



**Rúben Miguel
Paulo Menino**

**Reconhecimento Facial em Ambiente não
Cooperativo**

**Facial Recognition in a Non-Cooperative
Environment**



Rúben Miguel
Paulo Menino

Reconhecimento Facial em Ambiente não
Cooperativo

Facial Recognition in a Non-Cooperative
Environment

“The greatest challenge to any thinker is stating the problem in a way that will allow a solution”

— Bertrand Russell



Universidade de Aveiro
2022

**Rúben Miguel
Paulo Menino**

**Reconhecimento Facial em Ambiente não
Cooperativo**

**Facial Recognition in a Non-Cooperative
Environment**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Professor Doutor António José Ribeiro Neves, Professor Auxiliar da Universidade de Aveiro, e do Professor Doutor José Silvestre Serra da Silva, Professor associado com agregação da Academia Militar.

Dedico este trabalho aos meus familiares e amigos pelo incansável apoio.

o júri / the jury

presidente / president

Professor Doutor Joaquim João Estrela Ribeiro Silvestre Madeira
Professor auxiliar da Universidade de Aveiro

vogais / examiners committee

Professora Doutora Catarina Helena Branco Simões Silva
Professora Auxiliar do Departamento de Engenharia Informática da Fac. de Ciências e Tecnologia
da Universidade de Coimbra

Professor Doutor António José Ribeiro Neves
Professor auxiliar da Universidade de Aveiro

agradecimentos / acknowledgements

A concretização da dissertação de mestrado representa a etapa final de um importante ciclo na minha vida e não teria sido possível sem o apoio de algumas pessoas, e por este motivo gostaria de agradecer a todas elas. Gostaria de agradecer, assim, à minha família e amigos, pelo apoio, força e incentivo constante. Ao Professor Doutor António José Ribeiro Neves e ao Professor Doutor José Silvestre Serra da Silva, reconheço e agradeço toda a disponibilidade, orientação, acompanhamento e conhecimentos transmitidos durante este processo de aprendizagem. A todos os professores que me acompanharam e colegas de licenciatura e mestrado que de algum modo contribuíram para o sucesso e crescimento a nível pessoal e profissional. Agradecer também à Academia Militar de Lisboa pela cumplicidade e cedência de equipamentos necessários para o melhoramento e realização desta dissertação. Por fim, agradeço à Universidade de Aveiro pela excelência com que me acolheu ao longo do meu período académico. A todos, um especial obrigado!

Palavras Chave

deteção facial, reconhecimento facial, processamento de imagens, aprendizagem profunda, algoritmos, aquisição de imagens.

Resumo

Nos dias de hoje, o reconhecimento facial tornou-se uma marco bastante importante na área da informática e tem vindo a receber bastante atenção ao longo dos anos. O reconhecimento facial pode ser utilizado em bastantes áreas, porém uma área que tem estado em bastante crescimento é a área da segurança. Temas como acessos a instalações militares, identificação de grupos terroristas e pessoas que forçam e abusam da lei são alguns dos temas mais abordados. Apesar de ser um tema bastante estudado existem ainda algumas limitações, principalmente quando a aquisição das imagem são adquiridas de pessoas em ambientes não-cooperativos. O objetivo desta dissertação de mestrado é a investigação de vários métodos de deteção e reconhecimento facial. Apresenta um estudo sobre os mais importantes algoritmos, e técnicas de pré-processamento como frontalização e alinhamento facial de modo a conseguir comparar os níveis de precisão de cada um dos algoritmos. De modo a obter resultados foi efetuado um dataset de imagens na Universidade de Aveiro de vários espetros de cores. Foi possível observar que algoritmos baseados em redes neurais convolucional profundas têm uma precisão mais elevada em relação a vários métodos tradicionais. Foi ainda dado um primeiro passo no sentido de desenvolver um modelo de deteção facial em imagens térmicas, onde existiu uma melhoria de cerca de 30% em relação ao modelo original.

Keywords

face detection, face recognition, image processing, deep learning, algorithms, image acquisition.

Abstract

Nowadays, facial recognition has become very important in the field of computing and has been receiving a lot of attention over the years. Facial recognition can be used in many areas, but one area that has been growing a lot is security. Topics like access to military installations, identification of terrorist groups and people who force and abuse the law are some of the most discussed topics. Despite being a widely studied topic, there are still some limitations, especially when image acquisition is acquired from people in non-cooperative environments. The objective of this master's thesis is the investigation of various methods of detection and facial recognition. It presents a study on the most important algorithms, and pre-processing techniques such as frontalization and facial alignment in order to be able to compare the accuracy levels of each of the algorithms. In order to obtain results, an image dataset was carried out at the University of Aveiro of various color spectrum. It was possible to observe that algorithms based on deep convolutional neural networks have a higher precision compared to several traditional methods. A first step was also taken towards developing a model of facial detection in thermal images, where there was an improvement of about 30% compared to the original model.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	v
Acrónimos	vii
1 Introdução	1
1.1 Contexto e motivação	1
1.2 Objetivos	3
1.3 Estrutura da dissertação	3
2 Revisão de Literatura e Conceitos	5
2.1 Algoritmos de deteção de faces	7
2.1.1 Haar Cascade	7
2.1.2 Histograma de Gradientes Orientados	10
2.1.3 Dlib CNN	11
2.2 Reconhecimento facial	12
2.3 Técnicas e algoritmos de reconhecimentos de faces	13
2.3.1 Análise de Componentes Principais (PCA)	13
2.3.2 Análise discriminante linear (LDA)	13
2.3.3 EigenFaces	14
2.3.4 FisherFaces	16
2.3.5 Framework Deepface - OpenFace	17
3 Metodologia	19
3.1 Aquisição de imagens e input	20
3.2 Processamento de imagem	21
3.2.1 Frontalização	22
3.3 Reconhecimento Facial	23

3.3.1	Deteção de imagens e respetivo treino	23
3.4	Deepface Framework - Openface	25
3.5	Treino de classificador para deteção facial de imagens térmicas	26
3.5.1	Aquisição de dataset	26
3.5.2	Criação de arquivo de vetores de imagens positivas	26
3.5.3	Treino do classificador	27
4	Resultados e discussão	29
4.1	Datasets	29
4.1.1	Dataset criado na Universidade de Aveiro (UA)	29
4.1.2	Dataset online	31
4.2	Análise métodos deteção facial	31
4.2.1	Ângulos de rotação	31
4.2.1.1	Deteção em imagens térmicas	34
4.2.2	Tempo de processamento	35
4.3	Análise de métodos de reconhecimento	36
4.3.1	Verdadeiros Positivos, confiança e tempo de processamento	36
4.3.2	Pré-processamento	40
4.3.2.1	Frontalização da face	40
4.3.3	Deepface framework - OpenFace	41
4.4	Treino de classificador para deteção facial de imagens térmicas	43
4.4.1	Comparação classificadores	44
5	Conclusão	47
	Referências	49
	Autorização para utilização de imagens	51
	Artigo científico Conference International Society of Military Sciences	53

Lista de Figuras

1.1	Exemplo de reconhecimento facial em uma imagem.	2
1.2	Principais passos para um sistema de reconhecimento facial.	2
2.1	Diferentes categorias de desafios para detecção facial[4].	6
2.2	Diferentes técnicas para detecção facial.	7
2.3	Características Haar mais significativas em uma face[8]	8
2.4	Cálculo do gráfico da imagem integral [9].	8
2.5	Exemplo de características relevantes[10].	9
2.6	Classificador em cascata[11].	10
2.7	Orientação do histograma e formato das células e histograma de gradientes orientados calculado [9].	11
2.8	Diferentes abordagens de reconhecimento facial[15].	12
2.9	Diferença entre PCA e LDA[18].	14
2.10	Concatenação de uma face em vetor único.	15
2.11	Estrutura do OpenFace [26].	18
3.1	Fluxograma do sistema.	19
3.2	Organização das imagens para os algoritmos de detecção e reconhecimento.	20
3.3	68 coordenadas (x,y) na face de imagem de teste [28].	22
3.4	Exemplo de transformações após a aplicação da frontalização numa face. (a) Imagem de teste (id = 4) com um ângulo de rotação de -20° (b) Imagem "a" frontalizada (c) Imagem de teste (id = 2) com um ângulo de rotação de $+30^\circ$ (d) Imagem "c" frontalizada	23
3.5	Terminal após execução do script python.	23
3.6	Redimensionamento das imagens.	24
3.7	Exemplo de alinhamento de face através de trigonometria [24].	25
4.1	Setup para aquisição das fotografias	29
4.2	Ângulo de rotação de 20° e 90° em relação à posição da câmara	30
4.3	Exemplo de imagens adquiridas no dataset adquirido. (a) Visível; (b) Térmicas; (c) Infravermelho de ondas curtas (SWIR); (d) Infravermelho próximo (NIR))	30

4.4	Taxa de sucesso em função do ângulo de rotação para imagens do dataset efetuado. . . .	31
4.5	Exemplo de falsos positivos testados	32
4.6	Taxa de sucesso em função do ângulo de rotação para imagens do dataset online.	33
4.7	Diferença da taxa de sucesso relativamente a ambos os datasets.	33
4.8	Taxa de sucesso em função do ângulo de rotação para imagens térmicas do dataset efetuado.	35
4.9	Exemplo de imagens térmicas	43
4.10	Linha superior - imagens com objeto a ser detetado(faces) e Linha inferior - imagens sem objeto a ser detetado (face) [34], [35].	44
4.11	Taxa de sucesso em função do ângulo de rotação.	45
4.12	Exemplo de falsos positivos testados	45

Lista de Tabelas

4.1	Especificações dos equipamentos de imagem pertencentes à Academia Militar	30
4.2	Resultados dos testes para os 3 métodos de detecção a partir do dataset efetuado.	32
4.3	Resultados dos testes para os 3 métodos de detecção a partir do dataset online.	34
4.4	Resultados dos testes para os 3 métodos de detecção a partir do dataset efetuado em imagens térmicas.	34
4.5	Tempo de processamento relativo ao algoritmos de detecção facial.	35
4.6	EigenFaces sem frontalização	37
4.7	FisherFaces sem frontalização	37
4.8	Resultados do reconhecimento utilizando o método EigenFaces	38
4.9	Resultados do reconhecimento utilizando o método FisherFaces	39
4.10	Eigenfaces com frontalização	40
4.11	Fisherfaces com frontalização	40
4.12	Resultados do reconhecimento utilizando o método EigenFaces com frontalização da face	41
4.13	Resultados do reconhecimento utilizando o método FisherFaces com frontalização da face	41
4.14	Resultados do reconhecimento geral utilizando Openface	42
4.15	Resultados do reconhecimento utilizando Openface	42
4.16	Resultados dos testes para o método treinado.	44

Acrónimos

3D	Tridimensionais
AM	Academia Militar
CNN	Rede neural convolucional
Dlib	Biblioteca digital
HOG	Histograma de gradientes orientados
LDA	Análise discriminante linear
ML	Machine Learning
MMOD	Deteção do objeto de margem máximo
NIR	Infravermelho próximo
OpenCV	Biblioteca de computação visual de código aberto
PCA	Análise de Componentes Principais
RGB	Vermelho, Verde e Azul
SIFT	Transformação de recursos invariantes de escala
SURF	Características Robustas e Rápidas
SVM	Máquinas de Vetores de Suporte
SWIR	Infravermelho de ondas curtas
UA	Universidade de Aveiro
XML	Extensible Markup Language

Introdução

Neste capítulo é apresentada uma introdução ao tema da detecção e reconhecimento facial, apresentando tópicos como o contexto geral, motivação, os objetivos e a estruturação da dissertação.

1.1 CONTEXTO E MOTIVAÇÃO

A partir do reconhecimento facial, é possível identificar qualquer pessoa, que tenha um conjunto de imagens, por exemplo numa base de dados. Para ser possível a face ser detetada e posteriormente reconhecida, é necessário algo que capture a face da pessoa em questão. Um dos dispositivos eletrónicos mais utilizado para estas ações, são câmaras, nomeadamente de vigilância ou somente de captura. Um exemplo de um reconhecimento facial é possível ser observado na Figura 1.1

Nos dias de hoje, existem bastantes métodos para reconhecimento. Os métodos biométricos são os mais reconhecidos e utilizados, como por exemplo, na leitura de impressão digital, reconhecimento de voz e geometria da mão.

Estes métodos, consistem em várias tecnologias onde permite a identificação de uma pessoa, não só a partir da face mas de todas as características físicas presentes num ser-humano. Cada pessoa possui as suas próprias características físicas, como impressão digital, formato da mão e face.

Cada um destes métodos, possivelmente contém as suas vantagens em relação aos outros, porém, relativamente ao reconhecimento facial, além de necessitar somente de uma câmara e um software adequado para o reconhecimento, não necessita da colaboração da pessoa nem do contacto com hardware necessário para o reconhecimento, como no caso da impressão digital, e pelo formato da mão, o que leva a uma melhor segurança e consecutivamente um melhor conforto para a pessoa em questão. Além de possuir vantagens existem também

Nome da pessoa: 4 - Ruben Menino

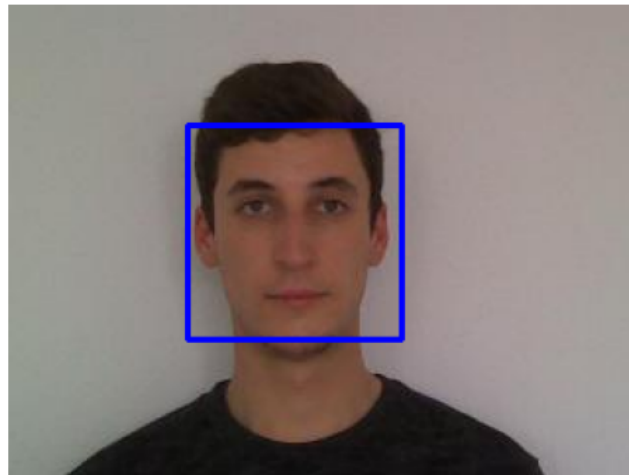


Figura 1.1: Exemplo de reconhecimento facial em uma imagem.

desvantagens. O preço de determinadas câmaras, pode variar consoante o que é necessário para poder realizar o reconhecimento. Fatores como, iluminação, distância de captura e preço, são fatores em ter em conta, numa escolha e utilização dessas câmaras.

O reconhecimento facial é uma área de investigação com um grande crescimento, devido a poder ser utilizado em bastantes áreas com vários propósitos.

A utilização do reconhecimento facial na área das forças de segurança, identificação de pessoas e também no controlo delas em determinados lugares como estádios e grandes eventos são alguns dos locais propícios à utilização destes métodos.

Podemos observar na Figura 1.2 os principais passos de um sistema onde seja possível efetuar o reconhecimento facial.

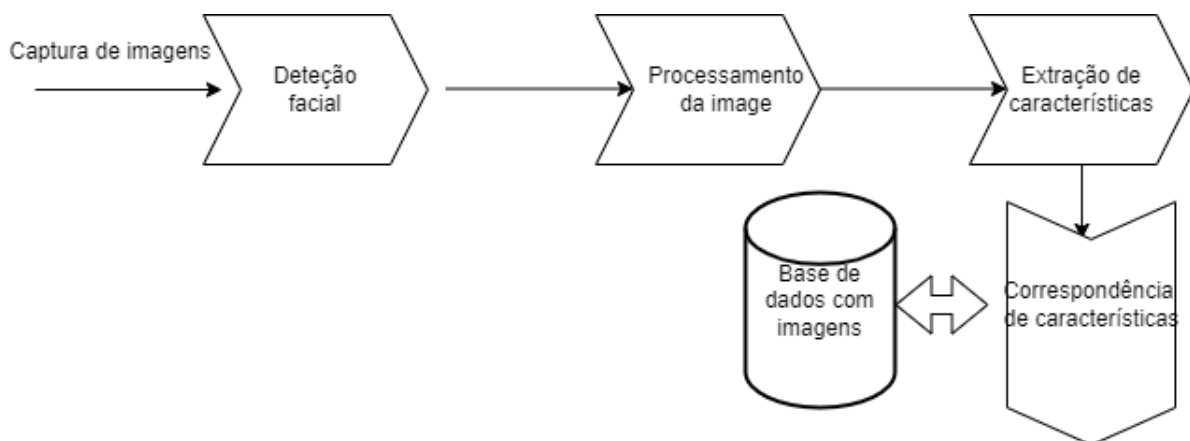


Figura 1.2: Principais passos para um sistema de reconhecimento facial.

Iniciando pela captura de imagens, é realizada posteriormente a deteção facial nessa própria imagem e o pré-processamento. Seguidamente é realizada a extração das características faciais mais importantes e uma comparação com as imagens treinadas da base de dados.

1.2 OBJETIVOS

Os principais objetivos desta dissertação são, a partir de ferramentas de computação visual, conseguir verificar o desempenho de vários algoritmos de detecção e de reconhecimento. Algoritmos de reconhecimento como EigenFaces, FisherFaces e OpenFace vão ser analisados e retiradas conclusões.

Os objetivos complementares nesta dissertação são:

- Aquisição e levantamento de um dataset facial com diferentes ângulos de rotação.
- Estudo do estado da arte acerca de detecção e reconhecimento facial
- Pré-processamento das imagens na tentativa de melhorar os algoritmos.
- Treino de classificadores para uma melhor detecção facial em imagens térmicas.

Foi ainda realizado um artigo científico com os estudos e resultados adquiridos, de modo a submeter à "Conference of the International Society of Military Sciences". É possível observar em Artigo científico Conference International Society of Military Sciences.

1.3 ESTRUTURA DA DISSERTAÇÃO

O seguinte documento está dividido consoante os seguintes capítulos:

- O Capítulo 2 é dedicado a explicar a detecção facial e o respetivo reconhecimento. São abordados vários algoritmos de detecção facial e reconhecimento e como é o seu funcionamento.
- O capítulo 3 apresenta a metodologia proposta nesta dissertação. Desde a aquisição do dataset de imagens, detecção de faces nessas imagens de entrada, processamento e posterior reconhecimento.
- O capítulo 4 apresenta vários resultados e discussão acerca desses. Como foi feita a aquisição de imagens, taxa de acerto em relação aos métodos de detecção e reconhecimento, pré-processamento das imagens, tempos de processamento e treino de classificador específico para imagens térmicas.
- O capítulo 5 apresenta várias conclusões gerais acerca do trabalho realizado e o que poderá ser possível realizar como trabalho futuro.

Revisão de Literatura e Conceitos

O principal objetivo da detecção facial é conseguir determinar a partir de uma imagem de entrada, a existência de faces ou não nessa mesma. Parece algo simples para uma análise humana, porém para computadores é uma área ainda em desenvolvimento e pesquisa, que se torna bastante mais complexo. É então uma técnica de computação visual e processamento de imagem em que o objetivo é gerar um output com a localização dessas mesmas faces. Além de ser já algo à partida complicado, existem ainda fatores que dificultam ainda mais esse processo. Fatores como a iluminação presente quando essa mesma imagem foi realizada, a oclusão e a pose, são a principal razão para a taxa e a precisão da detecção efetuada, ser inferior. Em ambientes estudados, com iluminação controlada iria se obter resultados mais favoráveis. Alguns dos exemplos podemos observar na figura 2.1.

1. **Iluminação** - Com iluminação quer se dizer variações de luz. A iluminação pode não ser uniforme na imagem. Alguma parte da imagem pode conter iluminação muito alta e outra pode conter iluminação muito baixa[1]. Por vezes, uma ligeira alteração na intensidade da iluminação pode fazer com que uma face não seja detetada, o que pode causar um impacto significativo nos resultados.
2. **Pose** - Quando existe uma variação do movimento da cabeça e o ângulo de visão da pessoa, sabemos que a face está numa determinada orientação, denominada pose.
3. **Oclusão** - As faces ocluídas são apenas parcialmente visíveis e as regiões ocluídas têm aparências arbitrárias que podem diferir do rosto normal[2]. Isto acontece quando existe algum objeto a tapar a face, como óculos, máscara ou cachecol.
4. **Expressão Facial** - Normalmente a detecção, tem em conta uma expressão facial dita normal, caso essa expressão seja bastante expressiva pode ser um fator que dificulte essa detecção.

5. **Resolução da imagem e distância** - Várias imagens podem estar desfocadas, com ruído ou baixo contraste[3]. Uma imagem de resolução baixa não fornece muitas informações, pois a maioria delas é perdida, logo vai complicar na hora da detecção da face. A distância também é um fator pois normalmente, quando maior a distância mais informações se perdem acerca da imagem capturada.



Figura 2.1: Diferentes categorias de desafios para detecção facial[4].

A partir da detecção facial podemos analisar e estruturar vários algoritmos como alinhamento facial, verificação e autenticação das faces, rastreamento da pose da cabeça, reconhecimento de gênero/idade, reconhecimento de expressão facial e muito mais[1]. Ao longo dos anos foram estudadas várias abordagens para conseguir identificar faces em imagens, algumas das quais baseadas em características e baseadas em imagens. Quando existe a tentativa de extrair recursos chave da imagem a ser analisada, como por exemplo boca, olhos e nariz quando se fala de uma face, e posteriormente combinar com o conhecimento das características faciais, estamos a falar de uma abordagem baseada em características. Relativamente à abordagem baseada em imagens, temos como exemplo as redes neurais onde existe avaliações de partes de uma determinada imagem onde é decidido se existe uma face nela ou não. É feita então a melhor correspondência entre as imagens que vão ser testadas e as imagens que vão ser treinadas para esse fim.

Podemos observar diferentes técnicas para detecção facial na seguinte figura 2.2. Como abordado acima, pode ser baseado em características, como cor de pele e emoções e utilizando algoritmos para esse fim, como por exemplo o Haar Cascade. Relativamente a ser baseado em imagens, temos também um outro leque de métodos, como redes neurais, técnicas como PCA e SVM e outros mais.

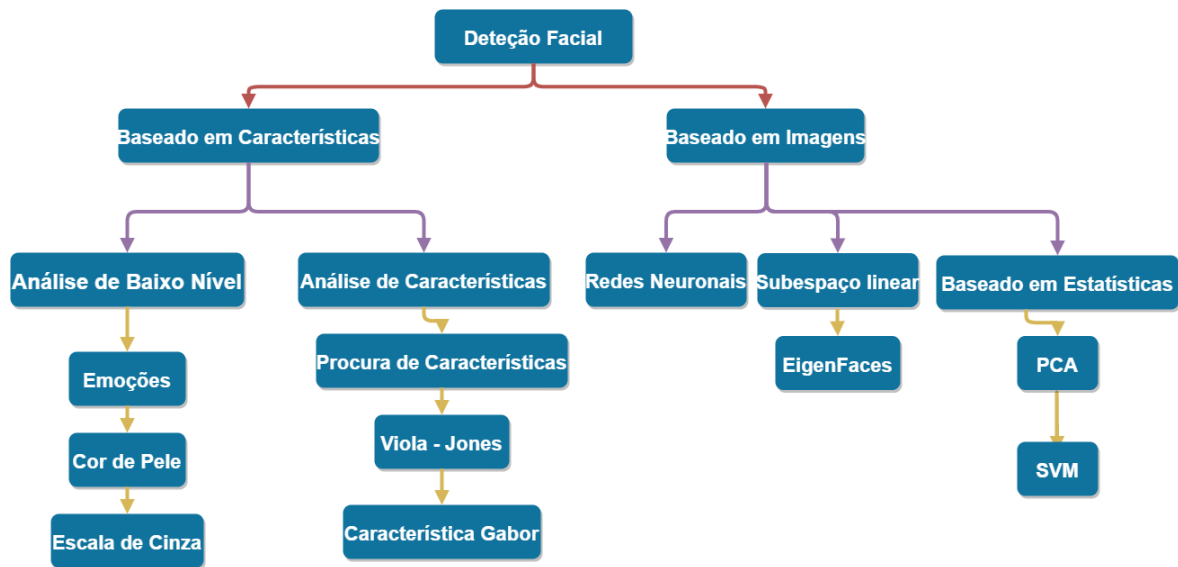


Figura 2.2: Diferentes técnicas para detecção facial.

2.1 ALGORITMOS DE DETECÇÃO DE FACES

Nesta seção, vai ser explicada detalhadamente três algoritmos para detecção facial: Haar Cascades, Histograma de gradientes orientados e Dlib CNN. Esses algoritmos serão usados durante a dissertação.

2.1.1 Haar Cascade

Nos dias de hoje, existem vários estudos acerca da detecção facial, especialmente utilizando o Haar Cascade Classifier. Este algoritmo foi desenvolvido por Paul Viola e Michael Jones [5] e descreve uma abordagem de Machine Learning (ML) para a detecção de faces e também de objetos que é capaz de conseguir processar imagens com uma taxa de acerto e uma velocidade bastante elevadas.

Este algoritmo é desenvolvido numa Biblioteca de computação visual de código aberto (OpenCV) e após ter uma imagem como método de entrada este algoritmo tem quatro fases definidas[6]. Estas fases são:

1. Seleção de características Haar
2. Criação de imagem integral
3. Treino de classificadores por um algoritmo de ML baseado no AdaBoost
4. Classificadores em cascata

Estas características consistem em um ou vários retângulos adjacentes que são utilizados para conseguir detetar vários pontos importantes numa face [7]. Existem pontos característicos que englobam todas as pessoas, como por exemplo a posição dos olhos, da boca e do nariz. É então realizado o scan dessas características, começando no canto superior esquerdo e terminando no canto inferior direito.

Cada característica fornece um valor, que é calculado pela subtração da soma das intensidades dos pixels que estão presentes na área onde o retângulo é branco e pela soma da intensidade

dos pixels que estão presentes na área onde está existente o retângulo preto. Assim, ao utilizar estas características e agrupando então um conjunto de pixels, torna-se mais rápido e com menos custo computacional do que ser feita a análise pixel a pixel.

Podemos observar na Figura 2.3 características que podem ser utilizadas para identificar na imagem os olhos e o nariz, pois normalmente existe sempre uma grande diferença na intensidade dos pixels onde os retângulos brancos e pretos estão colocados, daí ser possível conseguir identificar uma face mais facilmente.



Figura 2.3: Características Haar mais significativas em uma face[8]

Foi definida então posteriormente, um método onde é possível diminuir o tempo computacional, utilizando então um método denominado imagem integral, que pode ser calculado através de uma imagem, utilizando operações por pixel da imagem.

A localização x e y da imagem integral (II) é então calculado a partir da posição de cada pixel (x,y) , onde vai ser somada os valores dos pixels a cima e à esquerda do mesmo (IO), ou seja:

$$II(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} IO(i, j) \quad (2.1)$$

Na Figura 2.4 podemos observar um exemplo prático do algoritmo da imagem integral. O valor da imagem integral na posição 1 é A. O valor na posição 2 é $(A + B)$. O valor na posição 3 é $(A + C)$ e o valor na posição 4 é $(A + B + C + D)$.

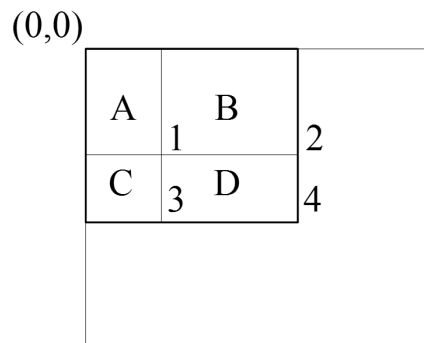


Figura 2.4: Cálculo do gráfico da imagem integral [9].

Mesmo assim, apesar de ter um custo computacional menor e ter uma velocidade mais eficiente, existem ainda bastantes características calculadas que são irrelevantes e inviáveis. Daí ser utilizado o algoritmo AdaBoost.

Este algoritmo é uma abordagem de ML utilizada para a redução de recursos e assim, conseguir determinar o menor número possível de conjunto de características do tipo Haar. É criado um classificador para conseguir classificar as imagens usando então este algoritmo, onde existe uma combinação de "classificadores fracos", para conseguir criar "classificadores fortes". A partir das várias características calculadas nas etapas anteriores, sabendo que nem todas fazem parte da face da pessoa, vão ser somente escolhidas algumas delas, onde seja possível ter uma maior precisão de uma deteção facial.



Figura 2.5: Exemplo de características relevantes[10].

Podemos observar na figura 2.5 que a partir de uma característica vertical o nariz é uma boa opção de escolha, porém o lado da face será irrelevante.

Como última fase do algoritmo Haar Cascade temos os classificadores em cascata. Apesar de termos várias fases, que já melhoraram bastante o algoritmo é possível através de classificadores de cascata, diminuir o tempo de computação e acerto de uma deteção facial. Consiste então na concatenação de classificadores mais completos em uma estrutura de cascata. Cada fase dos classificadores em cascata, treina os "classificadores fracos" para obter um "classificador forte". O principal objetivo é determinar se uma dada sub-janela, tem a presença de uma face ou não. Caso encontre uma face, passa para o seguinte estado, caso não exista, essa sub-janela é rejeitada e posteriormente descartada.

Podemos observar na Figura 2.6 o funcionamento da cascata e o processo de filtração de cada classificador.

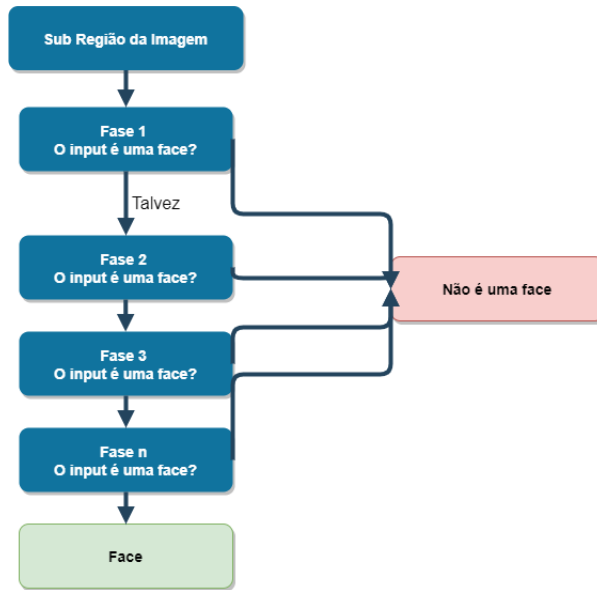


Figura 2.6: Classificador em cascata[11].

2.1.2 Histograma de Gradientes Orientados

O Histograma de gradientes orientados (HOG) é um descritor de recursos normalmente utilizado em computação visual, onde é possível fazer o processamento de imagens e normalmente utilizado para a detecção de objetos [12]. Neste caso, o "objeto" a ser detectado e analisado são as faces das pessoas. Este algoritmo é baseado nos objetos locais, que neste caso de detecção são as faces das pessoas e podem ser descritas pela distribuição de gradientes de intensidade local ou as direções das arestas das imagens. Este método divide-se em várias fases:

1. Cálculo das direções das arestas das imagens e gradientes dos pixels
2. Agrupamento de pixels em células
3. Agrupamento de células em blocos
4. Cálculo do histograma de orientação
5. Concatenação dos vários histogramas

Inicialmente cada imagem é dividida em pequenas partes, que são denominadas de células. Estas células têm a dimensão de um pixel, e obtendo essas células é calculada um histograma de direções de gradiente, agrupando depois num bloco. Podemos observar o formato possível da orientação do histograma e o formato das células e blocos na Figura 2.7.

Ao obter todos esses histogramas é representado então o descritor do algoritmo que possui as informações com os pontos de interesse. Após obtermos o conjunto de histogramas e para uma melhor precisão da detecção é realizada uma normalização das células, calculando a intensidade das várias regiões das imagens onde resulta uma melhoria de resultados relativamente às mudanças de sombras e também de iluminação.

O descritor Histogramas de Gradientes Orientados necessita agora de um método capaz de discriminar as características que foram previamente extraídas e conseguir então determinar a que classe pertencem (se é de facto encontrada ou não uma face).

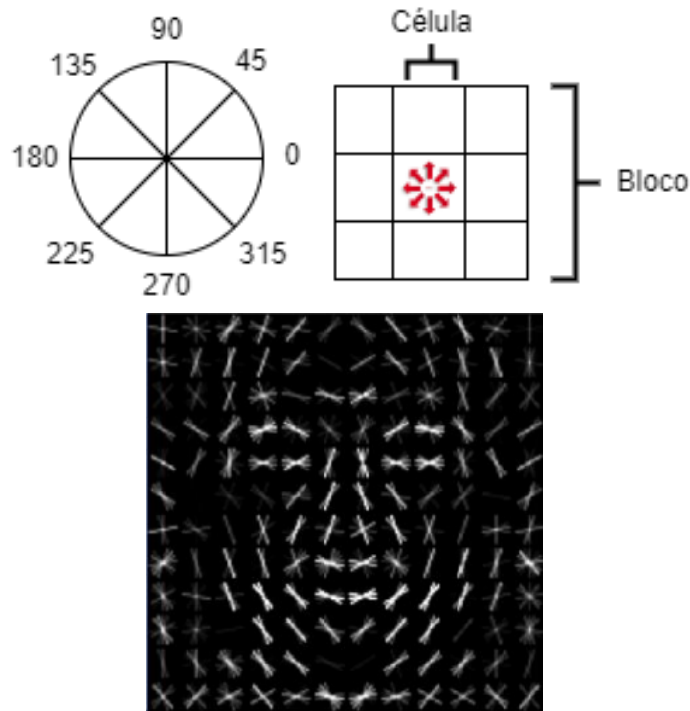


Figura 2.7: Orientação do histograma e formato das células e histograma de gradientes orientados calculado [9].

Utiliza-se então Máquinas de Vetores de Suporte (SVM). O conjunto de histogramas calculados anteriormente são então enviados para uma máquina de vetores de suporte, que funciona como um método de aprendizagem supervisionado, que consegue analisar os valores de input, tentar arranjar padrões nesses valores e finalmente conseguir classificar, neste caso, como existência ou não de uma face.

2.1.3 Dlib CNN

Uma Rede neural convolucional (CNN) é um tipo de rede neuronal artificial que vem sendo utilizado em reconhecimento, processamento e análise de imagens digitais [13]. Uma Rede neuronal Convolucional é um algoritmo de deep learning que possui a capacidade de diferenciar objetos e conceder relevâncias a partir de uma imagem de entrada. Neste caso, através de uma imagem de entrada de uma face. Sendo uma imagem uma matriz de valores de pixels, através da convolução é possível treinar um CNN para conseguir entender as características da imagem. Neste caso, o método de detecção é então baseado em deep learning e utiliza para detetar faces em imagens, o método Detecção do objeto de margem máximo (MMOD). Segundo estudos, necessita de um grande poder computacional e o tempo de processamento não é o menor relativamente a outros métodos, porém as taxas de acerto e os verdadeiros positivos são bastante elevados.

2.2 RECONHECIMENTO FACIAL

Nos dias de hoje, o reconhecimento facial é bastante importante, sendo mesmo um dos tópicos mais abordados em computação visual e reconhecimento de padrões[14]. Do ponto de vista do ser humano é bastante fácil e trivial reconhecer uma face de uma pessoa, porém, quando falamos de computação visual a matéria torna-se sempre mais complexa.

A partir deste reconhecimento é possível, através de uma imagem onde é detetada uma face de uma pessoa, e através de uma base de dados onde é possível comparar a face detetada com a face guardada nessa base de dados, conseguir através de algoritmos de reconhecimento determinar quem é a pessoa em questão. Existem vários sistemas para reconhecer faces de pessoas, estando elas classificadas como locais, holísticas ou híbridas, onde se pode observar na Figura 2.8 A primeira abordagem, **abordagem local**, não tem em conta a face total da pessoa, sendo classificada apenas por certas características faciais. Em relação às técnicas de keypoints, temos o exemplo da Transformação de recursos invariantes de escala (SIFT) e Características Robustas e Rápidas (SURF), onde o objetivo é encontrar uma elevada quantidade de características nas imagens, e como cada ponto chave é único, tenta fazer depois a suas correspondência. Acontece o contrário relativamente à **abordagem holística**, onde a face é utilizada completamente, sendo o verdadeiro input, para identificar características que correspondam a ele. Temos então vários casos como PCA, EigenFaces, SVM que vão ser abordados mais à frente na dissertação. Finalmente como ultima abordagem, a **abordagem híbrida**, trata de melhorar a precisão do reconhecimento facial, utilizando vários recursos globais e locais.

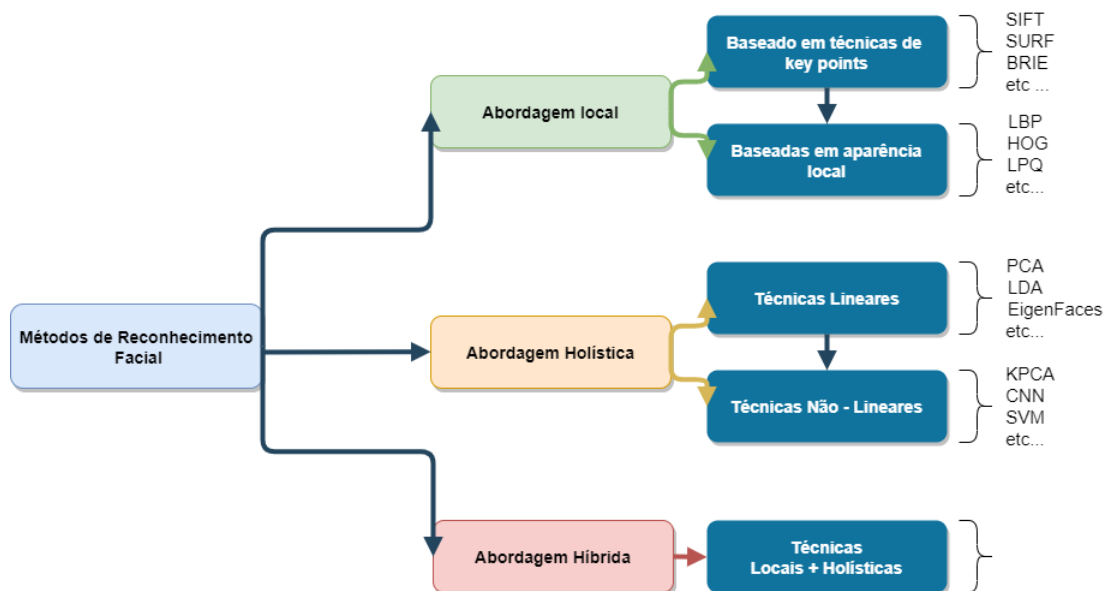


Figura 2.8: Diferentes abordagens de reconhecimento facial[15].

Esta tecnologia é usada com vários propósitos, sendo mais relacionada com temas acerca de segurança.

Esta tecnologia pode ser aplicada a uma grande variedade de cenários. De seguida pode-se

ver algumas das utilizações a partir de organizações e indivíduos de utilizarem estas tecnologia para diferentes casos de segurança [16].

1. **Autenticação de identidade** : Cada vez mais, o mundo está a tornar-se mais inteligente, logo possibilita a haver uma maior segurança. Utilizando métodos como reconhecimento facial para conseguir a autenticação de várias formas é um método onde existe uma maior segurança.

Exemplos como segurança em controlo de fronteiras em viagens, acesso de automóveis, controlo de pessoas em eventos é uma forma de, não só existir uma maior segurança, como diminuir o tempo que é necessário para efetuar essa verificação.

2. **Segurança** : Exemplos como alerta de terroristas e pessoas a causarem danos em manifestações são também casos de utilização de sistemas de reconhecimento facial.

2.3 TÉCNICAS E ALGORITMOS DE RECONHECIMENTOS DE FACES

Nesta subsecção, vão ser explicados dois algoritmos tradicionais de reconhecimento facial: EigenFaces e FisherFaces e várias técnicas utilizadas para facilitar e aumentar a taxa de reconhecimento nestes mesmo algoritmos.

2.3.1 PCA

Análise de Componentes Principais é um método ou técnica matemática, que a partir de um conjunto de dados e de uma análise, consegue gerar um grupo mais pequeno de componentes que conseguem sintetizar os dados originais, dando a possibilidade de diminuir a dimensionalidade dos dados e melhorar na deteção de padrões, preservando as partes mais importantes. Ao conseguir obter a redução da dimensionalidade, existe então a vantagem, de não trabalhar com todos os componentes e conseguir remover a informação e os componentes que não são úteis[17].

2.3.2 LDA

Análise discriminante linear, com o mesmo objetivo da Análise de Componentes Principais é uma técnica matemática que visa também em diminuir a dimensionalidade dos dados porém, neste caso tem em conta a classificação desses dados. O PCA tenta encontrar os principais eixos de variação dos dados em qualquer dimensão, porém não faz essa divisão em classes, ou seja, não é possível saber se esses dados pertencem ou não à mesma classe, somente tenta reduzir a dimensionalidade e remover a informação que não é útil. Podemos ver na Figura 2.9 do lado direito, a divisão realizada entre classes, o que não acontece relativamente ao PCA. Através da Análise discriminante linear, ao serem mapeados os dados, automaticamente caso sejam da mesma classe, são alocados próximos aos da própria classe, o que vai por sua vez, conseguir facilitar o estado e o processo da classificação desses dados.

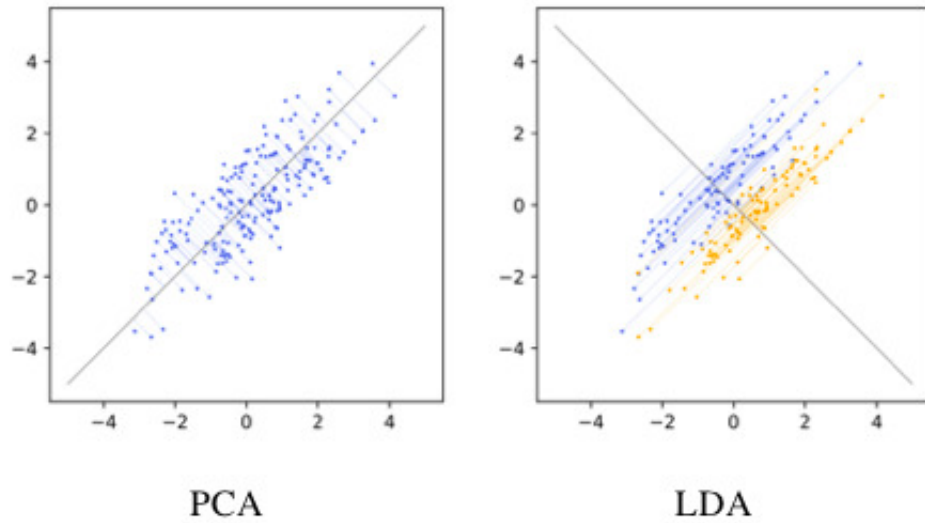


Figura 2.9: Diferença entre PCA e LDA[18].

2.3.3 EigenFaces

Vários métodos de reconhecimento facial, focam-se em tentar encontrar os pontos mais importantes e mais característicos da face das pessoas, como os olhos, o nariz, a boca, etc... Este método, ao invés de se concentrar nesses pontos principais, utiliza toda a informação da face[19]. É então formado um vetor chamado Eigenvector para cada uma das imagens dos indivíduos, onde todos os pixels dessas características faciais vão ser o tamanho do vetor EigenVector. Consideremos uma imagem de uma face com uma matriz de dimensão $N_x \times N_y$, ao concatenar essa matriz é possível chegar à matriz representada na Figura 2.10. Então, cada imagem pode ser vista como um ponto. Este conjunto de pontos que representam imagens são calculados através do PCA.

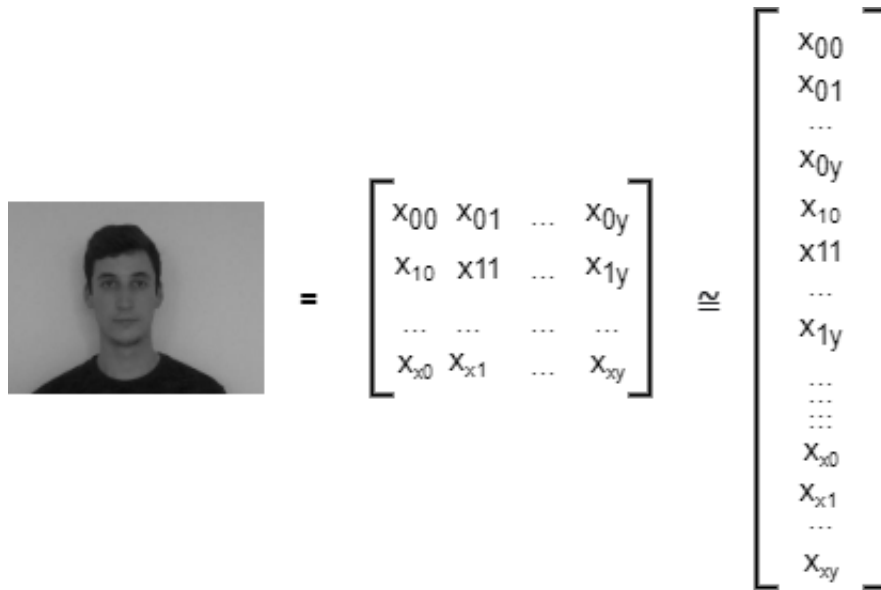


Figura 2.10: Concatenação de uma face em vetor único.

Por vezes, o número de pixels é bastante elevado, e trabalhar com todo esse número, iria ser bastante custoso computacionalmente e ter um tempo de resposta também muito elevado. Devido a isso é então utilizada a técnica de machine learning chamada PCA, onde vai então, a partir de um conjunto de dados e de uma análise, gerar um grupo mais pequeno de componentes dando a possibilidade de diminuir a dimensionalidade dos dados e também extrair as características de uma pessoa, baseado nas imagens de input. Cada vetor, Eigenvector descreve então a variação dos pixels em várias regiões de uma imagem, ou seja cada um dos vetores, está associado à variação dos pixels de cada característica facial.

Posteriormente, são avaliadas as fotos de treino, e extraídas as informações importantes, onde é criada uma face média com base no conjunto de dados de imagens. A face é reconhecida, projetando a imagem inicial, para o espaço das eigenfaces. Após decorrer a projeção, este método possibilita a classificação de faces através da comparação da distância Euclideana, entre os vetores eigenfaces no novo espaço e os eigenfaces gerados pela imagem de input.

Após ser feita essa comparação é nos fornecido o valor dessa distância. Caso esse valor seja 0, a mesma imagem está a ser analisada, quando mais se distanciar do valor nulo, maior existe uma diferença dos pixels nas imagens, logo pode ser possível deixar de reconhecer uma face, consoante o valor de threshold atribuído(distância limite)[20]. O algoritmo detalhado matematicamente vai ser referenciado de seguida [15], [21] :

Tendo então uma imagem com uma dimensão $N_x \times N_y$, e um conjunto de imagens de treino $X_1, X_2, X_3, \dots, X_N$, sendo N a quantidade de imagens desse conjunto. Como existem várias variâncias de informação redundante da face é relevante considerar uma média geral. É então calculada pela média aritmética de todos os vetores do conjuntos de imagens de treino. A face média do conjunto é então calculada por:

$$X = \frac{1}{N} \sum_{i=1}^N X_i \quad (2.2)$$

De seguida é calculada a matriz de covariância para conseguir representar o grau de dispersão dos vetores relacionados ao vetor médio.

$$Q = \frac{1}{N} \sum_{i=1}^N (X - X_i)(X - X_i)^T \quad (2.3)$$

Após ter a matriz de covariância, os eigenvectors e os seus valores são calculados pela seguinte expressão, onde V é o conjunto de eigenvectors Q associados ao eigenvalue λ .

$$CV = \lambda V, (V \in R_n, V \neq 0) \quad (2.4)$$

Posteriormente projeta-se todas as imagens de treino da pessoa correspondente i , para o sub-espço de eigenVector.

$$y_k^i = \omega^T(x_i), (i = 1, 2, 3, \dots, N) \quad (2.5)$$

onde y_k^i são as projeções de x , as eigenfaces.

2.3.4 FisherFaces

O método fisherface é também, como o eigenfaces um método de reconhecimento bastante popular e é baseado no mesmo princípio de similaridade do método Eigenfaces[22], onde tem em conta a variância luminosa e as expressões faciais. Ao invés de utilizar a técnica PCA, utiliza a LDA que visa em diminuir a dimensionalidade dos dados porém, neste caso tem em conta a classificação desses dados. Apesar de não existir um maior foco na variação das imagens, como no método anterior, tenta ao máximo separar as classes de teste e de treino, a partir das características mais relevantes da face da pessoa.

Este algoritmo segue então várias etapas. Inicialmente é calculada então a face média por classe e posteriormente a face média geral. Para calcular estes valores basta somar pixel a pixel de todas as imagens de uma determinada classe e de todas as imagens do conjunto de treino respetivamente. O algoritmo detalhado matematicamente vai ser referenciado de seguida [23]:

Tendo então uma imagem com uma dimensão $N_x \times N_y$, e um conjunto de imagens de treino $J_1, J_2, J_3, \dots, J_m$, sendo m a quantidade de imagens desse conjunto. Inicialmente é criado um vetor Γ , onde são concatenadas as linhas da imagem em questão J_j , com determinados pixels, onde é formado então um vetor. Sendo c_j a classe j e q_j o número de imagens na classe c , a face média por classe é calculada por:

$$\theta_{c_j} = \frac{1}{q_j} \sum_{k=1}^{q_j} \Gamma_k \quad (2.6)$$

Posteriormente, tendo este valor é possível calcular a matriz de dispersão intra-classe e inter-classe que referem o quanto as imagens de uma mesma pessoa diferem das outras e de pessoas diferentes, respetivamente.

$$S_W = \sum_{j=1}^c \sum_{\Gamma_k \in c_j} (\Gamma_k - \theta_{c_j})(\Gamma_k - \theta_{c_j})^T \quad (2.7)$$

$$S_B = \sum_{j=1}^c q_j (\theta_{c_j} - \theta)(\theta_{c_j} - \theta)^T \quad (2.8)$$

De seguida, semelhante ao PCA, deve-se calcular a matriz de projeção, que é uma matriz onde se tenta aumentar a relação entre as classes futuras e as classes das imagens projetadas.

$$W_{opt} = \underset{w}{\operatorname{argmax}} = \frac{w^T S_B W}{w^T S_W W} \quad (2.9)$$

Após todas as projeções necessárias é feito então o cálculo da similaridade, utilizando um classificador ou a distância euclidiana.

2.3.5 Framework Deepface - OpenFace

Um pipeline de um modelo de reconhecimento facial, normalmente consiste em quatro estágios, nomeadamente [24]:

1. Detecção
2. Alinhamento
3. Representação
4. Verificação

DeepFace é uma biblioteca reconhecimento facial onde possui várias features implementadas e a serem utilizadas. É uma biblioteca open-source onde lida com todos estes estágios, ditos anteriormente em segundo plano. Várias das features que podem ser utilizadas são: Verificação de faces, onde é possível comparar duas faces em imagens e verificar se essas duas imagens pertencem à mesma pessoa, e ainda reconhecimento da face, onde, ao invés de executar somente uma vez essas verificação da imagem, realiza essas verificações para todas as imagens para finalmente encontrar alguma semelhante num base de dados de imagens. Outra feature é a análise de atributos faciais [25], onde é possível, através das características visuais da face da imagem, extrair várias atributos como género da pessoa e idade. O método utilizado foi o Openface. Este método é um modelo para reconhecimento facial e utiliza uma rede neuronal convolucional profunda. É possível observar na Figura 2.11, as partes mais importantes da implementação do OpenFace.

Openface é um algoritmo deep learning de reconhecimento facial. Este utiliza tecnologia como Torch, FaceNet, dlib e scikit-learn. Este método é treinado com 500 000 imagens de várias base de dados de faces públicas. Como é possível observar na Figura 2.11. Torch é um framework que é utilizado pelo OpenFace na tentativa de obter o máximo de flexibilidade e velocidade ao construir os diferentes algoritmos. Neste caso consegue aperfeiçoar a rede neuronal quando está a efetuar a extração das características da face. FaceNet é uma abordagem na qual é gerada um mapeamento facial das faces através de uma rede neuronal

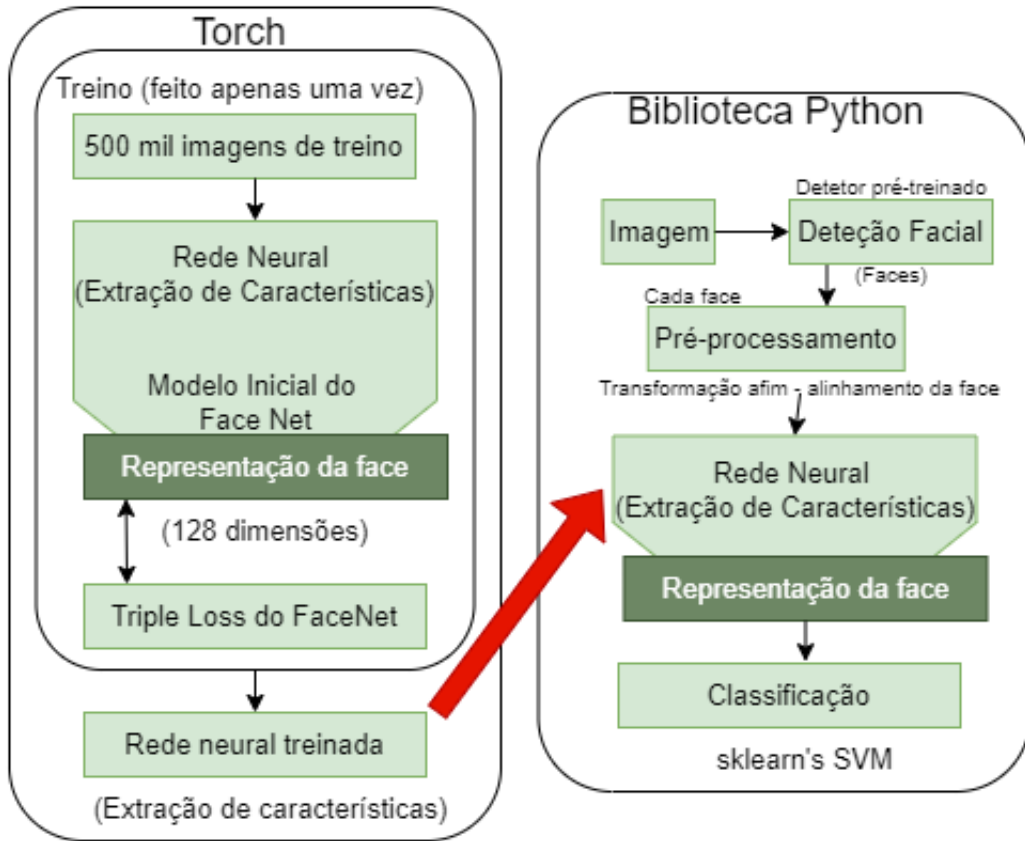


Figura 2.11: Estrutura do OpenFace [26].

convolucional e consegue fazer a assimilação das faces através da distância euclideana. É utilizada a triple loss do facenet, onde é possível através de várias imagens, coincidentes com a imagem a ser comparada e de pessoas diferentes, conseguir agrupar todas essas pessoas. É criado um vetor de 128 dimensões que representa o modelo geral da face e de seguida melhorada a eficiência do modelo. Após obter o modelo treinado, inicia-se o processo de deteção, através do algoritmo HOG e e processamento da imagem onde é possível observar com detalhe na secção 3.4. Como resultado é adquirida a classificação da pessoa a ser reconhecida.

Metodologia

Neste capítulo vai ser apresentada toda a metodologia para o estudo de vários algoritmos de detecção e de reconhecimento de faces em condições não cooperativas. Na figura 3.1 é apresentado em módulos o pipeline tradicional de um sistema de reconhecimento facial.

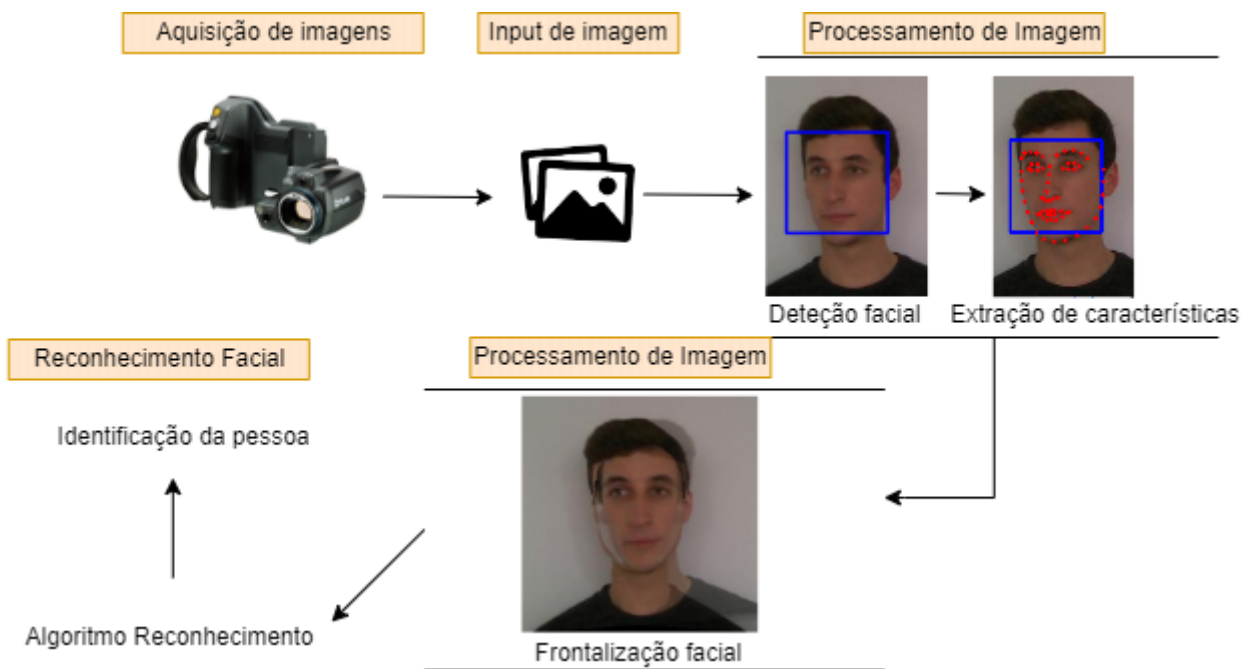


Figura 3.1: Fluxograma do sistema.

Para conseguir realizar o reconhecimento facial é necessário completar várias etapas. Os principais módulos são o processo de aquisição de imagens, onde se podem adquirir imagens de vários espectros possíveis além do visível. Tipicamente existe um pré-processamento das imagens onde é possível ajustar as imagens a diversos algoritmos e obter várias características da face da pessoa. De seguida, termina com um algoritmo de reconhecimento facial que vai comparar as características extraídas da imagem adquirida com as características das faces presentes na base de dados, e assim obter uma classificação da pessoa. O foco deste trabalho

consiste na exploração de algoritmos existentes na literatura para a deteção e reconhecimento em ambientes não-cooperativos. De forma a tentar melhorar a performance destes algoritmos foram testadas diversas abordagens de pré-processamento, nomeadamente frontalização e outras descritas nas secções seguintes.

Numa tentativa de melhorar os resultados era pretendido explorar a utilização de várias fontes de informação. Visto que não existia nenhum dataset que proporcionava as melhores condições para esta dissertação, neste trabalho foi proposta a criação de um dataset que foi realizado na Universidade de Aveiro com equipamento gentilmente cedido pela academia militar. De reforçar, que todos os participantes assinaram uma declaração para autorização do uso das imagens. Este documento situa-se em Autorização para utilização de imagens. Não foi possível no decorrer desta dissertação testar a fusão sensorial das várias imagens adquiridas, contudo foi dado um primeiro passo no sentido de desenvolver um modelo de deteção facial em imagens térmicas. Relativamente ao estudo efetuado na metodologia foram realizadas as seguintes procedimentos:

3.1 AQUISIÇÃO DE IMAGENS E INPUT

Inicialmente, antes de realizar qualquer processamento de imagens é necessário uma base de dados com algumas imagens para poder processá-las. Para realizar a base de dados foram então adquiridas várias imagens. Estas aquisições foram adquiridas de diferentes modos. Neste caso, todo o processo foi efetuado com várias câmaras que possuem a habilidade de capturar imagens em vários intervalos espectrais. O tipo de equipamentos utilizados e as condições de aquisição, estão mais detalhados na sub secção 4.1.1.

Ao serem necessárias a existência de imagens de treino para o reconhecimento é necessário obter também, além das imagens de treino, imagens de teste. Foram então divididas em diferentes pastas, como é possível observar na Figura 3.2, imagens para teste e imagens para treino. Nas imagens para treino, cada pessoa foi identificada por um determinado id, com as suas imagens das faces com diferentes ângulos de rotação.

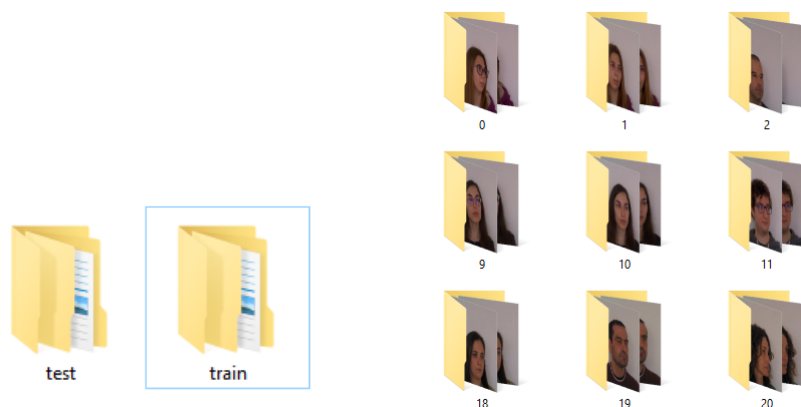


Figura 3.2: Organização das imagens para os algoritmos de deteção e reconhecimento.

A imagens de teste, são as imagens que vão servir de input para o sistema. Após ter a base de dados das imagens organizada é possível passar para a seguinte etapa.

3.2 PROCESSAMENTO DE IMAGEM

Detetar a face nas imagens é o seguinte passo do sistema. Após essa deteção irá ser realizada a extração dos pontos característicos da faces em questão. Tudo isto pertence ao processamento da imagem. As imagens adquiridas no espetro visível, possuem um padrão de cores Vermelho, Verde e Azul (RGB), ou seja, cada pixel da imagem possui 3 sub-pixeis com intensidades diferentes (vermelho, verde e azul). Ao ser recebida esta imagem, vai ser convertida para a escala de cinzentos. Várias das razões para se efetuar esta conversão deve-se ao facto de:

1. Escalas de cinza são uni-dimensionais, ou seja têm um canal apenas para processamento.
2. Muitos algoritmos da área de computação visual, só permitem a sua utilização em imagens em escala de cinza.

O processamento de uma imagem RGB, vai ser sempre 3 vezes com maior dimensão do que imagens em escala de cinza, o que num processamento pode levar a várias desvantagens. Para deteção facial, são utilizados diferentes métodos, tais como Haar Cascades, Histograma de Gradientes e um método de deep learning CNN. Toda a explicação e funcionamento destes três algoritmos estão explicados e detalhados na secção 2.1. Existem modelos disponibilizados pela OpenCV [27] e pela Biblioteca digital (Dlib), que são bibliotecas para processamento de imagens tanto em tempo real, como imagens base, que se tornou uma ferramenta bastante utilizada na visão computacional.

Relativamente ao algoritmo Haar, a deteção facial é realizada a partir de um ficheiro .xml, que funciona como um modelo treinado com um número elevado de faces utilizadas de uma base de dados pública. Este modelo requer imagens positivas, que são imagens onde há a existência de faces (objeto a ser detetado) e imagens onde não há a existência de nenhuma face. Após efetuar a deteção facial é nos fornecido a localização da face, na imagem de teste. Após obter a deteção da face é possível iniciar a extração dos pontos com as características principais da face. Partes da face como olhos, lábios e sobrancelhas são alguns dos pontos mais importantes e característicos de uma face humana. A biblioteca Dlib, que é uma biblioteca de ML, possui uma rede neuronal onde extrai as mais importantes características da face de uma pessoa.

Após se efetuar a deteção facial é possível proceder à extração dos vários pontos característicos da face. Como referenciado anteriormente, a biblioteca Dlib possui um modelo onde estima a posição das 68 coordenadas (x,y) que mapeia as características específicas da face humana. É possível observar na Figura 3.3 da esquerda, o modelo principal com a ilustração dos pontos mais característicos da face. Na imagem da direita observa-se a imagem de teste, com todos esses pontos implementados.

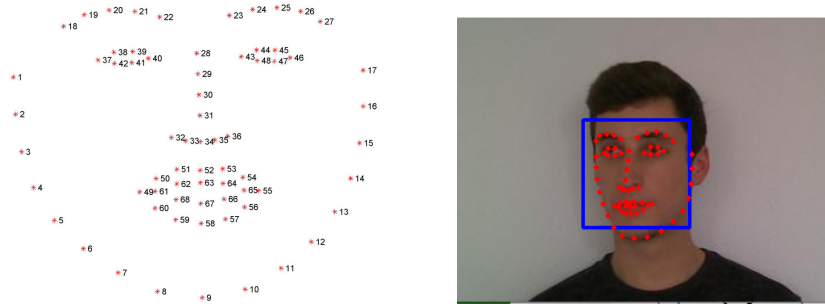


Figura 3.3: 68 coordenadas (x,y) na face de imagem de teste [28].

3.2.1 Frontalização

Ao executar o programa, e após ser obtida a detecção facial e os pontos característicos e antes de qualquer treino de dataset para posterior reconhecimento é feita uma frontalização da face da imagem que se quer testar, de modo a comparar essa imagem a uma presente no dataset de treino. O principal objetivo é acomodar a rotação da face para um ângulo de rotação de 0° , ou seja uma face frontal.

A frontalização é o processo, de sintetizar rostos com rotações não frontais, ou seja imagens retiradas em ambientes não cooperativos e transformá-as em imagens frontais [29]. Para este efeito foi utilizado um código open-source disponível no Github¹ por Douglas Souza. Enquanto muitos algoritmos de frontalização tentam para cada uma imagem de consulta aproximar formas faciais Tridimensionais (3D) para cada uma delas, este algoritmo utiliza uma única superfície 3D não modificada tentando uma aproximação com cada uma das imagens de teste. Esta versão originalmente, foi implementada numa versão Python 2.0, deste modo, foi necessário serem feitas alterações de modo a conseguir executar e implementar no presente sistema. É possível observar na Figura 3.4 um exemplo desta frontalização.

¹Face frontalization - <https://github.com/dougsouza/face-frontalization> [29]

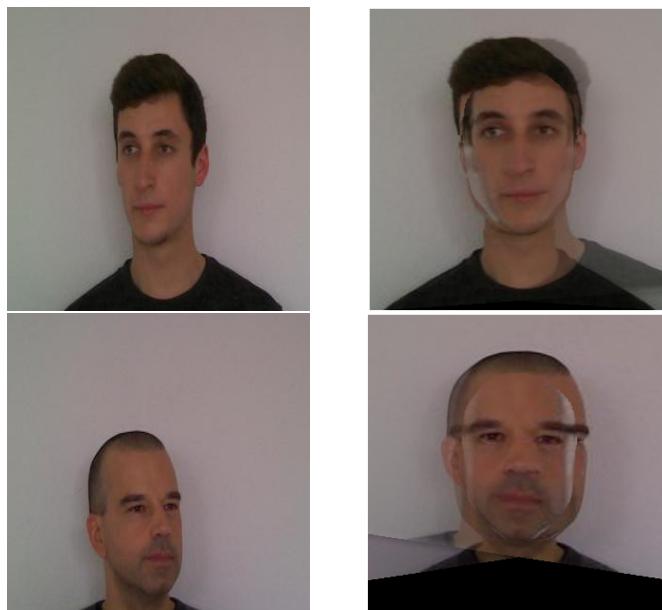


Figura 3.4: Exemplo de transformações após a aplicação da frontalização numa face. (a) Imagem de teste (id = 4) com um ângulo de rotação de -20° (b) Imagem "a" frontalizada (c) Imagem de teste (id = 2) com um ângulo de rotação de $+30^\circ$ (d) Imagem "c" frontalizada

3.3 RECONHECIMENTO FACIAL

Completando o processamento da imagem de teste, com a frontalização dessa mesma, de seguida é avançada mais uma etapa. Como os algoritmos vão iterar somente sobre pastas é necessários eliminar todos os ficheiros correspondentes da pasta "train" que não se identifiquem com pastas.

3.3.1 Detecção de imagens e respetivo treino

Nesta etapa, vai-se iniciar o treino das imagens com os respetivos algoritmos de reconhecimento. Cada um dos algoritmos, eigenfaces e fisherfaces pode ser executado com um determinado método de deteção, observável na Figura 3.5. É possível, após executar o script python escolher, qual o método de deteção pretendido (Haar, HOG, CNN) para detetar as faces nas imagens de treino e de seguida completar o treino dessas faces.

```
(TESE) C:\Users\Menino\Desktop\Tese-de-Mestrado\TESE\CODE>python recognize_detection.py 350 350
welcome
Choose one face detection method with Eigenfaces Recognition,
1 - Haar Cascade
2 - HOG
3 - CNN
Choose one face detection method with FisherFaces Recognition,
4 - Haar Cascade
5 - HOG
6 - CNN
7 - DeepFace
```

Figura 3.5: Terminal após execução do script python.

Após iterar sobre a pasta train, e as respetivas pastas com os identificadores das determi-

nadas pessoas é realizada então a detecção facial para cada uma das imagens.

Relativamente ao método Haar é utilizado um método da biblioteca OpenCV que trabalha com este tipo de dados.

O método é o seguinte e apresenta os seguintes parâmetros:

```
faces_haar = face_cascade.detectMultiScale(image, scaleFactor, minNeighbors,  
minSize)
```

1. *image* - Imagem a ser detetada
2. *scaleFactor* - Redimensionar a imagem detetada
3. *minNeighbors* - Define quantos objetos são vizinho para conseguir detetar ou não uma face
4. *minSize* - Tamanho mínimo possível

Em relação ao algoritmo HOG, o procedimento realizado com as imagens de treino é semelhante ao utilizado nas imagens de teste. Mais uma vez, para a detecção facial com um algoritmo CNN é utilizada um modelo treinado da biblioteca Dlib. Alguns destes algoritmos, necessitam que a imagem de teste e a imagem de treino tenham as mesmas dimensões. Logo, para os 3 algoritmos, após a detecção da face é realizado um redimensionamento da imagem da face detetada. É possível observar o processo na Figura 3.6.

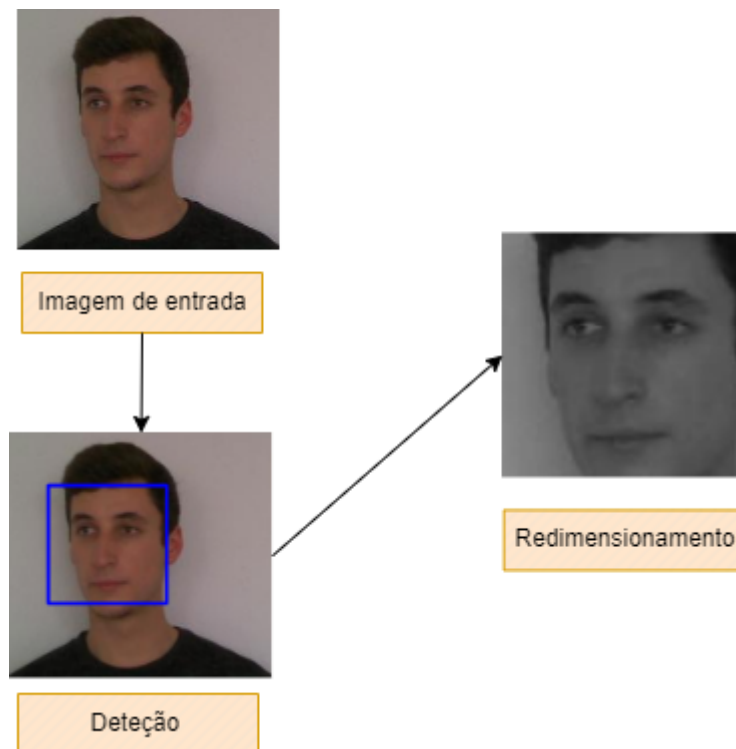


Figura 3.6: Redimensionamento das imagens.

Para iniciar o script é necessário colocar como argumento a dimensão da imagem. Como tanto a imagem de teste e as imagens de treino têm de possuir as mesmas dimensões foi definido um tamanho de 350x350 pixels.

Posteriormente é realizado o treino das imagens de todos os indivíduos, e é feita uma previsão de qual a face da pessoa que está a ser analisada, existindo um output com o id da pessoa prevista.

3.4 DEEPFACE FRAMEWORK - OPENFACE

A framework Deepface segue o seguinte pipeline de reconhecimento facial. Estágios como detecção, alinhamento, representação e verificação estão presentes nesta framework que vão ser detalhados de seguida. Inicialmente, para a detecção da face, foi utilizado um detetor facial pré-treinado que usa o HOG. De seguida, após obter a face e os pontos característicos é efetuada um alinhamento da face. O principal objetivo é, através destes pontos característicos, conseguir aplicar o alinhamento da face, até as coordenadas dos dois olhos permanecerem horizontais.

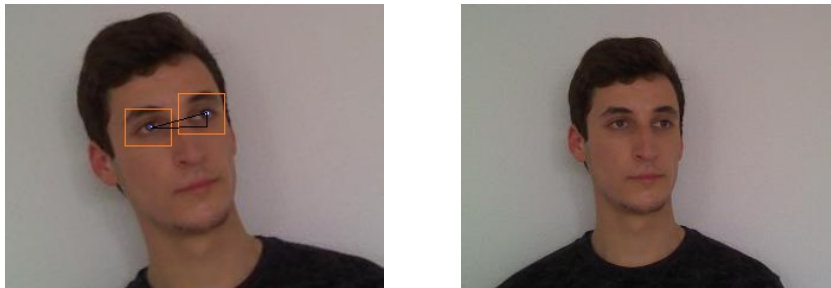


Figura 3.7: Exemplo de alinhamento de face através de trigonometria [24].

Normalmente, na anatomia de uma face, os olhos estão sempre na mesma linha horizontal, aplicando alguns cálculos de trigonometria através dos pontos fornecidos dos dois olhos é possível a partir de rotações alinhar a imagem. É possível observar um alinhamento de imagem na Figura 3.7. Após alinhamento é então realizada a verificação. Como estes modelos de reconhecimento são CNN são responsáveis por representar as imagens de teste como vetores. Ao obter estes vetores, vai ser possível a partir de cálculos de distâncias verificar se a imagem de teste vai coincidir com alguma imagem de uma determinada base de dados. O método de reconhecimento utilizado foi o OpenFace. Valores de confiança foram métricas utilizadas relativamente a este algoritmo de reconhecimento e são calculados através da distância euclideana comparando o valor de threshold definido com o valor da equação matemática referenciada na Equação 3.1.

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + ((p_n - q_n)^2)} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (3.1)$$

Em caso do valor gerado ser superior ao threshold, o resultado do reconhecimento será verdadeiro. No caso de ocorrer o contrário, o resultado é negativo. Apesar de não existir um valor certo de threshold, variando sempre de imagens para imagens, é possível chegar a um valor próximo.

3.5 TREINO DE CLASSIFICADOR PARA DETECÇÃO FACIAL DE IMAGENS TÉRMICAS

Relativamente ao treino de um modelo de deteção facial em imagens térmicas o percurso é o seguinte:

Para conseguir então a deteção facial em imagens térmicas é necessário criar um classificador e treinar um modelo com ML, onde seja aprendido a determinar certas características que com outro ficheiro XML não é possível, para conseguir posteriormente uma classificação. Os vários passos que são necessários para criar este classificador, são os seguintes:

1. Aquisição de dataset positivo e negativo
2. Criação de um arquivo de vetores baseado nas imagens positivas(ficheiros de treino)
3. Treinar o classificador

3.5.1 Aquisição de dataset

Inicialmente, como passo número **1** é necessário proceder a uma recolha de imagens. Para recolher estas imagens, e para ser possível treinar o algoritmo é necessário tanto imagens com o objeto a ser detetado (faces) como imagens sem esse mesmo objeto. [30]. Imagens positivas entende-se como imagens que possuem objetos que são pretendidos de detetar. Existindo uma variação desse objeto, como várias representações, várias condições de iluminação e diferentes perspetivas é sempre uma maior valia para a classe de treino, pois possui mais dados para analisar.

Quanto às restantes imagens é completamente o oposto das imagens anteriores. Nestas, possuem várias imagens onde o objeto que é pretendido detetar, não está presente. Relativamente ao dataset efetuado na universidade, devido a serem cerca de 200 imagens térmicas de faces, e serem necessário um número mais elevado de imagens de treino, para existir uma maior precisão de deteção, estas imagens vão ser utilizadas para imagens de teste.

Relativamente às imagens positivas é necessário delinear e anotar a área de interesse dessas imagens positivas. Após anotar estas imagens, foi gerado um ficheiro, onde é possível observar o nome relativo a cada imagem, e as coordenadas dos objeto alvo, definidas na anotação. Este ficheiro, 'info.txt' vai posteriormente ser utilizado para gerar o vetor das imagens positivas.

3.5.2 Criação de arquivo de vetores de imagens positivas

Após obter o ficheiro 'info.txt' com as anotações das faces das imagens positivas é necessário obter o ficheiro .vec que vai ser utilizado posteriormente, para conseguir treinar o classificador. Este ficheiro é gerado utilizando a função `opencv_createsamples` provindo da biblioteca `opencv`, onde produz um conjunto de dados de amostras positivas, com um formato onde vai ser possível ser treinado com uma função de treino.

Esse vetor foi então gerado com os seguintes parâmetros da aplicação:

```
createsamples.exe -info positive/info.txt -vec vector/vector.vec -num 1000 -w 24 -h 24
```

1. *info.txt* - Path para o ficheiro gerado 'info.txt' com as anotações das faces das imagens positivas

2. *vector.vec* - Path para o ficheiro .vec gerado
3. *1000* - Número de imagens positivas
4. *24* - Largura
5. *24* - Altura

Após obter o ficheiro dos vetores da delineação de todas as imagens positivas é possível avançar para o seguinte passo, que é o treino do classificador.

3.5.3 Treino do classificador

Após existir o vetor das imagens é possível então treinar o classificador. para isso é necessário recorrer novamente à biblioteca opencv que fornece a aplicação `opencv_traincascade`.

O classificador gerado, provém então dos seguintes parâmetros.

```
haartraining.exe -data cascades -vec vector/vector.vec -bg negative/bg.txt -npos  
1000 -nneg 1661 -nstages 15 -mode ALL -w 24 -h 24
```

1. *cascades* - Path para o ficheiro gerado
2. *vector.vec* - Path para o ficheiro .vec gerado anteriormente
3. *bg.txt* - Ficheiro com informações das imagens negativas
4. *npos* - Número de imagens positivas
5. *nneg* - Número de imagens negativas
6. *nstages* - Número de estágios do treino do classificador
7. *ALL* - Seleciona o tipo de conjunto de recursos Haar.

Resultados e discussão

Para testar alguns algoritmos de deteção facial, nomeadamente Haar Cascades, HOG e CNN foi desenvolvido um script python onde é possível de uma forma rápida e eficaz conseguir obter alguns resultados onde é possível comparar a partir de que ângulos de rotação da face a ser analisada é possível ser detetada.

Para conseguir obter uma deteção da face é necessário haver o input de uma imagem com uma face e posteriormente é nos dado um output com essa mesma deteção. Para isso foram utilizados dois datasets para serem então testados esses algoritmos.

4.1 DATASETS

4.1.1 Dataset criado na UA

Inicialmente, foi adquirido um novo dataset, realizado na Universidade de Aveiro, onde é possível melhorar a análise dos testes e conseguir comparar entre diversos datasets. Este dataset, foi realizado com o auxilio de 3 câmaras distintas, que podemos observar na Figura 4.1, onde cada uma delas tem uma função diferente. O principal objetivo era conseguir obter imagens com a rotação da face para diferentes ângulos, onde foi conseguido, olhando para diferentes pontos de contacto expostos na parede anteriormente à aquisição das fotografias.



Figura 4.1: Setup para aquisição das fotografias

No final, este dataset contou com um total de 19 pessoas distintas, onde duas dessas pessoas as fotografias foram adquiridas com e sem o uso de óculos para conseguir ter ainda

mais métodos de análise e de comparação. Foram adquiridas 11 poses diferentes, com ângulos entre os -90° e os 90° .

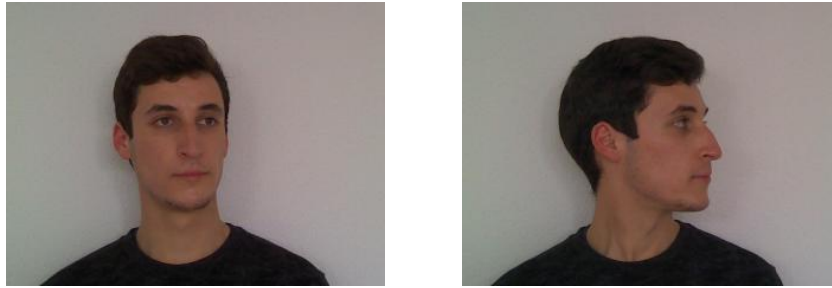


Figura 4.2: Ângulo de rotação de 20° e 90° em relação à posição da câmara

Para obter as imagens do dataset foram utilizadas as seguintes câmaras disponibilizadas pela Academia Militar (AM):

1. FLIR T440BX
2. QUEST Condor3 VNN-618
3. NIT WiDy C 640V-S

As especificações das mesmas estão presentes na Tabela 4.1.

Tabela 4.1: Especificações dos equipamentos de imagem pertencentes à Academia Militar

Equipamento	Resolução pixels	Banda espectral nm
FLIR T440BX (VIS e LWIR)	320 x 240	(450-520), (515-600), (600-690); (7 500-13 000)
QUEST Condor3 VNN-618	640 x 494	(400-670); (670-850); (850-1000)
NIT WiDy C 640V-S (SWIR)	640 x 512	(900-1 700)

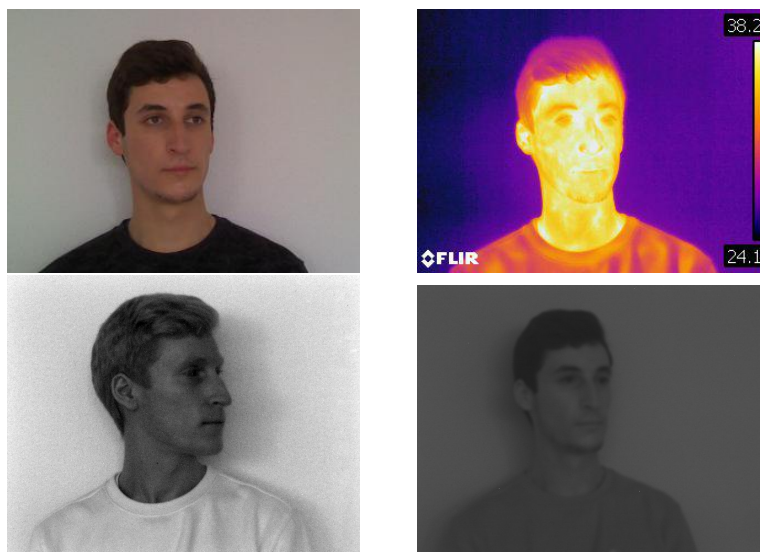


Figura 4.3: Exemplo de imagens adquiridas no dataset adquirido. (a) Visível; (b) Térmicas; (c) SWIR; (d) NIR)

. Podemos observar na Figura 4.3 todos os tipos de imagens adquiridas neste dataset.

4.1.2 Dataset online

Posteriormente foi utilizado um dataset disponível online, [31], onde era possível observar várias imagens de diversas pessoas. Este dataset continha um total de 6660 imagens de 90 pessoas distintas. Cada indivíduo possuía 74 imagens, onde eram capturadas de 5° em 5° , desde -90 até o oposto $+90^\circ$. Destas 90 pessoas, foram utilizadas apenas 21 para o teste de detecção facial dos vários algoritmos. Utilizaram-se também ângulos específicos de acordo ao que foram adquiridos no dataset criado na UA.

4.2 ANÁLISE MÉTODOS DETEÇÃO FACIAL

De forma a conseguir comparar e avaliar a metodologia proposta foram realizadas vários testes. Neste capítulo vão ser discutidos vários testes acerca da detecção facial.

4.2.1 Ângulos de rotação

Foram testados os algoritmos de detecção, para respetivos ângulos da face, nas imagens adquiridas no campo do visível. Podemos observar na Figura 4.4 os resultados da detecção desses mesmos 3 métodos para esse tipo de imagens.

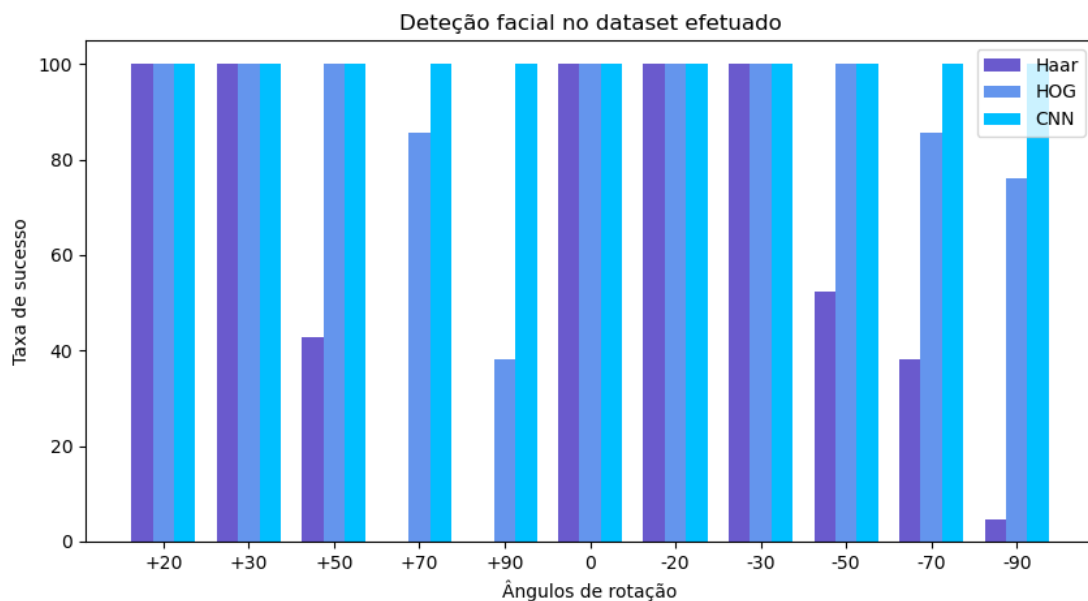


Figura 4.4: Taxa de sucesso em função do ângulo de rotação para imagens do dataset efetuado.

Conseguimos concluir por análise do gráfico de barras, que o método Haar consegue detetar faces frontais 100% das vezes, com ângulos compreendidos entre os -30° e os 30° . Podemos também observar na Figura 4.4 que os resultados para faces com mais de 50° de rotação e de -50° a -90° , apresentam resultados bastantes negativos, mesmo chegando a ter 0% de taxa de sucesso.

Pela tabela 4.2, das 221 imagens da totalidade do dataset, foram detetadas 127 faces, havendo ainda 3 falsos positivos, devido a deteções duplas, e deteções em espaços que não

determinam uma face, ocorrendo em imagens compreendidas entre os 70+ e os 90°. Este efeito deve-se ao facto que este tipo de deteção, não tem os melhores resultados nessa variação de ângulos. É possível observar na Figura 4.5 os tipos de falso positivos testados.

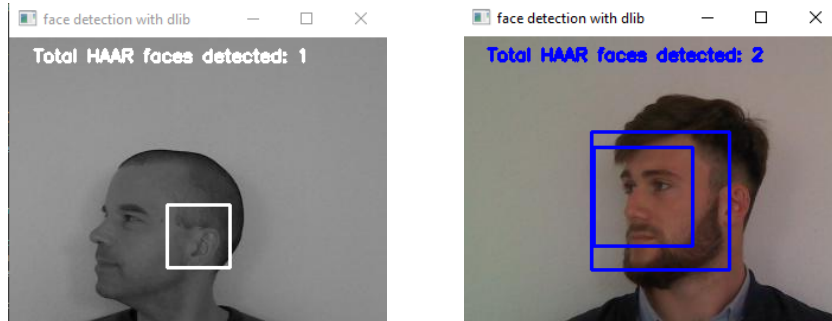


Figura 4.5: Exemplo de falsos positivos testados

Existem também diversas métricas que nos ajudam a identificar se o modelo está corretamente treinado ou não. Para isso, são apresentadas na Tabela 4.2 as métricas precisão e recall. Na métrica precisão que indica, das classificações positivas do modelo, quantas foram acertadas, no modelo Haar conseguimos ver um valor na ordem dos 97.64%, daí, a existência de poucos falso positivos.

Quanto à recall é uma métrica que indica, das amostras positivas existentes, quantas é que foram classificadas corretamente. Como o método haar não abrange uma grande variação de ângulos, conseguiu detetar por volta de metade das faces do dataset utilizado. Em relação ao algoritmo HOG, podemos observar que a taxa de sucesso é bastante melhor que o algoritmo Haar. Apresenta melhores resultados de deteção, tendo quase falhas, apenas quando a face está com a maior rotação, de 90°. Este método não detetou falsos positivos, o que leva a uma precisão de 100%. Detetou também apenas 199 das 221 faces do dataset, conseguindo um recall de 90.04%, não atingindo os 100% devido a 22 faces que não foram detetadas. Em relação ao último algoritmo CNN, conseguimos observar que possui uma taxa de deteção de 100% para qualquer ângulo da face, ou seja, consegue obter uma precisão e um recall máximo.

Tabela 4.2: Resultados dos testes para os 3 métodos de deteção a partir do dataset efetuado.

Algoritmo de deteção	Número de imagens	Número de deteções	Falsos Positivos	Verdadeiro Positivos	Precisão (%)	Recall (%)
Haar	221	127	3	124	97,64	54,7
HOG	221	199	0	199	100	90,04
CNN	221	221	0	221	100	100

. Todos estes valores, claro que podem ter uma variação, tanto por alteração da iluminação, com outros alguns fatores. Após a análise neste dataset, e como forma a conseguir comparar através de vários, para conseguir obter dados e resultados mais precisos, foi feita a mesma análise, porém desta vez a partir do dataset online. É de esperar que os resultados sejam bastante semelhantes observando na Figura 4.6.

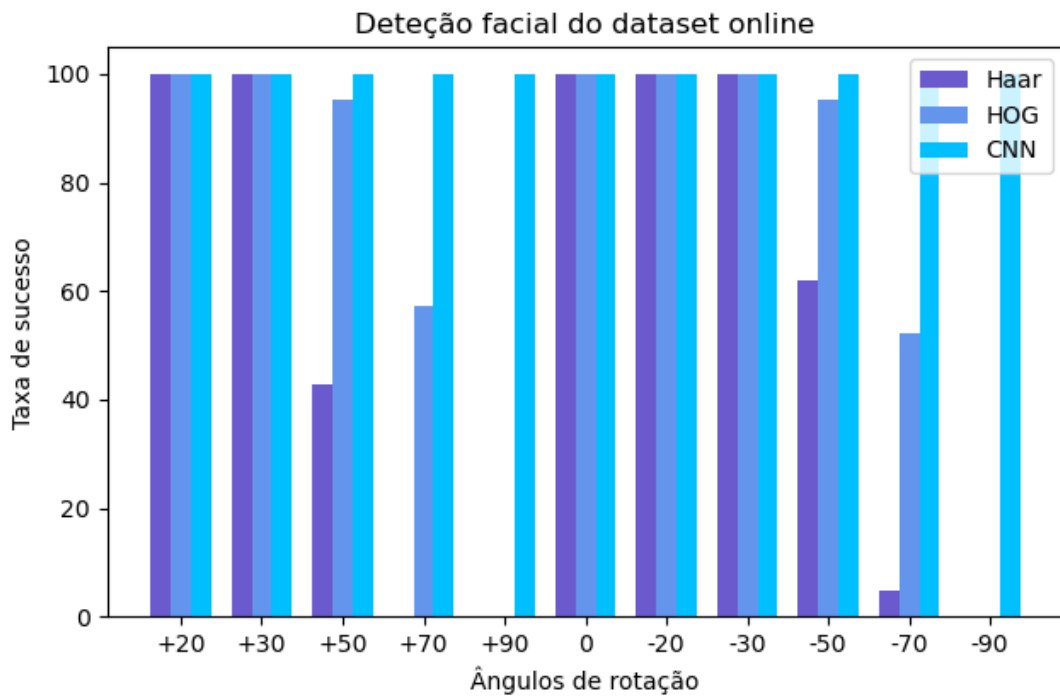


Figura 4.6: Taxa de sucesso em função do ângulo de rotação para imagens do dataset online.

Para conseguir ser de mais fácil análise e observação, realizou-se uma subtração dos valores referentes aos dois gráficos de barras. A partir da Figura 4.7 conseguimos observar melhor a diferença de valores entre a detecção dos vários datasets. Apresenta uma carga de informação menor aos outros gráficos, pois os valores coincidem e não existe tanta discrepância de resultados. Ainda que mínima, esta diferença que ocorre sempre para valores de ângulos superiores a 50° e inferiores a -50° pode dever-se ao facto das aquisições das fotografias, como posicionamento das câmaras e o ambiente em questão.

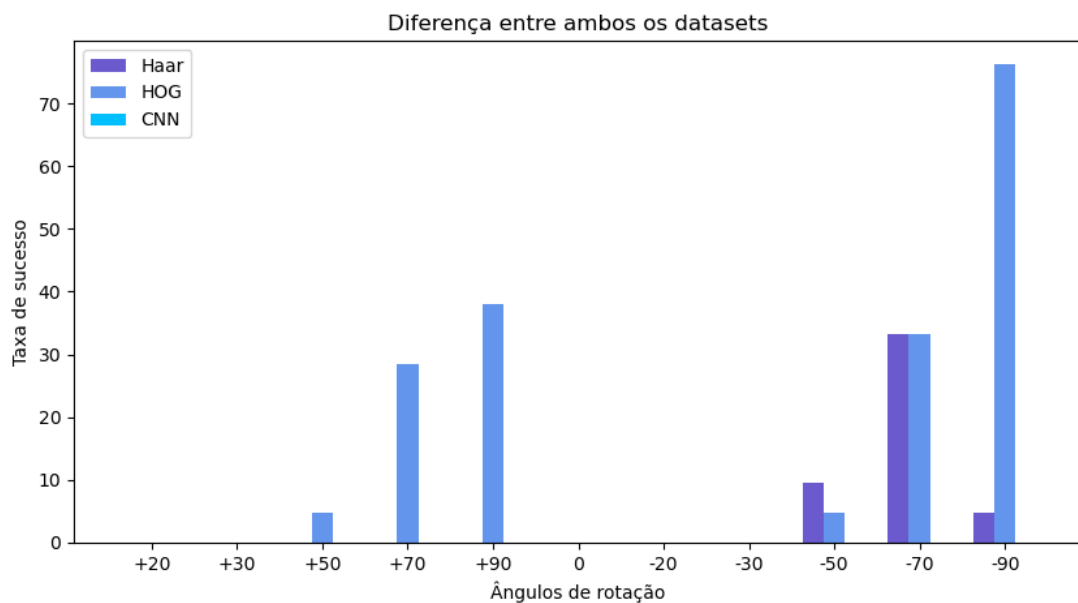


Figura 4.7: Diferença da taxa de sucesso relativamente a ambos os datasets.

Tabela 4.3: Resultados dos testes para os 3 métodos de detecção a partir do dataset online.

Algoritmo de detecção	Número de imagens	Número de detecções	Falsos Positivos	Verdadeiro Positivos	Precisão (%)	Recall (%)
Haar	221	133	3	130	97,74	58,82
HOG	221	175	0	175	100	79,19
CNN	221	221	0	221	100	100

Existe uma ligeira subida no número de detecções, observando pela Tabela 4.3, e no método Haar continuou a existência de falsos positivos .

Fatores como luminosidade, câmaras e mesmo a inclinação da cabeça podem levar a desvios quanto aos valores de detecção, o que faz com que os valores da taxa de sucesso não sejam 100% iguais, havendo sempre alguma discrepância de resultados.

4.2.1.1 Detecção em imagens térmicas

De forma a enriquecer, e obter mais métodos de comparação, após a análise dos algoritmos a imagens do espectro de luz visível, foi feita a comparação com esses mesmos algoritmos, a imagens do dataset efetuado com imagens térmicas. Pensando que por ventura, estes algoritmos estejam preparados para detetar faces em qualquer espectro, conseguimos identificar que somente funcionam bem em imagens do espectro visível. Apesar de por vezes serem identificadas faces nas imagens térmicas, estes algoritmos não estão treinados para estes tipos de imagens, o que vai levar a uma taxa de sucesso bastante baixa.

Pela Figura 4.8 conseguimos observar que apesar de existir pelo menos uma detecção facial para cada um dos ângulos de rotação da face, o máximo de taxa de sucesso dessas detecções não ultrapassa o valor de 57.14%.

Observa-se também, que ao contrário da análise das imagens do espectro visível, o método Haar, tem o maior número de detecções de faces em relação ao restantes métodos.

Tabela 4.4: Resultados dos testes para os 3 métodos de detecção a partir do dataset efetuado em imagens térmicas.

Algoritmo de detecção	Número de imagens	Número de detecções	Falsos Positivos	Verdadeiro Positivos	Precisão (%)	Recall (%)
Haar	221	58	1	57	98,28	26,24
HOG	221	38	0	38	100	17,19
CNN	221	48	0	48	100	21,72

Vê-se diretamente pela tabela 4.4 os valores da métrica recall bastantes baixos, o que indica logo uma baixa taxa de sucesso.

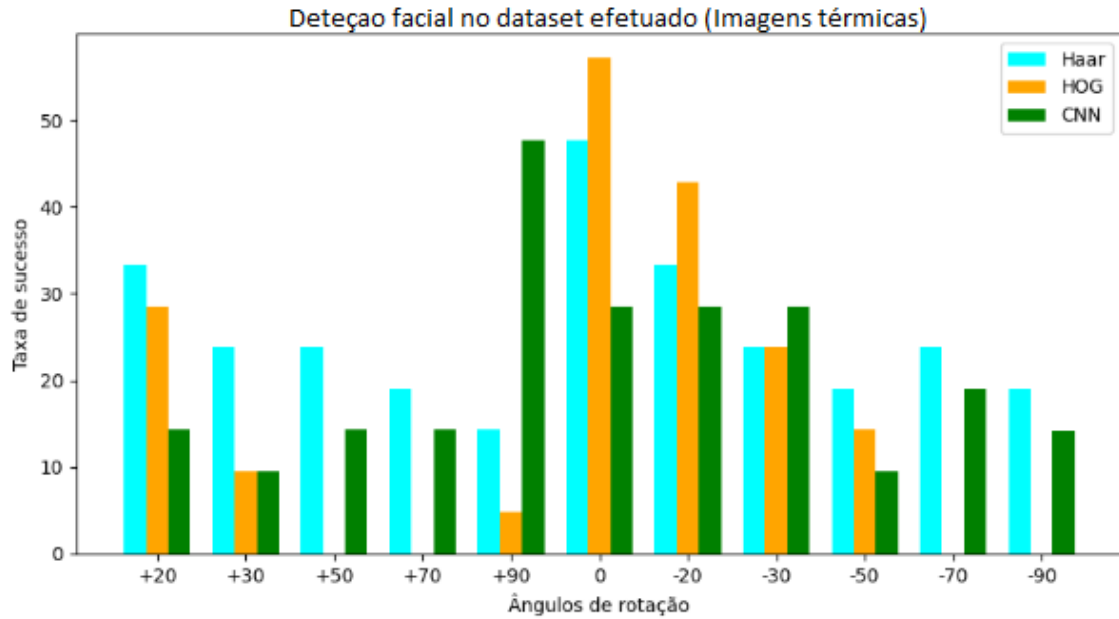


Figura 4.8: Taxa de sucesso em função do ângulo de rotação para imagens térmicas do dataset efetuado.

4.2.2 Tempo de processamento

Conseguimos concluir, que o método Haar apesar de ter uma performance em relação ao tempo, bastante superior aos outros dois métodos como se pode observar na tabela 4.5, e conseguir detetar faces frontais 100% das vezes, com ângulos compreendidos entre os -30° a 0° e os 30° a 0 , o mesmo não acontece com faces com maior ângulo de rotação. Podemos também observar na Figura 4.4 que os resultados para faces com mais de 50° de rotação e de -50° a -90° , apresenta resultados bastantes negativos, mesmo chegando a ter 0% de taxa de sucesso.

Quanto ao algoritmo HOG, a partir da análise da Figura 4.4, podemos observar que a taxa de sucesso é bastante melhor que o algoritmo Haar. Apesar de ser mais custoso em relação ao tempo, apresenta melhores resultados de deteção, tendo quase falhas, apenas quando a face está com a maior rotação, de 90° . Em relação ao último algoritmo CNN, conseguimos observar que possui uma taxa de deteção de 100% para qualquer ângulo da face. O tempo de processamento deste último algoritmo é bastante superior, cerca de 165x mais demoroso que o algoritmo Hog, porém tem uma taxa de acerto de 100%.

Tabela 4.5: Tempo de processamento relativo ao algoritmos de deteção facial.

Algoritmo	Tempo de processamento (segundos)
Haar	1,02
Hog	19,76
CNN	3142,79

4.3 ANÁLISE DE MÉTODOS DE RECONHECIMENTO

De forma a conseguir comparar e avaliar a metodologia proposta foram realizadas vários testes, de modo a consolidar e entender melhor os diversos tipos de reconhecimento facial. Neste presente capítulo vão ser discutidos vários testes realizados no âmbito da dissertação de mestrado.

4.3.1 Verdadeiros Positivos, confiança e tempo de processamento

Inicialmente, antes de serem apresentados resultados é necessário apresentar os parâmetros testados e utilizados por os vários algoritmos testados. Relativamente ao algoritmo Eigenfaces temos os seguintes parâmetros [32]:

- **int num_components = 0** : O número de componentes representa o valor em número, que é mantido para a Análise de Componentes Principais (PCA). O valor de default para esta variável é 0, o que significa de todos os componentes serão utilizados para a PCA. O número recomendado pela documentação do OpenCV é de 80.
- **double threshold = DBL_MAX** : O valor de threshold refere-se ao limiar de como é classificada a previsão da pessoa a ser reconhecida. O valor de default é DBL_MAX, o que significa que todos os resultados vão ser retornados, já que se apresentam todos abaixo do limiar máximo.

Em relação ao algoritmo Fisherfaces os parâmetros são os seguintes [33]:

- **int num_components = 0** :
O número de componentes representa o valor em número, que é mantido para a Análise Discriminante Linear (LDA). O valor de default para esta variável é 0, ou seja vai utilizar o número correto de classes (pessoas a reconhecer).
- **double threshold = DBL_MAX** : O valor deste parâmetro é novamente DBL_MAX e representa o threshold aplicado na previsão do reconhecimento. Caso a confiança resultante for menor que o valor definido é retornado -1, o que significa que a face não foi reconhecida.

Neste capítulo, para conseguir realizar o reconhecimento facial, foi o utilizado o dataset adquirido, com imagens no espetro do visível, e conseguiu-se resultados, relativamente aos algoritmos de reconhecimento EigenFaces, FisherFaces e OpenFace. Existiu ainda um processo de pré-processamento das imagens, na tentativa de uma melhoria dos algoritmos, em relação ao tempo de processamento, e aos verdadeiros positivos, do reconhecimento das faces. Foram realizados testes para cada um dos algoritmos, antes e após a frontalização da face, na tentativa de comparar o número de verdadeiros positivos, falsos positivos, percentagem de acerto, confiança e o tempo de processamento.

Na tentativa de obter um valor geral do comportamento do algoritmo, foi dividido o dataset em uma percentagem mais reduzida para imagens de teste e o restante para imagens de treino. Foram utilizados então dois datasets de treino, um com 200 imagens e outro com apenas 80, mantendo-se sempre as mesmas 16 imagens de teste. Relativamente à divisão do

dataset, foi efetuada aleatoriamente. Nestes teste, o número de Verdadeiros Positivos, refere-se à previsão correta da pessoa a ser reconhecida, enquanto que o falsos positivos refere-se ao não reconhecimento dessa pessoa. O acerto funciona em relação ao todo das imagens de teste e de treino, consoante a precisão do algoritmo.

Todos os testes efetuados, foram realizados como o método de detecção Histograma de Gradientes. Devido ao método Haar Cascades possuir uma taxa bastante inferior de detecção e ao método CNN possuir um tempo de processamento bastante elevado, foi escolhida este métodos para a determinada detecção e posteriormente o reconhecimento. Na tabela 4.6 é possível observar alguns resultados relativamente ao algoritmo EigenFaces sem a frontalização da face efetuada. Para este teste foram utilizados os parâmetros do método de reconhecimento eigenfaces (80, 6000), onde o número de verdadeiros positivos foi o mais elevado. À medida que era aumentado o valor de threshold o número de falsos positivos ia também aumentando. A taxa de acerto, ou seja o número de faces reconhecidas foi 50% para o dataset com 200 imagens de treino, e 43.75% para o outro dataset, com as 80 imagens.

Tabela 4.6: EigenFaces sem frontalização

Teste	Treino	Verdadeiros Positivos	Falsos Positivos	Acerto
16	200	8	8	50%
16	80	7	9	43,75%

Tabela 4.7: FisherFaces sem frontalização

Teste	Treino	Verdadeiros Positivos	Falsos Positivos	Acerto
16	200	11	5	68,75%
16	80	8	8	50%

Relativamente à tabela 4.7, onde foi utilizado o método fisherfaces é possível observar uma taxa de acerto de quase 70% para o dataset de treino de 200 imagens, e 50% para o dataset de 80.

De modo a aprofundar e obter mais resultados de comparação, foram realizados mais alguns testes. Para estes resultados, utilizou-se as 221 imagens para treino dos métodos de reconhecimento, e para teste, várias imagens de faces em determinados ângulos desse mesmo dataset. Cada pessoa no dataset está caracterizada por um número (utilizadas letras na tabelas seguintes para melhor compreensão, Tabelas 4.8–4.9), e o nome dessa respetiva pessoa, e todas as imagens vão sofrer o mesmo tipo de alterações para testes destes métodos de reconhecimento. Para este teste em específico, foram utilizadas duas imagens de 6 pessoas do dataset, nomeadamente com ângulos de rotação de $+20^\circ$ e -30° . Cada uma dessas imagens de teste, vai ser comparada com as imagens da base de treino, onde será possível observar, após a comparação, se essa imagem foi corretamente reconhecida pela pessoa em questão, qual o nível de confiança desse reconhecimento, e por fim, o tempo de processamento que levou desde o início da detecção, até ao final do reconhecimento.

Nesta experiência é possível observar, não só os verdadeiros positivos e os falso positivos, como também o nível de confiança e o tempo de processamento. Ambos os algoritmos,

Tabela 4.8: Resultados do reconhecimento utilizando o método EigenFaces

ID da Face	Classificada como	Confiança	Tempo de Processamento(s)
A (+20°)	A	4790	26,18
A (-30°)	S	7395	28,55
E (+20°)	C	6221	25,85
E (-30°)	E	3561	28,42
H (+20°)	H	4224	25,85
H (-30°)	H	8055	28,89
I (+20°)	L	6268	29,18
I (-30°)	N	8077	28,66
L (+20°)	L	5940	25,82
L (-30°)	L	4306	28,25
R (+20°)	P	7178	25,87
R (-30°)	R	2326	28,43

retornam um valor de confiança que representa a semelhança da imagem de teste em relação à imagem que foi classificada. De referir que, caso a imagem de teste esteja contida na base de treino, a confiança vai possuir um valor de 0, ou seja, quando maior a semelhança da imagem classificada e a de teste, mais próximo do valor nulo se encontra. Relativamente ao método Eigenfaces na Tabela 4.8 é possível observar valores de confiança com uma média dos 12 testes de 5695. Ao definir o threshold de 6000, todos os níveis de confiança abaixo desse limiar, vão obter supostamente um reconhecimento correto. Valores que estejam acima dos 6000, funcionam como uma face não reconhecida da base de dados de treino.

O mesmo acontece na Tabela 4.9, com o método Fisherface, porém os valores de confiança são inferiores aos do método Eigenfaces. Relativamente ao tempo de processamento EigenFaces tem uma média de 27.50s enquanto que FisherFaces possui 23.73s. Após o teste destes dois algoritmos, e a análise das tabelas 4.6–4.9 é possível retirar algumas conclusões acerca desta análise.

Observando os resultados obtidos é possível observar que tanto para EigenFaces e Fisherfaces, o valor de verdadeiros positivos e taxa de acerto aumenta, consoante o número de imagens no dataset.

À medida que o tamanho do dataset de treino seja inferior, o número de falsos positivos vai ter tendência a aumentar, ou seja, menos faces vão ser reconhecidas. O contrário acontece caso o número de imagens do dataset aumente, onde vai existir uma aumento de faces reconhecidas. É demonstrado então um fator bastante importante para o sucesso dos algoritmos e sucessivo reconhecimento. Relativamente ao nível de confiança e ao tempo de processamento, temos que o algoritmo Fisherfaces possui uma confiança e um tempo de processamento inferior ao algoritmo Eigenfaces, daí possível observar que Fisherfaces tem

Tabela 4.9: Resultados do reconhecimento utilizando o método FisherFaces

ID da Face	Classificada como	Confiança	Tempo de Processamento(s)
A (+20°)	B	3134	23,19
A (-30°)	A	2574	24,26
E (+20°)	R	2614	22,40
E (-30°)	E	1876	24,27
H (+20°)	H	2493	22,44
H (-30°)	H	1829	24,33
I (+20°)	A	2533	25,78
I (-30°)	I	2476	24,45
L (+20°)	L	3095	22,39
L (-30°)	L	2014	24,50
R (+20°)	B	2781	22,33
R (-30°)	R	1549	24,44

uma maior certeza que a face classificada é a face que está a ser testada.

4.3.2 Pré-processamento

4.3.2.1 Frontalização da face

Na tentativa de melhorar os valores de reconhecimento relacionado ao dois algoritmos testados anteriormente, foi realizada uma frontalização da face a ser treinada. Nas tabelas 4.10 e 4.11 conseguimos observar que os resultados do reconhecimento não são os mais satisfatórios com taxas inferiores a 38%, chegando mesmo a 0% no algoritmo Fisherfaces. Além disso, pode-se observar mais uma vez, que quanto maior o dataset de imagens de treino, maior sucesso o algoritmo vai apresentar.

Tabela 4.10: Eigenfaces com frontalização

Teste	Treino	Verdadeiros Positivos	Falsos Positivos	Acerto
16	200	6	10	37,5%
16	80	2	14	12,5%

Tabela 4.11: Fisherfaces com frontalização

Teste	Treino	Verdadeiros Positivos	Falsos Positivos	Acerto
16	200	1	0	6,25%
16	80	0	0	0%

Relativamente ao tempo de processamento, nas tabelas 4.12 e 4.13 assemelha-se ao analisado anteriormente, porém acresce determinados segundos relativamente ao tempo de processamento da rotação da face. O nível de confiança, aumentou também na comparação da imagem de teste que foi frontalizada à classificação.

Estes resultados, devem-se ao facto de as imagem de teste sofrerem bastantes deformações aquando a frontalização da face. Apesar de a olho humano, a face ficar mais reconhecível após a frontalização, quando testado numa máquina o mesmo não acontece. Devido à face média nos algoritmos serem calculados através dos pixels das imagens, estes vão variar bastante, pois apesar de ser efetuada a frontalização, existe bastante ruído presente nessas imagens.

Tabela 4.12: Resultados do reconhecimento utilizando o método EigenFaces com frontalização da face

ID da Face	Classificada como	Confiança	Tempo de Processamento(s)
A (+20°)	A	6833	29,07
A (-30°)	A	6255	29,59
E (+20°)	E	7105	29,78
E (-30°)	E	5676	29,35
H (+20°)	Q	8413	29,45
H (-30°)	B	7394	29,03
I (+20°)	D	7177	29,01
I (-30°)	M	6736	29,58
L (+20°)	L	6351	29,65
L (-30°)	L	5001	29,05
R (+20°)	R	7651	29,07
R (-30°)	R	6023	29,22

Tabela 4.13: Resultados do reconhecimento utilizando o método FisherFaces com frontalização da face

ID da Face	Classificada como	Confiança	Tempo de Processamento(s)
A (+20°)	R	2936	25,48
A (-30°)	P	2363	25,43
E (+20°)	C	1926	25,04
E (-30°)	E	1890	25,42
H (+20°)	N	2307	25,13
H (-30°)	Q	2154	25,19
I (+20°)	H	2212	25,44
I (-30°)	F	2601	25,19
L (+20°)	T	2614	25,12
L (-30°)	L	1894	25,27
R (+20°)	U	2420	25,10
R (-30°)	T	1826	25,46

4.3.3 Deepface framework - OpenFace

Avaliando pela tabela 4.14 pode-se observar os resultados dos verdadeiros positivo e concluir que a taxa de acerto é bastante elevada. Relativamente ao tempo de processamento, inicialmente é necessário efetuar o download do modelo treinado do algoritmo. Caso o dataset

de treino permaneça sempre igual, o tempo de processamento após a existência do ficheiro do modelo, iria ser por voltas dos 4 segundos, porém é sempre eliminada pois pode haver a alteração do dataset de treino.

Tabela 4.14: Resultados do reconhecimento geral utilizando Openface

Teste	Treino	Verdadeiros Positivos	Falsos Positivos	Acerto	Tempo de processamento(s)
16	200	14	2	87,5	251,931
16	80	13	3	81,25	107,28

Tabela 4.15: Resultados do reconhecimento utilizando Openface

ID da Face	Classificada como	Confiança
A (+20°)	A	0,0082
	I	0,1553
A (-30°)	A	0,0899
	A	0,0907
E (+20°)	E	0,0406
	E	0,0760
E (-30°)	E	0,0374
	E	0,0756
H (+20°)	H	0,0571
	H	0,0631
H (-30°)	H	0,0297
	H	0,0866
I (+20°)	I	0,1147
	I	0,1234
I (-30°)	I	0,0484
	I	0,0788
L (+20°)	L	0,0508
	L	0,0655
L (-30°)	L	0,0402
	L	0,0588
R (+20°)	R	0,0276
	R	0,1033
R (-30°)	R	0,0502
	R	0,0651

Este método retorna dois valores de confiança, onde classifica por ordem de semelhança de imagem, a qual classificação ela pertence. Os valores de confiança neste método de reconhecimento, aproximam-se praticamente do valor nulo, o que significa que a precisão da

previsão é quase certa. Definindo um valor de threshold de 0.1, maior parte das vezes vão possuir uma classificação correta.

4.4 TREINO DE CLASSIFICADOR PARA DETECÇÃO FACIAL DE IMAGENS TÉRMICAS

Como foi observado no Capítulo 4.2, foi utilizado um dataset adquirido na Universidade de Aveiro onde continha várias imagens de faces de vários espectros. Nesta secção vai ser analisada as imagens térmicas, como é possível observar na Figura 4.9

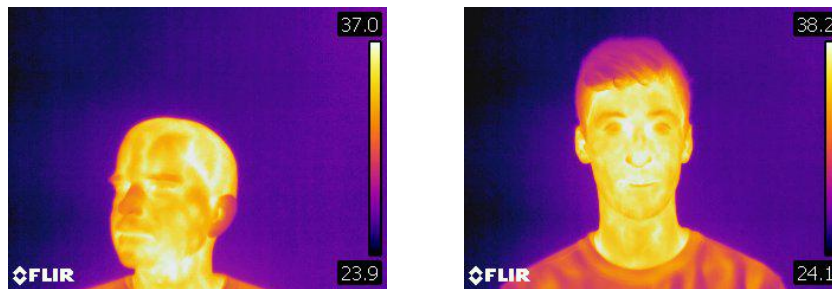


Figura 4.9: Exemplo de imagens térmicas

Foi possível chegar a resultados de deteção facial para imagens térmicas. Para essa deteção, essas imagens serviram de input para o algoritmo Haar Cascade, onde o output é nos dado, ao conseguir obter quantas faces foram localizadas e que graus de rotação essas mesmas pertenciam.

Previamente utilizamos “haar-like cascades features” para identificar as faces nas imagens visíveis. Utilizamos um ficheiro Extensible Markup Language (XML) (classificadores) já treinado, ou seja, já tinham características definidas para determinados objetos de deteção, neste caso, as faces das pessoas. Os resultados ao serem testados em imagens térmicas não foram os previstos, pois esse algoritmo, ao ser treinado com imagens no espectro visível, só vai conseguir obter resultados razoáveis, nesse mesmo espectro.

O dataset utilizado, na tentativa de treinar o classificador para faces em imagens térmicas é SF-TL54 Dataset [34]. Este dataset contém 1000 imagens de pessoas com faces de diferentes ângulos de exposição e rotação de face. Para o dataset de imagens sem faces foi utilizado outro dataset [35]. Este dataset possui cerca de 1700 imagens maioritariamente de carros e de gatos. É possível observar algumas imagens de treino na Figura 4.10.

Era possível utilizar as imagens por completo, porém os resultados não iriam ser os melhores, pois existiam áreas que não pertenciam a uma face.

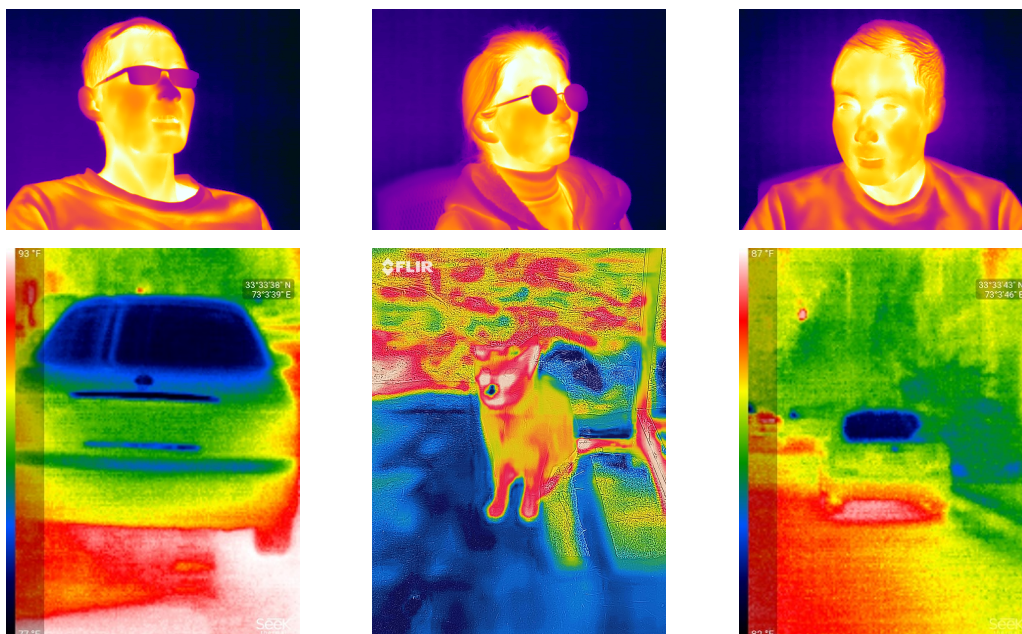


Figura 4.10: Linha superior - imagens com objeto a ser detetado(faces) e Linha inferior - imagens sem objeto a ser detetado (face) [34], [35].

4.4.1 Comparação classificadores

Após obter o classificador treinado é possível então comparar a diferença entre o classificar Haar da biblioteca OpenCV. Relativamente ao primeiro classificador, conseguiu observar-se que os valores da taxa de sucesso de deteção eram bastante baixos, tendo o valor máximo de taxa de sucesso na ordem dos 57%.

Relativamente ao classificador treinado, conseguimos observar pela Figura 4.11 que o máximo valor de taxa de sucesso ronda os 85%, o que incrementa bastante e melhora o algoritmo e a sua deteção.

Os ângulos de rotação onde este algoritmo deteta a face, continua a ser semelhante ao antigo, porém em números de deteção apresenta bastante melhores resultados. Das 221 imagens de teste utilizadas, que foram adquiridas na Universidade de Aveiro, 131 das faces foram detetadas. Porém, existiu um número de falsos positivos de 67, sendo possível observar na Tabela 4.16. Relativamente a valores de precisão, com este classificador, a percentagem vai ser menor, pois existem mais falsos positivos, porém quanto ao recall vai ser alterada de valores de cerca de 26% para 90%, pois existe um maior número de deteções faciais.

Tabela 4.16: Resultados dos testes para o método treinado.

Algoritmo de deteção	Número de imagens	Número de deteções	Falsos Positivos	Verdadeiros Positivos	Precisão (%)	Recall (%)
Haar	221	198	67	131	66.16	90.0

Grande maioria dos falsos positivos, a deteção pertence à face da pessoa, porém não à face completa. Por observação das imagens da Figura 4.12, na imagem da esquerda o quadrado de

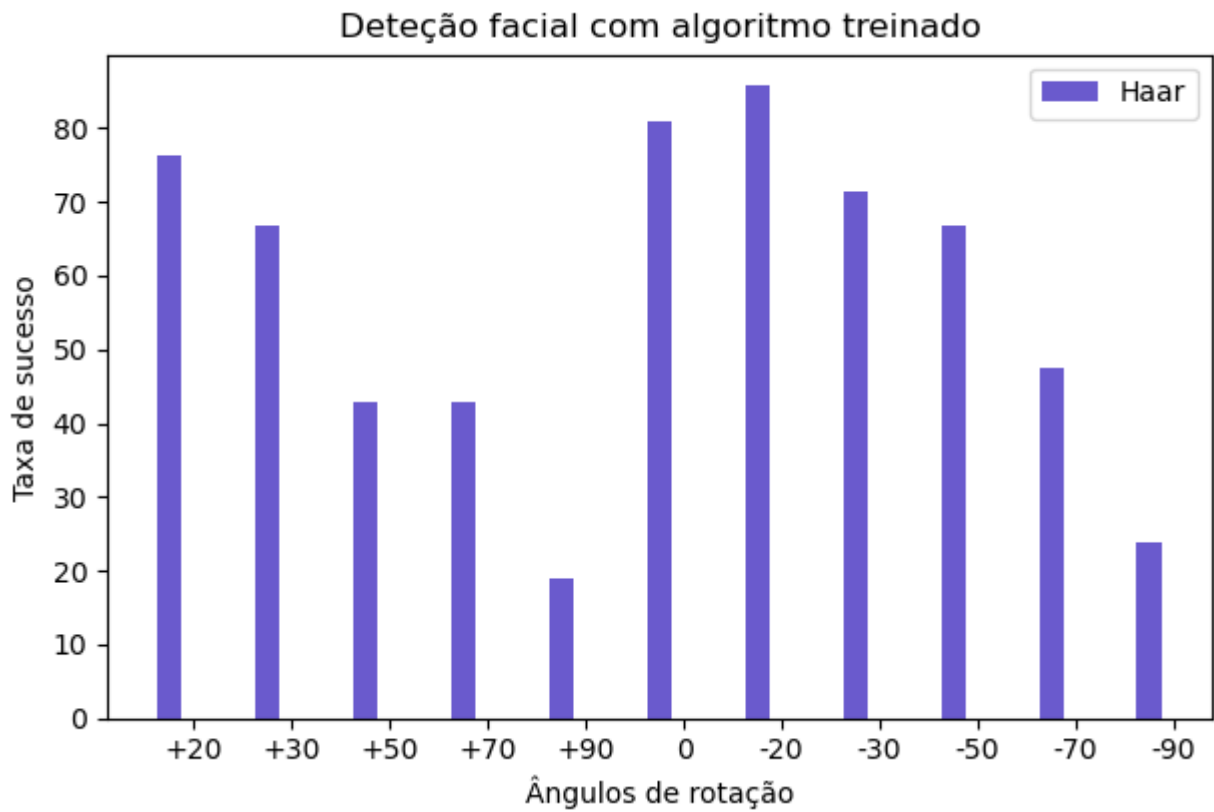


Figura 4.11: Taxa de sucesso em função do ângulo de rotação.

deteção está ligeiramente posicionado à direita do olho. Relativamente à imagem da direita ambos os quadrados, poderiam relacionar-se com a face, porém existe unicamente uma pessoa, onde deveria ser detetada somente uma face.

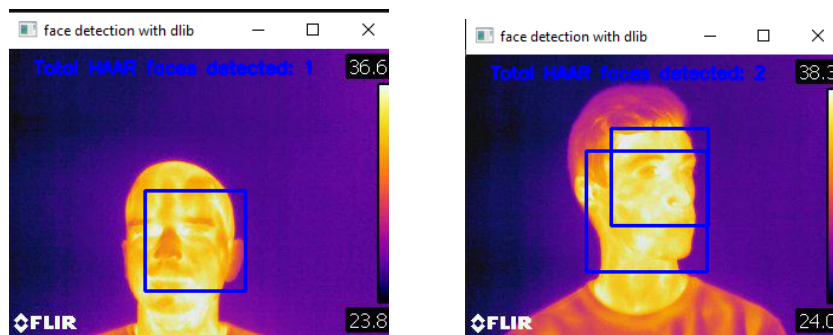


Figura 4.12: Exemplo de falsos positivos testados

Conclusão

A realização deste trabalho permitiu adquirir vários conhecimentos acerca do pipeline necessário para realizar um sistema de reconhecimento facial. Foi estudado e demonstrado que são necessárias várias etapas para conseguir chegar a uma classificação a partir do reconhecimento. Módulos como aquisição de imagens, métodos de pré-processamento, extrações das principais características das faces das pessoas e classificação são as etapas necessárias. Foi possível adquirir um dataset de imagens realizado na Universidade de Aveiro. Imagens em diferentes espectros de cores, com variações na rotação da face foram adquiridas e guardadas, caso haja necessidade para uso posterior em trabalhos de investigação. Relativamente aos resultados acerca da deteção facial, observou-se que nem todos os algoritmos têm o mesmo poder de deteção. Fatores como luminosidade, qualidade da câmara quando a aquisição das imagens e poses alteram o desempenho geral de cada um dos algoritmos. Relativamente aos algoritmos de deteção facial, apesar da rede neuronal Dlib CNN ter apresentado valores de 100% de deteção, o seu tempo de processamento era excessivamente elevado. O algoritmo HOG, apesar de possuir uma precisão ligeiramente menor, conseguiu ter tempos de processamento bastante inferiores.

Em relação aos métodos de reconhecimento, foi possível observar que métodos tradicionais da literatura como EigenFaces e Fisherfaces, apesar não possuírem taxas demasiado baixas, nos dias de hoje não são os melhores para um sistema de reconhecimento devido ao elevado número de falsos positivos. Realizar mudanças nas imagens, aplicando processos de pré-processamento como frontalização, dificulta ainda mais a classificação destes algoritmos. Alterações bruscas nos pixels da imagem, dificultam a extração das características faciais e existe um decréscimo na classificação. Relativamente ao número de imagens de treino, este é um grande fator quando se fala em reconhecimento facial. Quanto maior o número de imagens de treino no dataset, melhores resultados são adquiridos. Métodos baseados em Deep Learning como o Openface, obtêm resultados muito mais eficientes. São treinados por um grande conjunto de dados e as arquiteturas de rede neuronal não precisam de extração manual de recursos porque aprendem recursos diretamente dos dados. Aumentando a quantidade de dados no dataset

para extrair os recursos também é uma maneira de melhorar os resultados.

Atualmente, os sistemas de detecção facial multi-espectral continuam a ser bastante complexos, devido a diferentes fatores nesse tipo de imagem, torna-se difícil a detecção e a extração de características neste tipo de imagens. Através da literatura e após o teste de um classificador treinado a partir de imagens do espectro visível, foi dado um passo no sentido de desenvolver um modelo de detecção facial em imagens térmicas. Através de anotações nesse tipo de imagens, conseguiu-se uma taxa de detecção de cerca de 30% superior ao método já existente.

Como recomendações para trabalho futuro, é possível mencionar as seguintes opções: apesar de o dataset adquirido ser já uma grande valia para investigação na área científica é possível observar que um dos fatores mais importantes em métodos de reconhecimento facial é o tamanho do dataset. Estender o dataset com a aquisição de imagens de mais pessoas, com rotações de face menores (5° em 5°) e em diferentes ambientes e luminosidade iria ser possível obter maiores taxas de precisão. Também era possível uma investigação acerca de mais algoritmos deep learning de reconhecimento, de modo a conseguir comparar o desempenho entre eles. Visto que esta dissertação está relacionada com a Academia Militar, seria interessante aplicar os algoritmos usados, porém em casos de oclusão da face.

Referências

- [1] A. Kumar, A. Kaur e M. Kumar, «Face detection techniques: a review,» *Artificial Intelligence Review*, vol. 52, n.º 2, pp. 927–948, 2019.
- [2] Y. Chen, L. Song, Y. Hu e R. He, «Adversarial occlusion-aware face detection,» em *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, IEEE, 2018, pp. 1–9.
- [3] Y. Zhou, D. Liu e T. Huang, «Survey of face detection on low-quality images,» em *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, IEEE, 2018, pp. 769–773.
- [4] C. Rahmad, R. Asmara, D. Putra, I. Dharma, H. Darmono e I. Muhiqqin, «Comparison of Viola-Jones Haar Cascade classifier and histogram of oriented gradients (HOG) for face detection,» em *IOP conference series: materials science and engineering*, IOP Publishing, vol. 732, 2020, p. 012038.
- [5] P. Viola e M. Jones, «Rapid object detection using a boosted cascade of simple features,» em *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990517.
- [6] A. Srivastava, S. Mane, A. Shah, N. Shrivastava e B. Thakare, «A survey of face detection algorithms,» em *2017 International Conference on Inventive Systems and Control (ICISC)*, IEEE, 2017, pp. 1–4.
- [7] Y.-Q. Wang, «An analysis of the Viola-Jones face detection algorithm,» *Image Processing On Line*, vol. 4, pp. 128–148, 2014.
- [8] T. Soyata e N. Powers, «Face Recognition: A Tutorial on Computational Aspects,» em jan. de 2016, pp. 405–425, ISBN: 978-1-4666-8853-7. DOI: 10.4018/978-1-4666-8853-7.ch020.
- [9] L. Deng e Y. Zhou, «The Vehicle collision warning system detects the vehicle ahead,» *Journal of Physics: Conference Series*, vol. 1303, p. 012081, ago. de 2019. DOI: 10.1088/1742-6596/1303/1/012081.
- [10] V. HARIRAMANI, *Face Identification using Haar cascade classifier*, <https://medium.com/geeky-bawa/face-identification-using-haar-cascade-classifier-af3468a44814>, 2020.
- [11] A. Priadana e M. Habibi, «Face detection using haar cascades to filter selfie face image on instagram,» em *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIIT)*, IEEE, 2019, pp. 6–9.
- [12] T. Surasak, I. Takahiro, C.-h. Cheng, C.-e. Wang e P.-y. Sheng, «Histogram of oriented gradients for human detection in video,» em *2018 5th International conference on business and industrial research (ICBIR)*, IEEE, 2018, pp. 172–176.
- [13] R. N. Taha Anwar, *Effective Face Detection Algorithms in Python*, <https://bleedai.com/5-easy-effective-face-detection-algorithms-in-python/>, 2021.
- [14] 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)*, IEEE, 2018, pp. 471–478.
- [15] Y. Kortli, M. Jridi, A. Al Falou e M. Atri, «Face recognition systems: A survey,» *Sensors*, vol. 20, n.º 2, p. 342, 2020.
- [16] A. Microsoft, *O que é o reconhecimento facial?* <https://azure.microsoft.com/pt-pt/overview/what-is-face-recognition/#features>, 2022.

- [17] F. Jalled, «Face recognition machine vision system using Eigenfaces,» *arXiv preprint arXiv:1705.02782*, 2017.
- [18] C. Li, *Preprocessing Methods and Pipelines of Data Mining: An Overview*, jun. de 2019.
- [19] A. L. Machidon, O. M. Machidon e P. L. Ogrutan, «Face recognition using Eigenfaces, geometrical PCA approximation and neural networks,» em *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, IEEE, 2019, pp. 80–83.
- [20] A. Bhat, R. K. Jha e V. Kedia, «Robust Face Detection and Recognition using Image Processing and OpenCV,» em *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, 2022, pp. 1273–1278.
- [21] A. Tam, *Face Recognition using Principal Component Analysis*, <https://machinelearningmastery.com/face-recognition-using-principal-component-analysis/>, 2021.
- [22] R. S. Salunke e K. Pawar, «Face Recognition using Fisherfaces,»
- [23] N. M. C. Reddy e K. Kumar, «Comparison of HOG and Fisherfaces Based Face Recognition System Using MATLAB,» em *2021 2nd International Conference for Emerging Technology (INCET)*, IEEE, 2021, pp. 1–5.
- [24] S. I. Serengil e A. Ozpinar, «LightFace: A Hybrid Deep Face Recognition Framework,» em *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, 2020, pp. 23–27. DOI: 10.1109/ASYU50717.2020.9259802. URL: <https://doi.org/10.1109/ASYU50717.2020.9259802>.
- [25] S. I. Serengil e Ozpinar, «HyperExtended LightFace: A Facial Attribute Analysis Framework,» em *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, 2021, pp. 1–4. DOI: 10.1109/ICEET53442.2021.9659697.
- [26] B. Amos, B. Ludwiczuk, M. Satyanarayanan et al., «Openface: A general-purpose face recognition library with mobile applications,» *CMU School of Computer Science*, vol. 6, n.º 2, p. 20, 2016.
- [27] G. B. Garcia, O. D. Suarez, J. L. E. Aranda, J. S. Tercero, I. S. Gracia e N. V. Enano, *Learning image processing with OpenCV*. Packt Publishing Birmingham, 2015.
- [28] M. Palaniappan, K. Sowmia e S. Aravindkumar, «Real Time Fatigue Detection Using Shape Predictor 68 Face Landmarks Algorithm,» em *2022 International Conference on Innovative Trends in Information Technology (ICITIT)*, IEEE, 2022, pp. 1–5.
- [29] T. Hassner, S. Harel, E. Paz e R. Enbar, «Effective face frontalization in unconstrained images,» em *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4295–4304.
- [30] A. M. A. Hossen, R. A. A. Ogla e M. M. Ali, «Face detection by using OpenCV’s Viola-Jones algorithm based on coding eyes,» *Iraqi Journal of Science*, vol. 58, n.º 2A, pp. 735–745, 2017.
- [31] R. Lab, *Databases for Face Detection and Pose Estimation*, https://drive.google.com/drive/folders/1XLehM5DYqLqiAste0_h1PYZnavcCN0cR.
- [32] *EigenFaceRecognizer*, https://docs.opencv.org/3.4/dd/d7c/classcv_1_1face_1_1EigenFaceRecognizer.html.
- [33] *FisherFaceRecognizer*, https://docs.opencv.org/4.x/d2/de9/classcv_1_1face_1_1FisherFaceRecognizer.html#a0072e4c3f410250baf4b083296a41dfc.
- [34] A. Kuzdeuov, *SF-TL54 Dataset*, https://drive.google.com/drive/folders/1XLehM5DYqLqiAste0_h1PYZnavcCN0cR, 2021.
- [35] R. Z. Qirat Ashfaq Usman Akram, *Thermal Image dataset for object classification*, <https://data.mendeley.com/datasets/btmrycjbj/1>, 2021.

Autorização para utilização de imagens

AUTORIZAÇÃO PARA UTILIZAÇÃO DE IMAGENS

Eu, _____, declaro, para os devidos efeitos legais, que autorizo a utilização das fotografias captadas recolhidas no âmbito do estudo de mestrado do estudante Rúben Menino e na presença dos seus orientadores.

Mais declaro expressamente, que as referidas imagens e fotografias poderão ser utilizadas em qualquer meio de comunicação para fins didáticos, de pesquisa e divulgação de conhecimento científico, sendo vetada qualquer exposição publicitária ou de promoção das instituições ou membros envolvidos sem o meu prévio consentimento. Para caso de aplicação científica renuncio, desde já, a quaisquer direitos ou compensação de imagem que esta utilização possa eventualmente resultar.

Desta forma, autorizo:

Que as fotografias e imagens captadas poderão ser reproduzidas parcialmente, ou na sua totalidade, em qualquer suporte (papel ou digital) e integradas sem qualquer proteção da minha identidade.

Por ser verdade, e por nada haver a obstar, esta declaração vai ser assinada por mim,

Local e data:

_____, __/__/202_

Assinatura:

BI/CC nº:

**Artigo científico Conference
International Society of Military
Sciences**

Identification of people in the Wild

Ruben Menino (1), José Silvestre Silva (2) and António Neves (3)

(1) DETI – Dep. of Electronics, Telecommunications and Informatics, University of Aveiro, Portugal, ruben.menino@ua.pt

(2) Portuguese Military Academy & CINAMIL & LIBPhys-UC, jose.silva@academiamilitar.pt

(3) DETI & IEETA, University of Aveiro, Portugal, an@ua.pt

ABSTRACT

In recent years facial recognition has become increasingly important, with the main objective of identifying and recognizing a person. In this work, a study is carried out on several detection algorithms in various types of images, and recognition in visible spectrum images using various pre-processing methods. Experimental results on these algorithms are presented, as well as an improvement of about 30% in the detection of faces in thermal images.

KEYWORDS

Face detection, Face recognition, image processing, deep learning, algorithms, image acquisition

INTRODUCTION

Having an image of a person's face, it is possible to identify and recognize any person, who is in a set of images (Li et al., 2020). This theme is very important these days, due to the various places in which it can be used. It is a research area with a high growth due to the plurality that can be used, including the control of access to stadiums and important events and also increasing the security level in the access to military critical facilities. It is possible to recognize people previously flagged as terrorist groups, people who were aggressive and did not cooperate with security forces, and also recognize people who are considered a threat to national security. To achieve a facial recognition system, it is necessary to have a dataset, perform face detection, extraction and classification.

In the present work, a comparison was made between various detection and recognition methods. These results were motivating to carry out the improvement of one of the algorithms.

METHODOLOGY

The traditional pipeline of a recognition system can be seen in Figure 1. The main modules are the image acquisition process. Typically, there is a pre-processing of the images where it is possible to obtain various features of the person's face. Then it ends with a facial recognition algorithm where it will be possible to classify the input image. The focus of this work is the exploration of algorithms for recognition in non-cooperative environments. Since there was no dataset that provided the best conditions in this work, we proposed the creation of a dataset that was carried out at the University of Aveiro with equipment kindly provided by the military academy. In addition to imaging the visible spectrum, a first step was taken towards developing a model of facial detection in thermal images. Regarding the study carried out, the procedures are as follows:

The processing of the images begins, where the conversion to the gray color gamut and facial detection will be carried out. Several methods will be used to perform this detection such as Haar Cascades, Gradient Histogram and Dlib CNN. After each one of them is tested sequentially, the one that best adapts is chosen and used for the rest of the process. The next step is to extract the main features of the face (mouth, eyebrows, nose, eyes) in the image. A model from the Dlib library is used where the 68 most important coordinates of the face are returned (Álvarez, 2021). Through these characteristic points, facial

frontalization is performed, where the face is rotated in the image to an angle of 0° (front face) and then the same image is resized with the detected frontalized face (Souza, 2018). In the next module, facial recognition is carried out, where a recognition method transforms the characteristics of a face from a training database into a certain data set and is then able to classify and obtain a prediction of that face according to the data acquired after processing of image. The algorithms used for the methodology in Figure 1 were EigenFaces and FisherFaces. To compare various types of recognition methods it was performed using a deep learning algorithm. The HOG detection method was used, where it was possible to detect the face of the images and perform a face alignment through trigonometry, so that the horizontal line of the eyes remains horizontal and recognized through the OpenFace recognition method (Amos et al., 2016).

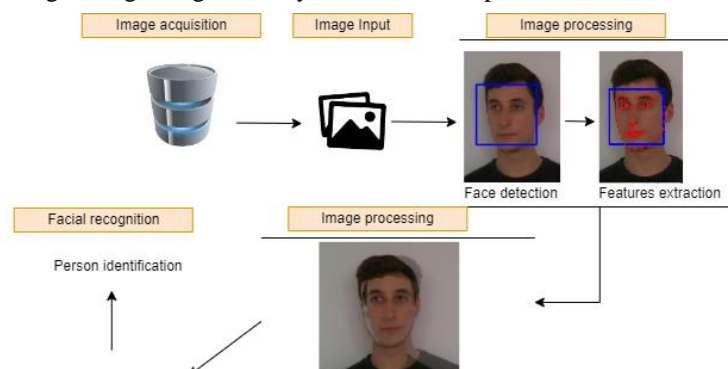


Figure 1- System Flowchart

Regarding the training of a facial detection model on thermal images, the steps are as follows:

OpenCV provides a Haar classifier for face detection, however this classifier is trained for images in the visible spectrum. Because of this, a classifier was created that can detect faces in thermal images.

Through images with the object to be detected (face) and images without the detection object, it was necessary to create a vector file based on the images with object, images that were annotated as positive images, with the help of OpenCV methods. After obtaining the file with the vector of the positive images, it is now possible to use test images to perform facial detection on these images.

RESULTS

1. Dataset creation

Several images were acquired at the University of Aveiro to use as training and testing of the algorithms. Three different cameras provided by the military academy were used, where facial images of people with different angles of rotation of the face, between -90° and 90° were captured. Images were acquired in various spectra (Visible, LWIR, SWIR, NIR).

2. Analysis of face detection methods

For the analysis of the detection methods, the created dataset and an online dataset were used. The detection was carried out with the 3 proposed algorithms, Haar Cascades, Hog and Dlib CNN for each of the datasets. It was possible to reach rotation values at which certain algorithms are no longer able to identify a face and even the processing times of these same algorithms. The Haar method, despite having the shortest processing time (1.02s), presented the worst results. Regarding the Gradient Histogram, it can detect faces in about 19.76s, achieving very positive detections. The Dlib CNN method presented 100% of facial detections, but a very high processing time, about 3142.79s. After obtaining data from the two datasets, it was possible to observe that the same angle was not always detected. Factors such as lighting at the time of capture, camera and face positioning are factors that change these results. Then, the same tests were performed, but in thermal images. It was possible to observe that in the 3 algorithms the detection results were below 57%. One of the main reasons is that these algorithms are trained for visible spectrum images and not for thermal images. After performing the training with the classifier created the detection values, improve from 57% to 85%.

3. Analysis of facial recognition methods

As for the recognition methods, the dataset made with images in the visible spectrum was used, and results from the Eigenfaces and Fisherfaces methods were acquired, with and without the frontalization of the face performed. Two types of tests were performed. First with a training database with 200 images and later another training database with 80 images, performing the tests on 16 test images. The eigenfaces recognition algorithm without frontalization obtained a hit rate of 50% and 43.75% for the training dataset of 200 and 80 respectively, while the fisherfaces method obtained 68.75% and 50%. It is possible to conclude right away that the number of images in the training dataset is a key factor for the success rate of the algorithms. The larger the dataset, the greater the number of true positives and the corresponding hit rate. Regarding the confidence level and processing time, the Fisherfaces algorithm has a lower confidence and processing time than the Eigenfaces algorithm, hence it is possible to observe that Fisherfaces is more certain that the classified face is the face that is being tested. The same tests were carried out but with the frontal face. The hit rate values were much lower in relation to the non-frontalized face. These results are due to the fact that the test images suffer a lot of deformations when frontalizing the face. Although to the human eye, the frontalized face becomes more recognizable, when tested on a machine the same does not happen. Due to the average face in the algorithms being calculated through the pixels of the images, these will vary a lot, because despite the frontalization being performed, there is a lot of noise present in these images. Using a deep learning method that uses deep neural networks, the values were much higher. As we can see in Table 1, we arrived at values with recognition rates of 87.5%.

Test	Train	True Positives	False Positives	Hit rate (%)	Processing time(s)
16	200	14	2	87,5	251,93
16	80	13	3	81,25	107,28

Table 1- Recognition results using deep learning

CONCLUSION

It was shown that a facial recognition system depends on several steps from image acquisition, methods to improve that image, extract important information and its detection and later a classification of that image.

In view of the results obtained in relation to face detection, it is possible to observe that there are several factors that alter the performance of the algorithm. Factors such as brightness, image acquisition camera quality and pose substantially alter these results. Although the Dlib CNN neural network presented more hits than the other methods, there are more factors that can be considered, in this case the processing time, where the HOG algorithm achieved a relatively high hit rate, but with a shorter processing time. Regarding recognition methods, it was possible to observe that methods such as EigenFaces and Fisherfaces, despite not having bad results, are not the best for a recognition system these days. Making sudden changes to the image like frontalization makes the algorithms worse due to the altered pixels of the images. It should be noted that the number of training images is a very important factor for the performance of the algorithm.

Methods based on Deep learning, have much superior results like OpenFace. They are trained by a large dataset and neural network architectures don't need manual feature extraction because they learn features directly from the data. It was also possible to train a Haar classification for thermal images that obtained a detection rate of 85% compared to 57% for images of the visible spectrum.

REFERENCES

- [1] Álvarez, C.(2021, April 13). Dlib-models.<https://github.com/davisking/dlib-models>
- [2] Amos, B., Ludwiczuk, B., & Satyanarayanan, M. (2016). Openface: A general-purpose face recognition library with mobile applications. *CMU School of Computer Science*, 6(2), 20.
- [3] Li, L., Mu, X., Li, S., & Peng, H. (2020). A review of face recognition technology. *IEEE access*, 8, 139110-139120.
- [4] Souza, D.(2018, February 18). Face Frontalization. <https://github.com/dougsouza/face-frontalization>