FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

Carolina Filipa dos Santos Pedro

PROVA DE VIDA PARA IMAGENS DE FACES

Dissertação no âmbito da Licenciatura de Engenharia Biomédica,
orientada pelo Professor Nuno Gonçalves e Luiz Schirmer.

Julho de 2022

Índice

Capítulo 1

Introdução

Nos últimos anos, os códigos tradicionais de acesso a sistemas, como pins e senhas, têm dado lugar à autenticação biométrica, em sistemas de diferentes cenários [1]. Existem não só aplicações em sistemas de segurança do dia-a-dia, como de pagamentos, *check-ins*, desbloqueio de telemóveis ou computadores, mas também em sistemas de identificação governamentais ou de identificação de indivíduos em contextos criminais.

A autenticação biométrica consiste na validação da identidade do utilizador através da extração e análise de características distintivas e únicas do corpo humano, as características biométricas, biológicas ou comportamentais. De acordo com a característica em que se centram podem-se distinguir vários tipos de métodos, sendo os mais comuns o reconhecimento facial, de impressão digital, padrões da retina ou íris, voz e assinatura [2].

Contudo, existem grandes preocupações em relação à sua segurança. Uma das principais passa pela vulnerabilidade a ataques de apresentação realizados por indivíduos que tentam contornar os sistemas entrando sem a devida permissão. Estas vulnerabilidades limitam a implementação deste tipo de tecnologia em condições não supervisionadas, pelo que um dos passos fundamentais é fazer prova de vida, de forma a verificar se a pessoa é um utilizador real que corresponde a credenciais armazenadas no sistema ou se se trata de uma tentativa de o falsificar.

1.1 Motivação

Entre os vários tipos de autenticação biométrica, o reconhecimento facial tem ganho especial destaque devido à sua interação pouco intrusiva, uma vez que não requer contacto físico nem interação humana direta. Além disso, está associado a baixos custos, rapidez e a uma grande adaptabilidade facilitada pela biometria facial natural [3].

Consequentemente, a última década compreendeu um rápido e substancial desenvolvimento deste tipo de autenticação biométrica em muitas áreas [4]. A par deste crescimento, surgem cada vez mais tentativas de contornar este tipo de sistemas, atacando-os, através do uso de máscaras 3D ou da apresentação de fotografias impressas, por exemplo. Existe, assim, uma enorme necessidade de desenvolver técnicas para detetar e impedir tentativas de reconhecimento não autorizadas, distinguindo apresentações autorizadas e reais de falsificações.

Deste modo, tem sido proposta uma grande variedade de algoritmos com base em diferentes abordagens e base de dados. Como incentivo ao desenvolvimento desta área, foram ainda criados concursos para o desenvolvimento dos modelos.

As primeiras estratégias apresentadas envolviam extração de características das imagens de forma tradicional e algoritmos básicos de *machine learning*, no entanto, atualmente, a maioria

das soluções tem por base *deep learning*, particularmente Redes Neurais Convolucionais (CNN's). Estas permitem uma melhoria da eficácia dos métodos, graças à sua capacidade de extrair características profundas das imagens de faces.

Em contrapartida, apesar do maior desempenho e robustez, as técnicas baseadas em CNN exigem uma grande quantidade de dados de treino para evitar o sobre-ajuste, o que traz um problema adicional, pois existem muitos tipos diferentes de ataques e, por vezes, são poucos os dados disponíveis para treinar esses algoritmos.

Desta forma, apesar da evolução ainda existe uma grande limitação quanto a soluções contra ataques de apresentação, principalmente na manutenção do seu desempenho na generalização para novas situações, pelo que ainda permanece um enorme desafio para esta área.

1.2 Objetivos

O principal objetivo deste trabalho passa por abordar a aplicação de métodos de *deep learning* em ferramentas de prova de vida, essenciais no desenvolvimento e aplicação dos sistemas de segurança com autenticação biométrica.

Assim, a parte inicial deste trabalho consiste num estudo sobre os conceitos básicos de *deep learning* e as suas aplicações previamente implementadas nesta área. Este passo é executado com vista ao desenvolvimento de um modelo de prova de vida, mais concretamente um algoritmo de *deep learning* com recurso a *transfer learning*. Finalmente, para avaliar a sua eficácia e precisão, recorre-se a uma base de dados, a *Wide Multi Channel Presentation Attack* (WMCA), formada por imagens de faces provenientes de diferentes modalidades e com diversos tipos de ataques de apresentação.

1.3 Estrutura do documento

As primeiras secções deste capítulo expõem um enquadramento do tema, as principais motivações e objetivos deste trabalho. Nesta secção é apresentada a estrutura deste documento, que é formado por 5 capítulos.

O capítulo seguinte contém o estado de arte de ataques de apresentação e prova de vida. Além disso, aborda algumas das diferentes estratégias desenvolvidas ao longo dos anos e as suas arquiteturas, a fim de entender as suas principais vantagens e possibilidades de melhoria.

Por sua vez, o capítulo 3 apresenta a descrição da arquitetura definida para o desenvolvimento do modelo e uma exposição das ferramentas utilizadas para tal. É ainda mencionada a importância das bases de dados e informações relativas à base de dados escolhida para o desenvolvimento deste trabalho, bem como as métricas selecionadas como fundamentais para avaliar o desempenho do modelo construído.

Segue-se o capítulo 4, no qual os detalhes de implementação do modelo proposto no capítulo anterior e os resultados obtidos são discutidos. Finalmente, o capítulo 5 consiste na apresentação das conclusões do trabalho baseadas nos resultados obtidos.

Capítulo 2

Estado de arte

2.1 Ataques de apresentação

Os ataques a sistemas de autenticação biométrica com reconhecimento facial podem ser divididos em indiretos e diretos [5]. Os primeiros são realizados por *hackers* que entram com sucesso nos sistemas e conseguem adulterá-los.

Por sua vez, os ataques diretos, os ataques de apresentação, ocorrem na fase de captura de dados de um sistema biométrico. Segundo Yu *et al.* [6], existem algumas formas de dividir este tipo de ataques. De acordo com o seu intuito, distinguem-se as ofuscações, ou seja, tentativas de ocultar a própria identidade e as personificações, isto é, tentativas de passar por outra pessoa. Destaca-se ainda o *face morphing*, no qual um indivíduo combina uma imagem do seu rosto com uma imagem de rosto alvo, resultando num rosto com as características de dois indivíduos diferentes. Esses ataques de falsificação desafiam fortemente os sistemas de prova de vida, pois normalmente correspondem a duas identidades diferentes [7].

Quanto à geometria podem ser divididos em ataques 2D e 3D. A apresentação de fotografias impressas ou representações de vídeos com rostos são os exemplos mais comuns de ataques de apresentação 2D. Os ataques 3D são mais sofisticados e realistas e podem envolver o uso de máscaras 3D personalizadas com uma dada identidade.

Existe ainda uma divisão entre ataques totais e parciais. Os primeiros consistem em ataques que envolvem a cobertura total da região do rosto, enquanto os segundos baseiam-se na cobertura de regiões específicas da face apenas. A apresentação de uma fotografia impressa com um corte parcial do rosto e a utilização de óculos são exemplos de ataques parciais.

2.2 Prova de vida

De acordo com o Glossário Biométrico [8], prova de vida é uma técnica usada para avaliar automaticamente se uma amostra biométrica apresentada é realmente de um sujeito biométrico real e fidedigno. Permite, assim, proteger sistemas contra ataques de apresentação.

2.2.1 Métodos de prova de vida

Os métodos de prova de vida têm como base a extração de características relevantes das imagens ou vídeos recebidos. O seu desenvolvimento deve ter em conta os tipos mais comuns de ataques de apresentação, como funcionam e as vulnerabilidades dos sistemas que exploram.

As abordagens mais básicas a este problema requerem a interação do utilizador, recorrendo a métodos de resposta ao desafio, nos quais o indivíduo deve cumprir um conjunto de movimentos previamente definido.

Métodos baseados em movimento

Baseados nos anteriores, surgiram os métodos que têm por base a deteção dos movimentos naturais da face e/ou entre o utilizador e o plano de fundo. Os primeiros trabalhos neste âmbito analisavam em grande pormenor o piscar de olhos [9][10]. No entanto, existem mais sinais e indícios de vida humana que se podem ter em consideração, nomeadamente movimentos da face e da cabeça, rastreamento do olhar e sinais fisiológicos remotos (rPPG - fotopletismografia remota).

Bharadwaj *et al.* [11] criaram um modelo baseado na intensificação dos movimentos faciais precisos, de tal forma que conseguiram exagerar as micro e macro expressões faciais dos rostos. Por conseguinte, distinguiram o padrão de movimento dos rostos reais de movimentos falsificados, identificando, assim, os ataques de apresentação.

Embora consigam bons resultados em caso de ataque com fotografias, estes sistemas geralmente falham em *replay attacks* ou ataques de máscara, principalmente pelo facto dos movimentos serem facilmente simulados por parte destes ataques. Além disso, apresentam um elevado custo computacional.

Métodos baseados em sensores e *hardware* extra

Normalmente, as câmaras que obtém informações de cor RGB são as mais comuns, contudo recorrendo a *hardware* adicional é possível obter informação adicional às imagens de cor, como informação de profundidade ou infravermelhos, por exemplo. Por conseguinte, é possível construir modelos de prova de vida aperfeiçoados. No entanto, apresentam grandes desvantagens, nomeadamente elevados custos, devido ao material necessário. Os sensores de profundidade apresentam falhas contra qualquer tipo de ataque de máscara 3D.

Lagorio *et al.* [12] propuseram um método de prova de vida baseado na estrutura 3D da face. Através do processamento da curvatura 3D dos dados adquiridos é possível que um sistema biométrico distinga um rosto real de um ataque de apresentação 2D, nomeadamente na apresentação de uma fotografia.

Métodos baseados em textura e qualidade de imagem

Existem também abordagens com base na análise da textura, que partem do pressuposto que ataques de apresentação apresentam menor qualidade de imagem, no que se refere à nitidez, textura e luminosidade. Por exemplo, *photo attacks* geralmente contêm defeitos de qualidade de impressão, além de refletirem a luz de maneira diferente a apresentações genuínas.

Määttä *et al.* [13] analisaram estas evidências, através de padrões de microtextura binários locais multi-escala. Estes foram posteriormente codificados num histograma e utilizados num classificador *Support Vector Machine* (SVM).

Peixoto *et al.* [14] desenvolveram um método baseado no processamento e análise das características multiespectrais da pele. Aplicaram o modelo com sucesso a imagens capturas sob condições de iluminação não controladas.

Autherith *et al.* [15] propuserem uma abordagem que consiste na análise das localizações de características faciais no momento da aquisição dos dados e na fotografia do passaporte, de forma a detetar alterações na geometria facial.

Galbally *et al.* [16] consideraram um espaço de características de 14 medidas gerais de qualidade de imagem e combinaram-nas com um classificador LDA simples para detetar as falsificações. O seu modelo apresenta como principais vantagens simplicidade, velocidade, não intrusão, facilidade de uso e baixa complexidade. Além disso, permitiu resolver um dos principais problemas associados a este tipo de métodos: a redução de eficácia à medida que a qualidade das imagens captadas diminui.

Wen *et al.* [17] desenvolveram um método de deteção de falsificações de rosto com base na Análise de Distorção de Imagem para extrair o vetor de características de cada imagem. O objetivo dos autores era projetar um sistema com boa capacidade de generalização, além de uma resposta rápida.

Apesar de alguns bons resultados, como os apresentados acima, estes métodos podem sofrer de má generalização, uma vez que as informações de textura variam de acordo com as câmaras e dispositivos de captura dos vídeos ou imagens. Além disso, devido à falta de um correlação explícita entre as intensidades dos pixéis e os diferentes tipos de ataques, extrair recursos de textura constitui um grande desafio.

É de salientar que é bastante comum encontrar algoritmos de prova de vida que combinam dois ou mais métodos de extração de recursos. Embora esta estratégia aumente a complexidade dos métodos, não é certo que aumente a sua eficácia na deteção de ataques, pelo que é fundamental ter alguma atenção.

Por exemplo, Yan *et al.* [18] propuseram um método que alia a análise da qualidade de imagem à análise dos movimentos naturais da face e da consistência entre o movimento do rosto em relação ao fundo. Apresentaram uma precisão de 100% de precisão quando aplicaram o modelo a um banco de dados de *photo attacks.*

Métodos baseados em *deep learning*

Recentemente, *deep learning* tem sido aplicado de forma eficaz à resolução de vários problemas de visão computacional, nomeadamente classificação de imagens e identificação de objetos. Com base neste facto, foram desenvolvidos bastantes modelos de prova de vida sustentados nestas abordagens, mais concretamente em CNN's.

Como trabalho inicial com redes neuronais profundas nesta área, Yang *et al*. [19] assumiram a prova de vida como um problema binário e apresentaram uma abordagem em que extraem as características das faces a partir de uma arquitetura CNN mais especificamente, um modelo *AlexNet*. Posteriormente, usam um SVM para classificar as imagens como falsificação ou apresentação fidedigna. Além disso, implementaram uma etapa de pré-processamento das imagens inovadora variando os tamanhos das caixas delimitadoras das faces e o número de quadros sucessivos usados na CNN.

No entanto, sistemas de prova de vida como estes têm tendência de sofrer sobre-ajuste, devido a possíveis limitações quanto à quantidade e diversidade de dados de treino. Como

consequência, por vezes não podem ser generalizados para outras situações, tornando-se ineficazes. A fim de contornar este problema, surgiram métodos com fusão entre várias técnicas.

Atoum *et al.* [20] propuseram uma nova abordagem de prova de vida baseado numa CNN de dois fluxos, usando informações de textura e mapas de profundidade. Ao contrário de outros métodos desenvolvidos anteriormente, utilizam não só as imagens de rosto completas, mas também pequenas porções das mesmas. Desta forma, o modelo distingue as falsificações sem aprender traços profundos ou de conjuntos de dados específicos, o que evita o sobre-ajuste, aumentando a generalização do modelo.

Gan *et.al* [21] analisaram as características espaço-temporais de quadros de vídeo usando 3D-CNN para detetar falsificações. Obtiveram bons resultados na deteção de ataques baseados na apresentação de vídeos comparativamente com as soluções apresentadas anteriormente, que se centravam mais em ataques de apresentação de fotografias.

Lucena *et al*. [22] propuseram uma abordagem que usa *transfer learning* de modelos pré-treinados em CNN's. Partiram da arquitetura VGG-16 pré-treinada no conjunto de dados *ImageNet* e modificaram as camadas finais totalmente conectadas, ajustando-as para o caso de prova de vida. Os seus resultados experimentais superaram quase todos os métodos anteriores aplicados às mesmas bases de dados.

Capítulo 3

Método

Este capítulo é dedicado à apresentação dos métodos utilizados na realização deste trabalho, tendo em conta os modelos expostos no estado da arte. A estratégia adotada centra-se no uso de *deep learning* devido às vantagens deste tipo de técnicas na área de prova de vida. Desta forma, primeiramente, são abordados alguns conceitos básicos das CNN's.

Para o funcionamento deste tipo de algoritmos é fundamental uma etapa de treino, com o objetivo de minimizar o erro entre a saída estimada pelo modelo e a saída real. Embora existam vários tipos de treino, foi escolhida uma aprendizagem supervisionada, na qual se recorre a dois conjuntos de dados rotulados: um conjunto de treino e um conjunto de validação. Este último é usado para ajustar os parâmetros do algoritmo e verificar se há *overfitting* da rede. Neste caso, o processo de treino é facilitado pelo uso de *transfer learning*. Após esta etapa, segue-se a fase de teste, na qual se aplica o modelo treinado no conjunto de imagens previamente definido para esta tarefa.

Finalmente, são apresentadas as métricas de avaliação do modelo escolhidas e a base de dados que permite a apreciação do desempenho.

3.1 Redes Neuronais Convolucionais

As CNN's, também designadas por ConvNet's, são um dos tipos de redes neuronais artificias mais utilizados atualmente, em grande parte devido ao seu elevado desempenho, principalmente em situações com dados multidimensionais, como é o caso das imagens. Embora as CNNs tenham ganho um grande destaque nos últimos anos, não é uma arquitetura nova, tendo sido propostas pela primeira vez em 1989 por Yann LeCun, conhecido como o pai das redes convolucionais. As CNN's têm como vantagem o facto de o pré-processamento exigido ser muito menor comparativamente a outros algoritmos de classificação.

A sua arquitetura é inspirada no padrão de conexões dos neurónios no cérebro humano e na organização do córtex visual, tendo como unidades básicas camadas de "neurónios", cada uma com a sua função específica. Os principais tipos de camadas são a camada de entrada, camadas de convolução, de agrupamento, totalmente conectadas e a camada de saída.

Quando uma CNN recebe como entrada uma imagem, esta é convertida num tensor. As camadas de convolução, tal como o seu nome indica, são responsáveis por realizar convoluções sobre os tensores, de forma a extrair os recursos. Para tal, apresentam um grupo de filtros, pequenas matrizes, também conhecidas como *kernels*. O processo inicia-se com a obtenção do produto entre todos os elementos do *kernel* e os do tensor de entrada. Os valores obtidos são somados, dando origem ao valor de saída na posição de correspondente no tensor de saída, ou seja, o mapa de características.

Este procedimento é repetido aplicando vários *kernels* para formar um número arbitrário de mapas de características, que representam diferentes características dos tensores de entrada;

diferentes *kernels* podem, portanto, ser considerados como diferentes extratores de características. O uso de várias camadas convolucionais permite que as primeiras extraiam os recursos de baixo nível, como limites, cores, orientação de gradiente, enquanto as seguintes extraem os detalhes de alto nível. Resumidamente, este processo depende do tamanho e do número de *kernels*, do passo e do preenchimento da imagem definidos previamente, sendo os *kernels* os únicos parâmetros aprendidos automaticamente durante o processo de treino na camada de convolução.

As camadas de agrupamento, também conhecidas como camadas de subamostragem, têm a função de compactar o tamanho espacial dos recursos extraídos. De facto, agrupam todos os mapas de características, sinalizando a presença das características das imagens em vez da sua localização exata. Por conseguinte, diminuem a complexidade computacional e o tempo necessário para o processo. Existem vários tipos de camadas de agrupamento, nomeadamente *Max-pooling*, *Global-pooling* e *Average Pooling*.

A última camada, a camada totalmente conectada, apresenta todas as suas unidades conectadas a unidades das camadas anteriores, pelo que tem a função de resumir os recursos das camadas anteriores, uma vez que têm acesso a todas as ativações dos neurônios da camada anterior, tal como o seu nome indica. De uma forma geral, no final encontra-se uma função de perda, como por exemplo a função *softmax*.

Após cada camada da rede pode seguir-se um processo de ativação, que introduz não linearidade ao sistema mapeando os recursos gerados em valores não lineares. As funções de não linearidade mais conhecidas são a Unidade Linear Retificada (ReLU), *tanh*, PreLU e *sigmoid*.

Partindo desta características das camadas existem várias opções de as organizar de forma a constituir uma CNN. As simples são formadas por uma sequência de várias camadas de convolução e de agrupamento, seguidas por uma ou mais camadas totalmente conectadas.

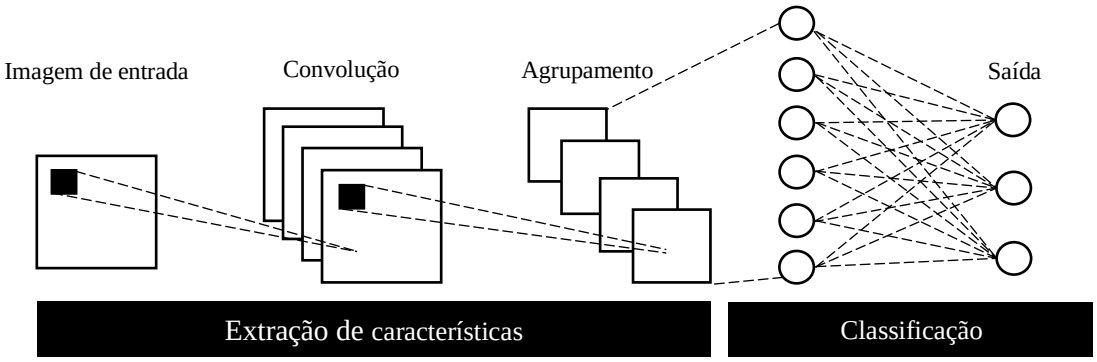


Figura 1 – Representação da arquitetura de uma CNN simples [23].

3.2 *Transfer learning*

Transfer learning é uma estratégia de *machine learning* que permite aplicar redes pré-treinadas na resolução de novos problemas. Possibilita reutilizar o conhecimento adquirido ao desempenhar uma tarefa numa nova, desde que estejam minimamente relacionadas. Assim, de

uma forma geral, o modelo pré-treinado é visto como ponto de partida, sofrendo algumas alterações para se adaptar ao novo problema.

Segundo Lucena *et al.* [22], existem duas possíveis abordagens de *transfer learning* para CNN's. A mais simples utiliza o modelo original para extrair características, usando a saída de uma camada escolhida como entrada para o modelo novo. A mais complexa ajusta o modelo original na totalidade ou parcialmente, treinando os seus pesos através de retropropagação.

Este método apresenta grande aplicabilidade, uma vez que, normalmente, não se treinam redes convolucionais inteiras do zero, pois é difícil ter um conjunto de dados suficientemente grande e este treino apresenta elevados custos computacionais. Além de permitir melhorar e acelerar estes processos o uso de *transfer learning* tem ainda como vantagem reduzir o sobre-ajuste de redes de grandes dimensões.

Existem diversas redes pré-treinadas disponíveis, cada uma com as suas vantagens, estando mais direcionadas para resolver um determinado tipo de problema. Variam também entre si no que toca à velocidade e exigências de poder de computação para serem executadas, além de apresentarem desempenhos distintos.

Um dos grandes desafios do uso de *transfer learnnig* é escolher acertadamente o algoritmo a adaptar ao novo problema e como o fazer sem alterar nenhum aspeto fundamental da rede original que possa influenciar negativamente o resultado. Para a realização deste trabalho foram selecionadas as *ResNets.*

3.3 *ResNets*

Propostas por He *et al.* [24] em 2015, as Redes Neuronais Residuais (*ResNets*) constituem um passo em frente em relação às restantes CNN's, apresentando algumas diferenças na sua arquitetura que permitem resolver certos problemas.

Um aumento das camadas de um CNN e consequente aumento da sua profundidade permite a computação de características mais discriminantes. Em contrapartida, redes muito profundas são difíceis de treinar, apresentando erros de treino crescentes devido a questões na função de otimização e no gradiente de fuga. Estes últimos são encontrados principalmente durante o treino de redes neuronais que envolvem aprendizagem baseada em gradientes e retropropagação. Ao longo do treino, o gradiente é retropropagado para as camadas anteriores, ou seja, usam-se gradientes para atualizar os pesos dos parâmetros da rede. Por vezes, devido a multiplicações repetidas o gradiente torna-se infinitamente pequeno, impedindo que os pesos mudem efetivamente de valor. Podem existir, assim, perdas de informação, pelo que a rede interrompe o processo, pois são propagados os mesmos valores, sem que nenhum trabalho útil seja feito. Por outras palavras, à medida que a rede se aprofunda, seu desempenho fica saturado ou até começa a degradar-se rapidamente.

As *ResNets* têm a capacidade de resolver estes problemas devido à introdução de uma nova camada de rede neuronal, o bloco residual. Tal como representado na figura abaixo, o bloco residual possui duas camadas convolucionais, cada uma delas seguida por uma função de ativação ReLu. Utilizando o "atalho" é possível saltar essas duas operações de convolução e inserir a entrada diretamente antes da última função de ativação ReLu.

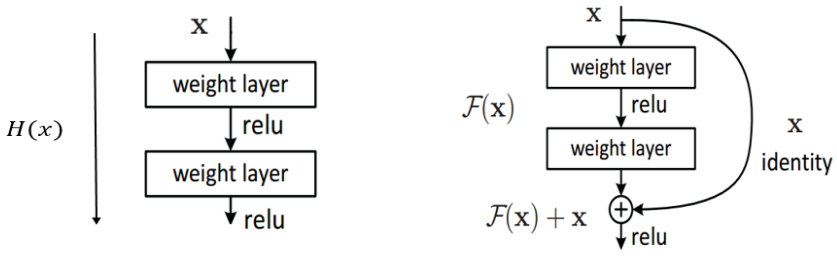


Figura 2 – Arquitetura de uma CNN normal (esquerda) e de uma CNN residual (direita) [24].

As *ResNets* podem ter tamanhos variáveis, dependendo do tamanho de cada uma das camadas do modelo e de número de camadas que apresenta. Para o desenvolvimento deste trabalho a rede escolhida foi a *ResNet-18* devido aos seus bons resultados em várias aplicações diferentes, incluindo reconhecimento facial e prova de vida [25][26][27].

Tal como a sua designação indica, a *ResNet-18* contém 72 camadas, das quais 18 são camadas profundas. Inicialmente, apresenta uma camada de convolução de passo 2 com 64 *kernels*, todos de dimensões 7x7. Segue-se uma outra camada de convolução com um passo de 2, mas com a função *max pool*, que realiza uma operação para o valor máximo para parte de um mapa de características e usa-o para criar um novo mapa com amostragem reduzida.

Na convolução seguinte há uma sequência de *kernel* 1x1,64 seguido de um *kernel* 3x3,64 e um *kernel* 1x1,256. Estas duas camadas são repetidas no total 2 vezes, dando origem a 4 camadas nesta etapa. Conforme ilustrado na tabela 1, seguem-se mais 3 convoluções, que dão origem a 8 camadas.

A rede termina com uma função *average poll* e uma camada totalmente conectada com 1000 conexões seguida de uma função ativação *softmax*.

Nome da camada	**Tamanho da saída**	***ResNet-18***
Conv1	112 x 112 x 64	7x7, passo 2
Conv2_x	56 x 56 x 64	3x3 M*ax pool*, passo 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix}$ x 2
Conv3_x	28 x 28 x 128	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix}$ x 2
Conv4_x	14 x 14 x 256	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix}$ x 2
Conv5_x	7 x 7 x 512	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix}$ x 2
	1 x 1 x 512	*Average pool*, 1000-d fc, *softmax*
FLOP's		$1{,}8 \times 10^9$

Tabela 1 – Arquitetura da *ResNet-18* pré-treinadas na base de dados *ImageNet* [24].

As 1000 conexões podem ser explicadas pelo facto de a rede ser pré-treinada com a base de dados *ImageNet*, composta por mais de 14 milhões imagens de 1000 classes diferentes. A função *average poll* realiza uma operação semelhante à *max pool*, no entanto recorre ao valor médio para cada uma das partes do mapa de características, pelo que extrai recursos com mais facilidade do que a *max pool*, embora menos pormenorizados. Já, a função *softmax* tem a capacidade de normalizar a saída da rede para uma distribuição de probabilidade sobre as classes de saída previstas.

3.4 Base de dados

O desenvolvimento de métodos de prova de vida exige bases de dados bem construídas a nível de qualidade e quantidade para que o seu desempenho possa ser avaliado. Posto isto, ao longo dos anos, a produção deste tipo de base de dados tem vindo a aumentar substancialmente. As particularidades das bases de dados, nomeadamente a quantidade de dados, o tipo de ataques de apresentação, a proporção de casos genuínos e falsificações, devem ser tidos em conta no desenvolvimento do modelo. A maioria apresenta uma ou duas modalidades e as informações de cor são as mais frequentes.

A base de dados WMCA foi produzida e disponibilizada pelo IDIAP no âmbito dos projetos *IARPA BATL* e *H2020 TESLA*, destinando-se à investigação de métodos de deteção de ataques de apresentação para sistemas de reconhecimento facial. Inclui 1679 pequenos vídeos, dos quais 347 são apresentações genuínas e os restantes, 1332, são tentativas de falsificação. Os dados pertencem a 72 indivíduos diferentes e foram registados em sete sessões diferentes a partir de quatro canais distintos: cor, profundidade, infravermelho e térmico.

Os ataques de apresentação presentes na base de dados incluem:

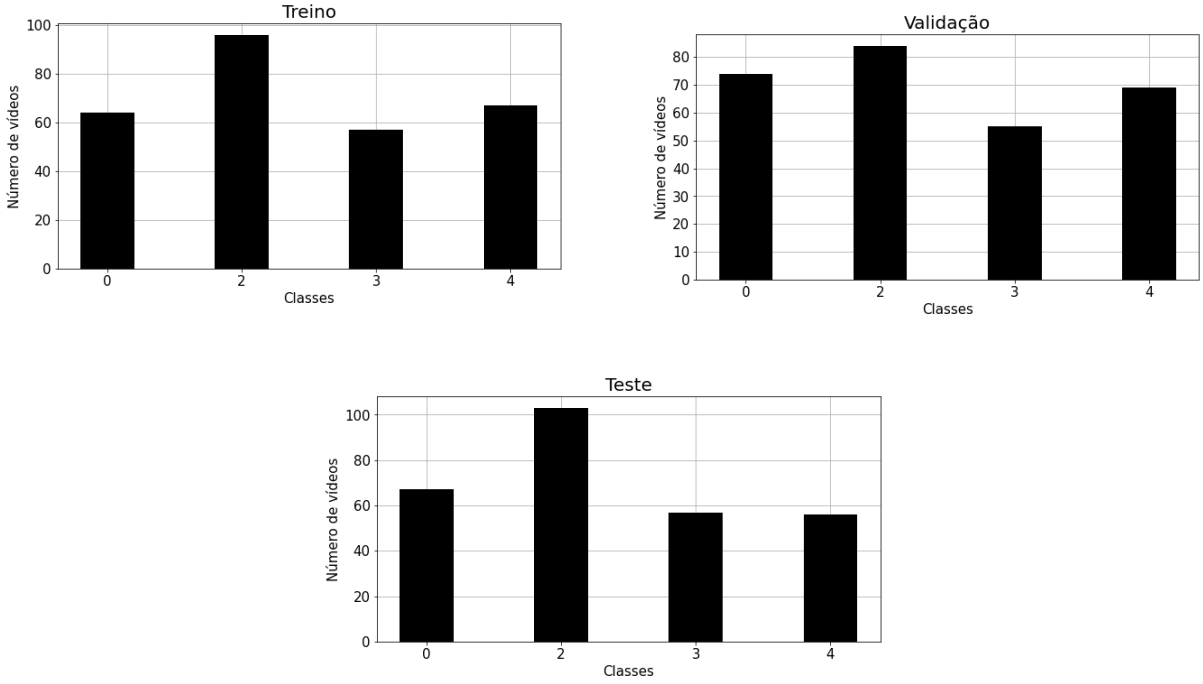
- uso de óculos (de papel ou com olhos falsos desenhados), que constitui um ataque parcial;
- uso de máscaras 3D realistas e personalizadas com base em identidades reais (rígidas de plástico, flexíveis de silicone ou de papel);
- uso de maquilhagem;
- *fake heads*, modelos de cabeças de manequins;
- print *attacks*, imagens de rostos impressas em papéis foscos ou brilhantes A4;
- *replay attacks*, fotos e vídeos eletrónicos. Alguns dos vídeos foram redimensionados para que o tamanho do rosto apresentado na tela fosse proporcional à realidade.

(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figura 3 – Exemplos de ataques de apresentação da base de dados WMCA: (a) óculos de papel, (b) óculos com olhos desenhados, (c) *print attack*, (d) *replay attack*, (e) *fake head*, (f) máscara rígida, (g): máscara rígida, (i) máscara flexível e (j) máscara de papel.

No entanto, na prática a base de dados fornecida é dividida apenas em 4 categorias: "0" para apresentações genuínas, "2" para *fake head attacks* e uso de máscaras, "3" para *photo attacks* e "4" para *video replay attacks*. As imagens estão previamente distribuídas pelas 4 classes de acordo com os gráficos seguintes.



Gráficos 1, 2 e 3 – Distribuição das imagens da base de dados WMCA consoante a sua classe.

Tal como já foi referido, atualmente, os métodos de prova de vida são fundamentais em sistemas de autenticação biométrica que estão implementados em inúmeros dispositivos do dia-a-dia. De uma forma geral, estes apresentam limitações claras, nomeadamente na capacidade de computação e armazenamento de dados. Em primeiro lugar, a qualidade das imagens recebida não é equiparável à qualidade de imagens obtidas nos ambientes controlados de

desenvolvimento dos modelos. Além disso, a quantidade de modalidades diferentes de imagens suportada é menor também.

Deste modo, é fundamental que os modelos desenvolvidos tenham em conta estas restrições, sejam leves e alcancem bons resultados usando informação básica obtida por equipamentos pouco avançados.

Boulkenafet *et al.* [28] avaliaram o desempenho de alguns algoritmos previamente desenvolvidos em condições dos dispositivos reais e concluíram que na maioria dos casos os recursos extraídos dos canais de cor conduzem a um melhor desempenho.

Posto isto, apesar da base de dados escolhida para o desenvolvimento do projeto apresentar informação obtida através de 4 canais diferentes, apenas são usadas informações relativas a canais de cor.

3.5 Métricas de avaliação

A avaliação de desempenho do modelo é uma etapa fundamental para determinar se o modelo foi bem construído e é passível de ser aplicado.

Em problemas de classificação binária podem-se definir falsos negativos (*false negative* -FN), falsos positivos (*false positive* - FP), verdadeiros negativos (*true negative* - TN) e verdadeiros positivos (*true positive* - TP). Tendo em conta que neste caso se pretende determinar quais os casos correspondem a ataques de apresentação, os verdadeiros positivos correspondem às situações em que o modelo determina de forma correta um ataque de apresentação, enquanto os falsos positivos correspondem a situações em que o modelo prevê um ataque de apresentação, mas erradamente. Os verdadeiros negativos estão associados a casos em que o modelo prevê corretamente a presença de uma apresentação genuína e os falsos negativos correspondem a casos em que esta mesma previsão está incorreta. Assim, valores elevados para verdadeiros negativos e verdadeiros negativos refletem um bom funcionamento do algoritmos. Pelo contrário, valores elevados de falsos positivos e negativos indicam erros no modelo, sendo a situação mais preocupante um elevado valor de falsos positivos, uma vez que pode levar a acessos indevidos aos sistemas.

Estes valores podem ser sumariados numa matriz confusão e permitem avaliar a exatidão, precisão e sensibilidade do modelo.

A exatidão representa a proporção de previsões corretas do modelo.

$$Exatidão = \frac{TP + TN}{TP + FP + TN + FN} = \frac{previsões\ corretas}{todas\ as\ previsões}$$

A precisão, também conhecida por valor preditivo positivo, consiste na proporção de identificações positivas realmente corretas.

$$Precisão = \frac{TP}{TP + FP}$$

A sensibilidade, por sua vez, permite determinar quão bom é o modelo a prever casos positivos, neste caso ataques de apresentação.

$$Sensibilidade = \frac{TP}{TP + FN}$$

A partir das últimas duas métricas apresentadas, é possível determinar a *f1-score*.

$$f1 - score = 2 \times \frac{precisão \times sensibilidade}{precisão + sensibilidade}$$

Para estas quatro métricas os valores variam de 0 a 1 (ou 0 a 100%), sendo que um valor mais próximo de 1 significa uma maior exatidão, precisão ou sensibilidade. Nos modelos de classificação com mais do que uma classe, pode-se calcular estas métricas sobre a classe que se pretende prever ou considerar cada classe em separado.

Recentemente, foi introduzida uma nova métrica padrão para esta área em ISO/IEC 30107-3 [29] a Taxa Média de Erros de Classificação (*Average Classification Error Rate* - ACER):

$$ACER = \frac{APCER + BPCER}{2}$$

APCER (*Attack Presentation Classification Error Rate*) representa o peso dos falsos negativos, enquanto a BPCER (*Bonafide Presentation Classification Error Rate*) representa o peso dos falso positivos.

$$APCER = \frac{FN}{FN + TP}$$

$$BPCER = \frac{FP}{FP + TN}$$

Capítulo 4

Experiência e resultados

Esta secção descreve os detalhes da implementação, os vários testes realizados e apresenta a avaliação da eficácia dos métodos propostos, tal como a influência dos vários detalhes nos resultados. O trabalho desenvolvido consiste num algoritmo em *Python*, utilizando *Pytorch*, uma estrutura de *machine learning* de arquitetura flexível, que é facilmente aplicada a modelos de prova de vida, uma vez que permite detetar e decifrar padrões e correlações em imagens de forma a efetuar classificações.

É fundamental referir que, de uma forma geral, a classificação de vídeos é semelhante à classificação de imagens, uma vez que os vídeos podem ser divididos em *frames*, isto é, imagens obtidas com uma certa taxa de captura.

A base de dados fornecida apresenta cada vídeo como um conjunto de 50 *frames*, para cada um dos canais de informação. Desta forma, ainda que inicialmente tenha sido pensada uma abordagem baseada em *PyTorchVideo*, tal não foi necessário, uma vez que a conversão de vídeos em *frames* já tinha sido previamente realizada. Existem, assim, duas alternativas de processar os *frames*: uma abordagem de *frame* único, que consiste no processamento de cada um individualmente, pelo que são avaliados separadamente *frames* do mesmo vídeo e uma abordagem em que são formadas sequências de *frames* para cada vídeo, havendo consideração dos *frames* anteriores em cada etapa.

Vídeo 1	**Vídeo 2**	**Vídeo 3**	…
Frame 1	*Frame* 1	*Frame* 1	…
Frame 2	*Frame* 2	*Frame* 2	…
Frame 3	*Frame* 3	*Frame* 3	…
…	…	…	…

Figura 4 - Ilustração do carregamento de *frames* de forma sequencial [30].

Frame 1	*Frame* 2	*Frame* 3	…	*Frame* 1	*Frame* 2	*Frame* 3	…

Figura 5 - Ilustração do carregamento de *frames* em forma de lista [30].

Foram realizados em primeiro lugar testes com base na abordagem de carregamento dos *frames* em forma de lista e seguiram-se testes baseados na outra abordagem, com recurso ao método *early fusion*, no qual é realizada uma combinação de dados obtidos através dos vários *frames* na fase inicial da rede.

Além disso, como já referido, a base de dados apresenta imagens de 4 classes distintas, pelo que os primeiros testes realizados passaram por tentativas de classificação multiclasse. No entanto, o problema base é distinguir ataques de apresentação de casos genuínos, ou seja, um problema binário, pelo que se seguiram testes direcionados para esta questão. Na prática, não é uma prioridade determinar o tipo de ataque, mas sim distinguir os ataques das apresentações fidedignas.

4.1 Classificação multiclasse

Analisando o gráfico 1 verifica-se que existe uma distribuição não uniforme das classes pelas imagens de treino, pelo que no desenvolvimento deste trabalho houve uma redução do número de imagens da classe "2", descartando a quantidade suficiente para obter uma distribuição mais equilibrada. É importante referir ainda que foram descartados metade dos *frames* disponíveis para cada vídeo, seguindo evidências de trabalhos anteriores, em que é utilizada uma quantidade próxima de *frames* com sucesso.

4.1.1 Pré-processamento

O pré-processamento das imagens da base de dados é um passo importante para que estejam prontas para serem aplicadas ao modelo desenvolvido, ou seja, é fundamental preparar as imagens para a rede tendo em conta os requisitos de entrada do modelo em causa.

Tal como já foi referido, os modelos *ResNet* foram pré-treinados para a base de dados *ImageNet*, pelo que, de uma forma geral, requerem, como entrada, tensores de dimensão [224, 224, 3] (224 x 224 pixéis e 3 canais de cor, RGB). Deste modo, alteram-se para estas dimensões e convertem-se para RGB todas as imagens. Segue-se a conversão das imagens para tensores e a normalização, de acordo com a média e o desvio padrão de pixéis.

Além disso, deve-se ter em conta que se pretende que o modelo tenha a capacidade de distinguir apresentações fidedignas de ataques de apresentação, independentemente da orientação das imagens fornecidas. Aprendendo, assim, com as imagens de entrada, mas não com a forma como são apresentadas. Deste modo, é necessário aplicar transformações aleatórias às imagens usadas na fase de treino, o que aumenta de forma artificial o número de dados. Para tal, para cada época, ou seja, para cada iteração por todas as imagens de treino, aplicou-se aleatoriedade em relação a algumas transformações. Mais especificamente, recorreu-se a uma espelhamento horizontal aleatório, com uma probabilidade de 50%.

As saídas são tensores, que podem finalmente servir de entrada à rede desenvolvida. No caso da abordagem de *frame* único as dimensões destes tensores são (*batch size*, canais, altura, comprimento), enquanto na abordagem sequencial as dimensões obtidas são da forma (*batch size*, *frames*, canais, altura, comprimento).

4.1.2 Implementação do modelo

A forma como é realizado o carregamento das imagens influencia as dimensões dos vetores de entrada para a rede, o que implica também alterações na construção da mesma.

Abordagem de *frame* único

Para adaptar a *ResNet-18* pré-treinada ao problema em questão, deve-se ter em conta que o objetivo final do modelo desenvolvido é distinto do objetivo do trabalho a partir do qual as redes foram pré-treinadas. Neste caso, o intuito é determinar se existe ataque de apresentação e em caso afirmativo de que tipo se trata. Desta forma, distingue-se do objetivo para o qual as

redes foram pré-treinadas, pelo que a saída da última camada totalmente conectada é alterada para 4 classes. Além desta camada é adicionada uma outra camada totalmente conectada seguida de uma função ReLu, com o intuito de aumentar a capacidade da rede.

Nos primeiros testes, foi ainda testada a inserção de uma camada de *dropout*¸ com uma probabilidade de 0,25, apenas ativa durante a etapa de treino. De uma forma geral, este tipo de camada tem a capacidade de ignorar aleatoriamente algumas unidades da rede, para simular o treino de um grande número de arquiteturas ao mesmo tempo, obrigando as suas unidades a se adaptarem a diferentes mudanças. Assim, regulariza o modelo, aumentando a sua generalização, embora diminua ligeiramente a exatidão máxima de treino. Esta diminuição provocou a existência de muitas épocas com a exatidão da validação superior à de treino, pelo que se optou pela eliminação da camada de *dropout,* apesar das suas vantagens.

Nos processos de treino e validação, utilizou-se 32 como tamanho de lote e 20 épocas. Tal como referido anteriormente, durante estas fases, o intuito do modelo é minimizar o erro entre a saída estimada pelo modelo e a saída real, que é representado pela função de perda, também designada por função de erro ou custo. Esta função permite calcular os pesos da rede que levam ao melhor resultado possível, ou seja, a um menor valor de perda. Desta forma, é importante escolher uma função que se adapte bem ao problema. Tendo em conta, que o objetivo passa por uma classificação multiclasse, recorre-se à *cross entropy*, que fornece valores de perda entre 0 e 1, sendo que quanto mais próximo do 0 for, menor é o erro associado ao modelo. O valor obtido representa a diferença entre a distribuição de probabilidade prevista pelo modelo e a distribuição real.

Ao treinar os modelos, foram testados vários optimizadores para alterar os atributos das redes neurais com o objetivo de reduzir as perdas no processo de treino. Face aos resultados, selecionou-se o optimizador *Adam* (*Adaptive Moment Estimation*), que se baseia num gradiente descendente. Tem como particularidade o facto de calcular taxas adaptativas individuais para os diferentes pesos da rede, enquanto a maioria dos outros optimizadores apresenta uma taxa adaptativa única. Essa taxa adaptativa, ou taxa de aprendizagem, tem o valor de 0,0005, um valor baixo, uma vez que é aplicado *transfer learning*.

Abordagem sequencial (*early fusion*)

Neste caso, a entrada da rede é alterada de forma a estar apta para a aplicação do método *early fusion*. Para tal, coloca-se a primeira camada de convolução da rede pré-treinada a aceitar entradas da forma (*batch size*, 3 x frames, altura, comprimento). Foi adicionada uma camada de *dropout*, com uma probabilidade de 0,25 e tal como no caso anterior a saída da camada totalmente conectada foi alterada para 4. Já à saída do modelo em si foi aplicada uma função *softmax*,

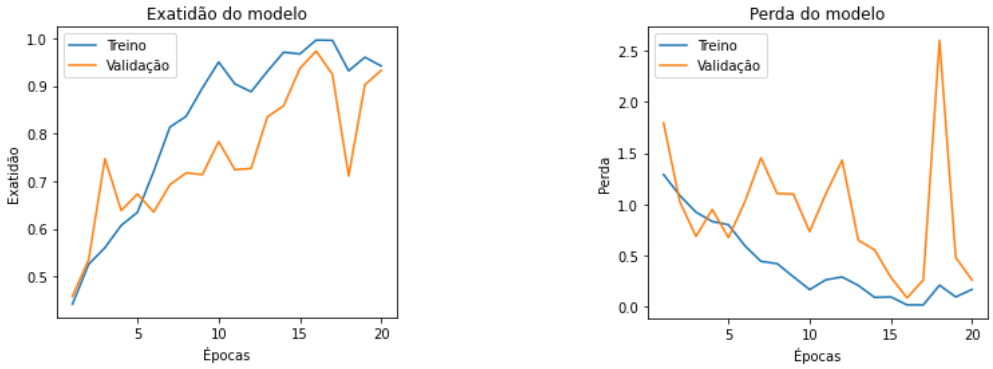
Nas etapas de treino foram usados o mesmo optimizador e função de perda que no caso anterior além do mesmo número para o tamanho de lote e de épocas.

4.1.3 Análise dos modelos

Para avaliar os modelos desenvolvido utilizam-se as primeiras métricas apresentadas no capítulo 3, direcionadas também para casos de classificação multiclasse. A fim de facilitar a análise da exatidão e da perda foram construídos gráficos que mostram a evolução de ambas as métricas durante as etapas de treino e a validação das duas experiências.

Abordagem de *frame* único

Os resultados abaixo referem-se ao processo de treino após a remoção da camada de *dropout*.



Gráficos 4 e 5 – Representação da exatidão (esquerda) e da perda (direita) ao longo das 20 épocas para o modelo de classificação multiclasse com abordagem de *frame* único.

Pela análise dos gráficos, verifica-se que apesar de a exatidão do treino ser superior à de validação na maioria das épocas, esta última não apresenta um comportamento esperado ao longos das 20 épocas, exibindo flutuações. O mesmo acontece com a função de perda de validação, embora diminua de certa forma como esperado, esta diminuição apresenta altos e baixos. Esta situação pode dever-se à remoção da camada de *dropout*. A sua utilização, neste caso apresentou desvantagens, tal como referido anteriormente, no entanto este problema não era tão evidente aquando da sua utilização. Conclui-se, assim, que o uso deste tipo de camadas influencia negativamente os resultados, mas a sua eliminação traz também problemas ao processo de treino.

Uma das causas da existência de flutuações dos valores da perda também pode ser o uso de um tamanho de lote muito baixo para a quantidade de dados em causa, no entanto testou-se um aumento para 128 e, posteriormente, para 256 e os resultados foram semelhantes.

Este fenómeno indicia maus resultados quanto ao desempenho do modelo nas imagens de teste, sendo estes apresentados abaixo, sob a forma de matriz confusão.

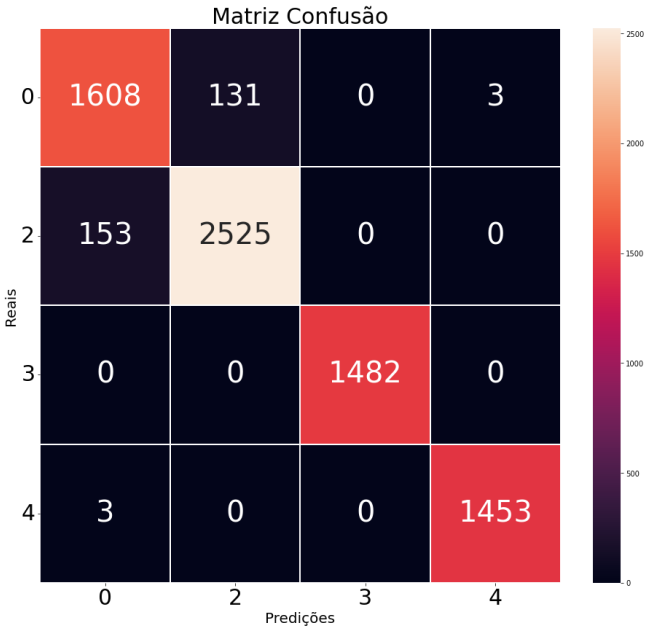


Figura 6 – Matriz confusão associada ao modelo de classificação multiclasse com abordagem de *frame* único.

Classes	Precisão	Sensibilidade	F1-score
0	0,91	0,92	0,92
2	0,95	0,94	0,95
3	1,00	1,00	1,00
4	1,00	1,00	1,00

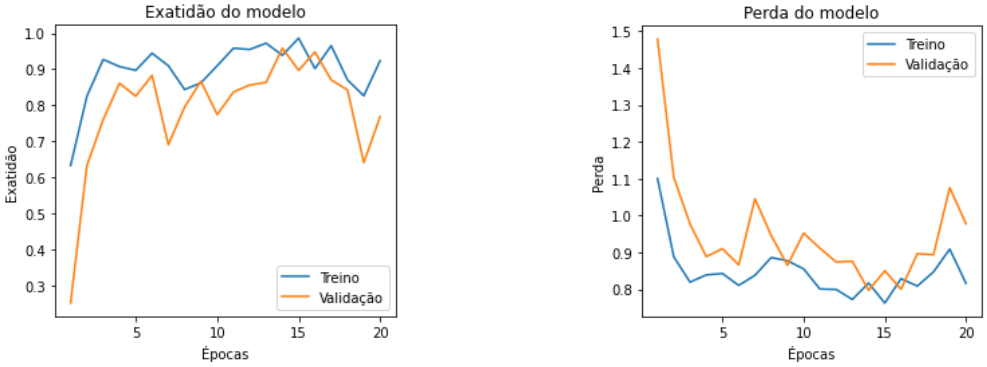
Tabela 2 – Precisão, sensibilidade e *f1-score* associadas ao modelo de classificação multiclasse com abordagem de *frame* único.

Conforme é possível verificar, ao aplicar o modelo treinado aos *frames* de teste, ao contrário do esperado, os resultados foram satisfatórios, com valores próximos de 1 para as precisões e sensibilidades.

No entanto, devido aos resultados da etapa de treino não se pode considerar um modelo confiável para aplicar em sistemas de prova de vida.

Estes resultados podem dever-se ao facto de os vídeos da base de dados darem origem a *frames* muito semelhantes, pelo que o carregamento de forma não sequencial resulta num modelo que processa imagens muito parecidas várias vezes, ficando sobre-ajustado a esses dados.

Abordagem sequencial (*early fusion*)



Gráficos 6 e 7 – Representação da exatidão (esquerda) e da perda (direita) ao longo das 20 épocas para o modelo de classificação multiclasse com abordagem de sequencial (*early fusion*).

Analisando os gráficos obtidos ao longo das 20 épocas de treino e validação, conclui-se que surge o mesmo problema da primeira abordagem: flutuações nos valores de exatidão e perda. Apesar disso, como esperado, têm tendências crescentes e decrescentes, respetivamente. As técnicas já usadas, nomeadamente o aumento do *batch size* e a aplicação de camadas de *dropout* com diferentes probabilidades, também foram testadas. Contudo, mais uma vez não resultaram numa melhoria dos resultados.

Para avaliar o desempenho do modelo e a influência deste fenómeno, executa-se a fase de teste, obtendo-se a matriz confusão, que permite calcular as métricas referenciadas no capítulo 3.

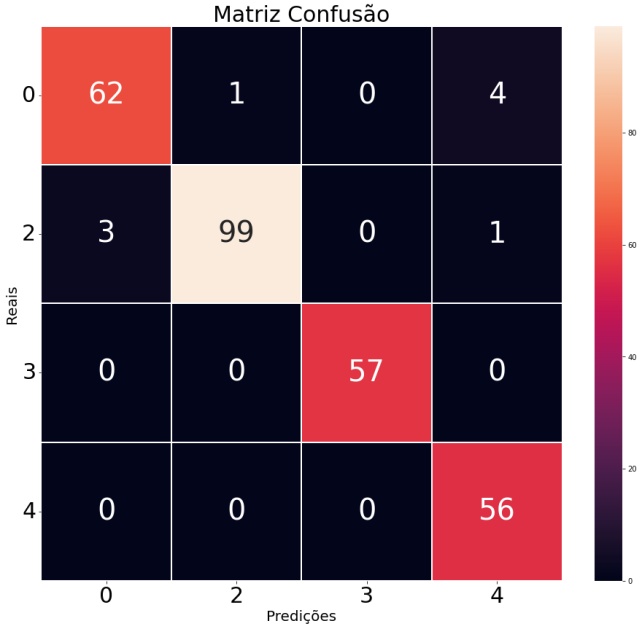


Figura 7 – Matriz confusão associada ao modelo de classificação multiclasse com abordagem sequencial (*early fusion*).

Classes	Precisão	Sensibilidade	F1-score
0	0,95	0,93	0,94
2	0,99	0,96	0,98
3	1,00	1,00	1,00
4	0,92	1,00	0,96

Tabela 3 – Precisão, sensibilidade e *f1-score* associadas ao modelo de classificação multiclasse com abordagem sequencial (*early fusion*).

Estes resultados evidenciam que apesar da fase de treino apresentar irregularidades, o modelo é capaz de generalizar para novas situações e obter resultados satisfatórios na classificação de novos vídeos. Apresenta, valores acima de 0,90 para a precisão, sensibilidade e *f1-score* associadas a todas as classes.

4.2 Classificação binária

Tal como nos casos anteriores, é necessário ter em atenção, a distribuição dos dados pelas classes. Tendo em conta a presença de um número muito superior de ataques de apresentação relativamente a apresentações genuínas, assim como é referido no capítulo anterior, é necessário descartar imagens de ataques de forma a equilibrar a distribuição de dados.

4.2.1 Pré-processamento

O pré-processamento das imagens para este caso é muito semelhante ao anterior. Da mesma forma, é fundamental, que exista uma distribuição equilibrada do número de imagens para cada classe, pelo que tendo em conta o referido no capítulo anterior, para a classificação binária é fundamental selecionar apenas algumas imagens das classes “2”, “3” e “4”, para constituir uma amostra de imagens de ataques de apresentação. A classe “0” continua a corresponder a apresentações genuínas e surge a classe “1” correspondente aos ataques.

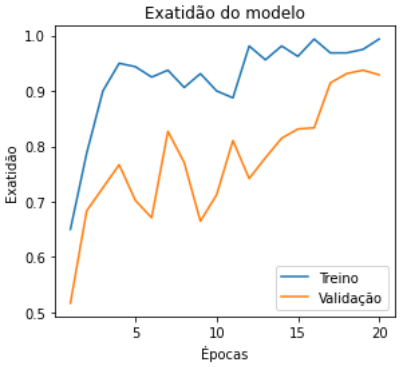
4.2.2 Implementação do modelo

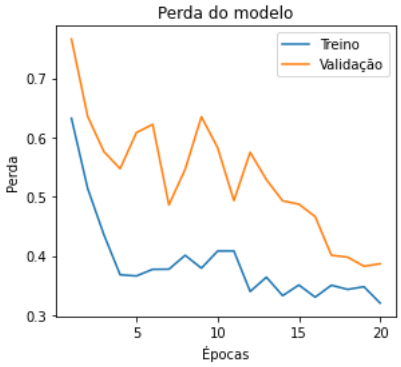
O pré-processamento com uma abordagem sequencial dá origem a tensores da forma (*batch size*, *frames*, canais, altura, comprimento), diferente da aceite pela rede pré-treinada, pelo que a estrutura da mesma foi modificada.

É importante referir que numa classificação binária, tal como o nome indica, existem apenas duas classes, pelo que apesar da estrutura da rede ser semelhante à anterior, a saída da última camada foi alterada para 2 classes. Posteriormente, foi implementada uma camada *dropout* de 0,25 de probabilidade e uma função de ativação *softmax*, embora se tenha testado uma abordagem distinta baseada na função *sigmoid*. O optimizador e a função de perda utilizados são os mesmo que na classificação anterior. Da mesma forma, nos processos de treino e validação, utilizou-se 32 como tamanho de lote e 20 épocas.

4.2.3 Análise do modelo

A avaliação deste modelo pode ser feita com recurso às métricas usadas anteriormente, mas também através da ACER, referenciada no capítulo 3 como uma boa métrica para classificações binárias de prova de vida.





Gráficos 8 e 9 – Representação da exatidão (esquerda) e da perda (direita) ao longo das 20 épocas para o modelo de classificação binária com abordagem sequencial (*early fusion*).

Analisando os gráficos 8 e 9, observa-se, tal como no caso anterior, flutuações no aumento das exatidões e na diminuição das perdas que constituem um alerta de que o modelo não está bem ajustado. Para o testar aplica-se a novos dados e obtém-se os seguintes resultados:

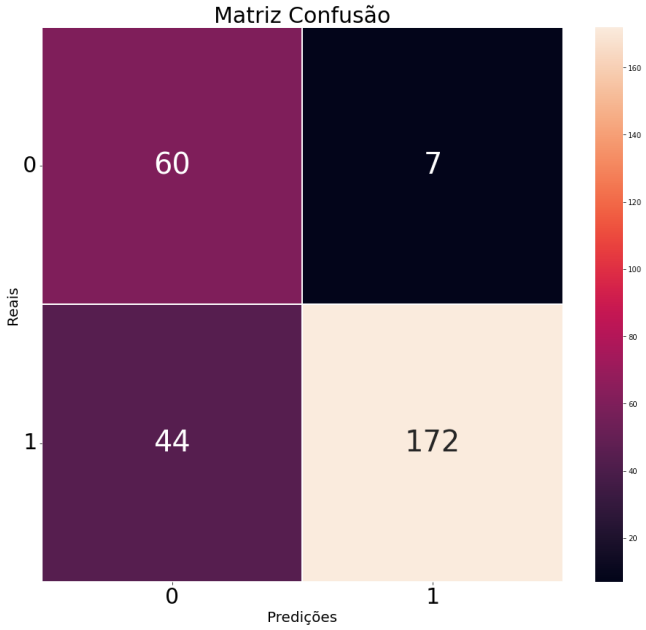


Figura 8 – Matriz confusão associada ao modelo de classificação binária com abordagem sequencial (*early fusion*).

Classes	Precisão	Sensibilidade	F1-score
0	0,58	0,90	0,70
1	0,96	0,80	0,87

Tabela 4 – Precisão, sensibilidade e f1-score associadas ao de classificação binária com abordagem sequencial (*early fusion*).

A partir dos valores da matriz confusão representada acima é ainda possível calcular a ACER.

$$APCER = \frac{FN}{FN + TP} = \frac{44}{44 + 172} = 0{,}204$$

$$BPCER = \frac{FP}{FP + TN} = \frac{7}{7 + 60} = 0{,}104$$

$$ACER = \frac{APCER + BPCER}{2} = \frac{0{,}204 + 0{,}104}{2} = 0{,}154$$

Com base nestes valores, conclui-se que o modelo apresenta um desempenho pior que os modelos apresentados com classificação multiclasse. O principal problema associado é o número de falsos negativos, ou seja, vídeos de ataques de apresentação que não são considerados como tal. Este fenómeno poder-se-ia dever a uma má distribuição das classes, com uma quantidade muito superior de dados de treino da classe "0", uma vez que levaria o modelo a detetar que uma predição "0" leva a uma boa eficácia por a probabilidade de estar correta ser bastante superior. Contudo, este aspeto foi tido em conta e foi realizada uma seleção dos vídeos a usar no treino, de forma a existir uma distribuição equilibrada tanto entre os casos genuínos e os ataques como quanto aos tipos de ataques de apresentação.

Capítulo 5

Conclusões

A aplicação da biometria em sistemas de autenticação requer formas de garantir a sua segurança e evitar tentativas de falsificação. Todos os anos surgem novas técnicas de ataques de apresentação com melhor qualidade, novos instrumentos e que exploram de formas diferentes as fraquezas dos sistemas, pelo que a sua deteção fica mais desafiante. Desta forma, existe uma constante necessidade de desenvolver novos modelos com o propósito de detetar estes ataques, impedindo o acesso de indivíduos não autorizados nos sistemas.

Esta trabalho consistiu na apresentação de modelos para detetar ataques a sistemas de reconhecimento facial, utilizando *deep learning*. Inicialmente, foram referidos os principais avanços nesta área, desde as abordagens mais tradicionais às mais inovadoras, tendo sido dada especial atenção aos métodos que utilizam CNN's. Em conformidade, foram apresentadas duas abordagens, distintas na forma como os *frames* são carregados e dois tipos de classificações: binária e multiclasse, recorrendo a uma CNN pré-treinada, a *ResNet-18*.

Os métodos desenvolvidos deviam estar associados a um baixo erro na deteção de ataques de apresentação e ter a capacidade de serem generalizados a novas situações.

No entanto, um dos principais problemas nos modelos deste género é precisamente o sobre-ajuste. Este leva os modelos a aprender os detalhes e o ruído dos dados de treino como conceitos, o que afeta negativamente o seu desempenho na aplicação a novos dados, uma vez que os conceitos aprendidos não se aplicam a novos dados e influenciam negativamente a capacidade de generalização dos modelos.

Assim sendo, os modelos foram construídos com vista a evitar este problema ao máximo. Uma das causas mais comuns para o sobre-ajuste é uma elevada complexidade do modelo, pelo que foi utilizada uma CNN pré-treinada através de *transfer learning* apenas com pequenas alterações nas camadas finais. Seguiu-se ainda a utilização de camadas de *dropout*, que tal como referido no capítulo anterior também visa a diminuição do sobre-ajuste.

A causa deste problema pode não estar associada na totalidade à arquitetura da rede desenvolvida, pode dever-se ainda à natureza da própria base de dados utilizada para a fase de treino. De forma a contornar alguns dos possíveis problemas, recorreu-se a um aumento de dados de treino de forma artificial com recurso a transformações aleatórias e à normalização das imagens.

Apesar da tentativa de minimização do sobre-ajuste com o intuito de obter modelos eficazes, os três modelos apresentaram bastantes evidências de problemas durante a fase de treino e de validação: flutuações nos valores de exatidão e da perda. Embora tenham sido testadas várias formas de as amenizar, tal não foi possível.

Ainda assim, na fase de teste, os modelos de classificação multiclasse apresentaram resultados satisfatórios, associados a números razoáveis de falsos positivos e negativos, que se refletem em valores elevados de precisão, exatidão, sensibilidade e *f1-score*. Desta forma, podem ser aplicados em sistemas reais não críticos. Já o modelo de classificação binária, apresentou uma

incapacidade de assumir uma grande parte de ataques de apresentação como tal, o que dificulta a sua aplicação em situações reais de reconhecimento facial, uma vez que leva a um elevado número de admissões de indivíduos não autorizados a sistemas, constituindo um grave problema de segurança dos mesmos.

Em suma, os modelos sugeridos poderiam ser alvo de algumas melhorias. Esta necessidade de evolução espelha precisamente o facto da deteção de ataques de apresentação em sistemas de reconhecimento facial ser ainda um problema em aberto, havendo uma constante necessidade de encontrar novas soluções.

Referências bibliográficas

[1] Z. Rui e Z. Yan, "A Survey on Biometric Authentication: Toward Secure and Privacy-Preserving Identification," IEEE *Access*, vol. 7, pp. 5994-6009, 2019, doi: 10.1109/ACCESS.2018.2889996.

[2] A. K. Jain, A. Ross e S. Pankanti, "Biometrics: a tool for information security", *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 125-143, 2006, doi: 10.1109/TIFS.2006.873653.

[3] J. Pereira, "Fingerprint Anti Spoofing – Domain Adaptation and Adversarial Learning", 2020.

[4] H. Zhou, A. Mian, L. Wei, D. Creighton, M. Hossny e S. Nahavandi, "Recent advances on singlemodal and multimodal face recognition: A survey", IEEE Trans. Human-Machine Syst., vol. 44, no. 6, pp. 701–716, 2014, doi: 10.1109/THMS.2014.2340578.

[5] D. B. Sousa, "Detection of Attacks to Face Recognition Systems", 2018.

[6] Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei, e G. Zhao, "Deep Learning for Face AntiSpoofing: A Survey", 2021, doi: 10.48550/arXiv.2106.14948.

[7] A. P. Graça, "Liveness Detection And Facial Recognition With Multi-Modal Features", 2021.

[8] "Biometrics Glossary", https://www.hsdl.org/?abstract&did=464490, 2008.

[9] G. Pan, L. Sun, Z. Wu e S. Lao, "Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera," *2007 IEEE 11th International Conference on Computer Vision*, pp. 1-8, 2017, doi: 10.1109/ICCV.2007.4409068.

[10] L. Sun, G. Pan, Z. Wu e S. Lao, "Blinking-Based Live Face Detection Using Conditional Random Fields", *Lecture Notes in Computer Science book series* (LNIP, volume 4642), pp. 252-260, 2007.

[11] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, e R. Singh, "Computationally efficient face spoofing detection with motion magnification", *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*. Work, pp. 105–110, 2013, doi: 10.1109/CVPRW.2013.23.

[12] A. Lagorio, M. Tistarelli, M. Cadoni, C. Fookes e S. Sridharan, "Liveness detection based on 3D face shape analysis", *2013 International Workshop on Biometrics and Forensics* (*IWBF*) pp. 1-4, 2013, doi: 10.1109/IWBF.2013.6547310.

[13] J. Määttä, A. Hadid e M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," *2011 International Joint Conference on Biometrics (IJCB)*, 2011, pp. 1-7, doi: 10.1109/IJCB.2011.6117510.

[14] B. Peixoto, C. Michelassi e A. Rocha, "Face liveness detection under bad illumination conditions," *2011 18th IEEE International Conference on Image Processing, 2011*, pp. 3557-3560, doi: 10.1109/ICIP.2011.6116484.

[15] S. Autherith e C. Pasquini, “Detecting Morphing Attacks through Face Geometry Features,” *J. Imaging*, vol. 6, no. 11, p. 115, 2020.

[16] J. Galbally e S. Marcel, "Face Anti-spoofing Based on General Image Quality Assessment," *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 1173-1178, doi: 10.1109/ICPR.2014.211.

[17] D. Wen, H. Han e A. K. Jain, "Face Spoof Detection With Image Distortion Analysis" *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 746-761, 2015, doi: 10.1109/TIFS.2015.2400395.

[18] J. Yan, Z. Zhang, Z. Lei, D. Yi e S. Z. Li, “Face liveness detection by exploring multiple scenic clues”, *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 188–193, 2012, doi: 10.1109/ICARCV.2012.6485156.

[19] J. Yang, Z. Lei, S. Z. Li, “Learn Convolutional Neural Network for Face Anti-Spoofing”, 2014, doi: 10.48550/arXiv.1408.5601.

[20] Y. Atoum, Y. Liu, A. Jourabloo e X. Liu, “Face Anti-Spoofing Using Patch and Depth-Based CNNs”, *2017 IEEE International Joint Conference on Biometrics (IJCB)*, 2017, pp. 319-328, doi: 10.1109/BTAS.2017.8272713.

[21] J. Gan, S. Li, Y. Zhai e C. Liu, “3D Convolutional Neural Network Based on Face Anti-spoofing”, *2017 2nd International Conference on Multimedia and Image Processing* (ICMIP), pp. 1-5, 2017, doi: 10.1109/ICMIP.2017.9.

[22] O. Lucena, A. Junior, V. Moia, R. Souza, S. Roberto, E. Valle e R. Lotufo, “Transfer Learning Using Convolutional Neural Networks for Face Anti-spoofing”, *Image Analysis and Recognition. ICIAR 2017. Lecture Notes in Computer Science, vol 10317*, pp. 27-34, 2017, doi: 10.1007/978-3-319-59876-5_4.

[23] V. H. Phung e E. J. Rhee, “A Deep Learning Approach for Classification of Cloud Image Patches on Small Datasets”, *Journal of Information and Communication Convergence Engineering*, pp. 173-178, 2018, doi: 10.6109/jicce.2018.16.3.173.

[24] K. He, X. Zhang, S. Ren e J. Sun, “Deep residual learning for image recognition”, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, 2016, doi: 10.1109/CVPR.2016.90

[25] H. Chen, G. Hu, Z. Lei, Y. Chen, N. M. Robertson e S. Z. Li, “Attention-based two-stream convolutional networks for face spoofing detection”, *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 578-593, 2020, doi: 10.1109/TIFS.2019.2922241.

[26] B. Chen, W. Yang e S. Wang, "Face Anti-Spoofing by Fusing High and Low Frequency Features for Advanced Generalization Capability", *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2020, pp. 199-204, doi: 10.1109/MIPR49039.2020.00048.

[27] Y. Huang, W. Zhang, e J. Wang, “Deep frequent spatial temporal learning for face anti-spoofing”, 2020, doi: 10.48550/arXiv.2002.03723.

[28] Z. Boulkenafet *et al*., "A competition on generalized software-based face presentation attack detection in mobile scenarios", *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 688-696, 2017 doi: 10.1109/BTAS.2017.8272758.

[29] ISO/IEC JTC1 SC37: "Information Technology - Biometric presentation attack detection - Part 3: Testingand Reporting. Technical report, International Organization for Standardization", 2016

[30] D. Karlsson, "Classifying sport videos with deep neural networks", 2017.