K. Jain, A. Ross, e S. Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, 2004.
- [43] Anil K. Jain.
- [44] A. Juhong e C. Pintavirooj. Face recognition based on facial landmark detection. Em *2017 10th Biomedical Engineering International Conference (BMEiCON)*, páginas 1–4, 2017.

- [45] Vahid Kazemi e Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
- [46] J. M. Keller, M. R. Gray, e J. A. Givens. A fuzzy K-nearest neighbor algorithm. *Transactions on Systems, Man, and Cybernetics*, SMC-15:580–585, 1985.
- [47] S. Khan, H. Rahmani, S. A. A. Shah, M. Bennamoun, G. Medioni, e S. Dickinson. 2018.
- [48] Salman Khan, Hossein Rahmani, Syed Afaq Ali Shah, e Mohammed Bennamoun. *A Guide to Convolutional Neural Networks for Computer Vision*. Número 1 em Synthesis Lectures on Computer Vision. Morgan Claypool Publishers, 2018.
- [49] Y. Kudo, H. Zwaan, T. Takahashi, R. Ishiyama, e P. Jonker. Tip-on-a-chip: Automatic dotting with glitter ink pen for individual identification of tiny parts. Em *ACM Int. Conf. on Multimedia Systems (MMSys'18)*, páginas 502–505, 2008.
- [50] Steve Lawrence, C Giles, Ah Tsoi, e Andrew Back. Face recognition: A convolutional neural network approach. *Neural Networks, IEEE Transactions on*, 8:98 – 113, 02 1997.
- [51] Y. Lecun, L. Bottou, Y. Bengio, e P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [52] Yunfeng Li, Zongying Ou, e Guoqiang Wang. Face recognition using gabor features and support vector machines. Em Lipo Wang, Ke Chen, e Yew Soon Ong, organizadores, *Advances in Natural Computation*, páginas 119–122, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [53] Chun-Fu Lin e Sheng-De Wang. *IEEE Transactions on Neural Networks*, 13(2):464–471, 2002.
- [54] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, e Le Song. Sphreface: Deep hypersphere embedding for face recognition. *CoRR*, abs/1704.08063, 2017. <http://arxiv.org/abs/1704.08063>.
- [55] D. G. Lowe. Object recognition from local scale-invariant features. Em *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, páginas 1150–1157 vol.2, 1999.
- [56] I. Masi, Y. Wu, T. Hassner, e P. Natarajan. Deep face recognition: A survey. Em *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, páginas 471–478, 2018.
- [57] Kishan Mehrotra, Chilukuri K. Mohan, e Sanjay Ranka. *Elements of Artificial Neural Networks*. Mit Press, 1997.
- [58] Prakash Nadkarni. *Chapter 10 - Core Technologies: Data Mining and “Big Data”*. Clinical Research Computing. Academic Press, 2016.
- [59] Loris Nanni, Stefano Ghidoni, e Sheryl Brahmam. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71:158 – 172, 2017. <http://www.sciencedirect.com/science/article/pii/S0031320317302224>.

- [60] W. Ouarda, H. Trichili, A. M. Alimi, e B. Solaiman. Face recognition based on geometric features using support vector machines. Em *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, páginas 89–95, 2014.
- [61] Wael Ouarda, H. Trichili, A. Alimi, e B. Solaiman. Face recognition based on geometric features using support vector machines. *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, páginas 89–95, 2014.
- [62] Ravikanth Pappu, Ben Recht, Jason Taylor, e Neil Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002. <https://science.sciencemag.org/content/297/5589/2026>.
- [63] João Pereira. Fingerprint anti spoofing – domain adaptation and adversarial learning. Tese de mestrado, Faculdade de Engenharia da Universidade do Porto, 2020.
- [64] D. M. W. Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness correlation. 2(1):37–63, 2011.
- [65] R. L. Rivest, A. Shamir, e L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978. <https://doi.org/10.1145/359340.359342>.
- [66] E. Rublee, V. Rabaud, K. Konolige, e G. Bradski. Orb: An efficient alternative to sift or surf. Em *Int. Conf. on Computer Vision (ICCV)*, páginas 2564–2571, Barcelona, Espanha, 2011.
- [67] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, e Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [68] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, e L. Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Em *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 4510–4520, 2018.
- [69] Chang Shu, Xiaoqing Ding, e Chi Fang. Histogram of the Oriented Gradient for Face Recognition. *Tsinghua Science & Technology*, 16:216–224, 04 2011.
- [70] Karen Simonyan e Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. Em *International Conference on Learning Representations (ICLR)*, San Diego, California, USA, May 2014. ICLR.
- [71] Lawrence Sirovich e M Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America. A, Optics and image science*, 4:519–24, 04 1987.
- [72] Shan Suthaharan. *Support Vector Machine*, páginas 207–235. Springer US, Boston, MA, 2016. [https://doi.org/10.1007/978-1-4899-7641-3\\_9](https://doi.org/10.1007/978-1-4899-7641-3_9).
- [73] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, e A. Rabinovich. Going deeper with convolutions. Em *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 1–9, Boston, MA, USA, June 2015. IEEE.

- [74] Igor V. Tetko, David J. Livingstone, e Alexander I. Luik. Neural network studies. 1. comparison of overfitting and overtraining. *Journal of Chemical Information and Computer Sciences*, 35(5):826–833, 1995. <https://pubs.acs.org/doi/abs/10.1021/ci00027a006>.
- [75] M. Turk e A. Pentland. Face recognition using eigenfaces. *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, páginas 586–591, 1991.
- [76] N. Vujovic e D. Brazakovic. Evaluation of an algorithm for finding a match of a distorted texture pattern in a large image database. *ACM Transactions on Information Systems*, 14(7):485–499, 1996.
- [77] N. Vujovic e D. Brzakovic. Evaluation of an algorithm for finding a match of a distorted texture pattern in a large image database. *ACM Transactions on Information Systems*, 16(1), 01 1998.
- [78] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Zhifeng Li, Dihong Gong, Jingchao Zhou, e Wei Liu. Cosface: Large margin cosine loss for deep face recognition. *CoRR*, abs/1801.09414, 2018. <http://arxiv.org/abs/1801.09414>.
- [79] Markus Weber. Face database collection of markus weber facedatabase collection of markus weber. <http://www.vision.caltech.edu/html-files/archive.html>.
- [80] Gideon Yuval. How to swindle rabin. *Cryptologia*, 3(3):187–191, 1979. <https://doi.org/10.1080/0161-117991854025>.
- [81] Sergey Zagoruyko e Nikos Komodakis. Wide residual networks. Em Edwin R. Hancock Richard C. Wilson e William A. P. Smith, organizadores, *Proceedings of the British Machine Vision Conference (BMVC)*, páginas 87.1–87.12. BMVA Press, September 2016. <https://dx.doi.org/10.5244/C.30.87>.
- [82] Matthew D. Zeiler e Rob Fergus. Visualizing and understanding convolutional networks. Em David Fleet, Tomas Pajdla, Bernt Schiele, e Tinne Tuytelaars, organizadores, *Computer Vision – ECCV 2014*, páginas 818–833, Cham, 2014. Springer International Publishing.
- [83] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, e Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016. <http://dx.doi.org/10.1109/LSP.2016.2603342>.
- [84] Q. Zhu, P. Zhang, Z. Wang, e X. Ye. A new loss function for cnn classifier based on predefined evenly-distributed class centroids. *IEEE Access*, 8:10888–10895, 2020.