UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DETECÇÃO DE ATAQUES A SISTEMAS DE RECONHECIMENTO FACIAL UTILIZANDO ABORDAGENS EFICIENTES DE APRENDIZADO DE MÁQUINA EM PROFUNDIDADE

GUSTAVO BOTELHO DE SOUZA

ORIENTADOR: PROF. DR. APARECIDO NILCEU MARANA
COORIENTADOR: PROF. DR. JOÃO PAULO PAPA

São Carlos – SP

Maio/2019

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DETECÇÃO DE ATAQUES A SISTEMAS DE RECONHECIMENTO FACIAL UTILIZANDO ABORDAGENS EFICIENTES DE APRENDIZADO DE MÁQUINA EM PROFUNDIDADE

GUSTAVO BOTELHO DE SOUZA

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação, área de concentração: Processamento de Imagens e Sinais - Algoritmos e Arquitetura.
Orientador: Prof. Dr. Aparecido Nilceu Marana
Coorientador: Prof. Dr. João Paulo Papa

São Carlos – SP

Maio/2019

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Tese de Doutorado do candidato Gustavo Botelho de Souza, realizada em 21/05/2019:

Prof. Dr. Aparecido Nilceu Marana
UNESP

Prof. Dr. Alexandre Luis Magalhães Levada
UFSCar

Profa. Dra. Heloisa de Arruda Camargo
UFSCar

Prof. Dr. Ivan Rizzo Guilherme
UNESP

Prof. Dr. Marcelo Andrade da Costa Vieira
EESC/USP

A meus queridos pais.

AGRADECIMENTOS

Esta tese é fruto de anos de esforço e dedicação aos estudos, bem como de contibuições, diretas ou indiretas, de meus familiares, professores, amigos, colegas de profissão e da insondável misericórdia e providência divina em minha história.

Agradeço a Deus, pela vida, pela graça de concluir mais esta etapa em minha trajetória acadêmica, profissional e pessoal, e pelas bençãos ao longo do caminho.

A Nossa Senhora, pelos cuidados e constante intercessão ao longo de toda minha vida.

A meus pais, Mauricio e Cidinha, meus primeiros mestres. Obrigado por tudo: pelo carinho, pelo incentivo, pelas orações, pela companhia e pelos braços sempre abertos para mim. Obrigado por eu ser quem sou. Que minha gratidão se eternize aqui. Amo vocês!

À minha irmã Natália, para sempre minha nenê. Obrigado pelas brincadeiras de criança, pelas risadas, pelos passeios, pela Amora (minha fiel companheira de estudos), pelo Kiwi (meu fiel escudeiro), pela presença e por eu ser para sempre seu Tato.

Ao meu amor, Fauninha, por tudo... Por estar, mesmo distante, ao meu lado, e por tornar meus dias mais doces. Por acreditar em mim, pelas orações e por me fazer sorrir sempre.

Aos meus avós, vô Tim e vó Ana, vô Dito e vó Minda, tia Lila (saudade), todos meus tios, primos, familiares e amigos. Obrigado por fazerem parte de minha vida!

Ao professor doutor Aparecido Nilceu Marana (Nilceu), meu verdadeiro *pai*fessor! Orientador qualis A1! Muito obrigado por ter me acolhido pela primeira vez naquela tarde de 2008 em sua sala, eu, um candidato a bacharel, iniciante nas pesquisas e na vida. Obrigado por me guiar desde então, com muito esmero, pelos horizontes do mundo acadêmico e me ajudar logística e intelectualmente muito além de suas atribuições. Um exemplo de pessoa, profissional e amigo.

Ao professor doutor João Paulo Papa (Papa) pelas valiosas sugestões, conversas e por toda ajuda e interesse em meus estudos desde antes do início do doutorado. Deus lhe pague!

Ao professor doutor Alexandre Luís Magalhães Levada (Levada), pelas aulas formais e

informais, por estar sempre disponível e pelo vasto conhecimento sempre compartilhado.

À Universidade Federal de São Carlos (UFSCar), em especial ao Departamento de Computação e ao Programa de Pós-Graduação em Ciência da Computação, por terem me acolhido tão bem nesta importante etapa de minha formação acadêmica e pela confiança em meus estudos e trabalhos. Por todo conhecimento obtido nas aulas, seminários, encontros, e por todo suporte prestado durante o doutorado. Muito obrigado!

À Nvidia pela concessão de GPUs para os estudos e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela bolsa do Programa de Doutorado Sanduíche no Exterior (PDSE) de número 88881.132647/2016-01.

Ao professor doutor Anil K. Jain e à Michigan State University (MSU), onde realizei o período de sanduíche do doutorado. Também aos professores doutores Arun Ross, Oleg Komogortsev e Vishnu Boddeti, pelas aulas, sugestões e pelos ricos conhecimentos adquiridos.

Ao Instituto das Apóstolas do Sagrado Coração de Jesus, onde passei bons anos de minha formação fundamental, e à Universidade Estadual Paulista (UNESP), por minha formação do ensino técnico ao mestrado em Ciência da Computação.

Por fim, agradeço, especialmente, ao Banco do Brasil (BB), particularmente à Diretoria de Tecnologia (Ditec), à Diretoria de Gestão de Pessoas (Dipes) e à Universidade Corporativa Banco do Brasil (UniBB), por esta oportunidade ímpar em minha vida de realizar o sonho do doutorado enquanto funcionário e pesquisador. Para mim, uma grande recompensa por meus esforços até então. Sou eternamente grato. É muito bom ser BB, é muito bom ser UniBB!

A todos meus colegas da Ditec, de todas as divisões, em especial meus orientadores técnicos, Sandro Carlos Vieira e Wagner Leão Costa Filho, mais que orientadores e gerentes, grandes profissionais e amigos. Vocês foram essenciais para a viabilização e sucesso deste projeto. Obrigado pela acolhida na Diretoria de Tecnologia do BB e pelo incentivo durante a jornada!

Agradeço ainda ao nosso diretor Gustavo Fosse, aos gerentes Fábio Castro, Rodrigo Mulinari, Auro Magnan, Geni Andrade, Élemer Carneiro, Mônica Luciana, Nilson Borges, Flávio Stutz, Emanuelle Oliveira, Klailton Ralf, Fabiana Lauxen e Pedro Pimentel, aos colegas Tiago Stutz, Johnatan Oliveira, Michel Duwe, Patrick Gomes, Antônio Carlos Júnior, Maurílio Marques e Ana Cláudia da Cruz, ao pessoal do Labbs pela parceria, do Administrativo e do Capital Humano da Ditec. A todos que fizeram parte de minha trajetória na Tecnologia do BB.

Aos colegas com quem já trabalhei um dia, em especial aos gerentes Iêda Theodoro, Enrico Tucunduva e José Adauto Ribeiro, pelo constante incentivo profissional. Obrigado por tudo!

Porque a Deus nenhuma coisa é impossível.

Lucas 1: 37

RESUMO

A Biometria despontou nas últimas décadas como uma robusta solução para os sistemas de segurança. Entretanto, apesar da maior dificuldade em burlar os sistemas biométricos, nos dias atuais, criminosos vêm desenvolvendo ataques, conhecidos como *spoofing* ou ataques de apresentação, simulando com precisão características biométricas de usuários válidos, como a imagem facial por meio de fotografias impressas em alta definição. Dentre as principais características biométricas, a face se apresenta como uma das mais vantajosas dada sua alta universalidade (todas as pessoas a possuem) e sua extração não intrusiva. Todavia, os sistemas de reconhecimento facial são os mais vulneráveis aos ataques de apresentação dada a alta disponibilidade de imagens faciais, hoje, na rede mundial. Neste contexto, técnicas anti-*spoofing* precisam ser desenvolvidas e integradas aos sistemas de reconhecimento pela face de forma que possam continuar operando em cenários reais. Métodos de Aprendizado de Máquina em Profundidade têm obtido resultados estado-da-arte em muitas áreas, inclusive na detecção de *spoofing* facial. Entretanto, os algoritmos propostos na literatura para tal fim se valem de redes neurais bastante profundas e complexas, sendo muito custosos, computacionalmente, e inviabilizando suas aplicações em ambientes com maiores restrições de *hardware*. Neste sentido, nesta tese são propostas novas arquiteturas de Aprendizado em Profundidade para detecção eficiente de *spoofing* facial. As abordagens propostas, dentre elas, adaptações nas Máquinas de Boltzmann Restritas (*Restricted Boltzmann Machines* - RBM), modelos neurais generativos enxutos convertidos em redes neurais discriminativas profundas, mudanças na arquitetura das Redes Neurais de Convolução (*Convolutional Neural Networks* - CNN), expandindo-as em largura ao invés de profundidade, bem como um novo algoritmo de treinamento para CNNs capaz de capturar informações de *spoofing* locais nas faces, possibilitaram reduzir a quantidade de parâmetros e de operações necessárias no processamento das imagens faciais e agilizar a convergência das redes neurais durante seus treinamentos, propiciando uma detecção de ataques de apresentação com taxas de acurácia compatíveis com o estado-da-arte, porém com menores custos computacionais.

Palavras-chave: *Spoofing*, Ataques de Apresentação, Reconhecimento Facial, Biometria, Máquinas de Boltzmann Restritas (RBM), Redes Neurais de Convolução (CNN), Aprendizado de Máquina em Profundidade, Eficiência Computacional.

ABSTRACT

Biometrics emerged, in the last decades, as a robust and convenient solution for security systems. However, despite the higher difficulty to circumvent the biometric applications, nowadays, criminals are developing attacks, known as spoofing or presentation attacks, precisely simulating biometric traits of legal users, such as the facial image with high-definition printed photographs. Among the main biometric traits, face is a promising one given its high universality (everyone has a face) and non-intrusive capture. Despite all this, face recognition systems are the ones that most suffer with such frauds given the high availability of facial images of people in the worlwide computer network. In this context, face spoofing detection techniques must be developed and integrated to the traditional face recognition applications in order to preserve their robustness in real scenarios. Deep Learning based methods have presented state-of-the-art performances in many areas, including face spoofing detection. However, the methods proposed in the literature so far present high computational costs, being not feasible in real situations, with significant hardware restrictions. In this context, in this thesis, efficient architectures of deep neural networks for face spoofing detection are proposed. Among the proposed approaches, modifications in the architectures of the Restricted Boltzmann Machines (RBM), generative and efficient models turned into deep discriminative neural networks, as well as modifications in the architecture of the Convolutional Neural Networks (CNN), expanding them in width instead of depth, and a novel training algorithm for CNNs, able to capture local spoofing cues of different parts of the faces, allowed a significant reduction on the amount of parameters and operations required for processing the facial images, as well as a faster convergence of the deep neural networks, allowing them to reach accuracy results, in attack detection, compatible with the state-of-the-art, at lower computational costs.

Keywords: Spoofing, Presentation Attacks, Face Recognition, Biometrics, Restricted Boltzmann Machines (RBM), Convolutional Neural Networks (CNN), Deep Learning, Computational Efficiency.

LISTA DE FIGURAS

1.1 Esquema do funcionamento de um sistema biométrico com o módulo de detecção de *spoofing* (em amarelo e tema desta tese) e o módulo de reconhecimento do indivíduo propriamente dito (blocos em azul). Fonte: Elaborada pelo autor. . . 27

1.2 Pesquisa efetuada no Google pelo nome "Gustavo Botelho de Souza", na seção de busca de imagens, e principais resultados retornados. Fonte: Elaborada pelo autor. 29

1.3 Imagens faciais retornadas por pesquisa no Google Imagens pelo nome "Angelina Jolie" com dimensões maiores que 1024×768 pixels. Fonte: Elaborada pelo autor. 30

1.4 Arquitetura da rede neural VGG-16, CNN com 16 camadas de convolução e completamente conectadas. Fonte: Adaptada de Prabhu (2018). 32

1.5 Volume de transações bancárias (em bilhões) por ano no Brasil e participação de cada canal de atendimento bancário. A participação do *mobile banking* cresce acentuadamente de ano em ano. Fonte: Fosse e Baptista (2018). 34

1.6 Dedos artificiais utilizados para fraudar sistemas biométricos em Ferraz de Vasconcelos (esquerda) e no porto de Paranaguá (direita). Fonte: Leite e Paes (2013) e Justi (2014). 35

2.1 Exemplos de características biométricas físicas, fisiológicas e comportamentais: (a) DNA; (b) Forma da orelha; (c) Face; (d) Termograma facial; (e) Termograma da mão; (f) Padrão de distribuição de veias na mão; (g) Impressão digital; (h) Forma de andar; (i) Geometria da mão; (j) Padrão da íris; (k) *Palmprint*; (l) Retina; (m) Assinatura; (n) Voz. Fonte: Adaptada de Jain, Ross e Prabhakar (2004). 40

2.2 Diferentes pontos de ataque que podem ser explorados nos sistemas biométricos tradicionais: do sensor à decisão final. Fonte: Adaptada de Ratha, Connell e Bolle (2001). 43

3.1 Filtros (pesos de conexões entre neurônios) aprendidos durante o treinamento de uma rede neural baseada nas Máquinas de Boltzmann Restritas na primeira (imagens maiores) e segunda (imagens menores) camadas. Diferentes e mais discriminativas características são capturadas a cada camada. Fonte: Adaptada de Krizhevsky (2009). 48

3.2 Exemplo de arquitetura de CNN. Fonte: Adaptada de LeCun et al. (1998). . . . 51

3.3 Exemplo de arquitetura de uma RBM. Fonte: Elaborada pelo autor. 52

3.4 Exemplo ilustrativo de uma DBN. A última camada corresponde a uma RBM com conexões não direcionadas. Fonte: Elaborada pelo autor. 56

3.5 Ilustração do processo de aprendizagem na DBM, considerando ambas as camadas adjacentes no treinamento da camada atual. Fonte: Elaborada pelo autor. 57

3.6 Arquitetura de uma rede DBM sendo treinada com um *patch* de impressão digital de tamanho 36×24. Fonte: Elaborada pelo autor. 59

3.7 Exemplo de convolução espaço-temporal utilizando a arquitetura C3D. O *kernel* de convolução possui 3 dimensões a fim de operar sobre mais de uma imagem por vez capturando variações dos valores dos pixels no tempo. Fonte: Elaborada pelo autor. 60

3.8 Esquema geral de arquitetura de uma rede neural recorrente. Fonte: Elaborada pelo autor. 60

3.9 Exemplo de arquitetura da rede neural LSTM. As operações indicadas em rosa são *pointwise*, isto é, ponto a ponto nos vetores. As operações em amarelo indicam ativações na rede. Os valores de x_t, h_t e C_t indicam os dados de entrada, saída e memória da rede neural, respectivamente. Fonte: Adaptada de Olah (2015). 61

4.1 Esquema do método para detecção de *spoofing* facial com base no MLBP. Fonte: Adaptada de Maatta, Hadid e Pietikainen (2011). 68

4.2 Esquema do LBP-TOP na captura de informações de textura espaco-temporais para classificação facial. Fonte: Adaptada de Pereira et al. (2012). 69

4.3 Deteção de *spoofing* facial utilizando informações de textura e cor. Fonte: Adaptada de Boulkenafet, Komulainen e Hadid (2015). 70

4.4 Abordagem proposta baseadas na seleção de *patches* da face para sua classificação. Fonte: Adaptada de Akhtar e Foresti (2016). 72

4.5 Deteção de *spoofing* facial a partir do CSURF (*Color Speeded-Up Robust Features*). Fonte: Adaptada de Boulkenafet, Komulainen e Hadid (2017). 73

4.6 Arquitetura da rede CNN acoplada à uma rede LSTM proposta por Xu, Li e Deng (2015). "FC" significa *Fully Connected*, isto é, camada com nós completamente conectados. Fonte: Xu, Li e Deng (2015). 74

4.7 Abordagem proposta por Patel, Han e Jain (2016a) com o treinamento inicial sobre imagens mais genéricas (objetos em cenas reais), depois com imagens faciais e finalmente com imagens de bases de anti-*spoofing* facial. Fonte: Adaptada de Patel, Han e Jain (2016a). 76

4.8 Arquitetura do método DPCNN: *fine-tuning* da VGG-Face; extração de características a partir das saídas da convolução; redução de dimensionalidade com o BPCA; e classificação por meio da SVM. Fonte: Adaptada de Li et al. (2016). 77

4.9 Abordagem proposta por Atoum et al. (2017) baseada em CNNs que analisam *patches* e profundidade da face para detecção de *spoofing*. Fonte: Adaptada de Atoum et al. (2017). 78

4.10 Exemplos de imagens faciais sintéticas (primeira linha) e reais (terceira linha) e seus respectivos mapas de ruídos (segunda e quarta linhas). Fonte: Jourabloo, Liu e Liu (2018). 79

4.11 Arquitetura baseada em CNN e LSTM proposta por Liu, Jourabloo e Liu (2018). A quantidade de filtros em cada camada é exibida junto ao seu respectivo bloco. No topo da LSTM aplica-se a Transformada Rápida de Fourier (FFT - *Fast Fourier Transform*) nos dados de saída. Fonte: Adaptada de Liu, Jourabloo e Liu (2018). 80

5.1 Exemplos de imagens da base de *spoofing* facial NUAA. As imagens faciais normalizadas que compõem a base são mostradas em tons de cinza juntamente com as respectivas imagens dos momentos de suas capturas. Fonte: Adaptada de Tan et al. (2010). 85

5.2 Exemplos de faces reais (imagens da linha superior) e sintéticas (imagens da linha inferior) nos vídeos da base Replay-Attack. Fonte: Adaptada de Chingovska, Anjos e Marcel (2012). 86

5.3 Imagens da base CASIA FASD. Fonte: Adaptada de Zhang et al. (2012). 86

6.1 Dada uma imagem facial e seu histograma LBP, após treinar a GB-RBM, um vetor de menor dimensão é obtido ao passar o histograma pela rede e observar as probabilidades de ativação de seus neurônios escondidos. Tal vetor reduzido é então apresentado à SVM. Fonte: Elaborada pelo autor. 89

6.2 Valores de acurácia para cada iteração dos experimentos, variando o tamanho dos vetores de características reduzidos pela GB-RBM e pela PCA. Fonte: Elaborada pelo autor. 90

7.1 Treinamento da arquitetura proposta: os tons de cinza das imagens de treinamento (pré-processadas pelo LBP e normalizadas) e os dois valores referentes às suas classes alimentam a GB-RBM discriminativa. Fonte: Elaborada pelo autor. 93

8.1 Arquitetura proposta da DDRBM. A BB-RBM discriminativa aprende a classificar imagens faciais convertidas pelo LBP (e normalizadas) a partir do treinamento baseado nas características extraídas pela camada escondida da GB-RBM das faces conhecidas e de seus *labels*. Fonte: Elaborada pelo autor. . . . 98

9.1 A operação de convolução na primeira camada da LBPnet atua convertendo a imagem para sua versão LBP (valores binários em preto são exibidos apenas para fins de elucidação, na verdade eles são automaticamente convertidos para decimais - tons de cinza) e realizando a convolução dos valores LBP com os respectivos *kernels* (exibidos em laranja). Fonte: Elaborada pelo autor. 103

9.2 Arquitetura da LBPnet. As duas primeiras camadas realizam operações de convolução e *pooling*. *CONV1* atua não só efetuando a convolução, mas convertendo os valores dos pixels para suas versões LBP primeiro. As dimensões dos *feature maps* de saída são exibidas em preto junto a cada camada. Fonte: Elaborada pelo autor. 104

9.3 Curvas ROC das CNNs propostas, LBPnet e n-LBPnet, do método baseado no MLBP, do método proposto pelos criadores da base NUAA, e do método baseado nos LLDs (*Low Level Descriptors*). Fonte: Elaborada pelo autor. . . . 107

10.1 Extração de características a partir de uma imagem facial e da VGG-Face. Fonte: Elaborada pelo autor. 111

10.2 Amostragem dos *frames* de vídeo de teste e classificação com base na classificação dos *frames*. Fonte: Elaborada pelo autor. 113

11.1 Arquitetura da PatchNet. Os *feature maps* da saída da operação *CONV2* são linearmente retificados e normalizados pela LRN (*Local Response Normalization*). Fonte: Elaborada pelo autor. 119

11.2 Uma imagem de face (96×96 pixels) obtida do conjunto de dados Replay-Attack dividida em nove *patches* (regiões menores não sobrepostas de 32×32 pixels). Fonte: Elaborada pelo autor. 120

11.3 Arquitetura da wCNN. Cada imagem facial é dividida em 9 *patches*, cada um passando por uma rede PatchNet, e então uma decisão final para a face (real ou sintética) é obtida pela função de integração. Fonte: Elaborada pelo autor. . . . 121

11.4 Curvas ROC para a wCNN e a PatchNet (trabalhando sobre as faces completas) para a base Replay-Attack. Quanto mais acima estiver a curva, melhor o método. Fonte: Elaborada pelo autor. 124

11.5 Curvas ROC para a wCNN e a PatchNet (trabalhando sobre as faces completas) na base de imagens CASIA. Fonte: Elaborada pelo autor. 125

12.1 Esquerda: face detectada com base nas características de Haar empregadas por Viola e Jones (2001). Centro e direita: exemplos de características de Haar. Fonte: Bradski (2000). 130

12.2 Autofaces, isto é, os autovetores mais discriminativos para descrever faces humanas de uma base de imagens faciais. Como pode ser visto, suas aparências lembram faces humanas e as diferentes regiões faciais são muito evidentes. Fonte: Dusenberry (2015). 131

12.3 Ilustração do processo de pré-treinamento local da lsCNN. Dada uma imagem facial, ela é dividida em suas 9 regiões principais, de $p1$ a $p9$, e 9 instâncias de uma CNN menor (PatchNet) são treinadas em cada uma delas. Fonte: Elaborada pelo autor. 135

12.4 Inicialização do modelo lsCNN com base nos pesos das 9 PatchNets. As linhas coloridas mais grossas representam 3×3 conexões e são inicializadas com os pesos aprendidos por cada PatchNet. A primeira PatchNet, por exemplo, inicializa os pesos entre os primeiros neurônios (primeiros *feature maps* - FM) nas camadas da lsCNN. As linhas finas pretas pontilhadas também indicam 3×3 conexões, mas inicializadas em zero, e as linhas finas verdes são inicializadas com valores aleatórios de uma distribuição normal com média zero e desvio-padrão de $0,01$). As linhas horizontais cinza tracejadas estão representadas apenas para uma melhor visualização do processo de inicialização. Fonte: Elaborada pelo autor. 136

12.5 Curvas ROC da lsCNN e de uma CNN com mesma arquitetura, mas tradicionalmente treinada, isto é, treinada nas faces em tons de cinza da base NUAA, sem a etapa de pré-treinamento local. Quanto mais acima a curva, melhor o método. Fonte: Elaborada pelo autor. 138

12.6 Imagens faciais segmentadas pela MTCNN da base Replay-Attack (primeira linha) e CASIA (segunda linha). Percebe-se que não há região de fundo para melhor análise das regiões faciais. Fonte: Elaborada pelo autor. 139

12.7 Curvas ROC para a lsCNN e a CNN tradicionalmente treinada sobre as bases Replay-Attack (à esquerda - sobre os vídeos do conjunto de validação) e a CASIA (à direita - sobre o conjunto de teste). As curvas da lsCNN, no geral, são ligeiramente superiores. Fonte: Elaborada pelo autor. 142

12.8 Pesos dos *kernels* 3×3 entre as camadas *CONV1* e *CONV2* da rede lsCNN para a base Replay-Attack. O quadrado azul indica, por exemplo, os 3×3 pesos do *kernel* entre o *feature map* de saída FM1 da camada *CONV1* e o *feature map* FM1 da camada *CONV2* (entre os neurônios da rede). Quanto mais claro o pixel, maior a magnitude do peso correspondente. Fonte: Elaborada pelo autor. 144

12.9 Pesos dos *kernels* entre as camadas *CONV1*-*CONV2*, *CONV2*-*CONV3* e *CONV3*-*CONV4* da rede lsCNN (ao topo) e da rede treinada tradicionalmente (na parte inferior) para a base Replay-Attack. Quanto mais claro o pixel, maior a magnitude do peso correspondente. Fonte: Elaborada pelo autor. 145

12.10 Pesos dos *kernels* entre as camadas *CONV1*-*CONV2*, *CONV2*-*CONV3* e *CONV3*-*CONV4* da rede lsCNN (ao topo) e da rede treinada tradicionalmente (na parte inferior). Ambas as arquiteturas treinadas sobre a base CASIA. Quanto mais claro o pixel, maior a magnitude do peso correspondente. Fonte: Elaborada pelo autor. 146

12.11 *Feature maps* gerados pela lsCNN (primeira linha) e pela CNN tradicionalmente treinada (segunda linha) na camada *CONV1* a partir das mesmas imagens da base Replay-Attack (dois *feature maps* mais à esquerda da linha) e da base CASIA (dois *feature maps* mais à direita da linha). Fonte: Elaborada pelo autor. 147

12.12 Acurácia média das 5 lsCNNs e das 5 CNNs tradicionalmente treinadas por iteração de treinamento, de 0 a 100.000, nas bases Replay-Attack e CASIA. Quanto mais alta a acurácia, melhor. Fonte: Elaborada pelo autor. 148

12.13 *Loss* médio das 5 lsCNNs e das 5 CNNs tradicionalmente treinadas, por iteração de treinamento, de 0 a 100.000, em cada base (Replay-Attack e CASIA). Quanto mais baixo o *loss*, melhor. Fonte: Elaborada pelo autor. 148

12.14 EER médio no conjunto de vídeos de validação e respectivo HTER médio no conjunto de vídeos de teste da base de imagens Replay-Attack bem como EER médio no conjunto de teste da base CASIA para as 5 lsCNNs avaliadas e as 5 CNNs tradicionais. Quanto menores os valores, melhor. Fonte: Elaborada pelo autor. 149

12.15 HTER e EER médio nas bases Replay-Attack e CASIA da lsCNN, da CNN tradicionalmente treinada e da lsCNN treinada sobre *patches* aleatórios. Quanto menores os valores, melhor. A lsCNN original apresentou o melhor desempenho. Fonte: Elaborada pelo autor. 151

12.16 Arquitetura proposta com a LSTM entre a camada FC (*Fully Connected*) e a *softmax*. A cada *frame* a LSTM recebe informações da lsCNN (da face atual) bem como de seu estado anterior a fim de gerar uma saída para a camada *softmax*. O estado da LSTM atual alimenta a análise futura do próximo *frame*. Fonte: Elaborada pelo autor. 157

LISTA DE TABELAS

2.1 Análise das principais características biométricas em relação às propriedades desejáveis. Fonte: Jain, Ross e Prabhakar (2004) 41

4.1 Síntese das abordagens apresentadas. Fonte: Elaborada pelo autor. 81

8.1 Resultados de acurácia (%) nos 10 experimentos realizados para a DDRBM e a GB-RBM discriminativa isolada. A média das acurácias obtidas e o desvio-padrão dos resultados de cada modelo são exibidos ao fim da tabela. Valores máximos de cada abordagem estão destacados. Fonte: Elaborada pelo autor. . . 100

9.1 Resultados (%) em diferentes métricas obtidos pela LBPnet, n-LBPnet, e outros métodos estado-da-arte na base NUAA. O melhor valor em cada métrica está em destaque. Fonte: Elaborada pelo autor. 107

10.1 Resultados (%) na base Replay-Attack em termos de *Half-Total Error Rate* (HTER). Quanto menor o HTER, melhor o método. Também foi calculado o valor de Acurácia (ACC) do método proposto. Os melhores valores estão destacados. Fonte: Elaborada pelo autor. 114

11.1 Quantidade de operações de multiplicação requeridas no *forward pass* de cada imagem facial pela wCNN, PatchNet, CNN baseada em *patches* aleatórios de Atoum et al. (2017), e VGG-face (após *fine-tuning*) de Li et al. (2016). Fonte: Elaborada pelo autor. 122

11.2 Resultados (%) em termos de EER no conjunto de validação e de HTER no conjunto de teste para a base Replay-Attack da arquitetura proposta (wCNN) e outras abordagens estado-da-arte. Quanto menores os valores, melhor o método. Os melhores valores estão destacados. Fonte: Elaborada pelo autor. 126

11.3 Resultados (%) na base CASIA da CNN proposta (wCNN) e de outros métodos estado-da-arte. Os melhores valores estão destacados. Fonte: Elaborada pelo autor. 127

12.1 Arquitetura proposta da lsCNN. A entrada da CNN são imagens faciais RGB (3 canais) com 96×96 pixels: estruturas de tamanho $3 \times (96 \times 96)$. Fonte: Elaborada pelo autor. 133

12.2 Arquitetura de cada CNN menor (PatchNet), parte da lsCNN, treinada em cada região facial, de $p1$ to $p9$ (*patches* com 32×32 pixels, também no espaço de cores RGB). Fonte: Elaborada pelo autor. 134

12.3 Resultados (%) na base Replay-Attack: *Equal Error Rate* (EER) no conjunto de validação e *Half-Total Error Rate* (HTER) nos vídeos de teste. Os melhores valores estão destacados. Fonte: Elaborada pelo autor. 140

12.4 Resultados (%) na base CASIA da rede proposta lsCNN e outros método estado-da-arte. Os melhores valores estão destacados. Fonte: Elaborada pelo autor. . . 141

12.5 Resultados (%) em termos de HTER de testes interbases (considerando as bases Replay-Attack e CASIA) da lsCNN, da CNN tradicionalmente treinada e de outros método estado-da-arte. Os três melhores valores em cada caso estão destacados. Fonte: Elaborada pelo autor. 143

12.6 Resultados (%) nas bases Replay-Attack e CASIA com a amostragem de *frames* dos vídeos. O valor $1/n$ indica a amostragem de um quadro a cada n quadros consecutivos. Os resultados originais das redes neurais (sem amostragem) também constam para fins de comparação. Fonte: Elaborada pelo autor. 152

12.7 Resultados (%) da lsCNN e da CNN tradicionalmente treinada sobre imagens no espaço RGB (original), HSV e YCbCr para comparação. Os melhores valores em cada espaço de cores estão destacados. Fonte: Elaborada pelo autor. . . 153

12.8 Resultados (%) da lsC3D e da CNN 3D tradicionalmente treinada na base Replay-Attack (e da lsCNN e CNN de mesma arquitetura originais). Melhores valores destacados. Fonte: Elaborada pelo autor. 154

12.9 Resultados (%) da lsC3D e da CNN 3D tradicionalmente treinada na base CASIA (e da lsCNN e CNN de mesma arquitetura originais). Melhores resultados destacados. Fonte: Elaborada pelo autor. 156

12.10 Resultados (%) da lsLSTM e de rede de mesma arquitetura tradicionalmente treinada nas bases Replay-Attack e CASIA (e da lsCNN e CNN de mesma arquitetura originais). Melhores resultados destacados. Fonte: Elaborada pelo autor. 156

LISTA DE SIGLAS

ACC - *Accuracy*

AUC - *Area Under the Curve*

BB - Banco do Brasil

BM - *Boltzmann Machine*

BN - *Batch Normalization*

BPCA - *Block Principal Component Analysis*

BRACIS - *Brazilian Conference on Intelligent Systems*

BSIF - *Binarized Statistical Image Features*

Caffe - *Convolutional Architecture for Fast Feature Embedding*

CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

CASIA - *Chinese Academy of Sciences - Institute of Automation*

CD - *Contrastive Divergence*

CDD - *Component Dependent Descriptor*

CF - *Color Frequency*

CIARP - *Iberoamerican Congress on Pattern Recognition*

CIS - *Computational Intelligence Society*

CNN - *Convolutional Neural Network*

CPU - *Central Processing Unit*

CSURF - *Color Speeded-Up Robust Features*

DBM - *Deep Boltzmann Machine*

DBN - *Deep Belief Network*

DCT - *Discrete Cosine Transform*

DDRBM - *Deep Discriminative Restricted Boltzmann Machine*

DLTP - *Dynamic Local Ternary Pattern*

DMD - *Dynamic Mode Decomposition*

DNA - *Deoxyribonucleic Acid*

DoG - *Difference of Gaussians*

DPCNN - *Deep Part Features from the Convolutional Neural Network*

EER - *Equal Error Rate*

ERMAC - Encontro Regional de Matemática Aplicada e Computacional

FAR - *False Acceptance Rate*

FASD - *Face Anti-Spoofing Database*

FBI - *Federal Bureau of Investigation*

FC - *Fully Connected*

FDR - *False Detection Rate*

FFT - *Fast Fourier Transform*

FM - *Feature Map*

FNDR - *False Non-Detection Rate*

FRR - *False Rejection Rate*

FVE - *Fischer Vector Enconding*

GLCM - *Gray Level Co-ocurrence Matrix*

GMM - *Gaussian Mixture Model*

GPU - *Graphics Processing Unit*

HMM - *Hidden Markov Model*

HOG - *Histogram of Oriented Gradients*

HSC - *Histogram of Shearlet Coefficients*

HTER - *Half-Total Error Rate*

IAPR - *International Association of Pattern Recognition*

IBM - *International Business Machines*

IEC - *International Electrotechnical Commission*

IEEE - *Institute of Electrical and Electronics Engineers*

IJCNN - *International Joint Conference on Neural Networks*

ISCAS - *International Symposium on Circuits and Systems*

ISO - *International Organization for Standardization*

LBP - *Local Binary Pattern*

LBP-TOP - *Local Binary Pattern from Three Orthogonal Planes*

LDA - *Linear Discriminant Analysis*

LLD - *Low-Level Descriptors*

LPQ - *Local Phase Quantization*

LRN - *Local Response Normalization*

LSP - *Local Speed Pattern*

LSTM - *Long Short-Term Memory*

LTP - *Local Ternary Pattern*

MBSIF - *Multiscale Binarized Statistical Image Features*

MF - *Mean-Field*

MLBP - *Multiscale Local Binary Pattern*

MLP - *Multilayer Perceptron*

MLPQ - *Multiscale Local Phase Quantization*

MRF - *Markov Random Field*

MSU - *Michigan State University*

MTCNN - *Multi-Task Convolutional Neural Network*

NUAA - *Nanjing University of Aeronautics and Astronautics*

PCA - *Principal Component Analysis*

PLS - *Partial Least Squares*

RBM - *Restricted Boltzmann Machine*

ReLU - *Rectified Linear Unit*

RNN - *Recurrent Neural Network*

ROC - *Receiver Operating Characteristics*

ROI - *Region of Interest*

SGD - *Stochastic Gradient Descent*

SIBGRAPI - *Conference on Graphics, Patterns and Images*

SIFT - *Scale-Invariant Feature Transform*

SURF - *Speeded-Up Robust Features*

SVM - *Support Vector Machine*

TAR - *True Acceptance Rate*

TDR - *True Detection Rate*

TNDR - *True Non-Detection Rate*

TL - *Transfer Learning*

UFSCar - Universidade Federal de São Carlos

UNESP - Universidade Estadual Paulista

UniBB - Universidade Corporativa Banco do Brasil

USSA - *Unconstrained Smartphone Spoof Attack*

VGG - *Visual Geometry Group*

WVC - *Workshop* de Visão Computacional

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO **26**

1.1 Motivação 26

1.2 Detecção de *Spoofing* Facial 28

1.3 Estado-da-arte e Eficiência Computacional 30

1.4 Objetivos 32

1.5 Justificativa 33

1.6 Organização da Tese 36

CAPÍTULO 2 – BIOMETRIA E ATAQUES A SISTEMAS BIOMÉTRICOS **38**

2.1 Biometria e Características Biométricas 38

2.2 Ataques a Sistemas Biométricos 42

CAPÍTULO 3 – APRENDIZADO DE MÁQUINA EM PROFUNDIDADE **46**

3.1 Motivação do Aprendizado em Profundidade 46

3.2 Redes Neurais de Convolução (CNN) 47

3.3 Redes Baseadas nas Máquinas de Boltzmann 51

3.3.1 Máquinas de Boltzmann Restritas (RBM) 52

3.3.2 Redes de Crença em Profundidade (DBN) 55

3.3.3 Máquinas de Boltzmann em Profundidade (DBM) 56

3.4 Redes Neurais Temporais 58

CAPÍTULO 4 – TRABALHOS CORRELATOS **63**

4.1 Métodos Baseados em Características *Handcrafted* 63

4.1.1 Detecção de Vida 63

4.1.2 Comparação da Face e do Plano de Fundo 66

4.1.3 Análise da Qualidade da Imagem Facial 67

4.2 Métodos Baseados em Aprendizado em Profundidade 73

4.3 Síntese dos Trabalhos 80

CAPÍTULO 5 – MATERIAL E METODOLOGIA **82**

5.1 Abordagens Propostas 82

5.1.1 Abordagens Baseadas em RBMs 82

5.1.2 Abordagens Baseadas em CNNs 83

5.2 Métricas de Avaliação 84

5.3 Bases de Imagens 85

CAPÍTULO 6 – DETECÇÃO DE ATAQUES E MÁQUINAS DE BOLTZMANN RESTRITAS **87**

6.1 Textura e Detecção de *Spoofing* Facial 87

6.2 Abordagem Proposta 88

6.3 Experimentos, Resultados e Discussão 89

6.4 Conclusões 91

CAPÍTULO 7 – DETECÇÃO DE ATAQUES E RBMS DISCRIMINATIVAS **92**

7.1 Abordagem Proposta 92

7.2 Experimentos, Resultados e Discussão 94

7.3 Conclusões 95

CAPÍTULO 8 – DETECÇÃO DE ATAQUES E RBMS DISCRIMINATIVAS PROFUNDAS **96**

8.1 Abordagem Proposta 96
8.2 Experimentos, Resultados e Discussão 98
8.3 Conclusões 100

CAPÍTULO 9 – DETECÇÃO DE ATAQUES E CARACTERÍSTICAS CONVOLUCIONAIS DE TEXTURA 102

9.1 LBPnet e n-LBPnet 102
9.2 Experimentos, Resultados e Discussão 105
9.3 Conclusões 108

CAPÍTULO 10 – DETECÇÃO DE ATAQUES E TRANSFERÊNCIA DE APRENDIZADO 109

10.1 Abordagem Proposta 109
10.1.1 Detecção, Normalização e Extração de Características das Faces 110
10.1.2 Treinamento da SVM e Calibração do Limiar de Aceitação 111
10.1.3 Avaliação do Desempenho 112
10.2 Experimentos, Resultados e Discussão 113
10.3 Conclusões 115

CAPÍTULO 11 – DETECÇÃO DE ATAQUES E REDE NEURAL DE CONVOLUÇÃO AMPLIADA EM LARGURA 116

11.1 Abordagem Proposta 116
11.1.1 PatchNet 118
11.1.2 wCNN 118
11.2 Experimentos, Resultados e Discussão 120
11.2.1 Análise da Eficiência 120
11.2.2 Análise da Acurácia 123
11.3 Conclusões 127

CAPÍTULO 12 – DETECÇÃO DE ATAQUES E APRENDIZADO DE CARACTERÍSTICAS LOCAIS PROFUNDAS **128**

12.1 Abordagem Proposta 128

12.1.1 Informações Espaciais e Regiões Faciais 130

12.1.2 lsCNN 132

12.1.2.1 Pré-Treinamento Local 133

12.1.2.2 *Fine-Tuning* Global 134

12.2 Experimento na Base NUAA 136

12.3 Experimentos nas Bases Replay-Attack e CASIA 137

12.3.1 Análise dos Pesos Sinápticos 143

12.3.2 Análise Estatística 146

12.3.3 *Patches* Aleatórios 150

12.3.4 Votação e Amostragem de *Frames* 151

12.3.5 Espaços de Cores HSV e YCbCr 152

12.3.6 Análise Temporal 153

12.3.6.1 lsCNN e C3D 154

12.3.6.2 lsCNN e LSTM 155

12.4 Conclusões 158

CAPÍTULO 13 – CONCLUSÃO **159**

13.1 Contribuições da Tese 161

13.2 Trabalhos Futuros 162

13.3 Publicações 162

13.4 Premiações 165

REFERÊNCIAS **167**

Capítulo 1

INTRODUÇÃO

Neste capítulo é apresentada, inicialmente, a motivação para a realização deste trabalho. Após, apresenta-se breve panorama do estado-da-arte relacionado ao tema desta tese, são definidas as hipóteses de pesquisa e explicitados os objetivos gerais e específicos da tese. Na sequência, apresenta-se breve justificativa para a realização deste trabalho baseada em casos de ataques reais a sistemas biométricos. Finalmente, apresenta-se a organização desta tese.

1.1 Motivação

O ser humano sempre teve a necessidade de identificar-se para obter acesso a informações ou lugares privilegiados. Desde os primórdios da humanidade, segredos mecânicos como chaves e cofres, guardavam os bens. Com o advento dos computadores, cartões magnéticos passaram a fazer parte da segurança das pessoas para proteger seus ativos. Entretanto, por se basearem em "algo que os indivíduos possuem" a fim de reconhecê-los, tais sistemas apresentam inúmeras fragilidades: as chaves e cartões podem ser roubados, esquecidos, perdidos, se deteriorarem ou, até mesmo, ser compartilhados indevidamente. As senhas, que também são antigas no processo de identificação humana, por permitirem o reconhecimento das pessoas por "algo que elas sabem", apresentam, da mesma forma, diversas fragilidades, como o esquecimento, a descoberta por outrem e também o compartilhamento indevido (JAIN; ROSS; PRABHAKAR, 2004; JAIN et al., 2004).

Na tentativa de trazer maior segurança e confiabilidade ao processo de identificação de pessoas, a Biometria despontou de forma a viabilizar o reconhecimento de indivíduos por meio de suas características físicas, fisiológicas ou comportamentais, tais como face, impressão digital, íris, termograma facial, padrão vascular da mão, forma de andar e de digitar, isto é, "pelo que eles são" (JAIN; ROSS; PRABHAKAR, 2004; JAIN; ROSS; NANDAKUMAR, 2011). Dois princi-

pais motivos apontam os métodos biométricos como substitutos para os métodos tradicionais de identificação humana baseados em conhecimento ou posses: (i) a pessoa a ser identificada é obrigada a estar presente fisicamente no local de identificação (onde está o sensor de captura da característica biométrica); e (ii) o reconhecimento biométrico elimina a necessidade de memorizar senhas e carregar chaves ou cartões, tornando o processo bastante conveniente aos usuários (JAIN; ROSS; PRABHAKAR, 2004).

Apesar da identificação biométrica dificultar bastante as fraudes, sabe-se que, diante do aumento de seu emprego nos sistemas de segurança atuais, em especial em aplicações comerciais, criminosos acabaram desenvolvendo e aprimorando meios de driblar tais mecanismos de identificação (MENOTTI et al., 2015). Em geral, os ataques aos sistemas biométricos se baseiam na reprodução de característica física ou comportamental de um usuário válido perante o sensor de captura (câmera, leitor de impressão digital, etc.) por meio das mais diversas técnicas, ataques conhecidos como *spoofing* (MENOTTI et al., 2015; SILVA; MARANA; PAULINO, 2015) ou ataques de apresentação (ISO; IEC, 2016). Diante disto, técnicas anti-*spoofing* se tornam indispensáveis aos sistemas biométricos nos dias atuais para inibir ou dificultar a atuação dos infratores, garantindo que a vantagem citada do uso da Biometria, isto é, que a pessoa a ser identificada realmente esteja presente no local de identificação, ainda seja válida. A Figura 1.1 ilustra o módulo de detecção de *spoofing* acoplado a um sistema biométrico tradicional.

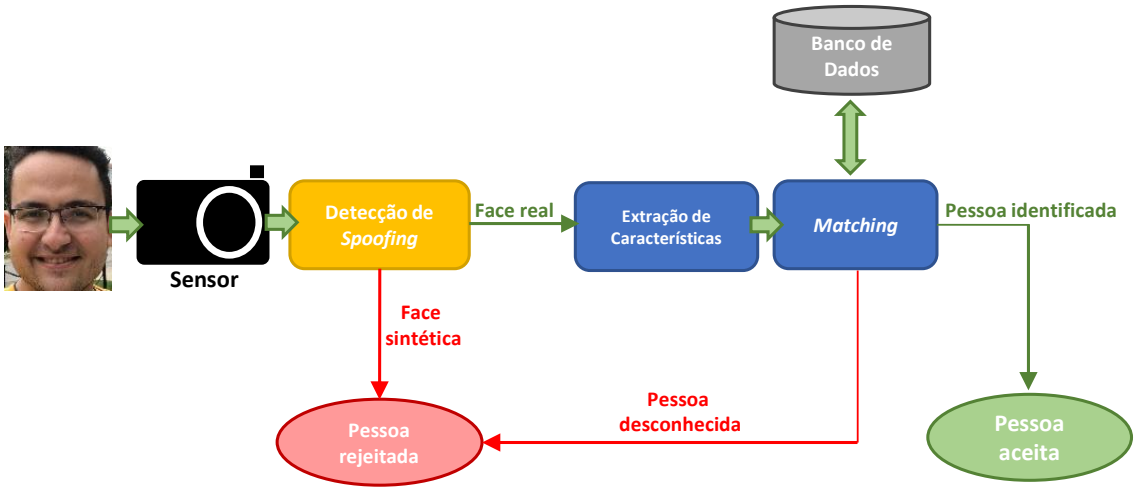


Figura 1.1: Esquema do funcionamento de um sistema biométrico com o módulo de detecção de *spoofing* (em amarelo e tema desta tese) e o módulo de reconhecimento do indivíduo propriamente dito (blocos em azul). Fonte: Elaborada pelo autor.

Um sistema biométrico tradicional possui apenas o módulo de reconhecimento (blocos em azul na Figura 1.1), responsável por reconhecer a pessoa como usuário válido ou não, por meio da extração de suas características biométricas e comparação (*matching*) com os registros da base de dados. Com a agregação de técnica anti-*spoofing* (bloco amarelo), ganha-se maior

robustez a fraudes: caso seja detectado um ataque, o sistema rejeita imediatamente a pessoa, antes de dar início ao processo de identificação. Um método de detecção de *spoofing*, acurado e eficiente, é essencial para agilizar a resposta da requisição de acesso. Um método de detecção de ataques computacionalmente custoso retarda a resposta do sistema como um todo ao usuário.

Vale ressaltar também que, podem ser considerados ataques de apresentação, as tentativas de burlar os sensores de captura não apenas com características sintéticas, moldadas a partir das reais, mas também qualquer tentativa de se passar por usuário válido do sistema, usando, inclusive, partes mutiladas de seu corpo, como dedos removidos da mão ou membros de cadáveres, por exemplo (ISO; IEC, 2016). Entretanto, doravante nesta tese, ao mencionar-se ataque de apresentação ou *spoofing*, estar-se-á fazendo referência ao uso de características sintéticas para burlar os sistemas biométricos (fotografias faciais impressas ou vídeos e fotografias exibidos em *displays*), estando assim em consonância com a maioria dos trabalhos da literatura e bases de dados públicas nela veiculadas. Além disto, ataques de outros tipos apresentam uma maior dificuldade de realização para os criminosos, desencorajando-os muitas vezes, em especial em se tratando de ataques a aplicações instaladas em ambientes comerciais.

1.2 Detecção de *Spoofing* Facial

No contexto dos sistemas de reconhecimento facial, técnicas robustas de anti-*spoofing* são ainda mais importantes visto que a face se configura como uma característica de fácil circunvenção, isto é, que pode ser passível de simulação com facilidade por meio, por exemplo, de fotografias impressas. Apesar da maior conveniência do reconhecimento facial para o usuário, a fraude pode ser uma das mais fáceis quando comparando-se com o reconhecimento biométrico baseado em outras características, como a impressão digital ou íris, por exemplo. Esta facilidade se acentua ainda mais nos dias de hoje dada a abundância de imagens faciais das mais variadas pessoas disponibilizadas em diversos ambientes públicos (em boa resolução), principalmente nas redes sociais. Como exemplo, a Figura 1.2 mostra uma consulta efetuada no Google, em janeiro de 2019, na seção Imagens, pelo nome "Gustavo Botelho de Souza", e os primeiros resultados encontrados. Pode-se perceber que, com exceção da imagem omitida pelo retângulo cinza, todas as demais trazem corretamente a face da pessoa buscada ou partes de seus artigos científicos (alguns inclusive contendo sua imagem facial).

Além dos artigos apontados como relevantes na Figura 1.2, percebe-se que o Google Imagens sugere também palavras-chave ligadas aos trabalhos do autor, o que permite a um possível atacante refinar ainda mais suas pesquisas e obter novas imagens faciais. Além disto, as ima-

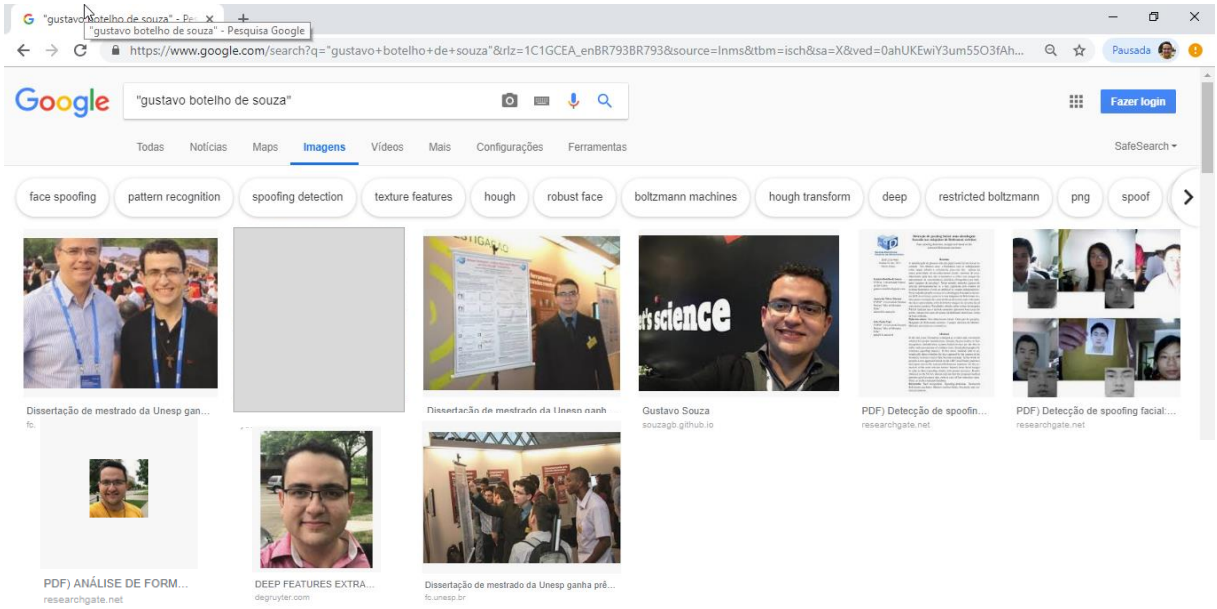


Figura 1.2: Pesquisa efetuada no Google pelo nome "Gustavo Botelho de Souza", na seção de busca de imagens, e principais resultados retornados. Fonte: Elaborada pelo autor.

gens apresentam sua face em diferentes ângulos, ambientes e com diferentes expressões, e a maioria delas possui dimensões superiores a 800×600 pixels, facilitando ainda mais o trabalho de reprodução facial para os criminosos. Na segunda linha da Figura 1.2 (imagens recuperadas mais abaixo na página de pesquisa) nota-se fotografias capturadas de redes sociais e de pesquisa como o *Research Gate*. Vale também lembrar que muitas outras imagens poderiam ser obtidas em pesquisas em outros *websites* de busca ou nas próprias redes sociais.

Para pessoas que têm participação intensa na mídia, o caso é ainda mais preocupante. A Figura 1.3 mostra os primeiros resultados trazidos pelo Google Imagens, em pesquisa realizada em janeiro de 2019, a partir do nome "Angelina Jolie", ao se selecionar a opção de retorno apenas de imagens com dimensões estritamente maiores que 1024×768 pixels. É possível observar que faces da famosa atriz são trazidas em diversos ângulos, poses, ambientes e em alta definição e com ricos detalhes, permitindo criar cópias impressas (ou mesmo digitais) bastante realistas.

Os ataques de *spoofing* a sistemas de reconhecimento facial são divididos em 2 grupos: ataques 2D e 3D (MARCEL; NIXON; LI, 2014). Nos ataques 2D, fotografias impressas em diversos tipos de papel (*print attack*) são apresentadas aos sensores de captura dos sistemas de reconhecimento. Em alguns casos, os olhos são recortados e o criminoso posiciona sua face logo atrás do papel, como que usando uma máscara, a fim de simular o movimento dos olhos e burlar possíveis mecanismos de detecção de ataques que se baseiam em movimentos oculares. Existem ainda ataques 2D em que as imagens ou vídeos faciais são exibidos às câmeras por

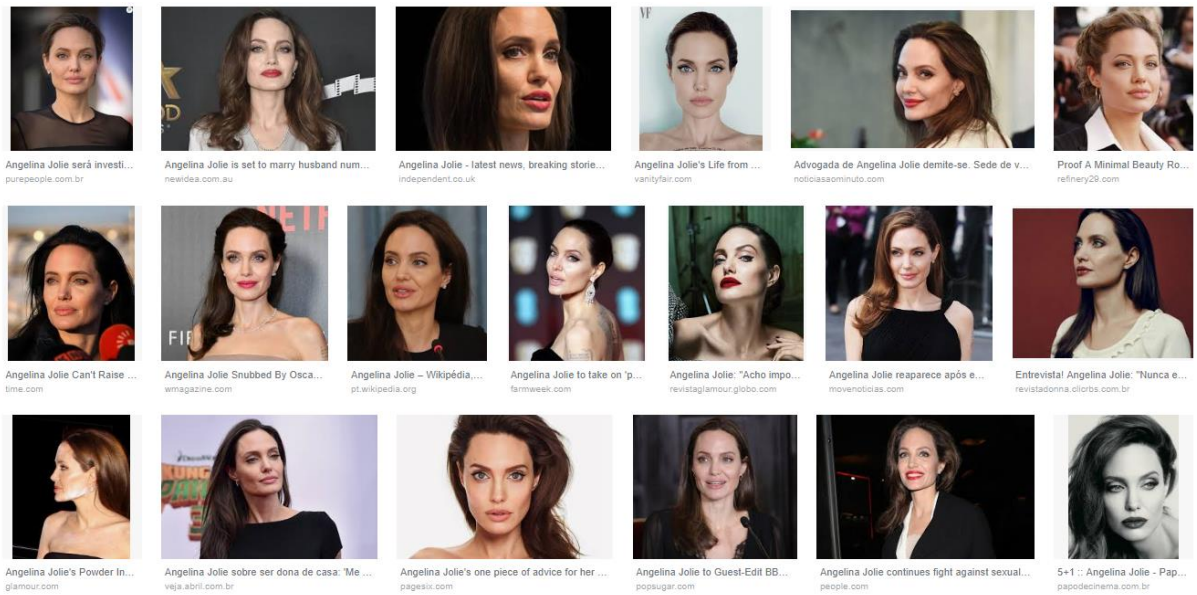


Figura 1.3: Imagens faciais retornadas por pesquisa no Google Imagens pelo nome "Angelina Jolie" com dimensões maiores que 1024×768 **pixels. Fonte: Elaborada pelo autor.**

meio de *displays* de dispositivos móveis (*replay attacks*). Nos ataques 3D, máscaras tridimensionais confeccionadas a partir das faces de usuários válidos são apresentadas aos sensores dos sistemas de reconhecimento. Nesta tese trata-se apenas a detecção de ataques faciais 2D visto que, atualmente, além da confecção de máscaras tridimensionais personalizadas e de qualidade ser bastante cara e requerer serviços especializados, o processo de geração de uma máscara 3D a partir de imagens faciais 2D apresenta alto grau de complexidade e são necessárias várias imagens faciais 2D, capturadas de diversos ângulos do usuário, não sendo uma técnica muito atraente aos criminosos dados os custos e esforços exigidos.

1.3 Estado-da-arte e Eficiência Computacional

Métodos baseados em diferentes princípios têm sido propostos para a detecção de ataques de *spoofing* 2D em sistemas de reconhecimento facial. Muitos deles trabalham com descritores de textura, cor e movimento, extraindo características referenciadas como *handcrafted*, isto é, pré-definidas matematicamente e deterministicamente (sem depender dos dados de treinamento do método) na elaboração das técnicas (MAATTA; HADID; PIETIKAINEN, 2012; CHINGOVSKA; ANJOS; MARCEL, 2012; WEN; HAN; JAIN, 2015; NOWARA; SABHARWAL; VEERARAGHAVAN, 2017). Entretanto, resultados recentes da literatura em diversas áreas, dentre elas, Visão Computacional, Processamento de Linguagem Natural e na própria Biometria, têm mostrado que os métodos de aprendizado multicamadas, isto é, métodos de Aprendizado de Máquina em Profundidade (do inglês, *Deep Learning*), os quais autoaprendem as melhores características de

alto nível para o problema com base nos dados de treinamento, têm alcançado taxas de acerto em patamares cada vez mais elevados, superando em muito as técnicas *handcrafted* tradicionais em diversas tarefas (LECUN; BENGIO; HINTON, 2015).

Ao extraírem informações de alto nível "autoaprendidas" dos dados sendo tratados, as técnicas de Aprendizado de Máquina em Profundidade, em especial as redes neurais profundas, conseguem obter alta abstração, poder de generalização e, consequentemente, robustez em suas aplicações (LECUN; BENGIO; HINTON, 2015). Dentre as arquiteturas de redes neurais profundas, são abordadas nesta tese as Redes Neurais de Convolução (do inglês, *Convolutional Neural Networks* - CNN) (LECUN et al., 1998), que atuam extraindo características de alto nível a partir de operações de convolução e amostragem (*pooling*) dos dados sob análise, e redes baseadas nas Máquinas de Boltzmann Restritas (*Restricted Boltzmann Machines* - RBM) (SMOLENSKY, 1986; HINTON, 2002), que podem ser interpretadas como redes neurais estocásticas baseadas em energia.

Apesar das maiores acurácias das redes profundas, os métodos estado-da-arte hoje para a detecção de *spoofing* facial que se valem de arquiteturas neurais multicamadas da literatura, em geral, empregam estruturas bastante complexas e computacionalmente custosas. No trabalho de Lucena et al. (2017), por exemplo, emprega-se uma rede neural de 16 camadas, denominada VGG-16 (SIMONYAN; ZISSERMAN, 2015), ilustrada na Figura 1.4, a qual apresenta mais de 31 milhões de parâmetros, para uma detecção robusta de *spoofing* facial. Os autores valem-se de uma GPU Nvidia Tesla K40, bastante potente e disponível apenas em ambientes experimentais, para treinar a rede neural. Li et al. (2016) também fazem uso de uma outra rede neural bastante onerosa em termos computacionais, baseada na arquitetura da VGG-16, a chamada VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015), igualmente com 16 camadas, mas treinada sobre imagens faciais de 2.622 pessoas para o reconhecimento de indivíduos. Os autores fazem o *fine-tuning* da VGG-Face em bases de imagens com faces reais e sintéticas e usam as saídas de camadas da rede como vetores de características para treinar uma SVM (*Support Vector Machine*) (CORTES; VAPNIK, 1995). No trabalho de Liu, Jourabloo e Liu (2018), da mesma forma, os autores se valem de CNN bastante complexa bem como de uma rede neural recorrente para estimar a profundidade facial e o fluxo sanguíneo a partir de imagens faciais 2D.

Algumas arquiteturas de CNN, como a MobileNet (HOWARD et al., 2017), foram propostas recentemente na literatura a fim de viabilizar um treinamento e teste mais eficientes. Entretanto, nenhuma investigação mais aprofundada utilizou tal arquitetura ou outras técnicas capazes de tornar os modelos das redes neurais profundas mais otimizados para a detecção de *spoofing* em sistemas de reconhecimento facial.

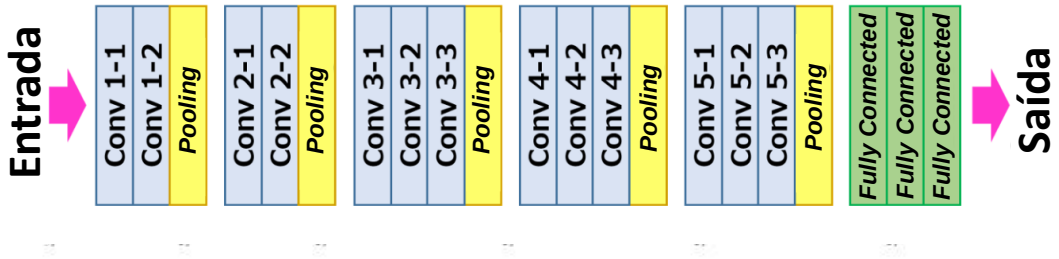


Figura 1.4: Arquitetura da rede neural VGG-16, CNN com 16 camadas de convolução e completamente conectadas. Fonte: Adaptada de Prabhu (2018).

De maneira similar, mesmo em abordagens *handcrafted*, isto é, que não fazem uso de redes neurais profundas, faz-se uma busca extremamente extensiva e bastante cara para a detecção de faces sintéticas, utilizando-se diversas características e custosos classificadores. Chingovska, Anjos e Marcel (2012), por exemplo, calculam diversas vezes o descritor LBP (*Local Binary Patterns*) (OJALA; PIETIKAINEN; HARWOOD, 1994, 1996), variando a vizinhança do pixels, sobre uma dada face e concatenam todos os histogramas gerados, formando uma estrutura com muitas dimensões, a qual é então apresentada a uma SVM. Wen, Han e Jain (2015), por sua vez, combinam vários descritores e utilizam diversas SVMs para a classificação das faces.

Todos esses trabalhos demonstram preocupação apenas com a obtenção de altas taxas de acurácia, porém, sem levar em consideração a questão da eficiência das técnicas propostas (em termos de armazenamento, de processamento e de tempo de execução) e suas aplicabilidades em situações reais, onde há, em geral, restrições severas de *hardware*. Há que se fazer mais, melhor, porém com menos recursos.

1.4 Objetivos

Neste contexto, nesta tese, investiga-se a hipótese da possibilidade de se obter resultados compatíveis com estado-da-arte na detecção de *spoofing* facial por meio de redes neurais profundas, porém a mais baixos custos computacionais. Visando proporcionar maior eficiência do que a obtida pelos métodos existentes, são propostas neste trabalho adaptações nas arquiteturas das redes profundas, bem como em seus algoritmos de treinamento. Também investiga-se o uso de características locais profundas da face na detecção de ataques, as quais são muito utilizadas no contexto da detecção e reconhecimento facial, mas pouco exploradas até então no âmbito da

detecção de *spoofing*, a fim de melhorar as taxas de acerto das abordagens propostas bem como torná-las ainda mais eficientes.

Em suma, tem-se os seguintes objetivos:

Objetivo Geral

A partir de estudos das arquiteturas de Aprendizado de Máquina em Profundidade, propor novas abordagens para a detecção de *spoofing* facial baseada em redes neurais profundas, considerando ataques 2D (faces impressas ou exibidas em *displays*), que apresentem acurácias compatíveis com o estado-da-arte, porém mais eficientes e possíveis de serem executadas em computadores com *hardwares* limitados.

Objetivos Específicos

i) Avaliar as arquiteturas baseadas nas Máquinas de Boltzmann Restritas (*Restricted Boltzmann Machines* - RBM) (SMOLENSKY, 1986; HINTON, 2002) na detecção de *spoofing* facial, seja como modelos generativos ou como modelos discriminativos;

ii) Avaliar as Redes Neurais de Convolução (*Convolutional Neural Networks* - CNN) (LECUN et al., 1998) na detecção de *spoofing* em sistemas de reconhecimento facial bem como técnicas de otimização de tais arquiteturas;

iii) Avaliar as redes neurais temporais, como a C3D (TRAN et al., 2015) e as redes neurais recorrentes LSTM (*Long-Short Term Memory*) (HOCHREITER; SCHMIDHUBER, 1997), em complemento às CNNs ou RBMs na detecção de *spoofing* facial;

iv) Propor novas arquiteturas profundas baseadas nas RBMs e CNNs de forma a manter os resultados estado-da-arte porém a mais baixos custos computacionais;

v) Avaliar o uso de informações profundas locais nas imagens faciais para uma detecção mais acurada e eficiente de *spoofing*, visto que tais informações são bastante exploradas no contexto dos métodos de detecção e reconhecimento facial.

1.5 Justificativa

Dentre as instituições que podem ser bastante beneficiadas por meio de métodos de detecção de *spoofing* facial acurados e eficientes encontram-se as do setor financeiro. No Brasil, conforme mostra a Figura 1.5, o principal canal para a realização de transações financeiras, com

participação crescente ano a ano, corresponde ao *mobile banking* (FOSSE; BAPTISTA, 2018). Em 2017, tal canal já era responsável por cerca de 36% do volume de transações financeiras no país. Neste contexto, o uso da face na autenticação de usuários vem se tornando uma tendência devido à sua conveniência e à existência de sensores de boa qualidade (câmeras) em quase todos os dispositivos eletrônicos, diferentemente de outros sensores. Entretanto, dadas as grandes vantagens que os criminosos podem obter ao fraudar os sistemas de reconhecimento facial das aplicações financeiras, tentativas de ataques não são raras e técnicas de anti-*spoofing* são essenciais para garantir a segurança dos clientes.

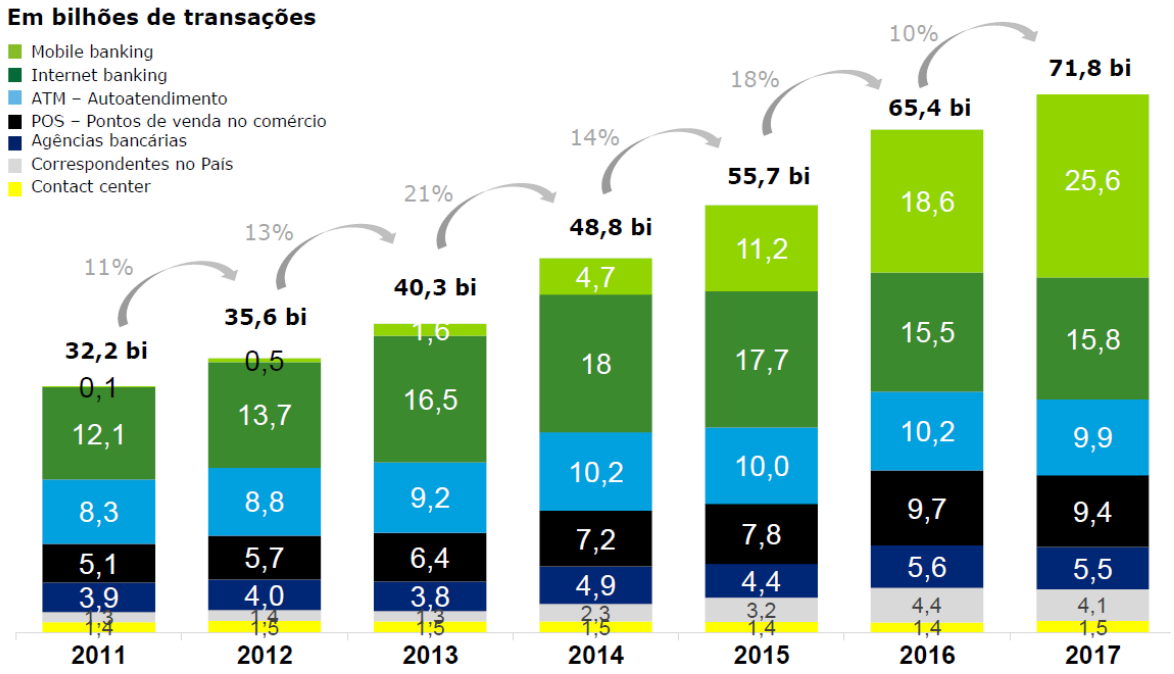


Figura 1.5: Volume de transações bancárias (em bilhões) por ano no Brasil e participação de cada canal de atendimento bancário. A participação do *mobile banking* cresce acentuadamente de ano em ano. Fonte: Fosse e Baptista (2018).

Seguindo esta linha, a Mastercard desenvolveu um aplicativo para realização de compras por meio de confirmação via *selfie*, o chamado *pay-by-selfie*, no qual, para confirmar que o usuário efetuou uma determinada compra, basta ele registrar uma *selfie* sua no momento (DEMARTINI, 2016). Apesar desta inovadora forma de confirmação de pagamentos, o aplicativo apresentava um mecanismo de detecção de *spoofing* facial não tão robusto baseado principalmente em movimentos oculares. A comunidade internacional demonstrou, pouco tempo após o lançamento do aplicativo, que era possível burlá-lo ao apresentar uma face impressa e esconder a região ocular com o dedo ou um lápis de cor próxima à da pele, de forma intermitente, ou usar máscaras 2D, impressas em papéis comuns, com a região ocular recortada e os olhos do atacante posicionados nas respectivas cavidades da máscara (BREITE, 2016).

Outros exemplos reais de fraudes em sistemas biométricos foram noticiados em nosso país, desta vez em sistemas baseados em impressões digitais, ressaltando a necessidade de técnicas anti-*spoofing* junto aos sistemas biométricos tradicionais. Em 2013, foi noticiado o caso de médicos da prefeitura de Ferraz de Vasconcelos, município do estado de São Paulo, que utilizavam dedos sintéticos para registrar o ponto de trabalho para seus colegas, burlando um sistema de reconhecimento biométrico baseado em impressões digitais (LEITE; PAES, 2013).

Outro caso famoso no Brasil, veículado em 2014 na mídia, ocorreu no porto de Paranaguá onde funcionários, também usando dedos sintéticos, registravam entrada e saída do trabalho para seus colegas (JUSTI, 2014). A Figura 1.6 mostra duas imagens divulgadas dos dedos sintéticos capturados pela polícia em ambos os casos. Como pode-se ver, os dedos artificiais apresentam qualidade rudimentar e mesmo assim conseguiam burlar os sistemas de reconhecimento.

Figura 1.6: Dedos artificiais utilizados para fraudar sistemas biométricos em Ferraz de Vasconcelos (esquerda) e no porto de Paranaguá (direita). Fonte: Leite e Paes (2013) e Justi (2014).

Vale citar ainda, como exemplos de fraudes em sistemas biométricos de empresas consolidadas, que logo após o lançamento do sensor leitor de impressões digitais da Apple, o *Touch ID*, a comunidade internacional demonstrou facilidade em burlá-lo com dedos sintéticos, inclusive obtidos de impressões digitais latentes, isto é, deixadas na superfície de vidro dos celulares, devido à ausência de mecanismos mais robustos de detecção de *spoofing* (SCHWARTZ, 2014). Foram obtidos acessos a funções de pagamento e informações pessoais contidas nos dispositivos fraudados. O mesmo ocorreu para o mecanismo de reconhecimento facial dos novos *smartphones* também da Apple, o *Face ID* (PYMNTS, 2017). Os autores do ataque mencionam, inclusive, que pessoas com alta exposição na mídia não devem usar tal mecanismo de segurança dadas suas imagens faciais em abundância na rede mundial de computadores.

É importante ressaltar, no caso dos ataques aos dispositivos da Apple, que alguns críticos questionam, em especial, a reprodutibilidade das fraudes em cenários reais por criminosos leigos. Entretanto, demontrou-se que, mesmo mecanismos hoje existentes e divulgados como à prova de fraudes, não são completamente invioláveis na prática, sendo a detecção de *spoofing* um campo de pesquisa ainda em aberto. Todos estes casos mostram também que a falta de técnicas anti-*spoofing* robustas e ao mesmo tempo eficientes, em especial no caso dos *smartphones* (plataformas com significativas restrições de *hardware*), pode facilmente ser explorada pelos criminosos.

1.6 Organização da Tese

Além deste capítulo inicial, esta tese de doutorado está dividida em outros 12 capítulos:

Capítulo 2 - Biometria e Ataques a Sistemas Biométricos Neste capítulo são definidos conceitos importantes acerca dos sistemas e características biométricos, com destaque para a face humana, bem como conceitos relacionados aos ataques a tais aplicações, com ênfase nos ataques aos sistemas de reconhecimento facial;

Capítulo 3 - Aprendizado de Máquina em Profundidade Neste capítulo apresenta-se referencial teórico referente ao Aprendizado de Máquina em Profundidade, com enfoque nas redes neurais profundas tomadas como base para as arquiteturas propostas nesta tese: as Redes Neurais de Convolução (*Convolutional Neural Networks* - CNN), redes baseadas nas Máquinas de Boltzmann Restritas (*Restricted Boltzmann Machines* - RBM) e as redes neurais temporais;

Capítulo 4 - Trabalhos Correlatos Neste capítulo são apresentados os principais métodos de detecção de *spoofing* facial da literatura, com destaque para os que se valem de Aprendizado de Máquina em Profundidade;

Capítulo 5 - Material e Metodologia Neste capítulo apresenta-se, de maneira geral, as abordagens propostas nesta tese para a detecção de *spoofing* facial 2D, bem como as bases de imagens e métricas de desempenho utilizadas em suas avaliações;

Capítulos 6 ao 12 - Trabalhos Desenvolvidos Nestes capítulos apresenta-se as abordagens eficientes baseadas em Aprendizado de Máquina em Profundidade para detecção de *spoofing* facial 2D propostas nesta tese e divulgadas na forma de artigos em conferências e periódicos qualificados;

Capítulo 13 - Conclusão Neste capítulo apresenta-se conclusão geral da tese com base nos estudos realizados, trabalhos desenvolvidos e resultados obtidos, bem como destaca-se as principais contribuições obtidas, possíveis trabalhos futuros, as publicações realizadas ao longo do doutorado e importantes premiações já recebidas.

Capítulo 2

BIOMETRIA E ATAQUES A SISTEMAS BIOMÉTRICOS

Este capítulo apresenta, de forma sucinta, os conceitos fundamentais relacionados à Biometria e às principais características biométricas, em especial a face. Também são abordados os tipos de ataques a sistemas biométricos, com destaque para os ataques de *spoofing* facial, e conceitos relacionados à detecção destas fraudes.

2.1 Biometria e Características Biométricas

A Biometria, do grego "bio" (vida) e "metria" (medida), pode ser definida como um ramo da Biologia que se volta ao estudo de medidas das características físicas, fisiológicas e comportamentais dos seres vivos. Nos últimos anos ela vem se traduzindo em métodos automatizados de análise destes traços em seres humanos a fim de poder identificá-los unicamente (JAIN; FLYNN; ROSS, 2008; JAIN; ROSS; NANDAKUMAR, 2011).

Com o aumento do poder computacional e o barateamento dos sensores nas últimas décadas, técnicas mais eficazes passaram a ser empregadas no reconhecimento automatizado de pessoas para diversos fins, dentre eles a segurança. Estas técnicas valem-se da Biometria para, por exemplo, permitir acesso de um indivíduo a uma repartição da empresa, a algum sistema ou informações sigilosas, ou ainda para outros fins como a identificação criminal (por exemplo, em um aeroporto visando detectar foragidos) e controle de ponto (JAIN et al., 2004). Baseado nas características físicas, fisiológicas e comportamentais das pessoas, os sistemas são capazes de identificá-las unicamente. Com isto, é possível reconhecer o indivíduo "pelo que ele é" e não "pelo que ele possui" (no caso das chaves e cartões) ou "pelo que ele sabe" (no caso das senhas), tornando os sistemas de identificação humana muito mais robustos e convenientes aos

usuários (JAIN; BOLLE; PANKANTI, 2006).

Os sistemas biométricos, comparados aos sistemas tradicionais de reconhecimento de pessoas baseados em conhecimento ou posses, apresentam maior robustez a fraudes, uma vez que o infrator tem de conseguir acesso à característica biométrica do usuário alvo para burlar um destes mecanismos de segurança, não sendo possível fraudar o sistema sem "conhecer" uma pessoa válida em sua base de dados (no caso das senhas, simples algoritmos de força bruta e técnicas heurísticas podem bastar para o criminoso), embora isto esteja mais fácil a cada dia dada a difusão da rede mundial de computadores e das redes sociais. Outrossim, não é necessário que a pessoa memorize algum código ou carregue algum pertence para que sua identificação aconteça, tornando o processo muito mais conveniente ao usuário: a "chave" é ele mesmo (JAIN; BOLLE; PANKANTI, 2006).

Na realidade, o uso de características físicas na identificação de pessoas já vem sendo realizado há bastante tempo, porém de forma manual. Na China antiga, a análise de impressões digitais já era utilizada para identificar indivíduos. Em 1900, um método de reconhecimento biométrico também baseado em impressões digitais, desenvolvido por Richard Edward Henry, foi adotado pela Scotland Yard, na Inglaterra. Vinte e quatro anos depois, o FBI (*Federal Bureau of Investigation*) norte-americano criou uma divisão para identificação de pessoas baseada em impressões digitais (JAIN et al., 2004).

Nas últimas décadas, os estudos acerca da Biometria apresentaram grande desenvolvimento. Além das impressões digitais, hoje, diversos sistemas biométricos que se valem de outras características como face, íris, voz, geometria das mãos, termograma facial e modo de andar, por exemplo, são empregados no reconhecimento de pessoas nos mais variados contextos. Centenas de companhias espalhadas pelo mundo estão envolvidas com pesquisa e desenvolvimento nesta área. Além disso, os custos dos equipamentos necessários para o reconhecimento biométrico estão cada vez mais baixos, como dito, tornando os sistemas biométricos mais acessíveis inclusive para pequenas e médias empresas (JAIN; ROSS; NANDAKUMAR, 2011). A Figura 2.1 ilustra algumas das principais características biométricas.

Nenhuma característica biométrica é ótima, ou seja, todas apresentam vantagens e desvantagens, dependendo da aplicação sob análise. Para cada contexto, uma característica biométrica pode ser mais adequada que outras (JAIN; ROSS; PRABHAKAR, 2004; JAIN et al., 2004; JAIN; BOLLE; PANKANTI, 2006). De qualquer modo, deve-se buscar empregar características que atendam ao máximo às seguintes propriedades: (i) **universalidade** - toda pessoa deve possuir a característica; (ii) **unicidade** - a característica deve ser única para cada pessoa; (iii) **permanência** - a característica não deve mudar com o passar do tempo; (iv) **coletabilidade** - a característica

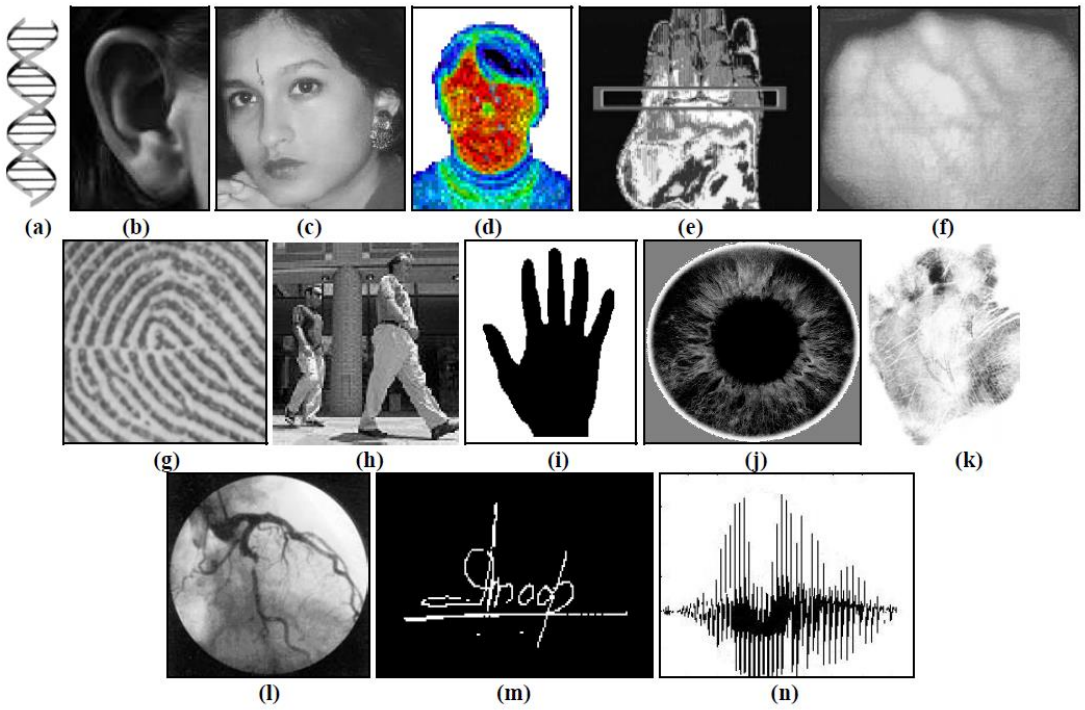


Figura 2.1: Exemplos de características biométricas físicas, fisiológicas e comportamentais: (a) DNA; (b) Forma da orelha; (c) Face; (d) Termograma facial; (e) Termograma da mão; (f) Padrão de distribuição de veias na mão; (g) Impressão digital; (h) Forma de andar; (i) Geometria da mão; (j) Padrão da íris; (k) *Palmprint*; (l) Retina; (m) Assinatura; (n) Voz. Fonte: Adaptada de Jain, Ross e Prabhakar (2004).

deve permitir ser medida quantitativamente; (v) ***performance*** - a característica deve propiciar uma identificação precisa, em tempo hábil, utilizando-se poucos recursos; (vi) **aceitabilidade** - sistemas biométricos que utilizam a característica devem ser aceitos facilmente pelos respectivos usuários; e (vii) **circunvenção** - a característica não deve permitir que o sistema biométrico seja facilmente fraudado (por forjamento de característica, por exemplo).

A Tabela 2.1 relaciona algumas das características biométricas quanto ao atendimento às propriedades citadas. O símbolo " ☺ " indica que a característica é boa em relação à respectiva propriedade, " 😐 " indica atendimento mediano e " ☹ " indica que a característica apresenta fragilidade em relação àquela propriedade.

Conforme pode-se observar na Tabela 2.1, a face apresenta bom atendimento aos requisitos de universalidade, coletabilidade e aceitabilidade. Ela é uma das características que melhor atende a tais requisitos. No tocante à universalidade, todas as pessoas a possuem, independentemente de alguma má formação ou anomalia, ao contrário das impressões digitais, por exemplo, que podem estar ausentes em certo indivíduo devido à falta dos dedos ou mesmo à realização de trabalhos manuais repetidos ou com substâncias degradantes da pele. Hoje, inclusive, surgem estudos do reconhecimento até mesmo de animais pela face, para controle e proteção ambiental,

Tabela 2.1: Análise das principais características biométricas em relação às propriedades desejáveis. Fonte: Jain, Ross e Prabhakar (2004)

Característica	Universalidade	Unicidade	Permanência	Coletabilidade	*Performance*	Aceitabilidade	Circunvenção
DNA	☺	☺	☺	☹	☺	☹	☺
Face	☺	☹	😐	☺	☹	☺	☹
Impressão Digital	😐	☺	☺	😐	☺	😐	😐
Íris	☺	☺	☺	😐	☺	☹	☺
Retina	☺	☺	😐	☹	☺	☹	☺
Palmprint	😐	☺	☺	😐	☺	😐	😐
Formato da orelha	😐	😐	☺	😐	😐	☺	😐
Geometria da mão	😐	😐	😐	☺	😐	😐	😐
Veias da mão	😐	😐	😐	😐	😐	😐	☺
Odor	☺	☺	☺	☹	☹	😐	☺
Termograma facial	☺	☺	☹	☺	😐	☺	☺
Voz	😐	☹	☹	😐	☹	☺	☹
Assinatura	☹	☹	☹	☺	☹	☺	☹
Padrão de digitação	☹	☹	☹	😐	☹	😐	😐
Modo de andar	😐	☹	☹	☺	☹	☺	😐

como no trabalho de Deb et al. (2018).

Em se tratanto da coletabilidade, a captura de imagens faciais, com qualidade suficiente para sua posterior análise e comparação, pode ser facialmente efetuada, nos dias de hoje, pela maioria das câmeras digitais existentes, presentes e acessíveis em praticamente todos os lugares e dispositivos eletrônicos, diferentemente da íris, por exemplo, que ainda requer sensores específicos e não tão abundantes para a coleta. A captura da imagem facial pode ser efetuada, atualmente, inclusive à distância e sem necessidade de muita colaboração do usuário. Na China, por exemplo, sistemas de câmeras inteligentes, posicionadas em locais públicos, já são capazes de identificar pessoas procuradas em meio a multidões de forma silenciosa (HIGA, 2018).

Em relação à aceitabilidade, principalmente devido à coleta não intrusiva como dito, as pessoas em geral não têm objeções quanto à captura de suas imagens faciais (a fim de serem identificadas). Muitas pessoas, inclusive, autocapturam sua imagem facial diversas vezes ao dia, sem nenhum incômodo, as chamadas *selfies*, para veiculá-las nas redes sociais, por exemplo. Isto não ocorre com a maioria dos sistemas biométricos baseados em impressões digitais e íris, onde, no primeiro caso, os usuários comumente se sentem desconfortáveis ao ter que tocar, em

geral, uma superfície de vidro, por medo de contaminação, ou, no segundo caso, por medo dos sensores causarem algum dano aos seus olhos, pois ainda necessita-se de certa proximidade e pelo fato de muitos sensores trabalharem com luz infravermelha.

Tudo isto torna a face muito conveniente aos usuários dos sistemas biométricos. Em relação à unicidade e permanência, é certo que as faces podem apresentar alta similaridade interclasses e sofrer grandes mudanças estruturais ou superficiais no decorrer do tempo, porém, já há alguns anos, algoritmos são capazes de atingir taxas de acerto próximas às dos sistemas de reconhecimento baseados em impressões digitais, como no trabalho de Parkhi, Vedaldi e Zisserman (2015). Estudos sobre o envelhecimento facial também são um ramo de pesquisa em pleno desenvolvimento (HAN; OTTO; JAIN, 2013; HAN et al., 2015; DEB; BEST-ROWDEN; JAIN, 2017; WANG et al., 2018). Em termos de *performance*, desde a proposta do método das autofaces (SIROVICH; KIRBY, 1987; TURK; PENTLAND, 1991), com a redução da dimensionalidade dos vetores de características por meio da transformada *Principal Component Analysis* (PCA) (PEARSON, 1901; HOTELLING, 1933), os sistemas de reconhecimento facial têm se tornardo mais eficientes.

O último quesito, a circunvenção, é provavelmente o mais crítico em sistemas de reconhecimento facial pois, em geral, simples fotografias impressas podem enganar a maioria das câmeras de captura. É neste ponto que sistemas anti-*spoofing* se tornam de grande valia. Dada a fácil circunvenção dos sistemas de reconhecimento facial, mesmo sendo bastante convenientes aos usuários, tais aplicações, sem métodos de contramedida a ataques, não seriam viáveis em situações práticas.

2.2 Ataques a Sistemas Biométricos

Na realidade, assim como ocorre com outros tipos de sistemas, existem muitas formas de se atacar um sistema biométrico tradicional (incluindo os de reconhecimento facial). Segundo Ratha, Connell e Bolle (2001), nestes sistemas existem 8 diferentes pontos que podem ser explorados durante uma tentativa de fraude. A Figura 2.2 ilustra tais pontos.

Basicamente, os ataques podem ser divididos em dois grandes grupos: ataques diretos (ponto "1" da Figura 2.2) e indiretos (pontos de "2" a "8"). Nos ataques diretos, tema desta tese, os fraudadores, em geral, produzem amostras sintéticas de características biométricas de usuários válidos, tais como fotografias impressas ou vídeos exibidos em *displays* de dispositivos móveis, no caso da face, a fim de obter acesso ao sistema. Os criminosos tentam enganar, com tais amostras, o sensor de captura, ponto mais vulnerável do sistema de reconhecimento biométrico (RATHA; CONNELL; BOLLE, 2001). Como dito, estes ataques são também

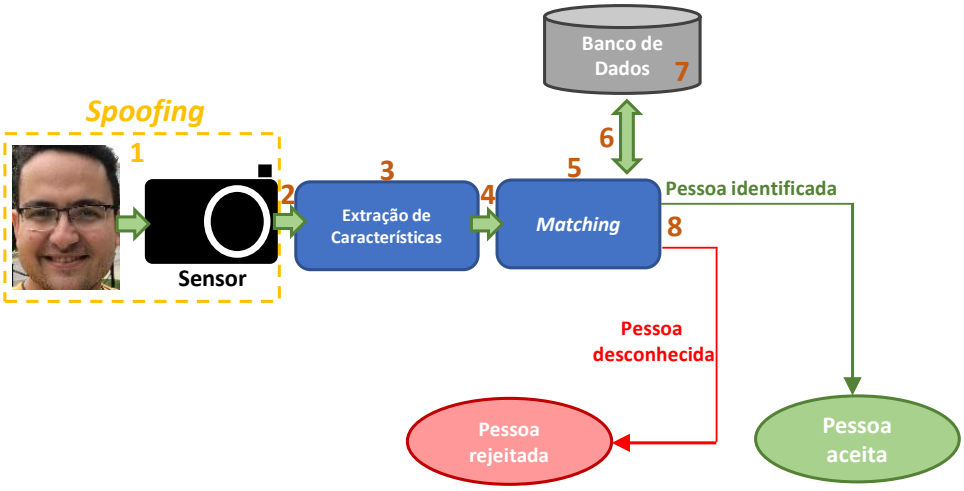


Figura 2.2: Diferentes pontos de ataque que podem ser explorados nos sistemas biométricos tradicionais: do sensor à decisão final. Fonte: Adaptada de Ratha, Connell e Bolle (2001).

denominados *spoofing*, ou ainda, ataques de apresentação, e, no caso de sistemas de reconhecimento facial, podem ser mais facilmente executados do que na maioria dos demais tipos de aplicações biométricas (PINTO et al., 2012). Nos ataques indiretos, os criminosos, após descobrirem informações sobre o funcionamento interno do sistema e baseados em alguma fragilidade, atuam modificando informações, seja da base de dados, dos algoritmos utilizados ou de mensagens trocadas internamente pelos módulos da aplicação (RATHA; CONNELL; BOLLE, 2001).

A vantagem dos ataques diretos para os infratores reside no fato de não ser necessário conhecer o funcionamento do sistema - seus algoritmos de extração de características, métodos de comparação e casamento de vetores, estrutura da base de dados, etc. (RATHA; CONNELL; BOLLE, 2001; GALBALLY; FIERREZ; GARCIA, 2007). Devido a esta facilidade, os ataques diretos correspondem à grande maioria das tentativas de fraudes registradas e, para detectar atividades mal intencionadas como estas, vêm sendo desenvolvidos variados métodos anti-*spoofing* baseados em diferentes princípios e técnicas. Entretanto, apesar dos avanços, a prevenção de tais fraudes ainda se apresenta com uma questão em aberto (PINTO et al., 2012; GALBALLY et al., 2014; PATEL; HAN; JAIN, 2016a).

Vale ressaltar que existem quatro categorias principais de métodos anti-*spoofing*: (i) métodos orientados a dados; (ii) métodos orientados ao modelamento do comportamento do usuário; (iii) métodos que requerem interação; e (iv) métodos que fazem uso de sensores adicionais (JAIN; FLYNN; ROSS, 2008; PINTO et al., 2012). As abordagens eficientes de anti-*spoofing* facial baseadas em Aprendizado de Máquina em Profundidade propostas nesta tese estão contempladas na primeira categoria: métodos orientados a dados. Segundo Menotti et al. (2015), como tais técnicas se baseiam em informações extraídas da própria característica biométrica sendo captu-

rada para o processo de reconhecimento, elas são mais convenientes uma vez que não necessita-se de novos sensores ou de comportamentos adicionais específicos dos usuários, diferentemente das demais categorias de métodos anti-*spoofing*.

Os métodos orientados a dados, em geral, operam extraindo medidas bastante variadas, tais como informações de textura, de cor, de movimento, etc., da característica biométrica sob análise. Tais medidas são apresentadas então a um classificador, que atua, dada uma tentativa de acesso ao sistema biométrico, rotulando-a como ataque ou não. No caso da detecção de *spoofing* facial, as medidas são extraídas das próprias imagens e vídeos faciais utilizados para o reconhecimento do indivíduo.

Como mencionado, nesta tese trata-se dos ataques de apresentação facial utilizando faces sintéticas em mídias 2D, isto é, faces impressas em papéis (*print attacks*) ou exibidas (às vezes na forma de vídeo) em *displays* de dispositivos móveis (*replay attacks*). Apesar de já ser possível fabricar máscaras 3D personalizadas e bastante realistas, o custo ainda é elevado além da confecção ser bastante complexa.

Pode-se agrupar os métodos orientados a dados para anti-*spoofing* facial 2D em três principais categorias, dependendo do tipo de medidas que extraem a partir das imagens das faces: (i) métodos que detectam a presença de vida, os quais tentam identificar padrões que somente faces reais de pessoas vivas apresentam, como mudanças na coloração da pele devido à circulação do sangue ou o piscar de olhos, movimento intrínseco ao globo ocular; (ii) técnicas que comparam características da face e do *background*, como a movimentação relativa da face em comparação ao plano de fundo da imagem, já que faces sintéticas 2D tendem a apresentar mesma movimentação que o cenário impresso ao seu redor ao se mover o papel ou *display* apresentado à câmera; e (iii) métodos que analisam a qualidade das imagens, por meio de descritores de cor e textura, por exemplo, uma vez que a distorções são encontradas na recaptura de imagens faciais sintéticas (PEREIRA, 2013).

Vale ressaltar que outros autores classificam os métodos orientados a dados de detecção de *spoofing* facial 2D de forma ligeiramente diferente. Patel, Han e Jain (2016b), por exemplo, categorizam as abordagens em seis classes: (i) métodos baseados na análise de movimentos faciais, como o piscar, movimentos da boca, entre outros; (ii) métodos baseados em textura; (iii) métodos que analisam a profundidade da face, por exemplo, estimando-a a partir de várias imagens 2D; (iv) abordagens que analisam a qualidade das imagens; (v) métodos baseados em domínios de frequência, que buscam ruídos gerados na recaptura das faces sintéticas pelo sensor do sistema; e (vi) métodos baseados na fusão de múltiplas destas características. Atoum et al. (2017) também mencionam brevemente três categorias de métodos anti-*spoofing* 2D para

sistemas de reconhecimento facial: (i) abordagens baseadas em texturas e cores; (ii) métodos baseados em movimentos; e (iii) métodos baseados em características profundas, que se valem de redes neurais em profundidade.

Capítulo 3

APRENDIZADO DE MÁQUINA EM PROFUNDIDADE

Este capítulo apresenta definições referentes às duas arquiteturas de redes neurais profundas tomadas como base para a proposta de novas abordagens eficientes de detecção de *spoofing* facial nesta tese: as Redes Neurais de Convolução (*Convolutional Neural Networks* - CNN) (LECUN et al., 1998), as quais são baseadas em camadas de convolução e amostragem, e as redes baseadas nas Máquinas de Boltzmann Restritas (*Restricted Boltzmann Machines* - RBM) (SMOLENSKY, 1986; HINTON, 2002), redes neurais estocásticas que trabalham com função de energia. Ao final são apresentadas redes neurais que trabalham com características temporais como a C3D (TRAN et al., 2015) e as redes LSTM (*Long-Short Term Memory*) (HOCHREITER; SCHMIDHUBER, 1997), também empregadas nesta tese.

3.1 Motivação do Aprendizado em Profundidade

Resultados teóricos indicam que a extração de características de alto nível a partir de um conjunto de dados, essenciais a sistemas que lidam com tarefas complexas nas áreas de Visão Computacional, Processamento de Linguagem Natural, dentre outras, requer arquiteturas profundas, isto é, multicamadas (LECUN; BENGIO; HINTON, 2015). Tais arquiteturas são compostas por múltiplas camadas de estruturas que realizam operações pré-definidas sobre o conjunto de dados inicial. Os resultados obtidos por uma camada inferior alimentam a próxima camada. Deste modo, aos dados originais vão sendo aplicadas uma repetição de operações até se obter características mais complexas, de alto nível, em geral bastante discriminativas e robustas a ruídos e distorções nos dados originais. A busca no espaço de hiperparâmetros das arquiteturas profundas pode ser complexa, mas métodos eficazes de inicialização e treinamento de tais estruturas têm sido propostos recentemente, os quais lidam com este problema com êxito, obtendo modelos que permitem melhores resultados que técnicas consideradas estado-da-arte em

muitas áreas (DENG; DONG, 2014; LECUN; BENGIO; HINTON, 2015). Este aprendizado baseado em camadas é conhecido como Aprendizado de Máquina em Profundidade ou ainda, do inglês, *Deep Learning*.

Segundo Bengio (2013), nos últimos anos, os emergentes algoritmos de Aprendizado de Máquina em Profundidade têm levado os sistemas de Aprendizado de Máquina tradicionais à descoberta de múltiplos níveis de representação de características, como mostrado na Figura 3.1, propiciando alcançar superiores resultados. O Aprendizado de Máquina em Profundidade tem atraído a atenção da comunidade científica e industrial do mundo todo. Empresas como Google, Facebook, Microsoft, Apple, IBM, dentre outras, têm investido em métodos de aprendizado multicamadas. A Google, por exemplo, utiliza técnicas de Aprendizado em Profundidade em seus *softwares* de identificação de objetos e busca de imagens (*Google Image Search*). Ela inclusive está aprimorando técnicas para classificação de vídeos baseadas em características profundas deles extraídas (ALECRIM, 2016). O Facebook, por sua vez, utiliza redes neurais profundas já há alguns anos para reconhecer as pessoas presentes em imagens pela face a fim de, por exemplo, facilitar a marcação de amigos nas fotografias. A empresa também já tem estudos relacionados ao uso das redes profundas para o reconhecimento de pessoas em imagens digitais mesmo quando suas faces estão oclusas, identificando-as por meio de suas aparências corporais (BBC, 2015; UOL, 2018).

3.2 Redes Neurais de Convolução (CNN)

As Redes Neurais de Convolução (CNN, do inglês *Convolutional Neural Networks*) (LECUN et al., 1998) são estruturas de aprendizado em profundidade muito utilizadas nos dias de hoje, em especial para se trabalhar com sinais 2D, particularmente imagens. Uma CNN é constituída por uma ou mais camadas onde espécies de filtros (operações de convolução e amostragem) são aplicados aos dados de entrada (SIMARD; STEINKRAUS; PLATT, 2003; CHELLAPILLA; PURI; SIMARD, 2006). O resultado de uma camada inferior serve de entrada para a camada imediatamente superior. Em contraste com as redes neurais completamente conectadas, de complexa construção e uso, as CNNs compreendem redes de topologia simplificada, com camadas de convolução e amostragem, como dito, e camadas opcionais no topo, constituídas de nós completamente conectados (CHELLAPILLA; PURI; SIMARD, 2006) como nas tradicionais redes MLP - *Multilayer Perceptron* (RUMELHART; HINTON; WILLIANS, 1986).

Dada uma imagem de entrada $\mathbf{I}$ com pixels $p = (p_x, p_y)$, em cada camada da rede, um conjunto de n filtros (*kernels*) de convolução 2D, denotados por $\mathbf{K}_i$, com $i = 1, 2, ..., n$, pode ser

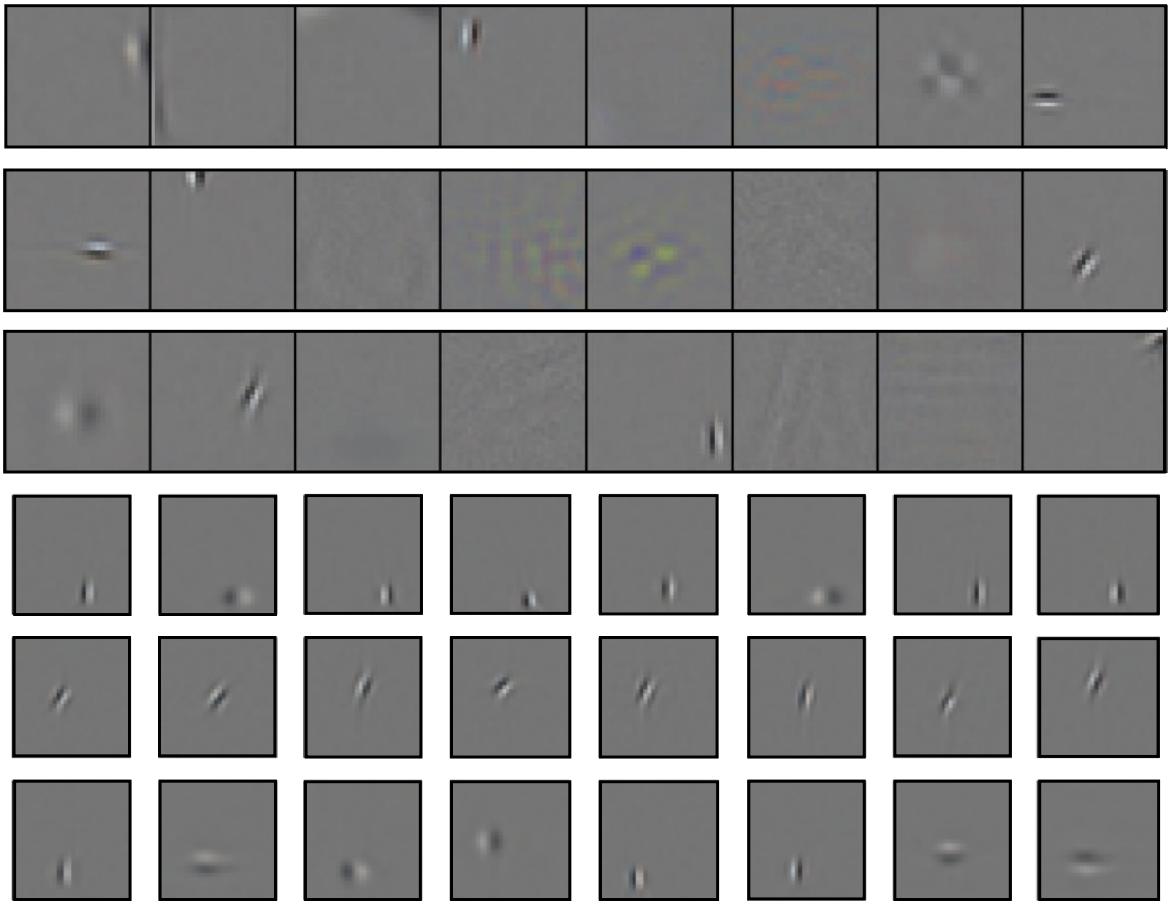


Figura 3.1: Filtros (pesos de conexões entre neurônios) aprendidos durante o treinamento de uma rede neural baseada nas Máquinas de Boltzmann Restritas na primeira (imagens maiores) e segunda (imagens menores) camadas. Diferentes e mais discriminativas características são capturadas a cada camada. Fonte: Adaptada de Krizhevsky (2009).

aplicado de modo a obter-se bandas $\mathbf{C}_i$, $i = 1,2,...,n$, da imagem original (também chamadas de *feature maps*):

$$\mathbf{C}_i(p) = \sum_{k_x=-2}^{2} \sum_{k_y=-2}^{2} \mathbf{K}_i(k_x,k_y) \cdot \mathbf{I}(p_x - k_x, p_y - k_y) \tag{3.1}$$

considerando-se, neste caso, filtros de dimensões 5×5 (índices k_x e k_y variando de -2 a 2).

Os pesos dos filtros podem ser vistos como pesos de conexões sinápticas entre neurônios de camadas adjacentes. Os neurônios responsáveis pelos valores de um mesmo *feature map* de saída de uma camada compartilham os mesmos filtros (mesmos *kernels*), ou seja, os mesmos pesos, em relação aos neurônios geradores dos *feature maps* da camada anterior. Na realidade, o resultado das operações de convolução, isto é, seus *feature maps* de saída, pode passar ainda por funções de ativação (ativações dos neurônios geradores dos *feature maps*), como a de retificação linear (*Rectified Linear Unit* - ReLU), dada por:

$$\mathbf{C}_i(p) = max\{0; \mathbf{C}_i(p)\} \tag{3.2}$$

Mais operações podem ser agregadas em cada camada a fim de melhorar o desempenho da CNN, por exemplo, funções de normalização dos valores dos *feature maps* de saída, como a *Local Response Normalization* (LRN) (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) e a *Batch Normalization* (BN) (IOFFE; SZEGEDY, 2015). A função LRN visa simular a competição por ativação que ocorre entre neurônios vizinhos do cérebro humano nos neurônios artificiais, em posições correspondentes em *feature maps* adjacentes, e é dada por:

$$\mathbf{C}_i(p) = \frac{\mathbf{C}_i(p)}{(1 + \alpha \sum_{j=i-t}^{i+t} \mathbf{C}_j(p)^2)^\beta} \tag{3.3}$$

onde t corresponde à quantidade de *feature maps* anteriores e posteriores a serem considerados como vizinhos do atual; α e β são parâmetros que controlam a magnitude da normalização; α é geralmente tido como $\frac{1}{2 \cdot t + 1}$ e β é geralmente tomado como $0,75$. Aplica-se tal equação para cada *feature map* C_i de saída da camada sob análise. Quando não há *feature maps* anteriores ou posteriores suficientes, *feature maps* com todos os valores em zero são virtualmente considerados para o cálculo da equação.

Já a função de normalização BN serve para normalizar os dados de saída de uma camada da rede para cada *batch* de dados de entrada, de maneira análoga à normalização que geralmente se faz nas amostras da base de dados sob análise antes de alimentar a rede neural, centralizando-as na origem do espaço de características. Em geral subtrai-se, dos dados de saída da camada, o valor de saída médio, e divide-se os mesmos por seu desvio-padrão. Deste modo, os dados gerados pela camada tendem a ficar centrados na origem e com dispersão dada por desvio-padrão (e variância) unitário.

Após a normalização e ativação (em geral, nesta ordem) das saídas de uma camada da rede (seus *feature maps*), operações de *pooling* (amostragem) espacial também são realizadas sobre tais estruturas a fim de se agregar maior invariância translacional e de escala. Em geral, emprega-se a operação de *pooling* máximo. Percorre-se cada *feature map* de saída $\mathbf{C}_i$ com uma matriz quadrada $\mathbf{\Lambda}$, cujas dimensões podem variar, bem como o passo dado de um posicionamento para outro. Em cada posição da matriz sobre a imagem, toma-se o valor máximo dentre os elementos sob a matriz como novo elemento de saída:

$$\mathbf{C}_i^{out}(q) = max_{\forall p \in \mathbf{\Lambda}_c}\{\mathbf{C}_i(p)\} \tag{3.4}$$

onde $\mathbf{\Lambda}_c$ corresponde ao posicionamento da matriz de amostragem $\mathbf{\Lambda}$ centrada na posição $c = (c_x, c_y)$ sobre o *feature map* $\mathbf{C}_i$; p representa todas as posições do *feature map* $\mathbf{C}_i$ sob a matriz $\mathbf{\Lambda}$ no momento; e q o novo elemento de saída (no novo *feature map* $\mathbf{C}_i^{out}$, obtido de $\mathbf{C}_i$).

Os resultados (estruturas de dados geradas, isto é, *feature maps* finais) são passados para camadas superiores da rede. Ao final, tem-se uma representação de alto nível dos dados originais, codificada em um vetor de características numérico (CHELLAPILLA; PURI; SIMARD, 2006; PINTO et al., 2009; BERGSTRA; YAMINS; COX, 2013; MENOTTI et al., 2015). Tal representação pode então alimentar um classificador a fim de identificar o padrão sob análise.

Vale ressaltar, porém, que muitas vezes a própria rede CNN pode ser transformada em um classificador ao acoplar-se camadas completamente conectadas em seu topo, tendo a última delas neurônios especiais, como unidades *softmax*, por exemplo. Basicamente tais unidades servem para indicar a qual classe pertence um sinal de entrada e, no caso de unidades *softmax*, seguem a função de ativação:

$$s_{max}(u_i) = \frac{e^{u_i}}{\sum_{j=1}^{n} e^{u_j}} \tag{3.5}$$

onde u_i corresponde à soma ponderada dos sinais de entrada no neurônio *softmax i* da camada de classificação da rede e n corresponde à quantidade de neurônios *softmax* existentes nesta camada (classes do problema sendo tratado). Devido ao denominador ser uma função de partição, a saída dos neurônios *softmax* estará sempre no intervalo $[0;1]$ e o neurônio cuja ativação for a máxima indica a classe do sinal de entrada.

A Figura 3.2 ilustra um exemplo de arquitetura CNN de duas camadas, tendo ao topo outras duas camadas completamente conectadas bem como uma camada de classificação, com 10 neurônios *softmax*. Conforme pode-se observar, dada uma imagem inicial, operações de convolução e amostragem são aplicadas seguidas por camadas de nós completamente conectados, obtendo-se deste modo um vetor de característica reduzido de alto nível para a mesma e a classificação da imagem de entrada na última camada (as dimensões e quantidades de *feature maps* gerados após cada operação da rede são indicadas na parte superior da figura).

Conforme mencionado por LeCun et al. (1998) e Simard, Steinkraus e Platt (2003), as redes convolucionais apresentam grande robustez a distorções, variações na escala e translação dos objetos nas imagens, justamente devido às operações com que trabalham. Os neurônios que respondem às operações de convolução, por exemplo, por estarem espacialmente distribuídos e compartilharem os mesmos filtros de convolução na geração de cada *feature map*, como mencionado, acabam detectando as características importantes na imagem mesmo que estas estejam

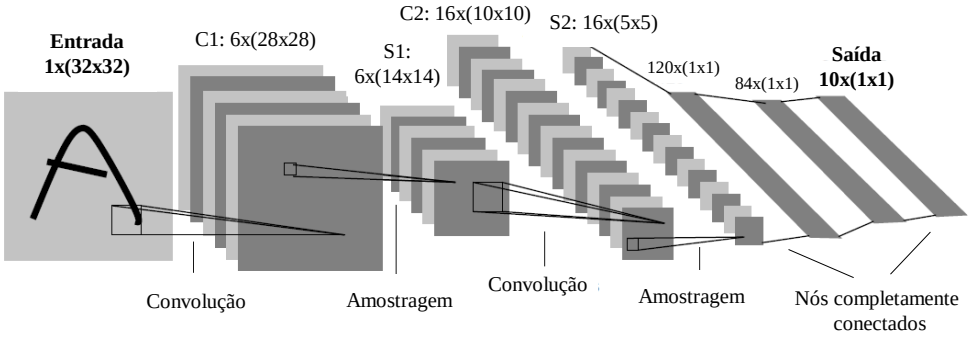


Figura 3.2: Exemplo de arquitetura de CNN. Fonte: Adaptada de LeCun et al. (1998).

deslocadas (a característica pode ficar transladada, mas continua presente no respectivo *feature map*). Os nós que respondem às operações de *pooling* espacial acabam por atenuar diferenças de escala e pequenas distorções nos objetos, preservando apenas os aspectos principais dos mesmos e o posicionamento relativo das características detectadas na operação de convolução.

Como em outras redes neurais, as CNNs também aprendem por *backpropagation* (LECUN et al., 1998). Os pesos dos filtros de convolução, como mencionado, podem ser vistos como os pesos sinápticos entre os neurônios das camadas da rede, os quais são inicializados e atualizados durante a aprendizagem. Menotti et al. (2015), por exemplo, inicializam tais pesos com valores amostrados de uma distribuição uniforme. Entretanto, em geral, utiliza-se uma distribuição normal, com média zero e desvio-padrão baixo, como no trabalho de Krizhevsky, Sutskever e Hinton (2012), de forma a evitar problemas com conexões de pesos muito elevados e dominantes (típicos de *overfitting*). Vale ressaltar que uma grande vantagem das CNNs em relação às redes tradicionais completamente conectadas reside no fato de que, por os neurônios geradores de cada *feature map* em cada camada compartilharem os mesmos pesos sinápticos (pesos dos filtros) em relação à camada anterior, como dito, a quantidade de parâmetros livres do modelo fica menor, tornando possível o treinamento mesmo com menos amostras do problema.

3.3 Redes Baseadas nas Máquinas de Boltzmann

As Máquinas de Boltzmann (BM, do inglês *Boltzmann Machines*) (HINTON; SEJNOWSKI, 1983) correspondem a redes neurais baseadas em energia compostas de unidades de processamento estocásticas, similares aos modelos de redes neurais conhecidos. Elas podem ser usadas no aprendizado de importantes aspectos de distribuições de probabilidade, a partir de amostras destas distribuições (ACKLEY; HINTON; SEJNOWSKI, 1985).

Por permitirem a existência de ligações entre todos os nós da rede, as BMs apresentam al-

goritmo de aprendizado complexo e bastante custoso, computacionalmente, sendo muitas vezes inviável. Entretanto, o aprendizado pode ser bastante facilitado ao se impor restrições em sua topologia, dando-se origem assim às chamadas Máquinas de Boltzmann Restritas (*Restricted Boltzmann Machines* - RBM) (SMOLENSKY, 1986; HINTON, 2002). Diferentemente das CNNs, as RBMs são, por natureza, modelos generativos de redes neurais. As subseções a seguir descrevem as RBMs e as redes profundas delas derivadas.

3.3.1 Máquinas de Boltzmann Restritas (RBM)

Uma RBM (*Restricted Boltzmann Machine*) (SMOLENSKY, 1986; HINTON, 2002) corresponde a um Campo Aleatório de Markov (MRF, do inglês, *Markov Random Field*) associado a um grafo não-direcionado bipartido. As RBMs podem ser vistas também como redes neurais de unidades estocásticas baseadas em energia. Os vértices (neurônios) se dispõem em duas camadas, uma visível e outra escondida, com arestas (sinapses) apenas entre vértices de camadas diferentes (TANG; SALAKHUTDINOV; HINTON, 2012; HINTON, 2012). Como as BMs, estas máquinas também podem ser utilizadas no aprendizado de aspectos de distribuições de probabilidade, de maneira mais limitada, mas ainda bastante eficaz e mais eficiente.

Os neurônios da camada visível são responsáveis pela observação, isto é, captura de informações do ambiente: por exemplo, pode-se ter um neurônio associado a cada pixel da imagem sob análise, isto é, extraindo características do pixel. Já na camada escondida, os neurônios modelam dependências e relações entre os vértices da primeira camada (por exemplo, dependências entre pixels da imagem sob análise). A Figura 3.3 ilustra a arquitetura de uma RBM. Pode-se observar as duas camadas (visível e escondida) e as ligações intercamadas apenas.

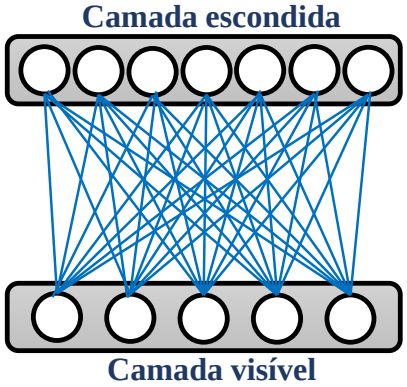


Figura 3.3: Exemplo de arquitetura de uma RBM. Fonte: Elaborada pelo autor.

Assumindo-se os neurônios da camada visível $\mathbf{v}$ e escondida $\mathbf{h}$ como unidades binárias, isto é, $\mathbf{v} \in \{0,1\}^m$ e $\mathbf{h} \in \{0,1\}^n$, onde m e n correspondem à quantidade de neurônios na camada visível e na escondida, respectivamente, tem-se a tradicional BB-RBM (*Bernoulli-Bernoulli*

Restricted Boltzmann Machine), uma vez que os neurônios seguem a distribuição de Bernoulli. A função de energia de uma BB-RBM é dada por:

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} v_i h_j w_{ij} \tag{3.6}$$

onde $\mathbf{a}$ e $\mathbf{b}$ correspondem aos *biases* da camada visível e escondida, respectivamente, e w_{ij} corresponde ao peso da conexão entre os neurônios i da camada visível e j da camada escondida.

A probabilidade da rede se encontrar em uma configuração $(\mathbf{v},\mathbf{h})$ é dada por:

$$P(\mathbf{v},\mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})} \tag{3.7}$$

onde Z corresponde à função de partição, isto é, um fator de normalização calculado com base em todas as configurações possíveis das unidades da camada visível e escondida. De maneira similar, a probabilidade marginal de uma configuração da camada visível é dada por:

$$P(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{3.8}$$

Uma vez que uma BB-RBM é um grafo bipartido, as ativações dos neurônios da camada visível e dos neurônios da camada escondida são mutuamente independentes, obtendo-se desta forma as seguintes probabilidades condicionais:

$$P(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{m} P(v_i|\mathbf{h}) \tag{3.9}$$

e

$$P(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{n} P(h_j|\mathbf{v}) \tag{3.10}$$

onde:

$$P(v_i = 1|\mathbf{h}) = \sigma\left(\sum_{j=1}^{n} w_{ij} h_j + a_i\right) \tag{3.11}$$

e

$$P(h_j = 1|\mathbf{v}) = \sigma\left(\sum_{i=1}^{m} w_{ij} v_i + b_j\right) \tag{3.12}$$

onde $\sigma(\cdot)$ corresponde nestes casos à função sigmoidal.

Seja $\boldsymbol{\theta} = (\mathbf{W}, \mathbf{a}, \mathbf{b})$ o conjunto de parâmetros de uma BB-RBM, eles são aprendidos por meio de um algoritmo de treinamento que visa maximizar o produtório das probabilidades de ocorrência de todos os dados (vetores) de treinamento $\mathscr{V}$, conforme segue:

$$\arg\max_{\boldsymbol{\theta}} \prod_{\mathbf{v} \in \mathscr{V}} P(\mathbf{v}) \tag{3.13}$$

Uma das abordagens mais empregadas para resolver este problema se dá por meio do método denominado Divergência Contrastiva (*Contrastive Divergence* - CD) (HINTON, 2002), o qual, em suma, simula o processo de amostragem de Gibbs (GEMAN; GEMAN, 1984) para a convergência da rede porém em uma única iteração, inicializando as unidades visíveis com um vetor de treinamento.

Vale ressaltar que, na presença de dados reais, como quando se trabalha com imagens em tons de cinza, deve-se empregar outro tipo de RBM, a chamada Gaussian-Bernoulli RBM (GB-RBM) (NAIR; HINTON, 2014), a qual modela o vetor de entrada por meio de unidades que seguem distribuição Gaussiana. Deste modo, a Equação 3.6 deve ser reescrita como:

$$E(\mathbf{v}, \mathbf{h}) = \frac{1}{2} \sum_{i=1}^{m} \frac{(v_i - a_i)^2}{\sigma_i^2} - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{v_i}{\sigma_i} h_j w_{ij} \tag{3.14}$$

Dada a modificação nas unidade visíveis, é necessário reformular suas probabilidades condicionais. Deste modo, a Equação 3.11 deve ser reescrita como:

$$P(v_i = 1 | \mathbf{h}) = \mathscr{N}\left(v_i \,\middle|\, \sum_{j=1}^{n} w_{ij} h_j + a_i; \sigma_i^2\right) \tag{3.15}$$

onde, em ambas as equações, σ^2 corresponde à variância da distribuição Gaussiana $\mathscr{N}$.

As RBMs têm sido muito estudadas uma vez que podem ser utilizadas em arquiteturas de Aprendizado em Profundidade como as Redes de Crença em Profundidade (do inglês, *Deep Belief Networks* - DBN) (HINTON; OSINDERO; TEH, 2006) e as Máquinas de Boltzmann em Profundidade (*Deep Boltzmann Machines* - DBM) (SALAKHUTDINOV; HINTON, 2009). Como na composição das camadas das redes CNNs, pode-se combinar, na forma de uma pilha, diversas RBMs, a fim de se obter representações em profundidade dos dados inicialmente apresentados à rede neural, na tentativa de aproximar seus resultados aos das Máquinas de Boltzmann originais.

Nestas arquiteturas, os neurônios da camada escondida da primeira RBM extraem características relevantes dos dados observados pelos neurônios de sua camada visível. Tais características servem de entrada para outra RBM, que extrai novas propriedades das características

obtidas anteriormente e as repassa a uma terceira RBM, e assim por diante, a fim de se obter representação de alto nível dos dados de entrada da primeira máquina (FISCHER; IGEL, 2012). Deste modo, este "empilhamento" de RBMs pode ser visto como uma rede neural *feedforward*, como funções que mapeiam observações para características aprendidas de alto nível. Pode-se também acrescentar uma última camada no topo da estrutura de forma a torná-la uma rede neural para classificação ou regressão (FISCHER; IGEL, 2012).

3.3.2 Redes de Crença em Profundidade (DBN)

As RBMs podem ser usadas para muitas tarefas, como a reconstrução de um dado sinal corrompido com ruídos (eliminação de informação indesejada) ou mesmo de um sinal com valores faltantes. Entretanto, a fim de aprender uma representação mais complexa e robusta dos dados, uma estrutura mais profunda faz-se necessária. Neste contexto e baseando-se na arquitetura das RBMs, as Redes de Crença em Profundidade (do inglês, DBN - *Deep Belief Networks*) (HINTON; OSINDERO; TEH, 2006) foram propostas, na tentativa de se aprender características de mais alto nível a partir das amostras de entrada e suas distribuições.

Basicamente, uma DBN consiste em uma pilha de RBMs (cada camada da DBN é composta por uma RBM) e, em geral, seu treinamento é realizado camada a camada, da base ao topo, considerando a camada escondida da RBM inferior (já treinada) como a camada visível da superior imediata (a treinar). Depois de encontrar os valores finais para os pesos e *biases* de todas as RBMs, a rede pode ser usada para eliminar ruído, realizar o *inpainting*, ou até mesmo para extrair características de alto nível (baseadas na camada escondida da última RBM), de forma muito mais acurada que quando se trabalhando com uma única RBM isolada. Isto ocorre porque cada RBM na pilha é independente das demais e o processo de treinamento isolado de cada uma delas gera probabilidades *a posteriori* complementares sobre os dados de entrada, aumentando o poder da estrutura.

Após o treinamento da DBN, é possível reconstruir sinais de entrada (como nas RBMs) realizando-se um *forward* e *backward pass* dos valores na rede, bem como adicionar uma camada *softmax* ao seu topo de forma a transformar o modelo em uma MLP (*Multilayer Perceptron*) (RUMELHART; HINTON; WILLIANS, 1986) completa para classificação. A Figura 3.4 mostra um esquema de arquitetura de uma DBN. As camadas de neurônios da rede são denotadas por $\mathbf{v}$ (camada visível), $\mathbf{h_1}$, $\mathbf{h_2}$ e $\mathbf{h_3}$ (camadas escondidas). Os conjuntos de pesos das conexões entre os neurônios de duas camadas adjacentes são denotados, na Figura 3.4, por $\mathbf{W}_l$ com $l = 1,2,3$ (rede com 3 camadas). As letras $\mathbf{a}$ e $\mathbf{b}$ representam os *biases* dos neurônios das RBMs empilhadas (apesar de constarem apenas na últimada camada da figura, cada RBM da pilha tem seus

próprios *biases*).

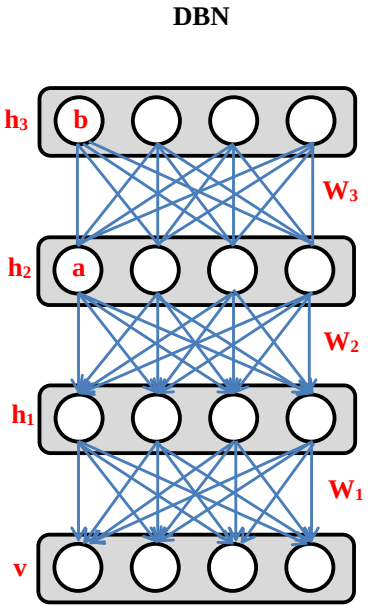


Figura 3.4: Exemplo ilustrativo de uma DBN. A última camada corresponde a uma RBM com conexões não direcionadas. Fonte: Elaborada pelo autor.

3.3.3 Máquinas de Boltzmann em Profundidade (DBM)

O modelo de uma Máquina de Boltzmann em Profundidade (DBM - *Deep Boltzmann Machine*) foi proposto por Salakhutdinov e Hinton (2009) na tentativa de aproximar o processo de aprendizagem ao das Máquinas de Boltzmann originais (completamente conectadas). Na DBM, o processo de aprendizagem de uma dada RBM da pilha considera informações em ambos os sentidos (camada superior e camada inferior), como mostrado na Figura 3.5. Como pode-se observar, quando se está analisando uma dada camada da rede, sua camada superior é considerada como de inferência complementar, sua camada inferior como de inferência inicial e a camada do meio sob análise como inferência final, no estado de equilíbrio.

Vale observar que, diferentemente de outros métodos de Aprendizado de Máquina, as DBMs (bem como RBMs e DBNs) permitem que se trabalhe com menos dados rotulados na tarefa de classificação, não degradando significativamente seus desempenhos e sendo esta uma de suas principais vantagens em relação a outros modelos (além da eficiência computacional). Como dito para as DBNs, pode-se acrescentar uma camada de classificação completamente conectada ao topo da DBM contendo neurônios *softmax* e formando uma MLP. O treinamento inicial da DBM (assim como da DBN) pode ser efetuado com amostras não rotuladas e somente após incluir a camada de classificação, em uma etapa final de ajuste fino dos parâmetros da MLP

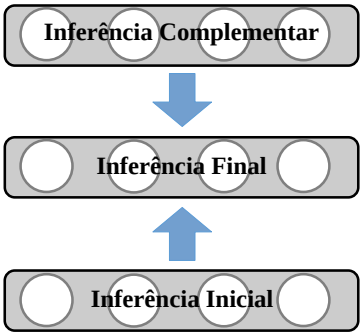


Figura 3.5: Ilustração do processo de aprendizagem na DBM, considerando ambas as camadas adjacentes no treinamento da camada atual. Fonte: Elaborada pelo autor.

formada, menor quantidade de dados rotulados faz-se necessária.

Salakhutdinov e Hinton (2012) propuseram o uso de um método denominado *Mean-Field* (MF) para tornar mais eficiente o aprendizado da DBM, que pode se tornar bastante complexo dadas as interações entre camadas da rede em ambos os sentidos. Basicamente, por meio desta técnica inspirada na física estatística, as probabilidades *a posteriori* são estimadas considerando as interações em ambos os sentidos com base em inferências parciais por meio de variáveis especiais, denominadas de *mean-field* (MACKAY, 2003).

Em termos gerais, a ideia é encontrar uma aproximação $Q^{MF}(\mathbf{h}|\mathbf{v};\boldsymbol{\mu})$ que melhor representa a distribuição das camadas escondidas da DBM, ou seja, $P(\mathbf{h}|\mathbf{v};\boldsymbol{\theta})$. Tal aproximação é calculada por:

$$Q^{MF}(\mathbf{h}|\mathbf{v};\boldsymbol{\mu}) = \prod_{l=1}^{L}\left[\prod_{k=1}^{F_l} q(h_k^l)\right] \tag{3.16}$$

onde L indica o número de camadas escondidas na DBM, F_l representa o número de nós na camada escondida l, e $q(h_k^l = 1) = \mu_k^l$. O objetivo é encontrar os parâmetros de *mean-field* $\boldsymbol{\mu} = \{\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, ..., \boldsymbol{\mu}^L\}$ de acordo com as seguintes equações:

$$\mu_k^1 = \sigma\left(\sum_{i=1}^{m} w_{ik}^1 v_i + \sum_{j=1}^{F_2} w_{kj}^2 \mu_j^2\right) \tag{3.17}$$

a qual representa a interação entre a primeira camada escondida da DBM e sua camada anterior (vísivel) e posterior, e onde $\sigma(\cdot)$ corresponde à função sigmoidal. Do mesmo modo, as interações entre as camadas escondidas l, $l-1$ e $l+1$ são dadas por:

$$\mu_k^l = \sigma\left(\sum_{i=1}^{F_{l-1}} w_{ik}^l \mu_i^{l-1} + \sum_{j=1}^{F_{l+1}} w_{kj}^{l+1} \mu_j^{l+1}\right) \tag{3.18}$$

onde w_{ij}^{l} corresponde ao peso entre os neurônios i da camada escondida $l-1$ e o neurônio j da camada escondida l.

Por fim, os parâmetros de *mean-field* referentes à camada escondida no topo da DBM são calculados por:

$$\mu_k^L = \sigma\left(\sum_{i=1}^{F_{L-1}} w_{ik}^L \mu_i^{L-1}\right) \tag{3.19}$$

representando a interação entre as últimas duas camadas da rede.

Vale observar que o treinamento das DBMs funciona basicamente em duas etapas: (i) *Greedy Pre-training*; e (ii) *Mean-Field Training*. Na primeira fase os parâmetros da rede são inicializados em um treinamento guloso *bottom-up*, isto é, treina-se cada RBM, sucessivamente, da primeira até a mais ao topo da pilha, de forma que o treinamento da RBM inferior (sua camada *hidden*) serve de entrada para o treinamento da próxima RBM (como para as DBNs).

Após este treinamento *bottom-up*, na segunda etapa do treinamento das DBMs, o método *Mean-Field* explanado atualiza os pesos da rede considerando influências não só *bottom-up* mas também *top-down*, como dito. A Figura 3.6 ilustra a arquitetura de uma rede DBM composta de 3 camadas (3 RBMs) sendo treinada com a apresentação de parte de uma imagem de impressão digital. As RBMs da DBM são do tipo BB-RBMs (Bernoulli-Bernoulli RBMs), isto é, suas camadas visíveis e escondidas são binárias. Uma Gaussian-Bernoulli RBM (GB-RBM) faz a interface entre os tons de cinza da imagem de entrada e os valores probabilísticos das BB-RBMs da DBM, que lidam apenas com valores de ativação binários. Os pesos $\mathbf{W}_l$ de cada camada l são mostrados em vermelho bem como os *biases* **a** e **b** da última camada (cada camada também tem seus *biases*).

3.4 Redes Neurais Temporais

Uma das principais arquiteturas de redes neurais profundas que captura informações temporais de sequências de imagens é a C3D (TRAN et al., 2015). Basicamente, a C3D pode ser vista como uma extensão temporal das CNNs tradicionais, que trabalham com imagens isoladas. Nesta arquitetura as operações de convolução e *pooling* ocorrem tanto na dimensão espacial quanto na temporal, isto é, os filtros de convolução e *pooling* possuem 3 dimensões e são aplicados sobre pixels vizinhos tanto espacialmente em uma dada imagem, quanto temporalmente em imagens subsequentes (pixels nas mesmas posições em imagens sucessivas). A Figura 3.7 ilustra exemplo de convolução espaço-temporal com filtro de dimensões $3 \times 3 \times 3$.

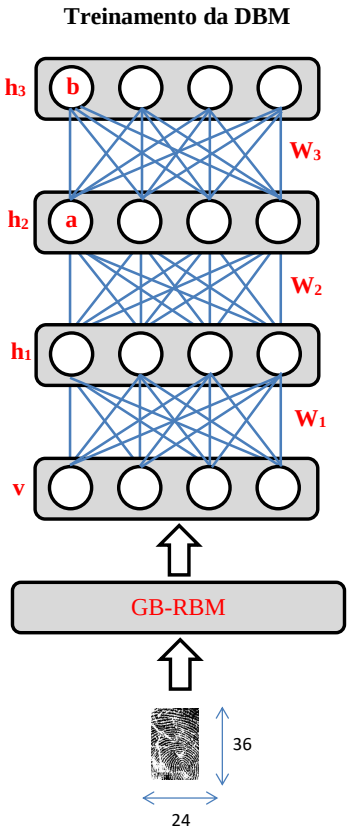


Figura 3.6: Arquitetura de uma rede DBM sendo treinada com um *patch* de impressão digital de tamanho 36×24. Fonte: Elaborada pelo autor.

As saídas de cada camada de convolução ou *pooling* correspondem também a sequências de *feature maps*, como mostrado na Figura 3.7, às quais novos *kernels* 3D de convolução ou *pooling* são aplicados, em camadas superiores da rede, a fim de continuar capturando informações temporais (cada vez de mais alto nível) a partir de tais sequências.

Uma outra categoria de redes neurais que podem ser empregadas na extração de características temporais mas que, ao contrário das redes *feedforward* (como as CNNs), apresentam retroalimentação, corresponde às redes neurais recorrentes. Tais redes, em geral, apresentam uma espécie de célula de memória interna que armazena o estado atual da arquitetura e que vai sendo atualizada a partir de novos padrões de entrada bem como do seu estado anterior (GRAVES, 2012). A rede proposta por Hopfield (1982), por exemplo, já apresentava um esquema simples de retroalimentação a fim de aprender padrões de dados dinâmicos. A Figura 3.8 ilustra o esquema geral da arquitetura de uma rede recorrente. Dada uma entrada e o estado atual da rede, obtém-se uma saída e atualiza-se o estado interno. A memória interna, na realidade, está codificada nos pesos internos da rede.

Dentre as arquiteturas de redes recorrentes hoje sendo estudadas, encontram-se as LSTMs

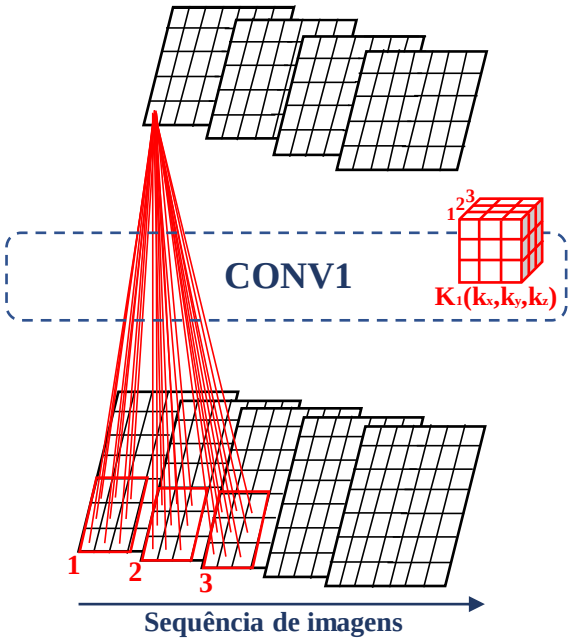


Figura 3.7: Exemplo de convolução espaço-temporal utilizando a arquitetura C3D. O *kernel* de convolução possui 3 dimensões a fim de operar sobre mais de uma imagem por vez capturando variações dos valores dos pixels no tempo. Fonte: Elaborada pelo autor.

(*Long Short-Term Memory*) (HOCHREITER; SCHMIDHUBER, 1997). As redes LSTMs foram propostas, como o próprio nome diz, para conseguir armazenar padrões temporais de mais longo prazo, visto que as rede recorrentes tradicionais, em geral, acabam capturando informações temporais de curtíssimo prazo (HOCHREITER; SCHMIDHUBER, 1997; GERS; SCHRAUDOLPH; SCHMIDHUBER, 2003). Basicamente, as unidades das redes LSTMs apresentam certa célula de memória interna a qual é atualizada com base em dois canais de entrada, o *Input Gate* e o *Forget Gate* e cujos valores são acessados pela próxima unidade por meio de um canal de saída denominado *Output Gate*. A Figura 3.9 ilustra a arquitetura de uma rede neural LSTM.

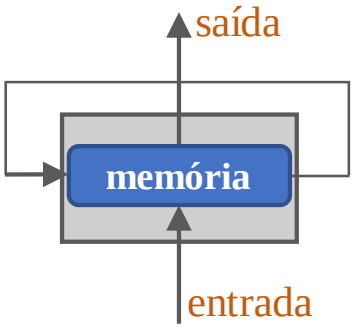
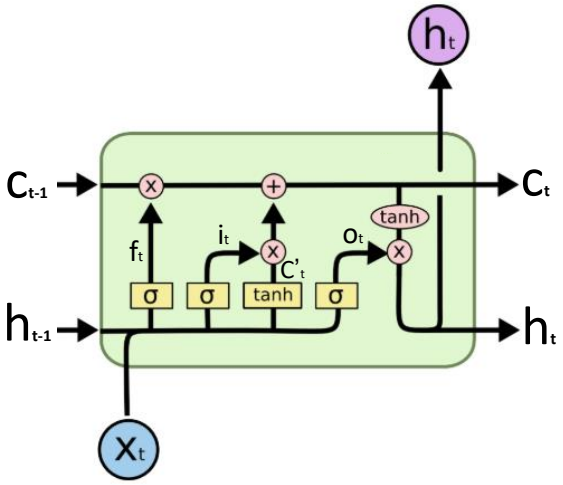


Figura 3.8: Esquema geral de arquitetura de uma rede neural recorrente. Fonte: Elaborada pelo autor.



,

Figura 3.9: Exemplo de arquitetura da rede neural LSTM. As operações indicadas em rosa são *pointwise*, isto é, ponto a ponto nos vetores. As operações em amarelo indicam ativações na rede. Os valores de x_t, h_t e C_t indicam os dados de entrada, saída e memória da rede neural, respectivamente. Fonte: Adaptada de Olah (2015).

O *Forget Gate* está localizado na parte esquerda da LSTM na Figura 3.9, com seu sinal $\mathbf{f}_t$ que controla o quanto a rede esquecerá da memória interna do modelo, C. Em termos matemáticos:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{3.20}$$

onde $\mathbf{W}_f$ e $\mathbf{b}_f$ representam os pesos e *biases* da rede no *Forget Gate* e $\sigma(\cdot)$ corresponde à função de ativação sigmoidal.

De forma similar no *Input Gate*, mais ao centro da LSTM na Figura 3.9, obtém-se:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{3.21}$$

$$\mathbf{C'}_t = tanh(\mathbf{W}_C[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \tag{3.22}$$

onde $\mathbf{W}_i$, $\mathbf{b}_i$, $\mathbf{W}_C$ e $\mathbf{b}_C$ representam os pesos e *biases* da rede no *Input Gate*.

Todos estes valores, $\mathbf{f}_t$, $\mathbf{i}_t$ e $\mathbf{C'}_t$, atualizam a memória da rede neural seguindo:

$$\mathbf{C}_t = \mathbf{f}_t \cdot \mathbf{C}_{t-1} + \mathbf{i}_t \cdot \mathbf{C'}_t \tag{3.23}$$

onde "$\cdot$" representa operações de multiplicação *pointwise* (ponto a ponto nos vetores).

Por fim, no *Output Gate*, tem-se a saída da rede neural:

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{3.24}$$

$$\mathbf{h}_t = \mathbf{o}_t \cdot tanh(\mathbf{C}_t) \tag{3.25}$$

onde $\mathbf{W}_o$ e $\mathbf{b}_o$ representam os pesos e *biases* da rede no *Output Gate*.

Por controlarem o quanto a memória interna pode ser sobreposta por uma nova entrada por meio do *Input* e *Forget Gate* é que as LSTMs conseguem manter dados de longo prazo armazenados internamente. Vale observar ainda que pode-se, por exemplo, acoplar uma rede LSTM no topo de uma rede CNN a fim de se analisar padrões temporais nas características de alto nível extraídas pela rede convolucional, tornando a análise temporal ainda mais robusta. Para problemas de classificação, em geral adiciona-se ainda, sobre a CNN e LSTM, uma camada completamente conectada com neurônios *softmax*.

Capítulo 4

TRABALHOS CORRELATOS

Neste capítulo são apresentados os principais métodos de detecção de *spoofing* facial 2D da literatura, a maioria deles referenciados nos capítulos subsequentes desta tese e comparados com as abordagens propostas, por representarem o estado-da-arte. Na Seção 4.1 são apresentados métodos baseados em características *handcrafted*, isto é, pré-definidas matematicamente e deterministicamente (sem depender dos dados de treinamento) na proposição das abordagens, e na Seção 4.2 são apresentados os métodos baseados em Aprendizado de Máquina em Profundidade, com características de alto nível autoaprendidas dos dados sob análise.

4.1 Métodos Baseados em Características *Handcrafted*

Como mencionado na Seção 2.2, os métodos de detecção de *spoofing* facial orientados a dados, isto é, que não necessitam de colaboração do usuário ou de sensores adicionais para detecção de ataques, podem valer-se de algoritmos que: (i) detectam a presença de vida (*liveness detection*), isto é, movimentos faciais como o piscar de olhos, etc.; (ii) avaliam características da cena capturada pelo sensor, comparando com as características da face apresentada, como por exemplo, diferenças de movimentação; ou (iii) se baseiam na avaliação da qualidade da imagem facial, uma vez que faces impressas ou vídeos exibidos à câmera do sistema biométrico são registrados com certas "distorções" em relação a imagens reais (alterações na refletância facial, etc.) (PEREIRA, 2013).

4.1.1 Detecção de Vida

Em se tratando de métodos que detectam a presença de vida, Pan et al. (2007) propuseram a extração de medidas, por meio do uso de um HMM (*Hidden Markov Model*) (BAUM; PETRIE,

1966; BAUM; EGON, 1967), do mapeamento dos estados dos olhos (abertos ou fechados) de uma face sob análise em um período de tempo. Com isto, é possível determinar se há ou não ataque baseado na presença ou ausência do piscar dos olhos na face diante do sensor. Os autores, após detectar e segmentar os olhos na sequência de imagens usando uma cascata de classificadores "fracos" baseada no contraste de pixels e regiões da imagem, inspirada no trabalho de Viola e Jones (2001), calculam medidas de abertura dos olhos, também com base em outro grupo de classificadores "fracos", similares aos primeiros e atentos às regiões das bordas dos olhos, as quais servem de base para o HMM.

Segundo Li (2008), o ato de piscar é intrínseco ao funcionamento do globo ocular e todo ser humano pisca ao menos uma vez a cada 20 segundos, o que torna tal característica discriminativa entre faces reais e fotografias impressas (ou mesmo faces de cadáveres). O autor também utiliza a detecção do piscar de olhos a partir da aplicação de filtros de Gabor (GABOR, 1946; DAUGMAN, 1985) nas imagens para identificar ataques com faces sintéticas estáticas (fotografias ou imagens exibidas em *display*, por exemplo).

Kollreider, Fronthaler e Bigun (2009) desenvolveram um método que analisa o movimento de diversos pontos (regiões) de controle na face usando um algoritmo de fluxo óptico. Baseado na movimentação horizontal e vertical dos pontos da face, determina-se se o sistema está sob ataque ou não. Basicamente, os autores extraem coeficientes relacionando a direção e intensidade da movimentação das regiões faciais internas com a movimentação das externas. Segundo eles, o padrão de tais movimentações difere de faces reais e impressas: regiões centrais em faces reais tendem a se mover mais e com direções diferentes das externas ao passo que, em faces impressas, as movimentações são homogêneas. Os coeficientes extraídos das imagens de vídeo são integrados para formar um *score* final referente à presença de vida na face sendo analisada.

Bao et al. (2009) analisam a movimentação facial a fim de identificar ataques por meio da diferença do padrão de movimentação de objetos 3D (faces reais) e 2D (faces impressas). Eles também utilizam o algoritmo de fluxo óptico, mas ao contrário de outros trabalhos que observam pontos específicos da face, eles analisam a movimentação da face como um todo. Como mostram os autores, o fluxo óptico global de faces impressas (2D) tende e diferir de faces reais (3D).

Tirunagari et al. (2015) propõem a detecção não só de de movimentos oculares mas também de outras partes da face, como exemplo, movimentos labiais, para a prevenção de ataques. Eles utilizam conceitos da dinâmica de fluidos para a análise e caracterização da movimentação dos pontos da face nas sequências de imagens. Empregam o método denominado DMD (*Dynamic Mode Decomposition*) (SCHMID, 2010) em dada sequência de *frames* para a extração do

comportamento dos pontos faciais ao longo do tempo, obtendo-se "imagens" (matrizes) com as principais diferenças de movimentação de tais pontos, e extraem características de textura (histograma de tons de cinza) das mesmas a partir do descritor LBP (*Local Binary Pattern*) (OJALA; PIETIKAINEN; HARWOOD, 1994, 1996), classificando o vídeo por meio de uma SVM (*Support Vector Machine*) (CORTES; VAPNIK, 1995).

Devido à sua grande utilização em métodos de anti-*spoofing* facial, dados os bons resultados que permite obter por sua sensibilidade às imperfeições geradas na recaptura das faces sintéticas (exibidas em *displays* ou papel, por exemplo), vale pontuar que o descritor LBP atua, basicamente, em imagens em tons de cinza, associando a cada pixel, um novo valor de intensidade, após comparar sua intensidade original com a dos pixels vizinhos. Na análise de um pixel p da imagem, compara-se seu tom de cinza com o tom de cada vizinho q. Caso o tom de q seja maior ou igual ao do p, associa-se o *label* "1" a q, caso contrário "0". Definido um raio de vizinhança R e um número de vizinhos P, após efetuar as comparações e percorrer os pixels vizinhos de p em sentido horário a partir de seu vizinho superior esquerdo, pode-se contruir um número binário com base nos *labels* associados a tais pixels. Este número é então convertido para decimal e associa-se o valor ao pixel central p sob análise. Em termos matemáticos o valor LBP de um pixel p é dado por:

$$LBP_{P,R} = \sum_{q=0}^{P-1} l(c_q - c_p) \cdot 2^q \tag{4.1}$$

onde c_p indica o tom de cinza de p, c_q o tom de cinza do vizinho q e $l(x)$ corresponde à função de limiarização definida por:

$$l(x) = \begin{cases} 1, & se\ x \geq 0, \\ 0, & se\ x < 0. \end{cases} \tag{4.2}$$

Após efetuar tal procedimento para todos os pixels de uma dada imagem, um histograma é construído com a frequência da ocorrência de cada decimal possível a fim de caracterizá-la. Tal histograma é que serve então de entrada, em geral, aos classificadores como as SVMs.

Kim, Suh e Han (2015) propuseram analisar a diferença de refletância da luz em objetos (faces) 2D e 3D baseado na ideia de que em faces impressas ou exibidas em *displays* (superfícies planas) a refletância da luz incidente se dá na mesma direção e sentido enquanto que em faces reais 3D, de maneira mais difusa. Eles calculam os chamados *Local Speed Patterns* (LSP) para caracterizar os pixels da face. Após estimar a diferença de intensidade de cada pixel entre uma imagem de referência e uma atual, eles geram os LSPs com base na comparação dos

pixels com seus vizinhos, de forma parecida com o LBP. Os códigos obtidos da região facial formam um vetor de características com base na frequência de suas ocorrências e alimentam um classificador SVM, treinado para distinguir códigos de faces impressas (ou em *displays*) 2D de faces reais 3D.

Liu et al. (2016) e Nowara, Sabharwal e Veeraraghavan (2017) utilizam métodos capazes de detectar a presença ou ausência de fluxo sanguíneo na face, com base na variação de sua coloração ao longo do tempo, a fim de identificar ataques (presença ou ausência de vida). Nowara, Sabharwal e Veeraraghavan (2017), por exemplo, após detectarem a face nas imagens, extraem medidas de cor baseadas em regiões internas das mesma, como testa e bochechas, bem como externas, a fim de se detectar se há variações de iluminação no ambiente. Caso os padrões das regiões internas sejam diferentes das externas, classifica-se a face nas imagens como real (utiliza-se uma SVM para tal fim), isto é, há pulsação e circulação de sangue na face enquanto que no fundo não, caso contrário, a face é classificada como sintética (caso em que as variações na coloração da pele possivelmente decorrem de variações no ambiente).

4.1.2 Comparação da Face e do Plano de Fundo

Em se tratando de métodos que analisam a face em comparação com o *background* para a detecção de *spoofing* facial, Anjos e Marcel (2011), por exemplo, propuseram um método de detecção de *spoofing* baseado em diferenças de movimento relativo entre a face e o fundo. O movimento acumulado para uma região de interesse (ROI - *Region of Interest*) é dado por:

$$M_D = \frac{1}{S_D} \sum_{(x,y)\in \mathbf{D}} |I_t(x,y) - I_{t-1}(x,y)| \tag{4.3}$$

onde **D** corresponde à região de interesse, S_D é a área de **D** em pixels, $I_t(x,y)$ é a intensidade do pixel na posição (x,y) em um *frame t* do vídeo, e t é o índice do quadro na sequência de vídeo. Medidas baseadas no valor M_D da região facial e do *background* em intervalos de 20 *frames* do vídeo (como M_D mínimo, máximo, etc.) alimentam um classificador *Multilayer Perceptron* (MLP) (RUMELHART; HINTON; WILLIANS, 1986) a fim de determinar se há ataque ou não.

Yan et al. (2012) também propõem, além da análise de outras características como o piscar de olhos, a comparação da movimentação da face e do *background*. Faces sintéticas (2D) tendem a apresentar, segundo eles, alta consistência em termos de movimentação em relação às regiões do entorno da face, enquanto que faces reais tendem a apresentar movimentação diferente (independente) do plano de fundo. Após a detecção da face baseada no método proposto por Viola e Jones (2001), os autores estimam as diferenças (taxas) de movimentação das duas

regiões nos quadros dos vídeos por meio de um modelo de mistura de Gaussianas (*Gaussian Mixture Model* - GMM) (STAUFFER; GRIMSON, 1999) e calculam a razão de movimentação entre face e *background*.

Xu, Li e Deng (2015) também argumentam que muitas pistas de ataques podem ser extraídas da região de *background*, isto é, do entorno da face, e por isso propõem o uso de imagens faciais ampliadas na detecção de ataques. Eles mostram que tal ampliação, até certo ponto, pode ser benéfica aos métodos de detecção de *spoofing* facial. Entretanto, ao ampliar demais a área sob análise, os métodos passam ter seus desempenhos deteriorados devido a muitas informações complexas e não úteis contidas nos pixels do plano de fundo. Patel, Han e Jain (2016a) também afirmam que importantes características podem ser extraídas do *background* a fim de se detectar ataques. Segundo os autores, da mesma forma que ocorrem distorções nos valores dos pixels da região facial ao se capturar uma face reproduzida em material sintético, o mesmo pode ocorrer para pixels do entorno da face (usualmente reproduzidos no mesmo material).

Apesar da existência destes trabalhos que se valem de informações do *background* para detectar ataques a sistemas de reconhecimento facial, não existem estudos mais aprofundados acerca da quantidade ideal de informações do fundo necessária para uma melhor acurácia dos métodos de anti-*spoofing*, apenas observações empíricas são reportadas.

4.1.3 Análise da Qualidade da Imagem Facial

Já em relação à análise da qualidade das imagens faciais para a detecção de ataques, Maatta, Hadid e Pietikainen (2011) propõem o uso da versão multiescala do LBP, denominada de MLBP (*Multiscale* LBP) (OJALA; PIETIKAINEN; MAENPAA, 2002), para detecção de *spoofing* facial em imagens isoladas. Neste método, o descritor LBP é aplicado à região facial com diferentes sistemas de vizinhança (variando o número de vizinhos e distância ao pixel central). Os histogramas gerados em cada sistema ao final são concatenados e alimentam uma SVM para a detecção de ataques. A Figura 4.1 ilustra a abordagem proposta. Tal método apresenta taxas de acerto bastante elevadas, também sendo tomado como referência na proposta de novas técnicas. Neste mesmo trabalho, os autores comparam a detecção de *spoofing* facial por meio dos descritores LBP, LPQ (*Local Phase Quantization*) (OJANSIVU; HEIKKILA, 2008) e *wavelets* de Gabor (MANJUNATH; MA, 1996), isoladamente, alimentando SVMs, e mostram a superioridade do LBP em relação aos demais descritores de textura avaliados.

Além da textura, alguns autores utilizam também descritores de formas e cores a fim de detectar *spoofing* facial (TRONCI et al., 2011). Outros trabalhos analisam a refletância da face valendo-se de técnicas como a normalização adaptativa de histogramas (borramento) (TAN et al.,

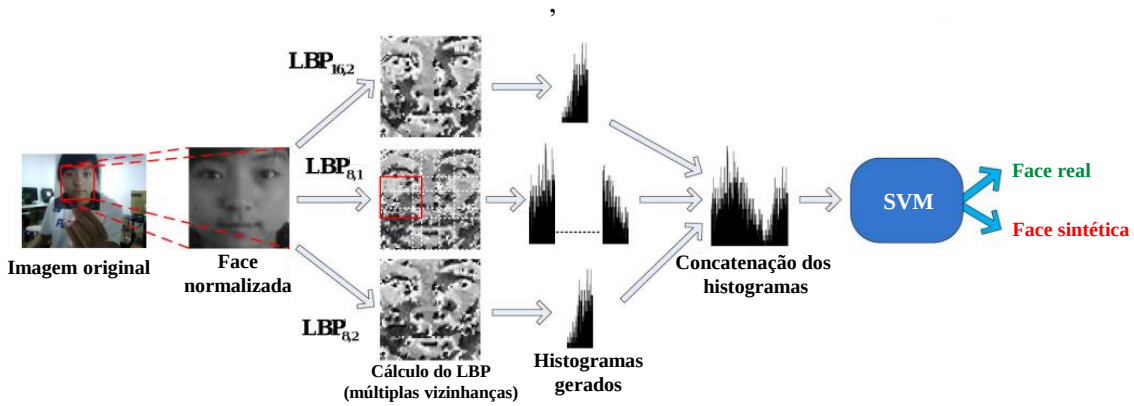


Figura 4.1: Esquema do método para detecção de *spoofing* facial com base no MLBP. Fonte: Adaptada de Maatta, Hadid e Pietikainen (2011).

2010; PEIXOTO; MICHELASSI; ROCHA, 2011). O brilho dos *displays* de exibição pode também ser usado para detectar a apresentação de característica sintética (PINTO et al., 2012).

Schwartz, Rocha e Pedrini (2011) utilizam uma abordagem baseada em descritores de textura e cor, denominados por eles de LLD (*Low-Level Descriptors*). A partir das imagens faciais, descritores como GLCM (*Gray Level Co-ocurrence Matrix*) (HARALICK; SHANMUGAM; DINSTEIN, 1973), HOG (*Histogram of Oriented Gradients*) (DALAL; TRIGGS, 2005), CF (*Color Frequency*) (SCHWARTZ et al., 2009) e HSC (*Histograms of Shearlet Coefficients*) (SCHWARTZ; SILVA; PEDRINI, 2011) extraem diferentes características das mesmas a fim de classificá-las por meio do método PLS (*Partial Least Squares*) (WOLD, 1985). Yang et al. (2013) igualmente extraem características de textura LBP (OJALA; PIETIKAINEN; HARWOOD, 1994, 1996), HOG (DALAL; TRIGGS, 2005) e LPQ (OJANSIVU; HEIKKILA, 2008) de regiões da imagem onde encontram-se os principais elementos faciais (olhos, boca, nariz, etc.) e codificam os valores obtidos em categorias, gerando um vetor final para a face o qual alimenta uma SVM para classificação. Tal descritor é denominado *Component Dependent Descriptor* (CDD). Arashloo, Kittler e Christmas (2015) também empregam características baseadas no LPQ (OJANSIVU; HEIKKILA, 2008) e medidas dinâmicas de textura baseadas no BSIF (*Binarized Statistical Image Features*) (KANNALA; RAHTU, 2012), que atua de forma similar ao LBP nas imagens.

Chingovska, Anjos e Marcel (2012) e Maatta, Hadid e Pietikainen (2012) também propuseram analisar a textura facial em quadros isolados de vídeo, a partir do descritor de texturas LBP (*Local Binary Pattern*) (OJALA; PIETIKAINEN; HARWOOD, 1994, 1996), para classificar as faces em reais e sintéticas. Uma classificação para o vídeo como um todo pode ser obtida por votação simples com base nas classificações de todos os seus *frames*.

Baseado na variação temporal do LBP proposta por Zhao e Pietikainen (2007), o LBP-TOP - *Local Binary Pattern from Three Orthogonal Planes*, Pereira et al. (2012) incrementam

o trabalho de Chingovska, Anjos e Marcel (2012) para atuar analisando sequência de *frames* na detecção de ataques, e não somente imagens isoladas. No LBP-TOP adiciona-se a dimensão "tempo" na comparação de certo pixel com seus vizinhos, isto é, também são considerados vizinhos de um pixel p, em um *frame* de índice t, os pixels próximos a p nos *frames* $t-1$ e $t+1$, por exemplo. Deste modo, a uma dada imagem facial, são associados 3 histogramas LBP (posteriormente concatenados). Cada um deles é obtido calculando-se o LBP para os pixels em um dos 3 planos ortogonais: o plano da imagem (plano xy), o plano formado pelos pixels de uma mesma linha de quadros consecutivos do vídeo (plano xt), e o plano formado por pixels de uma mesma coluna em quadros consecutivos (plano yt), conforme mostra a Figura 4.2 (pixel central p, tomado como base para o cálculo dos histogramas, representado por ponto branco).

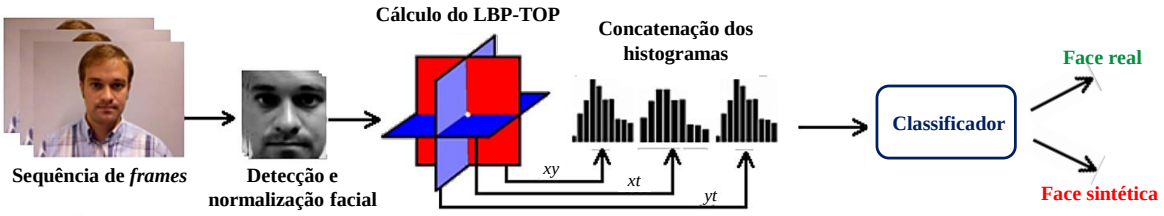


Figura 4.2: Esquema do LBP-TOP na captura de informações de textura espaco-temporais para classificação facial. Fonte: Adaptada de Pereira et al. (2012).

O incremento de informações temporais no método de detecção de *spoofing* facial baseado no LBP, que já apresentava boas taxas de acerto na detecção de ataques, melhorou ainda mais os resultados, tornando a técnica baseada no LBP-TOP, quando da sua proposição, um dos métodos considerados como estado-da-arte ao se trabalhar com imagens de vídeo.

Baseado na ideia de que as imagens ou vídeos em ataques concentram informações em bandas de frequência específicas, Zhang et al. (2012) propuseram um método de detecção de fraudes a partir de filtros DoG (*Difference of Gaussians*). O resultado da aplicação dos filtros, separadamente, à imagem facial, alimenta um classificador SVM. Li et al. (2004) também realizaram testes de detecção de *spoofing* com bons resultados a partir do princípio de que faces impressas apresentam menos componentes de alta frequência do que faces reais.

Pinto et al. (2012) também trabalham na detecção de *spoofing* a partir de ruídos deixados nas imagens no processo de captura. Eles utilizam uma técnica baseada na Transformada de Fourier discreta (COOLEY; TUKEY, 1965) e no ritmo visual (CHUNG et al., 1999) de forma a sumarizar diferenças de ruídos de vídeos de acessos válidos e ataques em imagens singulares. Em síntese, a partir de cada vídeo de treinamento, um novo vídeo é gerado com base nos ruídos detectados na sequência de imagens original e no espectro de Fourier. Cada vídeo resultante

é então sumarizado em uma única imagem em tons de cinza que captura o ritmo visual dos ruídos do respectivo vídeo de treinamento original. Descritores de textura (GLCM) extraem características a partir de tais imagens de ritmo visual, as quais alimentam um classificador SVM para treinamento. Dado um vídeo de teste, o mesmo processo é repetido e, com base em suas características de textura extraídas a partir do ritmo visual, o classificador determina se trata-se de ataque ou não.

Boulkenafet, Komulainen e Hadid (2015) combinam informações de textura e cor para detecção de *spoofing* facial utilizando uma variação do LBP para imagens coloridas apresentada por Choi, Plataniotis e Ro (2010). Eles alegam que as diferenças de textura entre faces reais e sintéticas, quando toma-se como base de avaliação apenas imagens em tons de cinza, isto é, considerando-se puramente informações de luminância, podem ser muito sutis para a detecção de ataques, visto que hoje as faces sintéticas apresentadas (fotografias impressas, vídeos em *displays*, etc.) são confeccionadas, em geral, em alta resolução. Neste sentido, ou autores propõem analisar também informações de crominância das faces dado que, segundo eles, faces sintéticas apresentam distorções nas cores (matizes) quando recapturadas pelas câmeras dos sistemas biométricos. Eles trabalham com modelos de cores que separam informações de luminância e crominância dos pixels, como o HSV (*Hue*, *Saturation* and *Value*) e o YCbCr, aplicando o LBP sobre os três canais das imagens faciais e capturando tais distorções de forma eficaz. Os histogramas dos três canais são concatenados e apresentados a classifcadores SVMs (com *kernels* lineares e de bases radiais). A Figura 4.3 ilustra este processo.

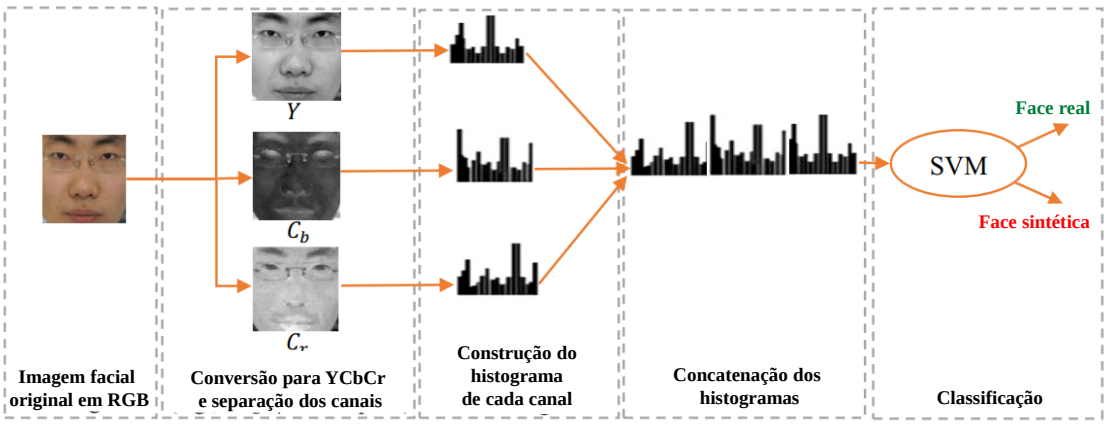


Figura 4.3: Deteção de *spoofing* facial utilizando informações de textura e cor. Fonte: Adaptada de Boulkenafet, Komulainen e Hadid (2015).

Wen, Han e Jain (2015) propuseram o uso de quatro diferentes características para detectar distorções no padrão das imagens faciais e, consequentemente, ataques de *spoofing*: reflexão especular, intensidade de borramento, momento cromático e diversidade de cores. Eles realizam

o treinamento, baseado nestes quatro tipos de informações extraídas das imagens, de várias SVMs, cada qual especializada em um tipo de ataque e depois fazem a fusão das classificações. Eles ampliam o método para trabalhar com vídeo por meio de um esquema de votação a partir da classificação de seus *frames*.

Patel et al. (2015) detectam a presença dos padrões de Moiré, "serrilhados" deixados quando da recaptura de vídeos exibidos a partir de *displays* digitais, para a detecção de *spoofing* por meio da extração de características LBP multiescalas (OJALA; PIETIKAINEN; MAENPAA, 2002) e SIFT (*Scale-Invariant Feature Transform*) (LOWE, 1999). Tais padrões não são encontrados em acessos válidos, isto é, na captura de imagens de faces reais. Deste modo, treinam uma SVM para determinar se em certo quadro do vídeo sob análise há ataque ou não. Após classificar cada *frame*, uma votação é feita para decidir se o vídeo é forjado ou se trata-se de um acesso válido ao sistema. Patel, Han e Jain (2016b) ampliam o trabalho anterior analisando uma série de características para a detecção de *spoofing* facial e por fim utilizam histogramas LBP e medidas de cores, como momentos das cores dos pixels da região facial, para alimentar uma SVM a fim de classificar faces em reais ou sintéticas.

Parveen et al. (2016) propuseram o uso do descritor DLTP (*Dynamic Local Ternary Pattern*) para extrair características das faces e detectar *spoofing*. Baseado no LTP (*Local Ternary Pattern*) (TAN; TRIGGS, 2010), uma adaptação do LBP que opera com três valores possíveis de *labels* ("-1", "0" e "1") ao invés de dois ("0" ou "1") nas comparações de pixels vizinhos, o DLTP estende tal método baseando-se na Lei de Weber (FECHNER, 1966), que diz que a variação de um estímulo só é notada se a magnitude for maior que uma razão em relação ao estímulo inicial, tornando o descritor dinâmico na comparação do pixel central e vizinhos em função dos tons de cinza da região em que opera no momento. Ambos os métodos se preocupam em tratar a questão de regiões homogêneas onde há certa presença de ruído, que pode degradar a performance do LBP. Os autores se valem de uma SVM para classificar a face como real ou sintética a partir das características extraídas pelo DLTP.

Akhtar e Foresti (2016) propõem um método para detecção de *spoofing* baseado em *patches* faciais, ao invés da face como um todo, na tentativa de obter um melhor desempenho por analisar apenas as partes faciais mais relevantes à identificação de ataques. Para um dado *frame* de um determinado vídeo, detecta-se e normaliza-se a face nesta imagem. Então, divide-se a região facial em um *grid* de *patches* a fim de selecionar os mais discriminativos para a classificação da face em real ou sintética. Para a seleção de tais *patches*, os autores avaliam uma série de métricas, parte delas baseadas em clusterização, agrupando os *patches* mais parecidos em termos de seus pixels e escolhendo os mais representativos de cada agrupamento, por exemplo. A

Figura 4.4 ilustra a abordagem proposta. Vale ressaltar que os *patches* mais discriminativos na abordagem podem variar de face pra face, isto é, não são sempre os mesmos. Após selecionar os *patches* mais relevantes de uma dada face, classifica-se cada um separadamente e toma-se uma decisão para a mesma com base na classificação da maioria de seus *patches*. Os autores avaliam diferentes classificadores no trabalho, dentre eles as SVMs (que apresentaram os melhores resultados).

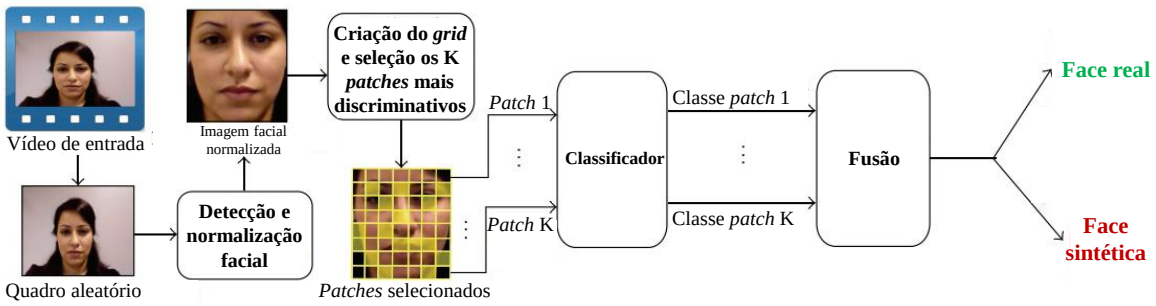
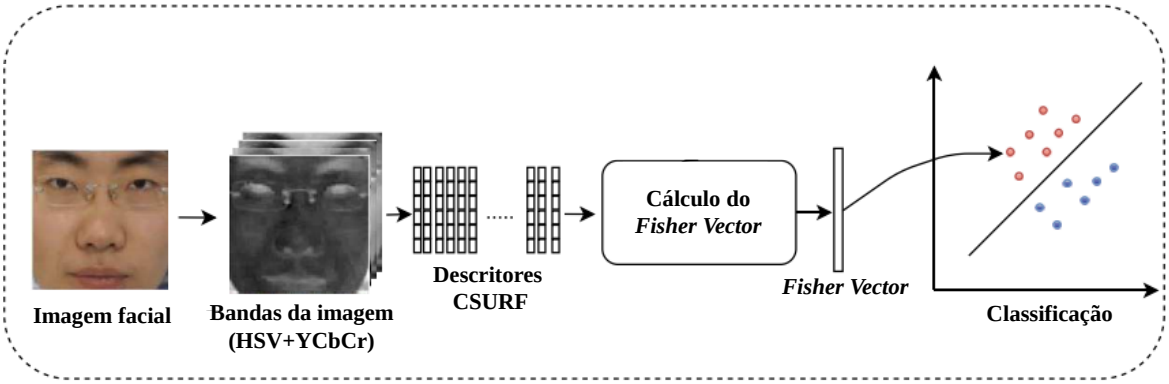


Figura 4.4: Abordagem proposta baseadas na seleção de *patches* da face para sua classificação. Fonte: Adaptada de Akhtar e Foresti (2016).

Boulkenafet, Komulainen e Hadid (2017) propõem, após detectar e normalizar as faces em 64×64 pixels, a aplicação do descritor SURF (*Speeded-Up Robust Features*) (BAY; TUYTELAARS; GOOL, 2006) em seus diversos canais, denominando-o de CSURF (*Color Speeded-Up Robust Features*) para a extração de características das imagens faciais coloridas. Eles fazem experimentos em múltiplos espaços de cores como RGB, HSV e YCbCr (combinando-os também - imagens compostas de 6 bandas). Então, para cada espaço de cor sob análise, eles aplicam uma técnica denominada de *Fisher Vector Enconding* (FVE) (PERRONNIN; SANCHEZ; MENSINK, 2010), onde os vetores de características originais são mapeados para espaços de maiores dimensões a fim de facilitar a classificação linear das amostras. Eles utilizam um classificador *softmax* com função de custo baseada em *cross*-entropia, e obtêm bons resultados tanto em experimento intrabase quanto interbases. A Figura 4.5 ilustra esquema de funcionamento do método. Na sequência deste trabalho, Boulkenafet, Komulainen e Hadid (2018) fazem uma comparação mais ampla de sete descritores de textura e cor na detecção de *spoofing* facial baseados no LBP e até mesmo no método SURF, sempre extraindo características a partir dos 3 canais das imagens faciais no espaço RGB, HSV e YCbCr (ou combinação destes - imagens com 6 canais, por exemplo, HSV+YCbCr).

,



,

Figura 4.5: Deteção de *spoofing* facial a partir do CSURF (*Color Speeded-Up Robust Features*). Fonte: Adaptada de Boulkenafet, Komulainen e Hadid (2017).

4.2 Métodos Baseados em Aprendizado em Profundidade

Por trabalharem com características autoaprendidas dos dados sob análise e mesclarem evidências de *spoofing* de diversas naturezas, tanto baseadas na qualidade das imagens, quanto na presença de vida e até mesmo em características do *background* (quando presente nas imagens faciais), não se categorizou as abordagens que se valem de Aprendizado de Máquina em Profundidade nas três classes principais de métodos de detecção de *spoofing* em sistemas de reconhecimento facial orientados a dados, mencionadas na Seção 4.1, podendo ser considerados, os métodos baseados em modelos profundos, abordagens híbridas.

Em um dos primeiros trabalhos utilizando CNN (*Convolutional Neural Network*) (LECUN et al., 1998) para anti-*spoofing* facial, Yang, Lei e Li (2014) empregam a AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), um modelo tradicional de CNN que contém 5 camadas de convolução e *pooling* bem como 3 camadas completamente conectadas no topo, bem referenciada por seu desempenho superior, na época de sua proposição, no reconhecimento de objetos em imagens de cenas reais. Após detectar as faces nas imagens utilizando o método de Viola e Jones (2001) e refinar tal detecção com base em características binárias locais da face (REN et al., 2014), os autores segmentam a região facial e treinam a rede neural. Então, após o treinamento da rede, para cada face, eles extraem suas características com base em um *forward pass* da imagem facial na rede neural e na ativação dos neurônios da última camada completamente conectada. Após extrair tal vetor de características para todas as faces de treinamento, uma SVM é treinada para classificar novas imagens. Os autores variam as proporções entre área facial e de *background* nas regiões de interesse (ROI - *Region of Interest*) que alimentam a rede neural e notam que a utilização de partes do entorno da face pode ajudar até certo ponto na detecção de *spoofing* facial (o uso de muita informação de *background* tende a "confundir" o classificador).

Em se tratando da análise de informações temporais para deteção de *spoofing* facial, Xu, Li e Deng (2015) utilizam uma CNN com duas camadas de convolução e *pooling* e uma camada completamente conectada para extrair características de alto nível das faces mas, no caso, acoplada a uma rede neural recorrente LSTM (*Long Short-Term Memory*) (HOCHREITER; SCHMIDHUBER, 1997) em seu topo, a fim de capturar informações temporais de longo prazo a partir de quadros de vídeos contendo faces reais e sintéticas. Como mencionado na Seção 3.4, diferentemente das redes convolucionais, o estado atual das redes recorrentes depende também dos estados anteriores e não só dos dados de entrada, mantendo desta forma uma certa memória da evolução das características de alto nível extraídas pela CNN. No trabalho proposto os autores integram a rede LSTM à rede CNN entre sua camada completamente conectada (que extrai as *features* de alto nível finais dos quadros de entrada) e uma camada com neurônios *softmax*. Deste modo, cada quadro que alimenta a CNN, atualiza o estado da rede LSTM com suas características de alto nível e a camada *softmax* classifica o vídeo com base nos estados da LSTM. Os autores mostram que, além da robustez das características profundas, características temporais podem também por vezes beneficiar os resultados na detecção de *spoofing* facial. Para os experimentos eles empregam uma placa gráfica Nvidia Tesla K40. A Figura 4.6 ilustra a arquitetura proposta.

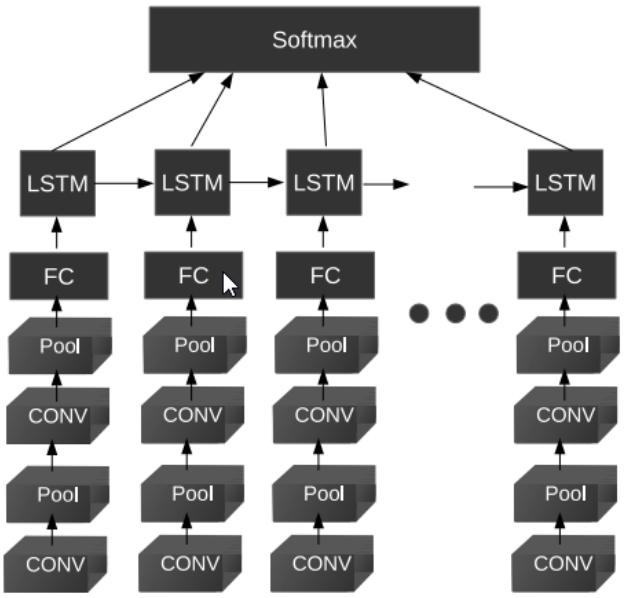


Figura 4.6: Arquitetura da rede CNN acoplada à uma rede LSTM proposta por Xu, Li e Deng (2015). "FC" significa *Fully Connected*, isto é, camada com nós completamente conectados. Fonte: Xu, Li e Deng (2015).

Menotti et al. (2015) propêm uma abordagem mais generalista e buscam encontrar uma arquitetura ótima de CNN para anti-*spoofing* facial com base em um algoritmo de busca aleatória

para exploração de valores para os hiperparâmetros da rede neural. Em uma primeira etapa, eles geram CNNs diferentes, variando suas arquiteturas (quantidade de camadas, neurônios, etc.), e observam seus desempenhos ao serem treinadas sobre bases de *spoofing* facial a fim de se extrair características para as faces a partir de suas camadas completamente conectadas e alimentar uma SVM. Em uma segunda etapa, os autores focam em inicializar os pesos dos filtros de convolução das melhores arquiteturas, antes inicializadas randomicamente, de maneira otimizada com base em uma arquitetura de CNN pré-treinada por eles sobre as bases de imagens de *spoofing* da literatura. Eles exploram apenas um pequeno intervalo de valores para os parâmetros das redes devido à complexidadade e onerosidade da tarefa (trabalham apenas com redes com poucas camadas de convolução) e obtém uma rede "ótima" para detecção de *spoofing* facial com apenas 2 camadas de convolução. São necessárias 5 GPUs, incluindo uma GPU Nvidia Tesla K40 com 12 GB de memória dedicada, para os experimentos.

Patel, Han e Jain (2016a) propõem trabalhar com redes neurais mais profundas do que as avaliadas por Menotti et al. (2015) a fim de tirar vantagem das não-linearidades nas camadas das CNNs, obtendo características mais profundas e robustas para classificar as faces em reais ou sintéticas. Os autores avaliam dois modelos tradicionais de CNN na detecção de *spoofing* facial, a CaffeNet (JIA et al., 2014) e a GoogleNet (SZEGEDY et al., 2014), ambas bastante profundas. Devido à grande quantidade de parâmetros das arquiteturas (em especial conexões sinápticas) e o tamanho restrito das bases de *spoofing* facial para treinamento satisfatório destas grandes arquiteturas, os autores primeiro treinam as redes em imagens de objetos em geral (em maior quantidade), fazem um primeiro *fine-tuning* em uma base de faces reais para reconhecimento facial e por fim um segundo *fine-tuning* nas imagens de bases de *spoofing* facial. A Figura 4.7 ilustra este processo de *fine-tuning* em etapas (*Transfer Learning*). Os autores utilizam uma GPU Nvidia Titan X nos experimentos.

Li et al. (2016) sugerem, ao invés de considerar somente a última camada completamente conectada da rede neural para a extração de características para as faces reais e sintéticas, utilizar também a saída 2D das camadas de convolução da arquitetura para representar as imagens faciais perante o classificador (no caso também uma SVM). Eles utilizam um modelo canônico de CNN denominado VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015), originalmente treinada para o reconhecimento facial (com 13 camadas convolucionais e 3 camadas completamente conectadas no topo), fazem o *fine-tuning* da rede sobre bases de imagens de *spoofing* facial encontradas na literatura (*Transfer Learning*) após substituir sua camada de 2.622 neurônios *softmax* por uma camada com apenas 2 neurônios *softmax* (dada a classificação da face em real ou sintética apenas) e, a partir de então, extraem características para as faces com base nas saídas das camadas de convolução da arquitetura (fazem testes considerando camadas diferentes

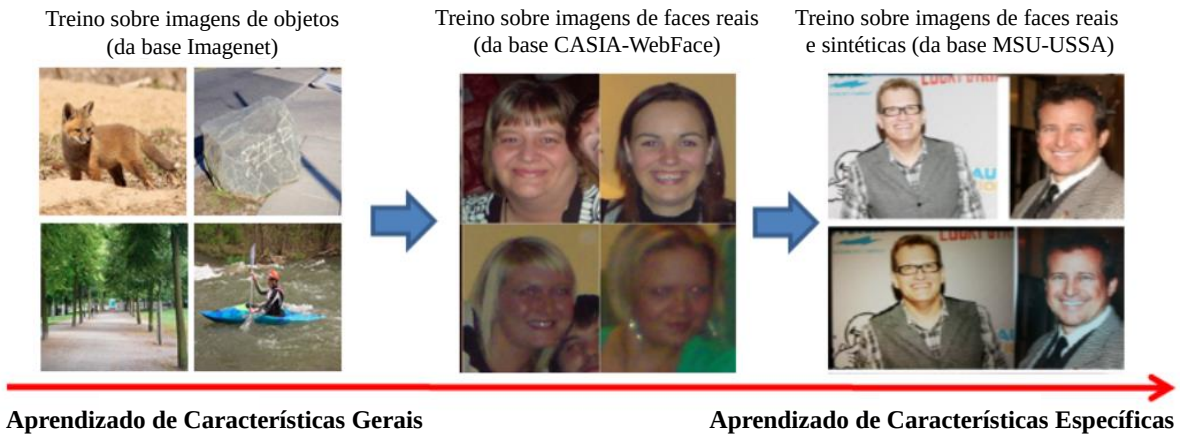


Figura 4.7: Abordagem proposta por Patel, Han e Jain (2016a) com o treinamento inicial sobre imagens mais genéricas (objetos em cenas reais), depois com imagens faciais e finalmente com imagens de bases de anti-*spoofing* facial. Fonte: Adaptada de Patel, Han e Jain (2016a).

em cada caso). Eles extraem a saída (*feature map*) média de dada camada de convolução para uma imagem de entrada.

Dadas várias imagens e após obter suas saídas médias na camada de convolução sendo considerada, seleciona-se 10% dos elementos de tais saídas com valores acima de um limiar pré-definido, obtendo-se por fim uma matriz de características, a qual tem sua dimensionalidade reduzida por meio da técnica que os autores denominam de BPCA (*Block Principal Component Analysis*), que divide tal matriz em partes e aplica a técnica PCA (PEARSON, 1901; HOTELLING, 1933) em cada uma delas separadamente. Então, as características finais obtidas nas várias aplicações do BPCA são apresentadas a uma SVM com *kernel* linear. Os autores denominam tal arquitetura como DPCNN (*Deep Part Features from the Convolutional Neural Network*), a qual é ilustrada na Figura 4.8.

Lucena et al. (2017) também realizam o *Transfer Learning*, mas a partir da rede neural VGG-16 (SIMONYAN; ZISSERMAN, 2015), já pré-treinada sobre as de imagens de objetos e cenas reais da base ImageNet (RUSSAKOVSKY et al., 2015). A CNN VGG-16 (SIMONYAN; ZISSERMAN, 2015), que deu origem inclusive à rede VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015), também possui 16 camadas, sendo 13 de convolução e 3 completamente conectadas no topo, incluindo a de classificação, com 1.000 neurônios *softmax*. Os autores eliminam as camadas completamente conectadas do topo do modelo inicial e as substituem por 2 camadas completamente conectadas, tendo uma 256 neurônios e a última apenas 1 neurônio com função de ativação sigmoidal (ao invés de *softmax*) a fim de classificar as imagens faciais em reais ou sintéticas. Eles realizam então o *fine-tuning* da rede neural sobre as bases de *spoofing* facial, ajustando os pesos com um *learning rate* baixo (de $0,0001$) e obtêm bons resultados, conforme

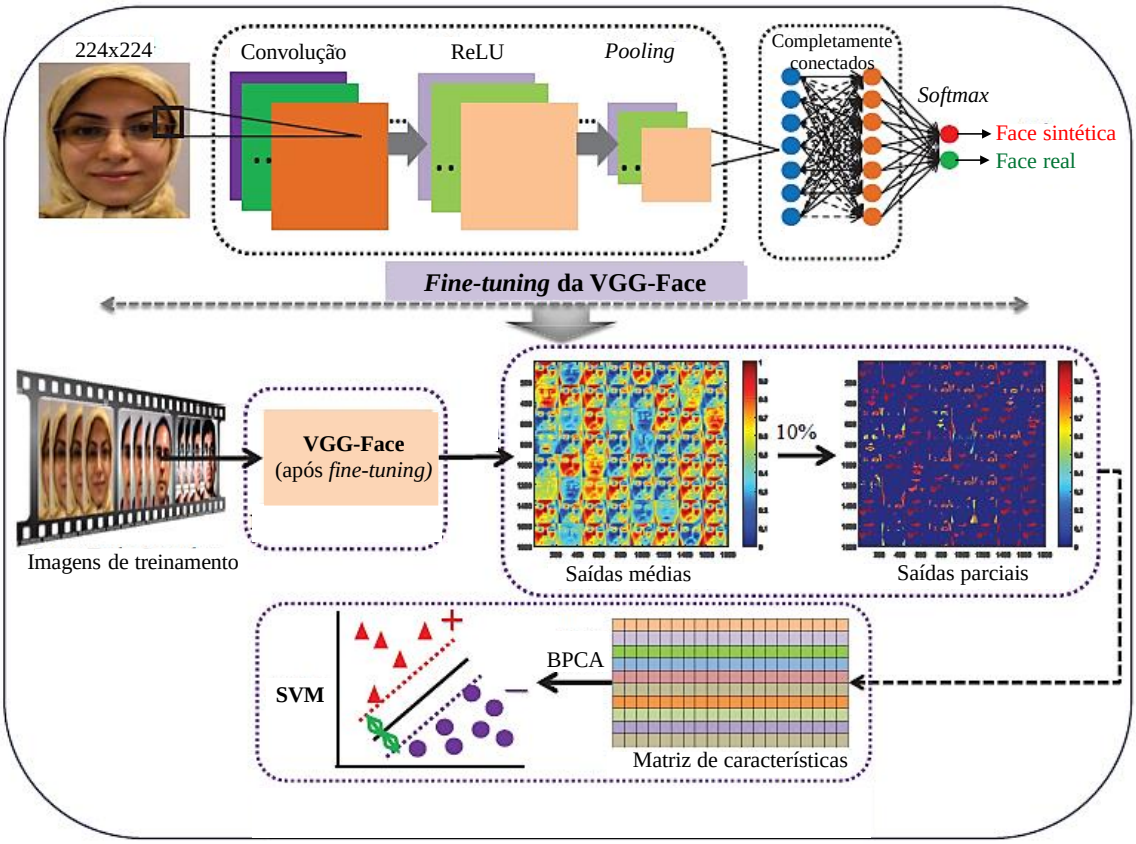


Figura 4.8: Arquitetura do método DPCNN: *fine-tuning* da VGG-Face; extração de características a partir das saídas da convolução; redução de dimensionalidade com o BPCA; e classificação por meio da SVM. Fonte: Adaptada de Li et al. (2016).

apresentado posteriormente no Capítulo 12. Os autores se valem de uma GPU Nvidia Tesla K40 para a realização dos experimentos.

Alotaibi e Mahmood (2017) também utilizam uma CNN de poucas camadas, apenas 3 camadas convolucionais (e de *pooling*), mas operando sobre imagens em tons de cinza transformadas com base no processo de difusão não-linear das mesmas, que tende a preservar as bordas de faces reais 3D enquanto que elimina as arestas de faces sintéticas 2D. Ao topo da CNN dois neurônios são responsáveis pela classificação das faces. Resultados inferiores ao estado-da-arte foram obtidos, porém sem o uso de processamento paralelo (empregou-se apenas uma CPU Intel i7 nos trabalhos).

Atoum et al. (2017), por sua vez, propõem um método que atua, para cada imagem, extraindo *patches* aleatórios da face a fim de calcular a probabilidade de cada um deles ser proveniente de face real ou sintética por meio de suas apresentações a uma CNN com 5 camadas de convolução e *pooling* e duas camadas completamente conectadas no topo, também treinada sobre *patches* facias aleatórios, isto é, extraídos de diferentes regiões das faces, randomicamente. Um *score* final é gerado para a face a partir da média das probabilidades de seus *patches*.

Além da classificação baseada em *patches*, os autores ainda trabalham com uma segunda CNN, mais profunda, capaz de estimar a profundidade (3D) da face a partir de sua imagem 2D, isto é, a profundidade de cada um de seus pixels, para sua classificação em real ou sintética. Faces sintéticas impressas ou exibidas em *displays*, por exemplo, tendem a apresentar homogeneidade de profundidade em todos os pixels.

Dada uma face de teste, após analisar cada *patch* extraído e encontrar um *score* médio para ela com base na primeira CNN, o mapa de profundidade da face é estimado com base na segunda CNN, seus valores (características) alimentam uma SVM gerando outro *score*, também indicando a probabilidade da face ser real ou sintética. Ambas as saídas, são, então, combinadas de forma a classificar a face sob análise em real ou sintética, conforme mostrado na Figura 4.9.

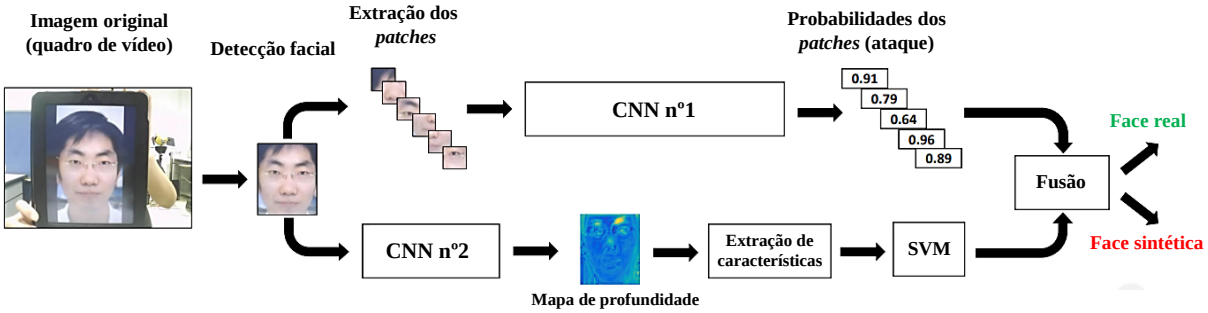


Figura 4.9: Abordagem proposta por Atoum et al. (2017) baseada em CNNs que analisam *patches* e profundidade da face para detecção de *spoofing*. Fonte: Adaptada de Atoum et al. (2017).

Jourabloo, Liu e Liu (2018) propõem uma arquitetura de CNN bastante profunda e capaz de aprender os sinais de ruídos gerados no processo de replicação da face por meio de materiais sintéticos. Faces reais tendem a não apresentar ruídos de replicação (mapas de ruídos nulos) enquanto que faces sintéticas apresentam ruídos de replicação (mapas de ruídos com valores não-nulos). Faces reais e sintéticas, com seus respectivos mapas de ruídos de replicação ideais (previamente estimados) são utilizados para treinar a rede e corrigir seus pesos. Dada uma face de treinamento como entrada, compara-se a saída obtida (mapa de ruídos estimado pela rede neural) com a saída esperada (mapa de ruídos ideal). Com isto a rede aprende a detectar faces sintéticas: quando, a partir de uma imagem facial, gera-se mapa de ruídos com valores significativos. A Figura 4.10 mostra imagens faciais reais e sintéticas e os mapas de ruídos de reprodução encontrados pela rede neural.

Sun, Sun e Li (2018) também avaliam informações temporais para a detecção de ataques de *spoofing* facial em vídeos. Eles trabalham com 4 arquiteturas de CNNs, com 5 camadas de convolução cada, a fim de avaliar qual a melhor forma de se capturar pistas de ataques espaciais e temporais: (i) CNN tradicional - que só trabalha com informações espaciais; (ii) CNN 3D

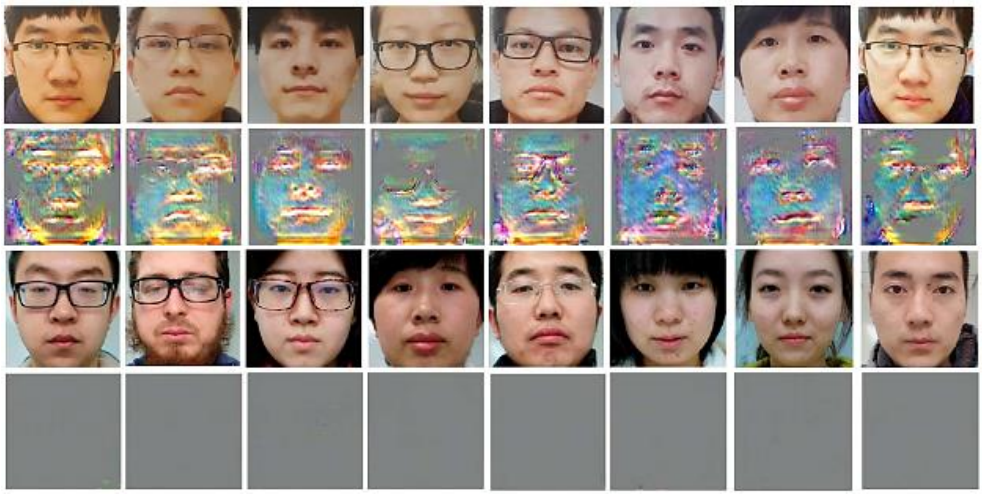


Figura 4.10: Exemplos de imagens faciais sintéticas (primeira linha) e reais (terceira linha) e seus respectivos mapas de ruídos (segunda e quarta linhas). Fonte: Jourabloo, Liu e Liu (2018).

- que apresenta operações de convolução 3D (sobre múltiplos *frames* do vídeo); (iii) CNN e LSTM - CNN tradicional com camada LSTM ao topo (abaixo da *softmax*); e (iv) CNN e LSTM Convolucional - idêntica à anterior exceto pelo fato de que a rede LSTM trabalha também com sinais 2D (com operações de convolução ao invés de produto interno). Nos experimentos realizados os melhores resultados reportados foram obtidos pela CNN tradicional e pela CNN com a LSTM comum no topo.

Liu, Jourabloo e Liu (2018), por sua vez, propõem o uso de duas redes neurais profundas, uma CNN e uma LSTM, a fim de aprender tanto informações de profundidade (3D) da face, estimando-as a partir de imagens 2D, seguindo o trabalho de Atoum et al. (2017), quanto do ritmo cardíaco, com base na variação da coloração facial nas imagens de vídeo, a fim de detectar ataques. Faces sintéticas (2D), diferentemente de faces reais, tendem a apresentar mapa de profundidade plano (todos os valores em zero) e ritmo cardíaco também nulo. A Figura 4.11 ilustra a arquitetura proposta. Na primeira parte, a CNN apresenta, após a primeira camada de convolução, 3 blocos de convoluções gerando *feature maps*, os quais são "concatenados" a fim de alimentar as camadas do topo desta rede. No topo, duas ramificações são encontradas: uma serve para treinar a CNN (a fim de detectar a profundidade das faces) com base nas imagens de de entrada e no mapa de profundidade esperado para elas (todo nulo para vídeos de ataque e com valores no intervalo $[0;1]$ para vídeos de treinamento de faces reais). A segunda ramificação serve para alimentar a rede neural recorrente (*Recurrent Neural Network* - RNN) LSTM.

Dados os *feature maps* de saída da CNN para a LSTM, na segunda ramificação da CNN, eles são alinhados com base na informação de profundidade 3D das respectivas faces de treinamento a fim de se extrair a coloração facial sempre da mesma região (etapa chamada de "registro da

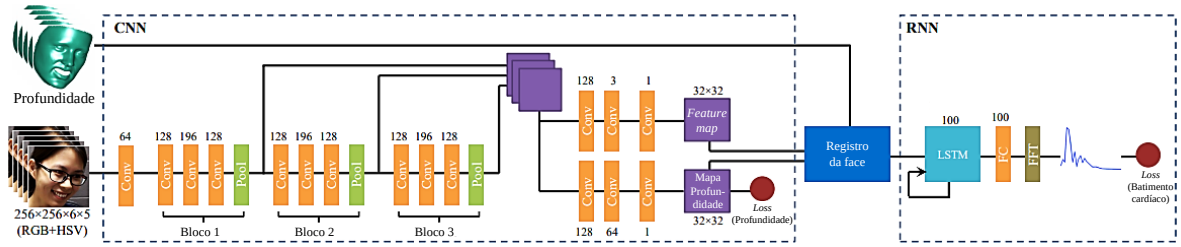


Figura 4.11: Arquitetura baseada em CNN e LSTM proposta por Liu, Jourabloo e Liu (2018). A quantidade de filtros em cada camada é exibida junto ao seu respectivo bloco. No topo da LSTM aplica-se a Transformada Rápida de Fourier (FFT - *Fast Fourier Transform*) nos dados de saída. Fonte: Adaptada de Liu, Jourabloo e Liu (2018).

face"). A rede recorrente LSTM é então alimentada (a qual possui 100 neurônios de memória), sua saída alimenta uma camada completamente conectada também com 100 neurônios e aplica-se então a Transformada Rápida de Fourier (*Fast Fourier Transform* - FFT) (COOLEY; TUKEY, 1965) sobre os valores de saída de tal camada obtendo um sinal de saída final o qual é comparado com o valor esperado para as faces de treinamento (pulsação normal para faces reais e nula para vídeos de faces sintéticas) a fim de treinar esta útima parte da arquitetura. Para calcular o *score* final de uma imagem de teste (que indica se trata-se de face real ou sintética), ela é passada pelas redes (*forward pass*) e funde-se as duas saídas (*score* relacionado à profundidade da face e *score* relacionado ao ritmo cardíaco) em uma soma ponderada.

4.3 Síntese dos Trabalhos

A Tabela 4.1 sintetiza as abordagens *handcrafted* e baseadas em Aprendizado em Profundidade apresentadas neste capítulo em termos das características que extraem das faces para detectar possíveis ataques.

Tabela 4.1: Síntese das abordagens apresentadas. Fonte: Elaborada pelo autor.

Característica Analisada	Abordagem	Autores
Métodos que detectam vida		
Piscar dos olhos	HMM	Pan et al. (2007)
	Filtros de Gabor	Li (2008)
Movimentos faciais	Fluxo óptico	Kollreider, Fronthaler e Bigun (2009)
		Bao et al. (2009)
	DMD/LBP	Tirunagari et al. (2015)
Refletância facial 2D/3D	LSP	Kim, Suh e Han (2015)
Fluxo sanguíneo	Descritores de cor	Liu et al. (2016)
		Nowara, Sabharwal e Veeraraghavan (2017)
Métodos que comparam face e fundo		
Movimento relativo face/fundo	Tons de cinza dos pixels	Anjos e Marcel (2011)
	GMM	Yan et al. (2012)
Métodos que analisam a qualidade da imagem facial		
Textura	MLBP	Maatta, Hadid e Pietikainen (2011)
	LBP	Chingovska, Anjos e Marcel (2012)
		Maatta, Hadid e Pietikainen (2012)
	LBP-TOP	Pereira et al. (2012)
	CDD	Yang et al. (2013)
	LPQ/BSIF	Arashloo, Kittler e Christmas (2015)
	DLTP	Parveen et al. (2016)
Cor	CSURF	Boulkenafet, Komulainen e Hadid (2017)
Textura e cor	LLD	Schwartz, Rocha e Pedrini (2011)
	LBP em cores	Boulkenafet, Komulainen e Hadid (2015)
	LBP e momento de cores	Patel, Han e Jain (2016b)
	LBP/SURF	Boulkenafet, Komulainen e Hadid (2018)
Informações de frequência	Altas frequências	Li et al. (2004)
	DoG	Zhang et al. (2012)
Ruídos	FFT e ritmo visual	Pinto et al. (2012)
	LBP/SIFT	Patel et al. (2015)
Outras	Brilho, borramento e cores	Wen, Han e Jain (2015)
	Patches aleatórios	Akhtar e Foresti (2016)
Métodos baseados em Aprendizado em Profundidade		
Tons de cinza/cor	CNN (AlexNet)	Yang, Lei e Li (2014)
	CNN (3 camadas)	Menotti et al. (2015)
	CNN (VGG-Face)	Li et al. (2016)
	CNN (GoogleNet)	Patel, Han e Jain (2016a)
	CNN (VGG-16)	Lucena et al. (2017)
	CNN (*Patches* aletórios)	Atoum et al. (2017)
Características temporais	CNN+LSTM	Xu, Li e Deng (2015)
	CNN+LSTM	Sun, Sun e Li (2018)
Características de difusão	Difusão não-linear+CNN	Alotaibi e Mahmood (2017)
Ruído	CNN (Mapas de ruídos)	Jourabloo, Liu e Liu (2018)
Profundidade/Fluxo sanguíneo	CNN+LSTM	Liu, Jourabloo e Liu (2018)

Capítulo 5

MATERIAL E METODOLOGIA

Neste capítulo descreve-se, de maneira geral, as abordagens eficientes baseadas em Aprendizado em Profundidade para detecção de *spoofing* facial 2D propostas, a fim de atender os objetivos desta tese, bem como o material e métricas utilizados em suas proposições e avaliações.

5.1 Abordagens Propostas

Como mencionado, tem-se como objetivo desta tese, a proposta de novas abordagens eficientes para a detecção de *spoofing* facial 2D a partir de duas arquiteturas principais de redes neurais profundas: as redes baseadas nas Máquinas de Boltzmann Restritas (*Restricted Boltzmann Machines* - RBM) (SMOLENSKY, 1986; HINTON, 2002) e as Redes Neurais de Convolução (*Convolutional Neural Networks* - CNN) (LECUN et al., 1998), arquiteturas detalhadas no Capítulo 3. A Seção 5.1.1 trata das propostas baseadas em RBMs e a Seção 5.1.2 das propostas baseadas em CNNs.

5.1.1 Abordagens Baseadas em RBMs

A fim de atender aos objetivos específicos (i) e (iv) desta tese, que consistem na avaliação de redes neurais baseadas em RBMs e na proposição de técnicas de otimização para tais arquiteturas aplicadas à detecção de *spoofing* facial, até mesmo dada sua baixa exploração no contexto de classificação de imagens, no Capítulo 6, emprega-se uma RBM tradicional (generativa), apenas extraindo características das faces e alimentando um classificador a fim de detectar ataques, com o intuito de verificar se tal rede neural realmente consegue capturar informações discriminativas das faces reais e sintéticas a ela apresentadas.

Nos Capítulos 7 e 8, emprega-se uma RBM discriminativa, capaz de classificar as faces

por si só, bem como propõe-se ainda um versão profunda de tal rede neural, a DDRBM (*Deep Discriminative Restricted Boltzmann Machine*), a fim de avaliar o ganho de desempenho ao se trabalhar com características mais profundas na detecção de *spoofing* facial. Em todos os casos trabalha-se com versões de textura das imagens faciais dada a melhoria de desempenho apresentadas em outros trabalhos, como no de Maatta, Hadid e Pietikainen (2011).

Dentre as motivações para para o emprego das RBMs na detecção de *spoofing* facial, pode-se mencionar sua grande eficiência computacional, permitindo treinar e testar as abordagens propostas em arquitetura computacional tradicional, isto é, utilizando uma CPU Intel i7 comum, sem a necessidade do uso de GPUs.

5.1.2 Abordagens Baseadas em CNNs

Visando atender aos objetivos específicos (ii) e (iv), que propõem a avaliação das CNNs na detecção de *spoofing* facial, bem como de técnicas de otimização de tais arquiteturas, no Capítulo 9, propõe-se uma CNN compacta, inspirada no trabalho de LeCun et al. (1998), com o intuito de extrair características convolucionais também de textura das faces, a fim de avaliar sua superioridade em relação às características de textura *handcrafted* (e às características de textura extraídas pela RBMs). Nos Capítulos 10 e 11, propõe-se novas arquiteturas de CNNs, trabalhando-se com bases de imagens maiores (imagens no espaço de cores RGB), juntamente com técnicas de otimização para as mesmas de forma a torná-las menos custosas computacionalmente, podendo-se citar: (i) técnica de Tranferência de Aprendizado (*Transfer Learning* - TL); e (ii) expansão de CNN compacta em largura ao invés de profundidade.

No Capítulo 12, além de atender aos objetivos específicos (ii) e (iv), contempla-se também os objetivos específicos (iii) e (v), que tratam do uso de informações locais da face para um treinamento mais eficiente das CNNs bem como do uso de características temporais na detecção de *spoofing* facial. Propõe-se um novo algoritmo de treinamento eficiente baseado em *patches* faciais locais, bem como incorpora-se redes neurais temporais como a C3D (TRAN et al., 2015) e a LSTM (*Long Short-Term Memory*) (HOCHREITER; SCHMIDHUBER, 1997) à CNN proposta, a fim de aprender características baseadas em múltiplos *frames*, ao se trabalhar com vídeos, com o intuito de tornar a abordagem ainda mais acurada.

Vale mencionar que em todos os casos, dadas as técnicas de otimização propostas para as CNNs, pode-se trabalhar utilizando-se apenas de uma simples GPU GeForce GTX 980, com somente 4 GB de memória dedicada.

5.2 Métricas de Avaliação

Em relação às métricas para avaliação das abordagens propostas, tanto baseadas em RBMs como em CNNs, em termos de acurácia são analisadas as seguintes medidas: acurácia global (ACC - *Accuracy*), curva ROC (*Receiver Operating Characteristcs*), AUC (*Area Under the Curve*), EER (*Equal Error Rate*) e HTER (*Half-Total Error Rate*).

A acurácia global corresponde ao total de faces reais e sintéticas corretamente classificadas em dado experimento. A curva ROC, por padrão, é obtida ao variar-se (no intervalo $[0;1]$) o limiar de aceitação do sistema, responsável por classificar as faces como reais ou sintéticas, e observar-se as taxas de TDR (*True Detection Rate*) e FDR (*False Detection Rate*) para cada valor do limiar. Considerando os valores de FDR no eixo das abscissas e os valores de TDR no eixo das ordenadas, quanto mais elevada a curva, melhor. A taxa FDR indica o percentual de faces reais errôneamente classificadas como sintéticas. Já a TDR, o percentual de faces sintéticas corretamente classificadas.

Da curva ROC, são extraídos os valores de AUC, que corresponde à área sob a curva ROC (quanto maior, melhor) bem como o EER, que corresponde ao ponto do sistema onde as taxas FDR (*False Detection Rate*) e FNDR (*False Non-Detection Rate* - percentual de faces sintéticas classificadas como reais) são iguais. Após fixar o limiar de aceitação das faces como reais ou sintéticas do sistema (ou tomá-lo como $0,5$) pode-se estimar a medida HTER, que corresponde à média entre FDR e FNDR, dado o limiar fixado.

Vale notar que ainda existe a taxa TNDR (*True Non-Detection Rate*), que corresponde ao percentual de faces reais corretamente classificadas. A TNDR pode também ser referenciada, em trabalhos publicados há mais tempo, como TAR (*True Acceptance Rate*) e a FNDR como FAR (*False Acceptance Rate*). Alguns autores constroem a curva ROC a partir das taxas de TAR e FAR. Por fim, a taxa FDR também pode ser referenciada, em trabalhos mais antigos, como FRR (*False Rejection Rate*).

Em relação às métricas de avaliação de desempenho em termos de eficiência, em alguns casos (como no Capítulo 11) estima-se a quantidade de operações de multiplicação requeridas por cada modelo neural para um *forward pass* da imagem facial na rede profunda a fim de identificar os mais eficientes. Em outros, por falta de informações na descrição das abordagens estado-da-arte profundas, apenas o *hardware* utilizado para treino e teste dos modelos pode ser usado como parâmetro de complexidade. Em alguns casos sequer detalhes da arquitetura ou do *hardware* utilizado para treinamento e teste da arquitetura profunda da literatura são fornecidos, dificultando a comparação em termos de eficiência computacional.

No caso do Capítulo 12, avalia-se a eficiência no treinamento dos modelos neurais (em número de iterações necessárias para a convergência), dada a abordagem de treinamento eficiente proposta baseada em *patches* locais.

5.3 Bases de Imagens

Uma das primeiras bases de imagens propostas para a detecção de *spoofing* facial é a NUAA (*Nanjing University of Aeronautics and Astronautics*) (TAN et al., 2010), a qual apresenta 3.491 imagens em tons de cinza obtidas de faces reais e sintéticas (fotografias) para treinamento (1.743 imagens de faces reais e 1.748 imagens de faces sintéticas) e 9.123 imagens em tons de cinza para teste de métodos anti-*spoofing* (3.362 faces reais e 5.761 faces sintéticas). As imagens da base foram obtidas por meio de diferentes *webcams* e a partir das faces de variados indivíduos em termos de idade e gênero bem como em diferentes sessões. Exemplos das imagens da base NUAA são mostrados na Figura 5.1. Percebe-se que é difícil, até mesmo visualmente, diferenciar as imagens faciais normalizadas reais e sintéticas da base.

Figura 5.1: Exemplos de imagens da base de *spoofing* facial NUAA. As imagens faciais normalizadas que compõem a base são mostradas em tons de cinza juntamente com as respectivas imagens dos momentos de suas capturas. Fonte: Adaptada de Tan et al. (2010).

Outra base de imagens bastante referenciada na literatura corresponde à Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012), que contém 360 vídeos (com *frames* em RGB) para treinamento (60 vídeos de faces reais e 300 de sintéticas), 360 vídeos para validação (60 de faces reais e 300 de sintéticas), a fim de calibrar o limiar do sistema usado para determinar se uma dada imagem facial (extraída de um quadro de vídeo) é real ou sintética, e um conjunto

de vídeos de teste com 80 vídeos de usuários reais e 400 vídeos de faces sintéticas. A Figura 5.2 mostra exemplos de faces reais e sintéticas presentes nos quadros dos vídeos do conjunto de treinamento desta base de imagens. Pode-se observar que também é difícil diferenciá-las, especialmente devido à alta variabilidade intraclasse e alta semelhança interclasses.

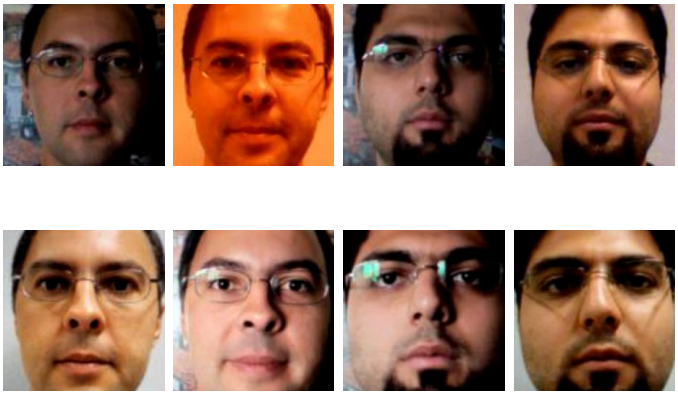


Figura 5.2: Exemplos de faces reais (imagens da linha superior) e sintéticas (imagens da linha inferior) nos vídeos da base Replay-Attack. Fonte: Adaptada de Chingovska, Anjos e Marcel (2012).

A base CASIA FASD (*Chinese Academy of Sciences - Institute of Automation - Face Anti-Spoofing Database*) (ZHANG et al., 2012), também empregada nos experimentos desta tese, por sua vez, apresenta vídeos (com *frames* em RGB) de 50 sujeitos, 12 vídeos por pessoa, sendo 3 reais e 9 de faces sintéticas. A base de dados é dividida em conjunto de treinamento (20 pessoas, ou seja, 240 vídeos) e conjunto de teste (30 indivíduos - 360 vídeos). Não há conjunto de validação definido explicitamente para essa base de imagens. A Figura 5.3 ilustra algumas imagens do processo de captura das faces da base CASIA.

Figura 5.3: Imagens da base CASIA FASD. Fonte: Adaptada de Zhang et al. (2012).

Capítulo 6

DETECÇÃO DE ATAQUES E MÁQUINAS DE BOLTZMANN RESTRITAS

Neste capítulo apresenta-se os resultados iniciais desta tese de doutorado obtidos a partir de um estudo realizado com as Máquinas de Boltzmann Restritas (*Restricted Boltzmann Machines* - RBM) (SMOLENSKY, 1986; HINTON, 2002) na detecção de *spoofing* facial a fim de avaliar o poder de aprendizado destas redes neurais em tal tarefa e prosseguir, posteriormente, com estudos mais aprofundados. Para tanto, compara-se uma pequena RBM, extraindo características de textura das faces por meio de seus neurônios escondidos, com a técnica PCA (*Principal Component Analysis*) (PEARSON, 1901; HOTELLING, 1933) também encontrando vetores de características enxutos para as faces, em ambos os casos utilizando uma SVM para classificar os vetores de características reduzidos encontrados e, consequentemente, classificar as faces como sendo reais ou sintéticas. O conteúdo deste capítulo descreve o artigo **"A Restricted Boltzmann Machine-Based Approach for Robust Dimensionality Reduction"**, publicado e apresentado no *Workshop* de Visão Computacional (WVC) de 2017.

6.1 Textura e Detecção de *Spoofing* Facial

Como para a detecção e o reconhecimento facial, a textura desempenha um papel importante na detecção de *spoofing* em sistemas de reconhecimento pela face (MAATTA; HADID; PIETIKAINEN, 2011). Entre os principais descritores de textura, o LBP (*Local Binary Pattern*) (OJALA; PIETIKAINEN; HARWOOD, 1994, 1996) e suas variações são os mais usados para tal fim devido aos bons resultados que permitem alcançar e aos seus eficientes funcionamentos, baseados em simples comparações de valores de pixels adjacentes.

Em síntese, e como mencionado na Seção 4.1, dado um sistema de vizinhança $\{P,R\}$, onde

P corresponde ao número de vizinhos a serem considerados e R ao raio da vizinhança, isto é, distância euclidiana entre o pixel central e seus vizinhos, o descritor LBP atua comparando o valor de tom de cinza de cada pixel p da imagem com a intensidade de seus vizinhos e associando um novo valor inteiro (também em escala de cinza) a p com base em tais comparações.

Em alguns métodos *handcrafted*, um histograma é construído com base nos valores dos pixels obtidos pelo LBP a partir da imagem facial original e, dado um conjunto de imagens conhecidas e seus respectivos histogramas, um classificador é treinado para prever as classes de novas faces (reais ou sintéticas). Alguns trabalhos, na tentativa de capturar informações de texturas locais na imagem, dividem ainda a face em *patches* e geram um histograma para cada um deles, os quais são concatenados ao final, melhorando ainda mais os resultados (MAATTA; HADID; PIETIKAINEN, 2011).

6.2 Abordagem Proposta

Como visto na Seção 3.3, as Máquinas de Boltzmann Restritas (RBM) (SMOLENSKY, 1986; HINTON, 2002) são modelos generativos de redes neurais, não apresentando neurônios *softmax*, por exemplo, para classificação de amostras. Neste trabalho, dadas imagens faciais em tons de cinza e seus histogramas de valores de pixels após a conversão pelo LBP, propõe-se avaliar, partindo de uma GB-RBM de dimensões reduzidas, sua capacidade de aprender informações de tais histogramas e extrair características compactas para compor os vetores de características das respectivas faces, vetores posteriormente apresentados a um classificador SVM a fim de detectar ataques. Compara-se a extração de características e redução da dimensionalidade dos histogramas LBP das imagens obtidas pela GB-RBM com a redução da dimensionalidade obtida por meio da técnica PCA (*Principal Component Analysis*) (PEARSON, 1901; HOTELLING, 1933), também alimentando outra SVM, no intuito de verificar qual dos métodos, GB-RBM ou PCA, aprende as melhores características dos histogramas.

Dado o conjunto de histogramas LBP das imagens faciais reais e sintéticas de treinamento (normalizados para média zero e variância unitária em todas as suas posições), alimenta-se então a GB-RBM a fim de treiná-la para extrair as características mais relevantes de tais estruturas e bem representar os histogramas em um espaço de menor dimensionalidade.

Depois de treinar a GB-RBM, como mostrado na Figura 6.1 de forma simplificada (pequenos vetores de características e poucos neurônios são mostrados), os histogramas originais são apresentados novamente a tal estrutura neural, realiza-se um *forward pass* dos mesmos na rede e as probabilidades de ativação dos neurônios de sua camada escondida são tomadas como

vetores de características de menor dimensionalidade para eles, os quais são apresentados então à SVM.

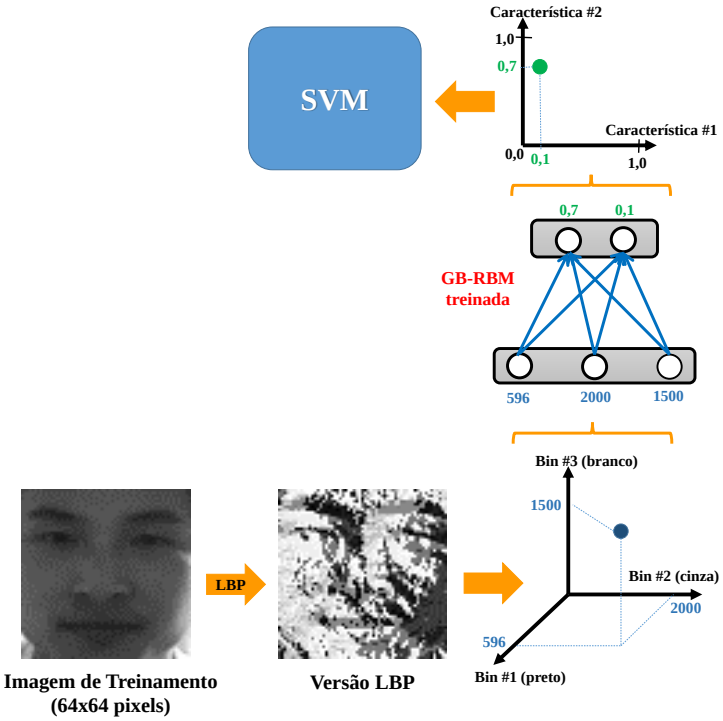


Figura 6.1: Dada uma imagem facial e seu histograma LBP, após treinar a GB-RBM, um vetor de menor dimensão é obtido ao passar o histograma pela rede e observar as probabilidades de ativação de seus neurônios escondidos. Tal vetor reduzido é então apresentado à SVM. Fonte: Elaborada pelo autor.

6.3 Experimentos, Resultados e Discussão

Avaliou-se a abordagem proposta sobre a base de imagens NUAA (*Nanjing University of Aeronautics and Astronautics*) (TAN et al., 2010), apresentada na Seção 5.3. Dadas as imagens de treinamento e seus histogramas LBP (considerando vizinhança $R = 1$ e $P = 8$, isto é, 3×3), treinou-se a GB-RBM por 100 épocas (*batches* com 100 amostras), utilizando *learning rate* de $0,0005$, *momentum* de $0,5$ (e $0,9$ nas 5 últimas épocas), bem como *weight cost* ou *weight decay* (fator de regularização da aprendizagem dos pesos sinápticos) de $0,0002$. Os pesos da GB-RBM foram inicializados seguindo uma distribuição gaussiana com média zero e desvio-padrão de $0,001$ e os seus *biases* foram inicializados em zero. Após este processo, fez-se um *forward pass* de todos os histogramas de treinamento pela rede a fim de obter suas versões reduzidas e treinar então a SVM (com *kernel* de base radial). Dados os vetores reduzidos de treinamento empregou-se uma abordagem de validação cruzada com 5 *folds* para treinar o classificador. Então, dadas as imagens de teste e seus histogramas LBP, repetiu-se o processo de redução de

dimensionalidade e apresentação dos vetores reduzidos à SVM a fim de classificá-los e calcular a acurácia para a abordagem sobre o conjunto de teste.

O processo de redução de dimensionalidade dos histogramas pela GB-RBM e classificação pela SVM (incluindo o treinamento destas estruturas) foi repetido 9 vezes. Em cada uma delas a GB-RBM possuía menos neurônios escondidos a fim de analisar a acurácia da SVM frente ao grau de compactação dos histogramas LBP pela rede neural (começando com 256 neurônios escondidos - tamanho original dos histogramas). Ao mesmo tempo, aplicou-se 9 vezes a técnica PCA sobre os histogramas LBP de treinamento com o intuito de reduzi-los a vetores menores, de mesmas dimensões que os gerados pela GB-RBM, e treinar outra SVM de forma análoga a fim de poder comparar o desempenho de tal técnica com a baseada na GB-RBM. A Figura 6.2 mostra a acurácia das duas abordagens na base de teste (compactação pela GB-RBM ou pela PCA e classificação pelas SVMs) em cada uma das 9 iterações.

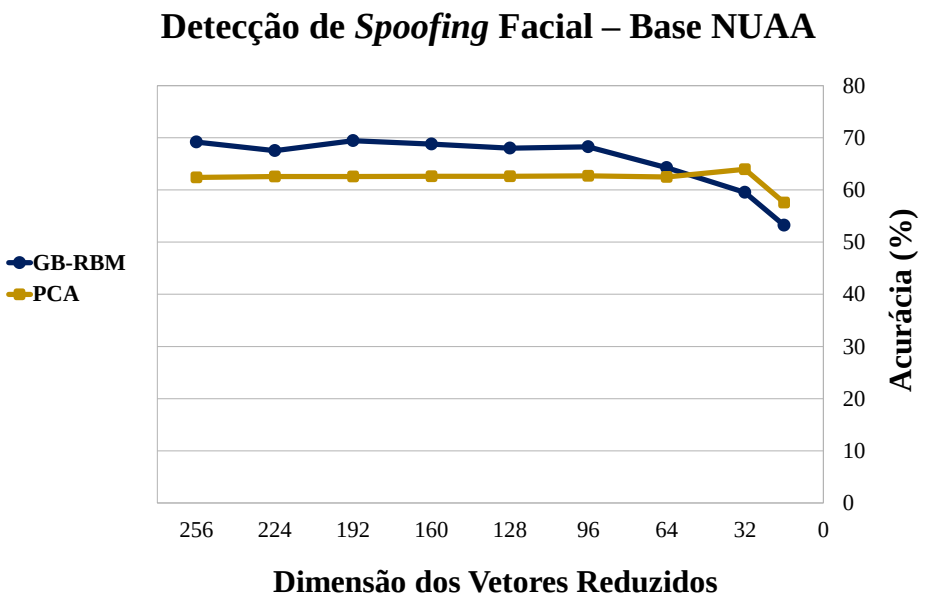


Figura 6.2: Valores de acurácia para cada iteração dos experimentos, variando o tamanho dos vetores de características reduzidos pela GB-RBM e pela PCA. Fonte: Elaborada pelo autor.

Pode-se perceber que a GB-RBM conseguiu capturar melhores caracterísicas dos histogramas LBP do que a técnica PCA em todos os casos, exceto quando a compactação é muito grande e a rede perde sua capacidade de aprendizado devido ao fato de haver poucos neurônios escondidos para a complexidade do problema sendo tratado. Vale ainda observar que a RBM, implementada a partir dos códigos disponibilizados por Salakhutdinov (2015), pôde ser treinada e testada em uma CPU Intel i7 em questão de poucos minutos, sem a necessidade de placas gráficas (GPUs). Utilizou-se a implementação da SVM da biblioteca LibSVM (CHANG;

LIN, 2011), operando também sobre CPU.

6.4 Conclusões

Com base nos experimentos realizados percebe-se que a GB-RBM capturou características mais relevantes dos histogramas LBP do que a técnica PCA. Deste modo, evidencia-se a alta capacidade de aprendizado e extração de características das RBMs sobre os dados a elas apresentados. Obviamente, representar as imagens faciais apenas por seus histogramas LBP gerais não é a melhor opção para a detecção de ataques dada a perda de muitas informações importantes, necessárias para uma classificação mais eficaz das faces, como informações espaciais (resultando em acurácias não tão elevadas).

Observou-se que a SVM se comportou melhor sobre os vetores compactados pela GB-RBM do que pela PCA, comprovando o bom desempenho de tais redes. Esse resultado corrobora ainda a hipótese de que as GB-RBMs podem ser usadas nos problemas de redução de dimensionalidade, possivelmente operando melhor que técnicas tradicionais na literatura, como a PCA.

Capítulo 7

DETECÇÃO DE ATAQUES E RBMS DISCRIMINATIVAS

Neste capítulo apresenta-se a abordagem proposta nos artigos **"Detecção de Ataques a Sistemas de Reconhecimento Facial: uma Abordagem Baseada nas Máquinas de Boltzmann Restritas"**, publicado e apresentado no Encontro Regional de Matemática Aplicada e Computacional (ERMAC) de 2017, e em sua versão estendida, sob o título de **"Detecção de *Spoofing* Facial: uma Abordagem Baseada nas Máquinas de Boltzmann Restritas"**, publicada na Revista Eletrônica Paulista de Matemática em 2017, para detecção de *spoofing* facial a partir das Máquinas de Boltzmann Restritas discriminativas (LAROCHELLE; BENGIO, 2008; HINTON, 2012). Dados os bons resultados apresentados pelas RBMs no capítulo anterior, explora-se uma nova abordagem baseada no descritor LBP (*Local Binary Pattern*) (OJALA; PIETIKAINEN; HARWOOD, 1994, 1996) e nas Máquinas de Boltzmann Restritas discriminativas, RBMs modificadas que já atuam como classificadores, até então não avaliadas no contexto de detecção de *spoofing* facial, para a extração das características de textura mais relevantes das faces apresentadas e detecção de ataques de forma acurada e eficiente. Resultados obtidos sobre a base de imagens NUAA (TAN et al., 2010) indicam que a abordagem proposta apresenta boas taxas de acerto bem como eficiência computacional.

7.1 Abordagem Proposta

Baseado nas RBMs tradicionais, que são por natureza modelos generativos, propõe-se o uso de uma versão adaptada de tais redes neurais, seguindo Hinton (2012) e Montavon, Orr e Müller (2012), tornando-as modelos discriminativos, podendo assim aprender as características mais relevantes das faces bem como já classificá-las em reais ou sintéticas. Propõe-se uma GB-RBM

discriminativa, que na verdade é idêntica à arquitetura de uma GB-RBM tradicional, exceto pelo fato de que são inseridos, na camada visível da rede, dois neurônios (no caso deste trabalho) especiais que servem para representar a classe do vetor (imagem) de entrada: se trata-se de face real ou sintética.

No treinamento, conforme mostra a Figura 7.1, após converter cada imagem facial conhecida para sua versão baseada no LBP e normalizá-la (elementos seguindo média zero e variância unitária), os valores de seus pixels servem de entrada para os nós visíveis tradicionais da GB-RBM e os dois valores referentes à classe da imagem ("1 e 0" se face real, ou "0 e 1" se sintética) são considerados como ativações para os dois neurônios especiais. Dadas todas as imagens de treinamento e suas respectivas classes, procede-se então com a Divergência Contrastiva (CD - *Contrastive Divergence*) (HINTON, 2002) a fim de ajustar os parâmetros, isto é, o modelo interno da rede, à distribuição de tais amostras conhecidas (diminuindo a energia do sistema para as imagens de treinamento e suas classes) a fim de poder classificar amostras de teste posteriormente.

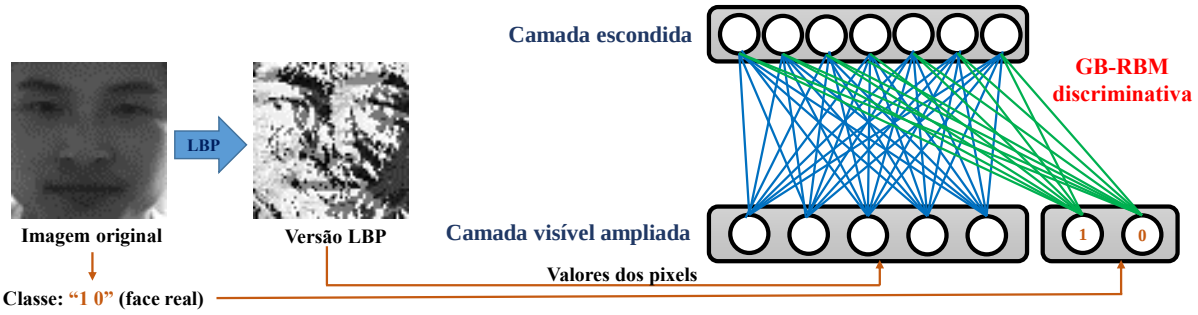


Figura 7.1: Treinamento da arquitetura proposta: os tons de cinza das imagens de treinamento (pré-processadas pelo LBP e normalizadas) e os dois valores referentes às suas classes alimentam a GB-RBM discriminativa. Fonte: Elaborada pelo autor.

Para determinar a classe de uma dada imagem facial de teste, aplica-se o LBP sobre a mesma, servindo seus pixels (após normalização) de entrada para os nós visíveis tradicionais da GB-RBM, e verifica-se qual configuração dos dois neurônios especiais (se "1 e 0", ou "0 e 1") apresenta maior probabilidade de ocorrência segundo a distribuição de probabilidades aprendida pela rede. Para isto, calcula-se a energia livre (HINTON, 2002, 2012) fornecida pela GB-RBM discriminativa ao se apresentar a imagem de teste e os valores "1 e 0" e ao se apresentar a mesma imagem e os valores "0 e 1" à sua camada visível. A configuração dos dois nós adicionais responsável pela energia livre mínima (a rede "acredita" que tal configuração seja a mais adequada com base no que aprendeu) indica a classe da imagem de teste.

Vale observar que o modelo de RBM discriminativa pode ainda aplicar funções de ativação *softmax* nos neurônios representantes das classes da imagem bem como trabalhar com uma ma-

triz de pesos sinápticos separada entre eles e os neurônios escondidos (LAROCHELLE; BENGIO, 2008). Entretanto, a arquitetura simplificada aqui apresentada torna os trabalhos ainda mais eficientes, tratando os neurônios da classe como neurônios visíveis comuns (sem ativação *softmax* e com uma única matriz de pesos de conexões sinápticas entre a camada visível e escondida da GB-RBM discriminativa - englobando todos os neurônios da camada visível), e ainda assim produzindo bons resultados.

7.2 Experimentos, Resultados e Discussão

A arquitetura proposta foi avaliada sobre a base de imagens em tons de cinza NUAA (TAN et al., 2010), apresentada na Seção 5.3. A GB-RBM discriminativa avaliada apresentava 4.098 neurônios visíveis (4.096 para os pixels das imagens de tamanho 64×64 - redimensionava-se as imagens para 66×66 pixels de modo que, ao aplicar o LBP, com vizinhança $R = 1$ e $P = 8$, voltassem a possuir 64×64 pixels, e 2 neurônios adicionais para identificar suas classes) e 2.000 neurônios escondidos. A rede foi treinada sobre 6.982 imagens faciais (3.491 imagens de treinamento originais da base acrescidas de suas versões equalizadas com base em seus histogramas) por 10 épocas (com *batches* de 100 imagens), utilizando *learning rate* de $0,001$, *momentum* de $0,5$ (e $0,9$ nas 5 últimas épocas), bem como *weight cost* ou *weight decay* (fator de regularização) de $0,0002$. Os pesos da GB-RBM discriminativa foram inicializados seguindo uma distribuição gaussiana com média zero e desvio-padrão de $0,001$. Todos seus *biases* foram inicializados em zero.

O método proposto obteve acurácia de $93,6\%$ na classificação das faces sobre as 9.123 imagens de teste (considerando o limiar de aceitação das faces como verdadeiras de 0,5), enquanto a técnica proposta pelos próprios autores da base NUAA, que se vale de descritores *handcrafted*, obteve acurácia de $92,0\%$. Em complemento, as taxas de Falsa Não-Detecção (FNDR - *False Non-Detection Rate*) e Falsa Detecção (FDR - *False Detection Rate*) da abordagem proposta foram de apenas $7,24\%$ e $4,97\%$, respectivamente. A GB-RBM discriminativa, implementada a partir dos códigos disponibilizados por Salakhutdinov (2015), pôde ser treinada e testada sobre uma simples CPU Intel i7 em questão de poucos minutos, sem a necessidade de processamento paralelo (GPU).

7.3 Conclusões

Com base nos resultados obtidos, pode-se confirmar que as RBMs discriminativas podem aprender e extrair importantes características discriminativas a partir de amostras conhecidas de diferentes classes dos problemas com que lidam, neste caso, detecção de *spoofing* facial, apesar de até então não terem sido usadas para tal fim, propiciando um bom desempenho mesmo em tarefas complexas como o teste sobre a base NUAA, onde há grande similaridade interclasses e variabilidade intraclasse.

A rede apresentada, baseada nas Máquinas de Boltzmann Restritas, ao aprender boas características de textura a partir de informações extraídas pelo descritor *Local Binary Pattern* (LBP) das faces reais e sintéticas, obteve acurácia superior na detecção de ataques à do método proposto pelos próprios autores da base de imagens analisada, que se vale apenas de descritores *handcrafted*.

Capítulo 8

DETECÇÃO DE ATAQUES E RBMS DISCRIMINATIVAS PROFUNDAS

Como mostrado no capítulo anterior, entre as arquiteturas de redes neurais que podem ser aplicadas na detecção de *spoofing* facial estão a Máquinas de Boltzmann Restritas (RBM) discriminativas (LAROCHELLE; BENGIO, 2008; HINTON, 2012) que, além de eficientes, alcançam bons resultados na detecção de ataques. No entanto, sabe-se que redes neurais mais profundas apresentam melhores desempenhos em muitas tarefas (LECUN; BENGIO; HINTON, 2015). Neste contexto, neste capítulo propõe-se um novo modelo neural denominado de Máquina de Boltzmann Restrita Discriminativa e Profunda (DDRBM - *Deep Discriminative Restricted Boltzmann Machine*), com duas RBMs empilhadas, aplicado na detecção de ataques de *spoofing* facial. Resultados também sobre a tradicional base de imagens NUAA (TAN et al., 2010) mostram uma melhora significativa de desempenho na tarefa de detecção de *spoofing* quando comparado ao uso de uma única RBM discriminativa, sendo inclusive mais estável em relação a variações nos dados (ordem de apresentação) bem como aos valores de inicialização dos pesos do modelo no treinamento.

Este capítulo refere-se ao conteúdo do artigo **"Deep Discriminative Restricted Boltzmann Machine (DDRBM) for Robust Face Spoofing Detection"**, publicado em 2019 no periódico *Progress in Human-Computer Interaction*.

8.1 Abordagem Proposta

Como dito, uma RBM (SMOLENSKY, 1986; HINTON, 2002) corresponde a uma rede neural estocástica baseada em energia e apresenta duas camadas: a visível e a escondida. Existem conexões apenas entre neurônios de camadas distintas. Elas aprendem distribuições de pro-

babilidade dadas amostras dessas distribuições e, inserindo-se neurônios especiais na camada visível para representar as diferentes classes das amostras por meio de *labels* binários, dá-se origem ao modelo chamado de RBM discriminativa (LAROCHELLE; BENGIO, 2008; HINTON, 2012), podendo-se classificar novas amostras com base na sua similaridade às distribuições aprendidas das classes do problema.

Neste trabalho, de maneira similar às DBNs (HINTON; OSINDERO; TEH, 2006) e DBMs (SALAKHUTDINOV; HINTON, 2009), redes profundas baseadas nas RBMs, propõe-se uma nova arquitetura denominada *Deep Discriminative Restricted Boltzmann Machine* (DDRBM), também composta por uma pilha de RBMs, porém de treinamento mais simples: para a classificação de amostras não é necessário treinar a estrutura e formar uma MLP (*Multilayer Perceptron*) (RUMELHART; HINTON; WILLIANS, 1986) ao final (que também precisa passar por uma etapa de *fine-tuning*), uma vez que a RBM discriminativa do topo da DDRBM já realiza a classificação. A DDRBM é composta por uma pilha de 2 RBMs (pode-se acrescentar mais RBMs no meio da pilha), sendo a da base uma GB-RBM tradicional (generativa) para capturar as informações dos pixels das imagens (tons de cinza) e a do topo uma BB-RBM discriminativa, capaz de aprender características mais profundas do que quando se utiliza uma única RBM e classificar as faces de forma mais robusta (a BB-RBM discriminativa do topo é treinada sobre as características da camada escondida da RBM da base, aprendendo traços de alto nível para a classificação).

Depois de aplicar o descritor LBP (*Local Binary Pattern*) (OJALA; PIETIKAINEN; HARWOOD, 1994, 1996) nas imagens faciais de treinamento e obter suas representações baseadas em textura, os valores de seus pixels são normalizados para a média zero e a variância unitária (considerando todas as imagens de textura de treinamento) para servir como entrada para a GB-RBM da base, que é, então, treinada por meio do método da Divergência Contrastiva (CD - *Contrastive Divergence*) (HINTON, 2002), a fim de aprender a distribuição dos vetores de treinamento (imagens faciais), podendo capturar suas características relevantes (neste passo, não são empregadas informações sobre as classes das imagens faciais). Depois de treinar a GB-RBM, cada imagem facial (já convertida em sua versão baseada em LBP e com valores normalizados) passa através da GB-RBM inferior e as probabilidades de ativação dos neurônios da camada escondida da GB-RBM são tomadas como uma nova representação para a mesma, e são usadas para treinar a segunda RBM da pilha, a BB-RBM discriminativa. A BB-RBM discriminativa é treinada também por meio do método CD baseado nas novas representações das imagens faciais de treinamento (reais e sintéticas) e suas classes, se a face é real, os valores "1 e 0" são tomados como ativações dos dois neurônios adicionais de sua camada visível, e "0 e 1", caso contrário, como ilustrado na Figura 8.1.

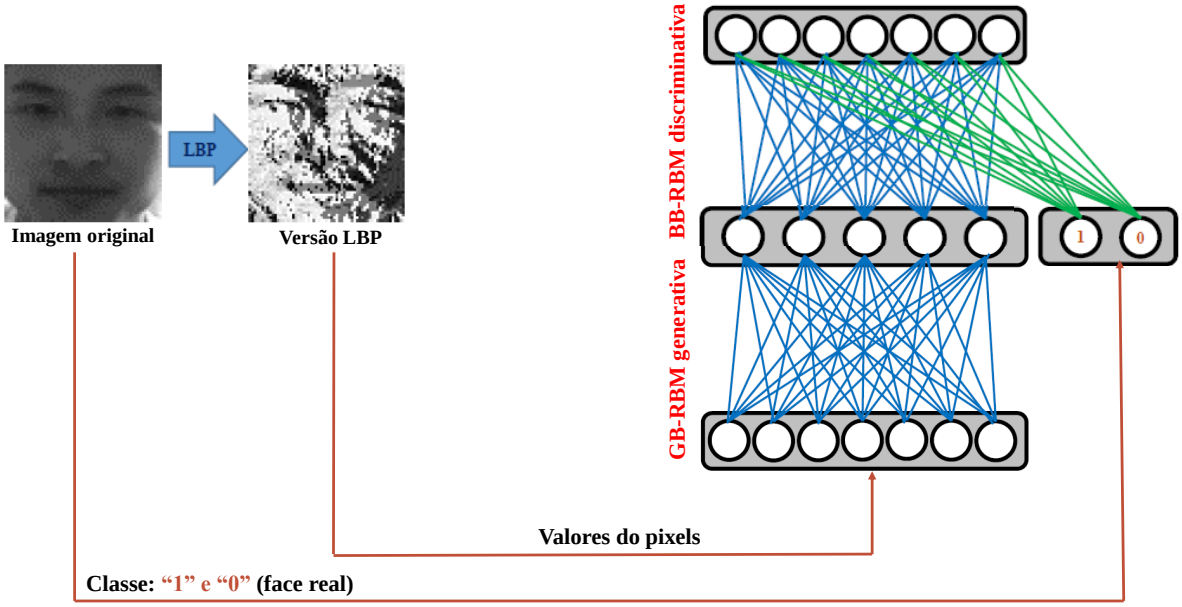


Figura 8.1: Arquitetura proposta da DDRBM. A BB-RBM discriminativa aprende a classificar imagens faciais convertidas pelo LBP (e normalizadas) a partir do treinamento baseado nas características extraídas pela camada escondida da GB-RBM das faces conhecidas e de seus *labels*. Fonte: Elaborada pelo autor.

Para determinar a classe de uma imagem facial de teste, sua representação baseada em LBP é gerada, os valores de seus pixels também são normalizados para média zero e variância unitária, e um *forward pass* de tal imagem baseada em textura normalizada é realizado na GB-RBM, obtendo a representação de alto nível da mesma. Então, tal vetor de característica de alto nível é apresentado como entrada para a BB-RBM discriminativa e é verificada qual configuração de seus dois neurônios adicionais de entrada (usados para representar a classe do vetor de entrada), ou seja, "1 e 0" (face real), ou "0 e 1" (face sintética), apresenta a maior probabilidade de ocorrência de acordo com a distribuição aprendida pela BB-RBM discriminativa. Para isto, como no capítulo anterior, a energia livre (HINTON, 2002, 2012) da BB-RBM discriminativa é calculada para ambas as configurações destes dois neurônios adicionais. A configuração com a menor energia livre indica a classe da imagem de teste.

8.2 Experimentos, Resultados e Discussão

A nova arquitetura proposta foi avaliada na base de imagens NUAA (TAN et al., 2010), apresentada na Seção 5.3. Por se tratar de base com imagens em tons de cinza, elas foram convertidas para suas versões LBP considerando um raio de vizinhança $R = 1$ e $P = 8$ pixels vizinhos (vizinhança 3×3). Diferentemente dos experimentos do capítulo anterior, para fins

de simplicidade, as imagens originais do conjunto de dados, com tamanho 64×64 pixel, não foram redimensionadas. Portanto, após a aplicação da LBP, suas representações resultantes apresentavam 62×62 pixels. Neste sentido, a GB-RBM da base do modelo DDRBM proposto apresentava $62 \times 62 = 3.844$ neurônios visíveis e 3.844 neurônios ocultos (a fim de manter a dimensionalidade do vetor de características para a segunda RBM da pilha e poder comparar a DDRBM com uma GB-RBM discriminativa simples com o intuito de verificar se havia realmente ganho de desempenho com o aumento de profundidade). A BB-RBM discriminativa do topo apresentava $3.844 + 2 = 3.846$ neurônios visíveis (dados os dois neurônios adicionais para os rótulos das imagens) e 2.000 neurônios ocultos.

Tanto a GB-RBM generativa quanto a BB-RBM discriminativa da DDRBM foram treinadas por 10 épocas com *momentum* de $0,5$ (e $0,9$ nas últimas 5 épocas), *weight cost* de $0,0002$ e *learning rate* de $0,001$ e $0,01$, respectivamente. Os pesos da GB-RBM foram inicializados seguindo uma distribuição gaussiana com média zero e desvio-padrão de $0,001$. Os pesos da BB-RBM discriminativa foram inicializados também com base em uma distribuição gaussiana, mas com média zero e desvio-padrão de $0,01$. Todos os *biases* foram inicializados em zero.

É importante esclarecer que antes de treinar as duas redes neurais empilhadas da DDRBM, realizou-se também um aumento do *dataset*, gerando uma segunda versão de cada imagem de treinamento aplicando uma função de equalização baseada no trabalho de Chen, Er e Wu (2006), os quais se valem da *Discrete Cosine Transform* (DCT), obtendo um total de 6.982 imagens de treinamento, a fim de melhorar tal processo. Também avaliou-se o desempenho de uma única GB-RBM discriminativa classificando as imagens da base NUAA para fins de comparação. Tal rede apresentava os mesmos $3.844 + 2 = 3.846$ neurônios visíveis (dois neurônios adicionais para os rótulos) e 2.000 neurônios ocultos e foi treinada nas 6.982 imagens faciais igualmente por 10 épocas, com *learning rate* de $0,001$, mesmos valores de *momentum* e *weight cost* da DDRBM. A GB-RBM discriminativa também foi inicializada seguindo uma distribuição gaussiana com média zero e desvio-padrão de $0,001$. A Tabela 8.1 mostra os resultados obtidos pela DDRBM proposta e a GB-RBM discriminativa simples na classificação das imagens de teste da base de dados NUAA. O experimento, para cada modelo, foi repetido 10 vezes a fim de verificar a robustez das estruturas a diferentes ordens de apresentação das imagens de treinamento e diferentes valores de inicialização obtidos das distribuições gaussianas.

Como se pode observar na Tabela 8.1, a DDRBM apresentou uma acurácia média maior (92,35%) do que a GB-RBM discriminativa sozinha (taxa de acurácia média de 91,34%) na detecção de ataques, bem como um desempenho mais estável nos 10 experimentos: desvio-padrão de 0,84%, contra 1,21% da GB-RBM discriminativa isolada. Estes resultados indicam

Tabela 8.1: Resultados de acurácia (%) nos 10 experimentos realizados para a DDRBM e a GB-RBM discriminativa isolada. A média das acurácias obtidas e o desvio-padrão dos resultados de cada modelo são exibidos ao fim da tabela. Valores máximos de cada abordagem estão destacados. Fonte: Elaborada pelo autor.

Experimento	GB-RBM	DDRBM
1	90,92	**93,82**
2	**92,67**	92,55
3	92,00	90,91
4	91,75	92,62
5	90,35	91,92
6	88,41	92,59
7	91,20	91,56
8	91,15	91,47
9	92,39	93,33
10	92,55	92,76
Média	**91,34**	**92,35**
desvio-padrão	**1,21**	**0,84**

que o aumento em profundidade e o trabalho com características de alto nível beneficiaram a detecção de *spoofing*. A DDRBM, implementada a partir dos códigos disponibilizados por Salakhutdinov (2015), pôde ser treinada e testada sobre uma CPU tradicional (Intel i7) em questão de poucos minutos (sem uso de GPU), dada a simplicidade da arquitetura (mais simples até que a DBN e DBM).

8.3 Conclusões

Neste capítulo foi proposta a Máquina de Boltzmann Restrita Discriminativa e Profunda (*Deep Discriminative Restricted Boltzmann Machine* - DDRBM) para detecção de *spoofing* facial. A DDRBM é composta por uma pilha de duas RBMs, uma GB-RBM tradicional na base e uma BB-RBM discriminativa no topo. Depois de treinar a GB-RBM da base, as versões baseadas em LBP das imagens faciais da base NUAA são passadas por ela e as probabilidades de ativação dos neurônios de sua camada escondida são usadas para treinar a segunda rede neural juntamente com os rótulos das respectivas imagens de treinamento.

Com os resultados obtidos nos 10 experimentos realizados pode-se notar que o aumento em profundidade do modelo DDRBM, em comparação com o modelo baseado em uma única GB-RBM discriminativa, e o uso de características de mais alto nível melhoraram as taxas de detecção de ataques, aumentando a taxa de acurácia média e diminuindo seu desvio-padrão. Apesar dos bons resultados obtidos com a DDRBM, das vantagens do uso de redes neurais

profundas baseadas em RBMs, como a grande eficiência computacional e a possibilidade de se utilizar um conjunto menor de amostras rotuladas para o treinamento das redes em alguns casos, decidiu-se iniciar novas investigações relacionadas à detecção de *spoofing* facial, porém valendo-se das Redes Neurais de Convolução (CNN - *Convolutional Neural Networks*) (LECUN et al., 1998), uma vez que foram encontrados na literatura vários trabalhos que mostravam resultados, em geral, bastante superiores das CNNs em tarefas de classificação de imagens, dadas suas operações 2D de convolução e amostragem bem como seu caráter intrinsecamente discriminativo. Nos próximos capítulos são apresentados os trabalhos realizados e os resultados obtidos referentes ao uso de CNNs para a detecção de *spoofing* facial.

Capítulo 9

DETECÇÃO DE ATAQUES E CARACTERÍSTICAS CONVOLUCIONAIS DE TEXTURA

A maioria das abordagens de anti-*spoofing* facial, em especial quando se trabalha com imagens em tons de cinza, como na base NUAA (TAN et al., 2010), extrai informações *handcrafted* de textura das faces a fim de detectar ataques. Grande parte delas emprega o descritor LBP (*Local Binary Pattern*) (OJALA; PIETIKAINEN; HARWOOD, 1994, 1996) e suas variações para tal fim. No entanto, resultados recentes indicam que características de mais alto nível (profundas) se apresentam mais robustas para tais tarefas complexas (LECUN; BENGIO; HINTON, 2015). Neste sentido, neste capítulo é apresentada uma nova abordagem para detecção de faces sintéticas em imagens em tons de cinza, que extrai características convolucionais de textura profunda das imagens por meio da integração do descritor LBP a uma Rede Neural de Convolução (CNN - *Convolutional Neural Network*) (LECUN et al., 1998). A nova abordagem proposta neste capítulo, bem como os resultados obtidos, foram publicados no artigo **"Deep Texture Features for Robust Face Spoofing Detection"**, apresentado no *IEEE International Symposium on Circuits and Systems* em 2017, e em sua versão ampliada, publicada no periódico *IEEE Transactions on Circuits and Systems II*.

9.1 LBPnet e n-LBPnet

Baseado no compacto e eficiente modelo de CNN conhecido por LeNet-5 (LECUN et al., 1998), neste trabalho uma nova arquitetura de CNN, denominada de LBPnet, é proposta, integrando o descritor LBP na primeira camada da LeNet-5, a fim de se extrair características de textura profundas, ao invés de histogramas *handcrafted*, a partir das imagens faciais e detectar *spoofing* facial de maneira mais robusta, sem deixar de ser eficiente.

A primeira camada da rede LBPnet incorpora informações do LBP da seguinte forma: a operação de convolução atua não apenas efetuando a convolução dos pesos dos respectivos *kernels* com os valores dos pixels da imagem, mas, antes disso, calculando novos valores para os pixels com base no LBP, ou seja, a convolução é executada nos valores LBP dos pixels e não em seus valores originais de tons de cinza. Isso melhora muito os resultados da rede neural proposta, já que o método herda o poder de detectar as pistas de *spoofing* facial do descritor LBP, mas em uma arquitetura mais profunda, trabalhando com características de textura de mais alto nível e mais robustas a ruídos, distorções nas imagens, dentre outros. A Figura 9.1 mostra a convolução da primeira camada da LBPnet.

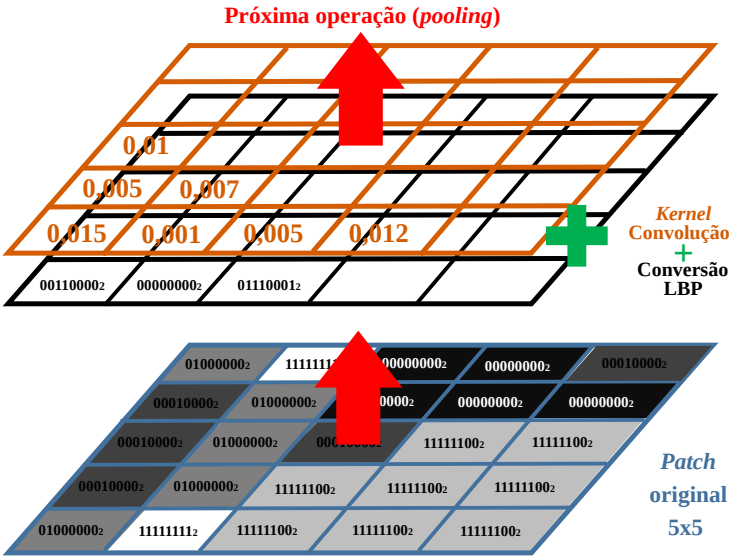


Figura 9.1: A operação de convolução na primeira camada da LBPnet atua convertendo a imagem para sua versão LBP (valores binários em preto são exibidos apenas para fins de elucidação, na verdade eles são automaticamente convertidos para decimais - tons de cinza) e realizando a convolução dos valores LBP com os respectivos *kernels* (exibidos em laranja). Fonte: Elaborada pelo autor.

A LBPnet apresenta a seguinte configuração, da base ao topo, herdada principalmente da LeNet-5: (i) duas camadas de convolução e *pooling* - a primeira camada é modificada, como dito, incorporando-se o descritor LBP na convolução; (ii) uma camada completamente conectada com funções de ativação ReLU (*Rectified Linear Unit*), que recebe os valores da camada imediatamente inferior e executa uma retificação (eliminação de valores negativos - que se tornam nulos) nos sinais; e (iii) uma camada totalmente conectada com dois neurônios responsáveis pela classificação das faces em reais ou sintéticas seguindo a função de ativação *softmax*.

Um esquema da arquitetura da LBPnet é mostrado na Figura 9.2. Dada uma imagem facial em tons de cinza detectada e normalizada (neste trabalho redimensionada para 66×66 pixels),

a operação de convolução na primeira camada, *CONV1*, encontra os valores baseados no LBP dos pixels e produz 20 saídas com tamanho 60×60, fazendo a convolução dos valores LBP com 20 *kernels* de convolução diferentes de tamanho de 5×5 - cada *kernel* gera uma saída e é aplicado com um passo de 1 pixel à imagem facial. Os valores dos *kernels* (que correspondem a pesos de conexões sinápticas) K_i, com $i = 1, 2, ..., 20$, são inicializados com pesos inversamente proporcionais ao tamanho das matrizes de entrada e saída da operação de convolução. Os *biases* dos neurônios responsáveis pelos *feature maps* e saída são inicialmente definidos como zero. Os *kernels* de pooling têm dimensão de 2×2 e passo de 2 pixels por vez. Aplica-se a operação de *pooling* máximo na LBPnet.

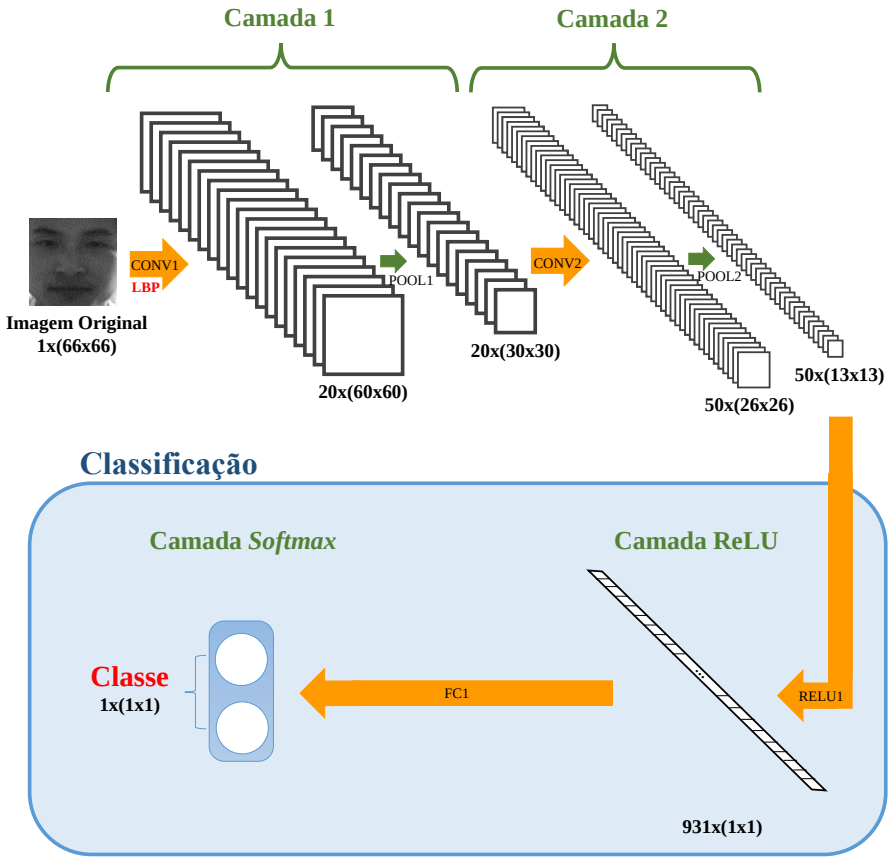


Figura 9.2: Arquitetura da LBPnet. As duas primeiras camadas realizam operações de convolução e *pooling*. *CONV1* atua não só efetuando a convolução, mas convertendo os valores dos pixels para suas versões LBP primeiro. As dimensões dos *feature maps* de saída são exibidas em preto junto a cada camada. Fonte: Elaborada pelo autor.

Uma segunda versão da LBPnet, denominada de LBPnet normalizada (n-LBPnet), também é proposta. A arquitetura n-LBPnet é bastante semelhante ao primeiro modelo, no entanto, uma etapa de normalização é incluída na segunda camada da rede entre a convolução e a operação de *pooling*. Primeiro, a saída de *CONV2*, ou seja, cada valor em dado *feature map* C_i, com $i = 1, 2, ..., 50$, é linearmente retificado, como na camada ReLU da rede. Depois, aplica-se

a normalização LRN (*Local Response Normalization*) (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), explanada na Seção 3.2, considerando 5 *feature maps* vizinhos por vez, $\alpha = 0,2$ e $\beta = 0,75$.

9.2 Experimentos, Resultados e Discussão

As redes propostas, LBPnet e n-LBPnet, foram avaliadas na tradicional base de detecção de *spoofing* facial NUAA (TAN et al., 2010), detalhada na Seção 5.3, com imagens obtidas de faces reais e sintéticas. As imagens normalizadas da base (com tamanho de 64×64 pixels) já são fornecidas pelos autores para tornar a comparação de métodos anti-*spoofing* mais justa, evitando que diferentes técnicas de pré-processamento afetem os resultados. Estas imagens normalizadas foram usadas nos experimentos. Elas foram apenas redimensionados para 66×66 pixels antes de alimentar a LBPnet e a n-LBPnet visto que o descritor LBP reduz as dimensões da imagem em 2 pixels, retornando ao tamanho original de 64×64 pixels (consideramos uma vizinhança de $P = 8$ e $R = 1$ para o LBP). Como observação, também aumentou-se o conjunto de treinamento (dobrando seu tamanho) considerando as 3.491 imagens normalizadas iniciais e suas versões equalizadas (simples equalização de histograma) para evitar a falta de dados durante o treinamento das redes e possível *overfitting*.

A LBPnet e a n-LBPnet foram implementadas usando o *framework* Caffe (JIA et al., 2014). Como dito, os pesos dos *kernels* (conexões sinápticas) de ambas as redes foram inicializados com valores inversamente proporcionais ao tamanho dos *feature maps* de entrada e saída em cada operação de convolução (*xavier filler*), enquanto os *biases* dos neurônios foram inicializados em zero. As redes foram treinadas por 200 iterações por meio do método Gradiente Descendente Estocástico (SGD - *Stochastic Gradient Descent*) (HINTON, 2012) com os seguintes parâmetros: 64 imagens por *batch* (imagens normalizadas com valores no intervalo $[0;1]$), *learning rate* inicial de $0,01$, *momentum* de $0,9$ e *weight decay* de $0,004$. A política de decaimento do *learning rate* foi "inv", ou seja, a taxa real de aprendizado para uma dada iteração de treinamento k é dada por:

$$lr_k = lr_0(1 + \gamma \cdot k)^{-\delta} \tag{9.1}$$

onde lr_0 corresponde ao *learning rate* inicial, $\gamma = 0,0001$ e $\delta = 0,75$.

A fim de verificar os desempenhos da LBPnet e da n-LBPnet e compará-los com os métodos estado-da-arte também avaliados sobre a base NUAA (TAN et al., 2010), diversas métricas, utilizadas em diferentes trabalhos, foram estimadas, como a curva ROC (*Receiver Operating Cha-*

racteristics), a AUC (*Area Under the Curve*), a EER (*Equal Error Rate*), a HTER (*Half-Total Error Rate*) e a Acurácia (ACC).

Em relação às curvas ROC, a Figura 9.3 mostra as taxas de *True Acceptance Rate* (TAR) *versus False Acceptance Rate* (FAR) dos seguintes métodos: (i) n-LBPnet; (ii) LBPnet; (iii) o método baseado no MLBP (*Multiscale* LBP) (MAATTA; HADID; PIETIKAINEN, 2011); (iv) o melhor método do artigo original da base NUAA (TAN et al., 2010) - esta melhor abordagem opera baseada em DoG (*Diference of Gaussians*, ou ainda Diferença de Gaussianas) com um classificador de regressão logística bilinear esparso; e (v) a abordagem baseadas em descritores de baixo nível (LLD - *Low Level Descriptors*) (SCHWARTZ; ROCHA; PEDRINI, 2011). Quanto mais acima estiver a curva ROC, melhor a abordagem. Como pode ser visto, as redes profundas propostas superaram significativamente a melhor técnica do artigo original da base de imagens NUAA (TAN et al., 2010) e a abordagem LLD (SCHWARTZ; ROCHA; PEDRINI, 2011) (ambas baseadas em características de texturas *handcrafted*), apresentando curvas bem mais altas.

Apesar da abordagem baseada no MLBP (MAATTA; HADID; PIETIKAINEN, 2011) também apresentar um bom resultado, este é inferior aos obtidos pela LBPnet e n-LBPnet. O método MLBP é custoso computacionalmente pois se baseia em vários histogramas *handcrafted* das faces, obtidos variando-se a vizinhança do LBP a fim de caracterizá-las. Além disto, utiliza uma SVM (CORTES; VAPNIK, 1995) para a classificação, cujo treinamento também é demorado dada a alta dimensionalidade dos histogramas gerados quando concatenados. Os modelos propostos, LBPnet e n-LBPnet, por outro lado, utilizam características de alto nível (profundas) obtidas de apenas um sistema de vizinhança fixo do LBP. Disto conclui-se que as características convolucionais de textura profundas são uma boa fonte de informações para anti-*spoofing* facial em comparação com as características tradicionais de textura *handcrafted*.

Em relação às medidas de Acurácia (ACC), *Area Under the Curve* (AUC), *Equal Error Rate* (EER), *False Acceptance Rate* (FAR), *False Rejection Rate* (FRR) e *Half-Total Error Rate* (HTER), a Tabela 9.1 mostra os resultados da LBPnet, n-LBPnet e outras abordagens da literatura como: LBP+SVM, LPQ (*Local Phase Quantization*)+SVM e Gabor+SVM - os três métodos apresentados por Maatta, Hadid e Pietikainen (2011); MLBP (MAATTA; HADID; PIETIKAINEN, 2011); DLTP (*Dynamic Local Ternary Pattern*) (PARVEEN et al., 2016); MLPQ/MBSIF (*Multiscale LPQ with Multiscale Binarized Statistical Image Features*) (ARASHLOO; KITTLER; CHRISTMAS, 2015); CDD (*Component Dependent Descriptor*) (YANG et al., 2013); NUAA Original (TAN et al., 2010); e LLD (SCHWARTZ; ROCHA; PEDRINI, 2011). Os valores de EER e AUC são usualmente obtidos das curvas ROC e os valores de ACC, FAR, FRR e HTER são os obtidos quando o limiar de aceitação do sistema está em $0,5$. Quanto mais altos os valores de ACC e

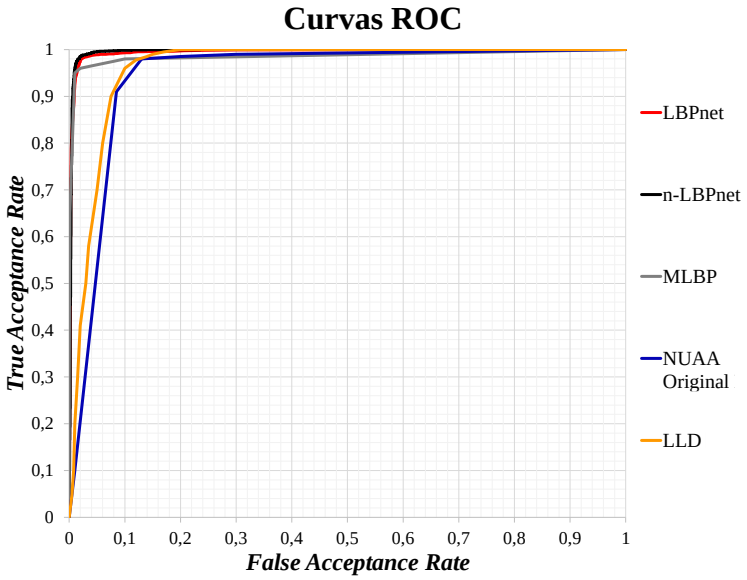


Figura 9.3: Curvas ROC das CNNs propostas, LBPnet e n-LBPnet, do método baseado no MLBP, do método proposto pelos criadores da base NUAA, e do método baseado nos LLDs (*Low Level Descriptors*). Fonte: Elaborada pelo autor.

AUC, e quanto mais baixos os valores de EER, FAR, FRR, e HTER, melhor. Alguns valores estão ausentes pois não foram informados pelos autores dos métodos.

Tabela 9.1: Resultados (%) em diferentes métricas obtidos pela LBPnet, n-LBPnet, e outros métodos estado-da-arte na base NUAA. O melhor valor em cada métrica está em destaque. Fonte: Elaborada pelo autor.

Método	ACC	AUC	EER	FAR	FRR	HTER
LBPnet	97,6	99,3	2,1	2,8	1,6	2,2
n-LBPnet	**98,2**	99,6	**1,8**	1,9	**1,5**	**1,7**
LBP+SVM	-	-	2,9	-	-	13,2
LPQ+SVM	-	-	4,6	-	-	-
Gabor+SVM	-	-	9,5	-	-	-
MLBP	98,0	99,0	-	**0,6**	4,4	2,5
DLTP	94,5	95,2	-	3,2	3,8	3,5
MLPQ/MBSIF	-	-	**1,8**	-	-	-
CDD	97,7	**99,8**	1,9	-	-	-
NUAA Original	92,0	95,0	-	-	-	-
LLD	-	96,6	8,2	-	-	-

Como pode-se observar, as CNNs propostas, no geral, superaram todas as demais técnicas avaliadas sobre a base NUAA (TAN et al., 2010). Em relação às curvas ROC, a n-LBPnet teve melhor desempenho que a LBPnet, indicando que a normalização dos valores dos *feature maps* teve efeito positivo na arquitetura da rede. A rede n-LBPnet obteve também os melhores resultados em 4 das 6 métricas avaliadas na Tabela 9.1 e a LBPnet obteve desempenho bastante próximo. O método baseado no MLBP também apresentou bons resultados dada sua aborda-

gem multiescala. Entretanto, tal método pode se tornar inviável dependendo da aplicação dado seu considerável custo computacional.

Em relação aos demais métodos da Tabela 9.1, o MLPQ/MBSIF e CDD, apesar de apresentarem bons resultados, também combinam uma série de características *handcrafted* a fim de caracterizar as faces, sendo computacionalmente custosos. Por fim. é possível observar ainda, na Tabela 9.1, que o descritor LBP apresentou, por si só um bom EER, mais baixo que outros importantes descritores de textura como LPQ e Gabor, justificando mais uma vez sua escolha na integração com as CNNs propostas, LBPnet e n-LBPnet.

Uma vantagem de se detectar ataques de *spoofing* em imagens estáticas (como na base NUAA) consiste no fato de que, geralmente, ela pode ser executada em tempo real e até mesmo se houver apenas uma pequena sequência de imagens da face (ou apenas uma única imagem) disponível. Tudo isso torna os métodos propostos ainda mais interessantes para aplicações reais.

9.3 Conclusões

Neste capítulo, duas Redes Neurais de Convolução (CNN) baseadas no LBP, LBPnet e n-LBPnet, são propostas para detecção de *spoofing* em sistemas de reconhecimento facial, as quais apresentaram ótimos resultados sobre a base de imagens NUAA (TAN et al., 2010), superando outras técnicas estado-da-arte avaliadas sobre esta base. Com as mais altas curvas ROC, baixas taxas de EER e altas acurácias, as redes LBPnet e n-LBPnet propostas configuram alternativas eficazes para a detecção de *spoofing* em aplicações de reconhecimento de faces, em especial quando se tem apenas imagens em tons de cinza. Além disto, as CNNs propostas são mais eficientes do que outras técnicas de ponta que combinam muitas informações *handcrafted* para detectar ataques: a LBPnet e a n-LBPnet empregam o descritor LBP trabalhando com uma única vizinhança e classificam as faces ao passar as imagens pelas poucas camadas das redes (simples multiplicações de matrizes). As CNNs propostas puderam ser treinadas e testadas sobre uma simples GPU Nvidia GeForce GTX 980 (com apenas 4 GB de memória dedicada).

Com base em tudo isso, é possível afirmar também que as características convolucionais de textura profundas são ricas fontes de informação para detecção de *spoofing* facial, em especial quando se tem imagens em tons de cinza apenas, propiciando melhores resultados do que os de métodos *handcrafted* (ou mesmo a combinação deles, que pode se tornar impraticável). A integração do descritor LBP em uma arquitetura de aprendizagem profunda é uma alternativa eficiente e robusta para prevenir tais atividades criminosas.

Capítulo 10

DETECÇÃO DE ATAQUES E TRANSFERÊNCIA DE APRENDIZADO

Neste capítulo, propõe-se uma abordagem acurada e eficiente para detecção de *spoofing* facial baseada em Transferência de Aprendizado a partir da rede neural VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015), bastante profunda e previamente treinada em grandes conjuntos de imagens para o reconhecimento facial (beneficiando a detecção de *spoofing* facial por as imagens serem de domínios próximos - faces), usada apenas para extrair características profundas das imagens da base de *spoofing* facial Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012) (a rede não é treinada sobre esta base). Tal abordagem, conhecida como Transferência de Aprendizado "off the shelf" permite tirar proveito da arquitetura neural profunda já treinada, evitando possível *overfitting* (por o modelo ser muito grande) bem como economizando tempo e processamento (não é sequer possível treinar a VGG-Face ou testar a arquitetura completa de uma só vez, necessitando fazer o *forward pass* das imagens faciais por etapas, sobre *hardware* restrito como com uma placa gráfica Nvidia GeForce GTX 980). O capítulo baseia-se no artigo **"Efficient Transfer Learning for Robust Face Spoofing Detection"**, publicado e apresentado no *Iberoamerican Congress on Pattern Recognition* (CIARP) em 2017.

10.1 Abordagem Proposta

Embora as arquiteturas de Aprendizado de Máquina em Profundidade permitam que se trabalhe com maior acurácia em tarefas complexas, um problema com as abordagens profundas é, muitas vezes, a falta de dados para treinamento, dada a enorme quantidade de parâmetros dos modelos (cada vez maiores), bem como o custo computacional envolvido neste processo, exigindo *clusters* de computadores ou de placas gráficas. Neste sentido a Transferência de

Aprendizado (TL, do inglês *Transfer Learning*) surgiu como uma alternativa para lidar com tais problemas. Na TL, pode-se optar por fazer um *fine-tuning* sobre a base de dados alvo da rede pré-treinada, o que ainda é um problema em termos de custos computacionais, ou então utilizar a arquitetura pré-treinada apenas para extrair características das imagens da base de dados alvo, atuando a rede de forma similar a um extrator de características convencional, capaz de obter características de alto nível para serem apresentadas a um classificador. Este esquema de TL, embora muitas vezes não permita obter taxas de acerto tão elevadas como quando retreinando ou refinando o modelo profundo como um todo, é bem mais eficiente e permite alcançar resultados muito superiores aos das técnicas *handcrafted*. Além disto, pode-se trabalhar com redes bastante profundas mesmo sobre *hardware* restrito uma vez que não é necessário treinar seus parâmetros (e pode-se testar a rede neural por partes, fazendo o *forward pass* das imagens faciais camada a camada).

Neste trabalho, em suma, utiliza-se a rede neural VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015), já pré-treinada em um problema de domínio similar (reconhecimento de faces), tendendo a não reduzir muito seu desempenho, para a detecção de ataques de *spoofing* facial. A abordagem proposta para detecção de faces sintéticas apresenta as seguintes etapas: (i) Detecção e normalização da face; (ii) Extração de características profundas (pela VGG-Face); (iii) Treinamento de uma SVM com base nas características extraídas para as faces de treinamento; e (iv) Calibração do limiar de detecção de face sintética do sistema.

10.1.1 Detecção, Normalização e Extração de Características das Faces

Dado o modelo VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015), já treinado, para cada quadro dos vídeos de treinamento da base Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012), recorta-se a região facial com base nas anotações que os próprios autores da base forneceram em relação a tal região de interesse a fim de evitar que problemas de segmentação possam atrapalhar os resultados. Aumenta-se ou diminui-se alguns pixels na largura e altura das regiões faciais a fim de todas conterem 112×112 pixels, facilitando seus redimensionamentos para 224×224 pixels, dimensões necessárias para alimentar a VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015).

Na verdade, por restrições de *hardware* (memória), faz-se uma amostragem dos quadros dos vídeos, avaliando-se apenas uma imagem a cada 8 quadros subsequentes para fins de eficiência e a fim de facilitar o treinamento da SVM na classificação das faces, obtendo desta maneira cerca de 11.650 imagens faciais de treinamento. Tais imagens faciais (com 224×224 pixels) resultantes são então normalizadas (subtração da imagem média da base de dados na qual a VGG-

Face foi treinada) e apresentadas à VGG-Face, realizando-se um *forward pass* das mesmas até a camada "Pool5" da rede, última camada antes dos neurônios completamente conectados do modelo (quase no topo da rede).

Dada uma imagem de entrada apresentada à rede, os 512 *feature maps* resultantes da camada "Pool5", com tamanho de 7×7, são também regularmente amostrados (um quarto deles) e seus valores, após concatenados, formam o vetor de características da imagem de treinamento apresentada à rede neural. A Figura 10.1 ilustra a geração do vetor de características para uma dada imagem facial. Os *feature maps* de saída da camada "Pool 5" com bordas em vermelho são os amostrados para se gerar o vetor.

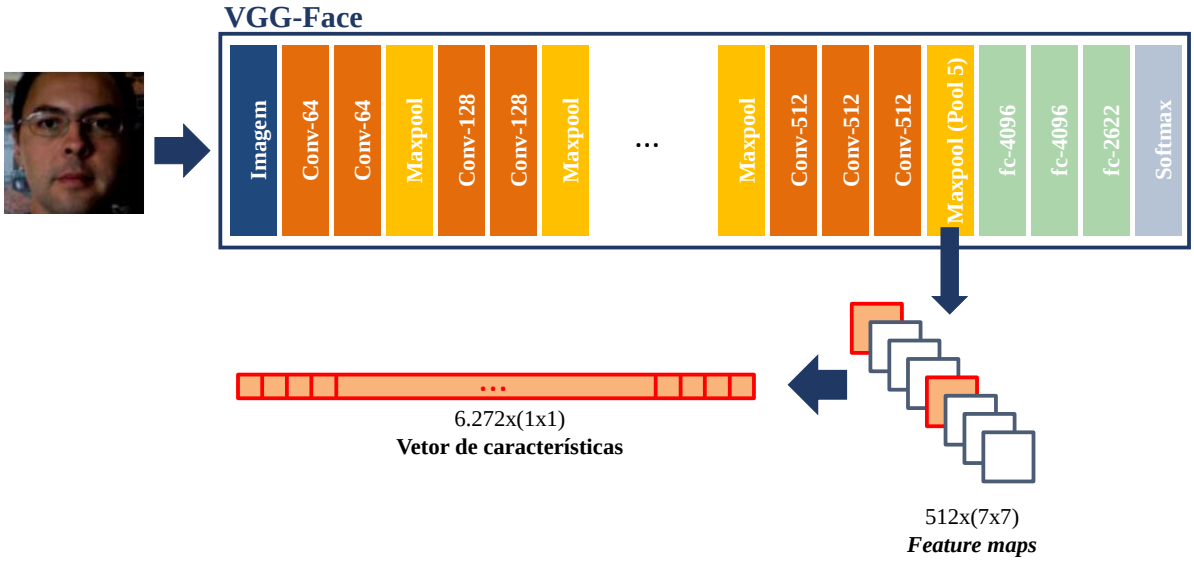


Figura 10.1: Extração de características a partir de uma imagem facial e da VGG-Face. Fonte: Elaborada pelo autor.

10.1.2 Treinamento da SVM e Calibração do Limiar de Aceitação

Dados todos os vetores de características extraídos por meio da VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015) das imagens de treinamento, isto é, dos vídeos de treinamento contendo faces reais e sintéticas da base Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012), e seus respectivos rótulos de classe, normaliza-se seus valores para o intervalo $[0;1]$ e treina-se um classificador SVM de base radial. Dados os vetores de treinamento, realiza-se o treinamento de tal classificador usando o esquema de validação cruzada com 5 *folds*.

Na verdade, uma versão probabilística do classificador SVM (CHANG; LIN, 2011) é empre-

gada. Após o treinamento, dada uma imagem facial de um vídeo sob análise, o classificador emite uma probabilidade indicando a similaridade de tal face com as classes real e sintética. Para calibrar o limiar τ, usado para determinar se uma face de teste é real ou sintética, dada a probabilidade gerada pela SVM, usa-se os vídeos do conjunto de validação da base Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012), variando τ no intervalo $[0;1]$ e observando a acurácia na classificação dos mesmos.

Para cada vídeo de validação, também são amostrados quadros em posições espaçadas regularmente, usando a proporção de um quadro a cada 8 deles, obtendo-se quase 11.650 imagens faciais, que alimentam o modelo VGG-Face (SIMONYAN; ZISSERMAN, 2015) após a detecção e normalização facial, a fim de extrair seus vetores de características e gerar suas probabilidades de pertencer a cada classe (face real ou sintética). Para cada imagem facial, se sua probabilidade de saída for maior que o limiar τ sendo considerado, classifica-se a face como real, e sintética caso contrário.

Finalmente, um esquema de votação é aplicado para determinar a classe final de cada vídeo de validação, ou seja, dadas as classificações de seus quadros, se pelo menos metade deles forem considerados como de faces sintéticas, o vídeo é classificado como ataque, caso contrário o vídeo é tomado como de face real. Varia-se τ no intervalo $[0;1]$, em etapas de $0,01$, e calcula-se a *False Non-Detection Rate* (FNDR) e a *False Detection Rate* (FDR) do método com base nos vídeos de validação. Fixa-se o parâmetro τ ao se obter os mesmos valores para FNDR e FDR (EER - *Equal Error Rate*). A Figura 10.2 mostra o esquema de classificação de um vídeo a partir da classificação de seus *frames* amostrados.

10.1.3 Avaliação do Desempenho

O mesmo processo feito para os vídeos de treinamento e validação é então aplicado aos vídeos de teste após a fixação do valor de τ. Para cada vídeo de teste, seus quadros também são amostrados na razão de um a cada 8 *frames* (considerando todos os vídeos de teste, obtém-se um total de 15.500 *frames*), detecta-se e normaliza-se as faces neles e tais imagens alimentam então a VGG-Face (SIMONYAN; ZISSERMAN, 2015) a fim de extrair seus vetores de características e também classificá-las comparando suas probabilidades de saída da SVM com o valor de τ. Se maior que τ, a face de teste é considerada como real, ou sintética, caso contrário. O mesmo esquema de votação baseado na classificação de seus *frames* também é aplicado para determinar a classe final dos vídeos de teste: se de face real ou sintética. Novamente, se pelo menos metade dos *frames* classificados de um determinado vídeo forem considerados como de face falsa, o vídeo é classificado como ataque, ou real caso contrário.

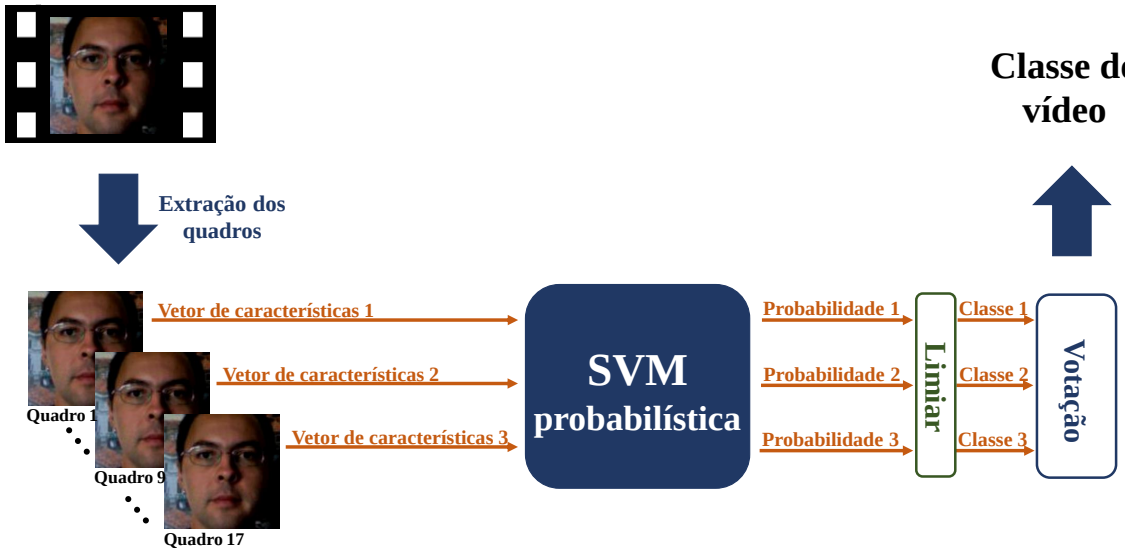


Figura 10.2: Amostragem dos *frames* de vídeo de teste e classificação com base na classificação dos *frames*. Fonte: Elaborada pelo autor.

10.2 Experimentos, Resultados e Discussão

A abordagem proposta para detecção de *spoofing* facial utilizando Transferência de Aprendizado foi avaliada na tradicional base Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012), apresentada na Seção 5.3, maior que a base NUAA (TAN et al., 2010) e com imagens faciais em RGB (mesmo espaço de cores das imagens faciais originalmente usadas para treinar a rede neural VGG-Face).

Empregou-se o *framework* Caffe (JIA et al., 2014) para carregar o modelo treinado da VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015) e executar o *forward pass* das imagens faciais. Dados os vetores de características das faces nos quadros de treinamento (de todos os vídeos de treinamento), utilizou-se a biblioteca LibSVM (CHANG; LIN, 2011) para normalizar seus valores no intervalo $[0;1]$ e treinar uma SVM com *kernel* de base radial. Foi realizado um esquema de validação cruzada com 5 *folds* para treinar o classificador. Depois de treinar a SVM probabilística, empregou-se os vídeos do conjunto de validação para encontrar o valor ótimo para o limiar de aceitação da face como verdadeira τ.

Então, após encontrar o limiar ótimo $\tau = 0,18$, classificou-se os quadros amostrados dos vídeos de teste e os próprios vídeos com base em um esquema de votação simples de seus *frames*. Como na literatura, os resultados de *Half-Total Error Rate* (HTER) da abordagem proposta (média entre FNDR e FDR, calculados sobre a base de teste) e de outros importantes métodos são mostrados na Tabela 10.1. Como observação, como a rede neural VGG-Face,

neste caso, comportou-se como um verdadeiro extrator de características, mas extraindo características de alto nível, comparou-se tal abordagem a métodos baseados em outros extratores de características *handcrafted* importantes da literatura a fim de verificar a superioridade das características profundas mesmo quando a rede não é treinada na base sobre a qual é avaliada. Os métodos da literatura avaliados foram: (i) *Diffusion Speed Model* (KIM; SUH; HAN, 2015); (ii) LBP+LDA (CHINGOVSKA; ANJOS; MARCEL, 2012); (iii) LBP+SVM (CHINGOVSKA; ANJOS; MARCEL, 2012); (iv) LBP-TOP+SVM (PEREIRA et al., 2012); e (v) *Non-Linear Diffusion* (ALOTAIBI; MAHMOOD, 2017).

Tabela 10.1: Resultados (%) na base Replay-Attack em termos de *Half-Total Error Rate* (HTER). Quanto menor o HTER, melhor o método. Também foi calculado o valor de Acurácia (ACC) do método proposto. Os melhores valores estão destacados. Fonte: Elaborada pelo autor.

Method	ACC	HTER
Abordagem Proposta	**93,95**	16,62
Diffusion Speed Model	-	12,50
LBP+LDA	-	17,17
LBP+SVM	-	16,16
LBP-TOP+SVM	-	**7,60**
Non-Linear Diffusion	-	10,00

É importante observar na Tabela 10.1 que a abordagem eficiente proposta, baseada no simples *forward pass* (simples multiplicações de matrizes) das imagens faciais em uma CNN já pré-treinada, apresenta um resultado HTER próximo aos dos demais métodos: com um HTER de 16,62 %, supera a técnica LBP+LDA, baseada no descritor LBP e fica relativamente próxima das demais abordagens. Os demais métodos avaliados, apesar de extrairem características *handcrafted* são computacionalmente bastante caros: o LBP-TOP, por exemplo, extrai vários histogramas LBP para muitos conjuntos de quadros dos vídeos (uma janela é deslocada sobre todos os quadros) e concatena esses histogramas gerando grandes vetores de características para serem processados por uma SVM. Nesse método, a fim de classificar cada vídeo, um esquema de votação também é realizado.

O método proposto requer apenas treinar uma SVM baseada em vetores de características robustas extraídas diretamente pelo modelo VGG-Face (SIMONYAN; ZISSERMAN, 2015) já pré-treinado, sendo eficiente e economizando tempo de processamento e consumo de *hardware*: a extração de características, baseada na camada "Pool5" da VGG-Face, levou cerca de 0,4 segundos por quadro. É importante notar que os resultados obtidos (HTER e ACC) do método proposto podem ser consideravelmente melhorados, considerando mais quadros dos vídeos em vez de um *frame* em cada 8 deles, bem como usando todos os *feature maps* de saída (de 7×7 posições) da camada "Pool5" para representar as imagens.

Além disso, como em outros trabalhos, pode-se usar as camadas superiores totalmente conectadas da rede VGG-Face (SIMONYAN; ZISSERMAN, 2015) ou até mesmo uma combinação de *feature maps* de diversas camadas da CNN, incluindo as inferiores, para melhor representar as faces. Também, a outra abordagem mencionada de Transferência de Aprendizado, isto é, refinar a própria rede profunda na base de imagens de *spoofing* facial, poderia ser realizada para melhorar ainda mais os resultados. Apesar disso, é importante notar que todas essas abordagens mencionadas requerem mais tempo de processamento e recursos computacionais do que a arquitetura proposta e, por esse motivo, foram descartadas devido às restrições de *hardware* (utilizou-se uma placa de vídeo Nvidia GeForce GTX 980).

10.3 Conclusões

Neste capítulo, apresenta-se uma abordagem eficiente e também robusta para detecção de *spoofing* facial baseada em Transferência de Aprendizado, aplicando um modelo já treinado de uma CNN bastante profunda, a VGG-Face, na classificação dos vídeos da base Replay-Attack. A abordagem proposta, devido ao seu trabalho com uma arquitetura profunda, é muito robusta, apresentando resultados próximos às técnicas de ponta na tradicional base avaliada. Além disso, como precisa-se apenas treinar uma SVM com base nas características de alto nível extraídas pela VGG-Face a partir das imagens faciais de treinamento, basta executar um *forward pass* de tais imagens no modelo treinado (camada a camada), economizando recursos de *hardware*.

Como a VGG-Face foi treinada em imagens de um domínio similar (imagens de faces para reconhecimento facial) em relação ao problema sendo tratado, a robustez das características de alto nível extraídas é fortemente preservada. A CNN atua como um verdadeiro extrator de características de alto nível com resultados relativamente melhores que extratores *handcrafted* computacionalmente caros. Com base em tudo isso, a abordagem proposta é adequada para aplicações reais ou em casos em que não há servidores de alta tecnologia disponíveis: pode-se empregar uma CNN de ponta na detecção de *spoofing* que sequer poderia ser utilizada se fosse preciso treiná-la trabalhando-se com uma GPU Nvidia GeForce GTX 980, como a utilizada.

Capítulo 11

DETECÇÃO DE ATAQUES E REDE NEURAL DE CONVOLUÇÃO AMPLIADA EM LARGURA

Recentemente, apesar de algumas abordagens baseadas em CNNs (*Convolutional Neural Networks*) (LECUN et al., 1998) terem alcançado resultados estado-da-arte na detecção de *spoofing* facial, na maioria dos casos, as arquiteturas propostas são muito profundas, sendo inadequadas para dispositivos com restrições de *hardware*, que sequer conseguem treiná-las ou executá-las. Neste capítulo, propõe-se uma arquitetura eficiente para detecção de ataques baseada em uma CNN ampliada em de largura ao invés de profundidade, denominada wCNN (*Width-extended* CNN). Cada parte da wCNN é treinada, separadamente, em uma determinada região da face, e suas saídas são combinadas para decidir se a face apresentada ao sensor é real ou sintética. A abordagem proposta, que aprende características locais relativamente profundas de cada região facial devido à sua arquitetura ampliada em largura, apresenta acurácia mais elevada do que métodos de última geração, incluindo a bem referenciada VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015) (com *fine-tuning*), sendo muito mais eficiente em relação a recursos de *hardware* e tempo de processamento como mostrado.

O conteúdo deste capítulo refere-se ao conteúdo do artigo **"Efficient Width-Extended Convolutional Neural Network for Robust Face Spoofing Detection"**, publicado e apresentado na *Brazilian Conference on Intelligent Systems* (BRACIS) em 2018, evento realizado na IBM *Research*, em São Paulo.

11.1 Abordagem Proposta

A informação espacial, ou seja, a distribuição e relação entre os valores dos pixels em posições vizinhas no sistema de coordenadas 2D da imagem é extremamente importante em

tarefas envolvendo faces, como detecção e reconhecimento facial (TURK; PENTLAND, 1991; CHIACHIA et al., 2014). Os diferentes padrões de cada região facial, isto é, a distribuição dos elementos faciais, codificam informações ricas e discriminativas a fim de distingui-las de outros objetos, e também para distinguir uma determinada face de outras. Em relação à detecção de *spoofing* facial e a importância de cada região facial para tal tarefa, diferentes abordagens baseadas em características *handcrafted*, como de textura (MAATTA; HADID; PIETIKAINEN, 2011; CHINGOVSKA; ANJOS; MARCEL, 2012) e de qualidade da imagem (AKHTAR; FORESTI, 2016), indicam que diferentes pistas de *spoofing* podem ser extraídas de distintas regiões faciais.

Algumas abordagens baseadas em CNNs foram recentemente propostas para detecção de *spoofing* facial (LI et al., 2016). No entanto, a maioria utiliza arquiteturas muito profundas para alcançar bons desempenhos, aumentando bastante a complexidade dos modelos, a quantidade de dados rotulados necessária para seus treinamentos bem como o consumo de recursos de *hardware*, especialmente a memória requerida nas placas gráficas para carregar tais arquiteturas para treinamento e teste, sendo inadequadas em cenários reais com limitações de *hardware*, por exemplo em plataformas embarcadas ou dispositivos móveis. Além disto, nenhuma das abordagens profundas propostas se volta a aprender características locais de *spoofing* nas diferentes regiões faciais: todas trabalham com imagens faciais inteiras, aprendendo características holísticas, ou com *patches* extraídos de regiões aleatórias das faces.

Com base em tudo isto, neste trabalho propõe-se uma arquitetura CNN ampliada em largura ao invés de profundidade, denominada wCNN (*Width-extended* CNN). Na wCNN, diferentemente de outras abordagens que aumentam a capacidade do modelo em profundidade, aumenta-se em largura um modelo de CNN compacto, permitindo treiná-lo e testá-lo mesmo usando dispositivos com restrições de *hardware*. Cada parte do modelo wCNN pode ser treinada separadamente e até mesmo em momentos diferentes, em cada região facial principal (aprendendo pistas de *spoofing* locais). Então, dada uma face desconhecida, ela pode ser classificada como real ou sintética com base nas saídas de todas as partes da rede neural proposta. Dividir a arquitetura em largura também evita o crescimento exponencial do número de parâmetros da CNN. Além disso, como cada parte da wCNN tem como entrada um pequeno *patch* facial (de dimensões reduzidas), ao invés da imagem facial completa, gera-se menores *feature maps* nas camadas da rede e, consequentemente, reduz-se a quantidade de operações necessárias para o *forward pass* das imagens faciais pela arquitetura.

Os resultados obtidos nos experimentos mostram que a wCNN, mesmo rodando em um *hardware* limitado (computador com uma única GPU Nvidia GeForce GTX 980, com apenas 4 GB de memória dedicada), apresenta desempenho compatível com o estado-da-arte, isto é,

abordagens tais como a baseada no *fine-tuning* da VGG-Face (LI et al., 2016), a CNN profunda proposta por Atoum et al. (2017), bem como outras arquiteturas complexas que trabalham com várias características de textura e vários espaços de cores, como nos trabalhos de Boulkenafet, Komulainen e Hadid (2015, 2017).

11.1.1 PatchNet

Como mencionado, neste trabalho, propõe-se uma CNN ampliada em largura, expandindo uma CNN menor, a qual é inspirada no modelo compacto apresentado no Capítulo 9, aqui denominada de PatchNet. Como mostrado na Figura 11.1, o modelo PatchNet apresenta duas camadas com operações de convolução e *pooling* na parte inferior, com *kernels* e *strides* de 5×5 e 2×2 elementos, 1 e 2 pixels, respectivamente. As operações de *pooling* retornam os valores máximos sob os *kernels* (*pooling* máximo) e, no topo da rede, uma camada totalmente conectada com 931 neurônios com ativações ReLU e uma camada *softmax* com 2 neurônios também estão presentes.

Como ilustra a Figura 11.1, para acelerar o treinamento da PatchNet e melhorar sua acurácia, após a operação *CONV2*, uma retificação de sinal (ReLU - *Rectified Linear Unit*) nos valores dos *feature maps* e uma operação de normalização LRN (*Local Response Normalization*) (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) são realizadas. Os valores nos *feature maps* resultantes de *CONV2*, C_i, com $i = 1, 2, ..., 50$, após serem retificados pela operação ReLU, são normalizados com base nas posições correspondentes dos *feature maps* vizinhos pela LRN, considerando 5 *feature maps* vizinhos por vez, $\alpha = 0,2$ e $\beta = 0,75$.

11.1.2 wCNN

Baseado no modelo compacto da PatchNet, neste trabalho propõe-se a wCNN, uma arquitetura de CNN eficiente e ampliada em largura projetada para detecção de *spoofing* facial, capaz de obter resultados compatíveis com o estado-da-arte, mesmo quando usando um *hardware* inferior. A expansão da rede PatchNet em largura (em vez de profundidade) permite poupar computação e aprender pistas de *spoofing* locais de diferentes regiões das faces.

Basicamente, dado um conjunto de imagens faciais de treinamento, cada imagem é dividida em nove *patches* principais $(p_1, p_2, \ldots, p_9)$, como mostrado na Figura 11.2. Estas nove regiões também são adotadas em outros trabalhos (JAIN; ROSS; NANDAKUMAR, 2011). Após dividir cada imagem facial de treinamento em suas nove regiões principais, cada *patch* é usado para treinar uma instância diferente do modelo PatchNet, ou seja, treina-se nove instâncias desta rede

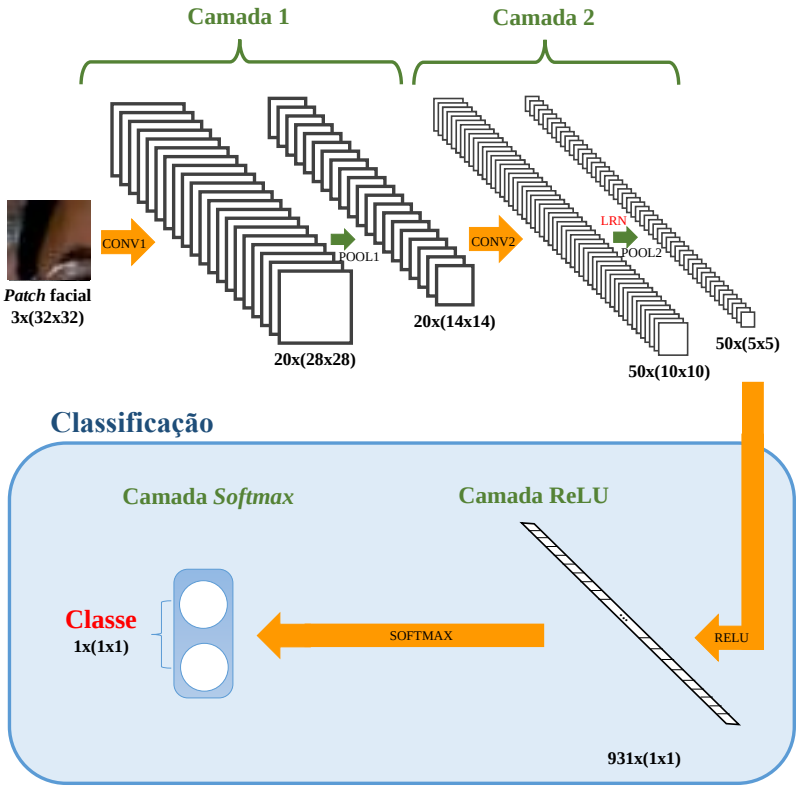


Figura 11.1: Arquitetura da PatchNet. Os *feature maps* da saída da operação *CONV2* são linearmente retificados e normalizados pela LRN (*Local Response Normalization*). Fonte: Elaborada pelo autor.

compacta. Todos os *patches* p_i das faces são usados para treinar a i-ésima instância PatchNet.

Em seguida, compõe-se a wCNN integrando, no topo, as saídas das nove instâncias da PatchNet, aplicando uma função de integração, no caso deste trabalho, um esquema de maioria de votos, responsável por uma classificação final de toda a face de entrada em real ou sintética. Esse modelo maior pode ser visto como uma rede neural ampliada em largura única, como mostrado na Figura 11.3. As instâncias do modelo PatchNet, como dito, não compartilham parâmetros, permitindo que sejam treinadas separadamente (mesmo em momentos diferentes), reduzindo a quantidade de processamento, tempo e requisitos de *hardware* e tornando-as especializadas em cada região facial. O limiar de aceitação da face na função de integração para a votação dos *patches* é ajustado com base nas probabilidades de saída das nove PatchNets treinadas e nas classes das imagens de validação.

Após o treinamento de toda a wCNN, dada uma imagem de face de teste, ela também é dividida em nove regiões e cada região é passada através de sua respectiva PatchNet. A decisão final para toda a face é tomada, como dito, com base na maioria dos votos das nove PatchNets, dado o limiar de aceitação da função de integração otimizado.

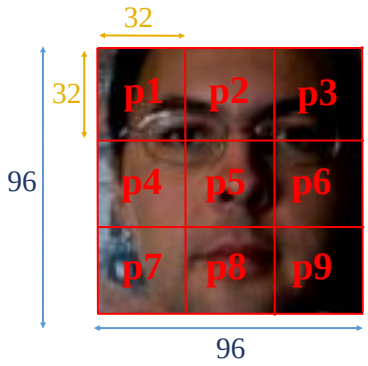


Figura 11.2: Uma imagem de face (96×96 **pixels) obtida do conjunto de dados Replay-Attack dividida em nove *patches* (regiões menores não sobrepostas de** 32×32 **pixels). Fonte: Elaborada pelo autor.**

11.2 Experimentos, Resultados e Discussão

A CNN proposta foi avaliada e comparada com CNNs estado-da-arte tanto em termos de eficiência como de acurácia. A subseções 11.2.1 e 11.2.2 descrevem tais análises.

11.2.1 Análise da Eficiência

A fim de avaliar a eficiência da wCNN proposta em termos de processamento necessário para a detecção de *spoofing* facial, comparou-se seu desempenho com CNNs detentoras de resultados estado-da-arte como a VGG-Face (com o *fine-tuning*) (LI et al., 2016), e uma CNN robusta, baseada em *patches* aleatórios faciais para a detecção de ataques, proposta por Atoum et al. (2017).

Em vez de apresentar os tempos de processamento, apresenta-se a quantidade de operações de multiplicação requeridas pelas CNNs adotadas na passagem de cada imagem facial (ou *patches*) pela arquitetura para sua classificação, uma vez que essa medida é independente do *hardware* utilizado para comparação. Por razões de simplicidade, não contou-se o número de operações de adição realizadas pelas CNNs, uma vez que tais operações não são tão custosas quanto as multiplicações e estão presentes em menor número.

Como o *forward pass* das imagens através das redes neurais serve de base para o algoritmo de *backpropagation*, o treinamento das CNNs também é geralmente muito mais complexo para as arquiteturas com *forward passes* mais caros computacionalmente. Sua complexidade tende a aumentar substancialmente, pois o algoritmo de *backprogation* calcula derivadas parciais para todos os pesos da rede.

A wCNN proposta trabalha com imagens faciais menores (9 *patches* de 32×32, formando

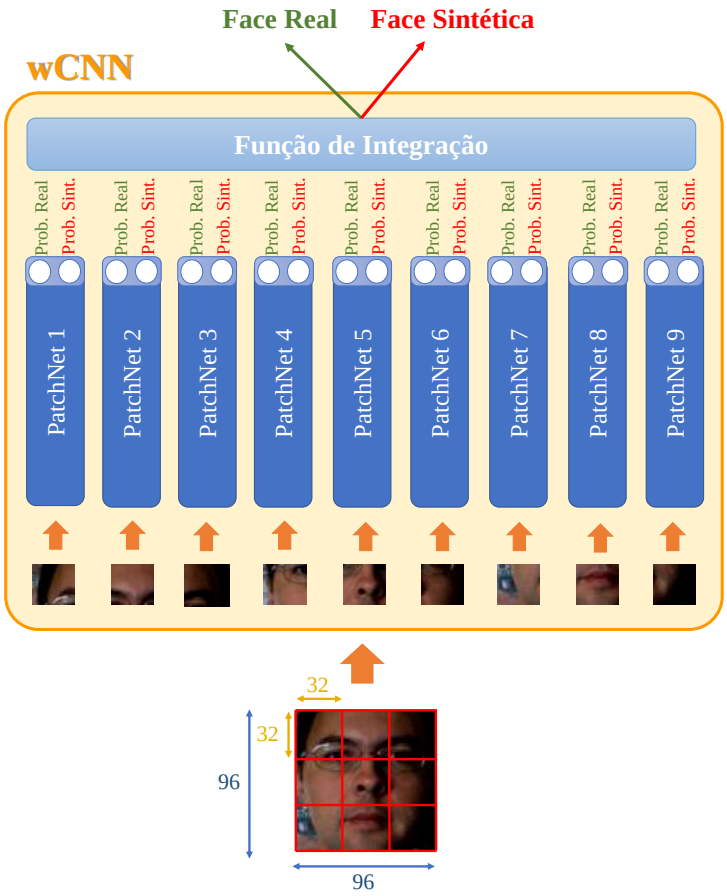


Figura 11.3: Arquitetura da wCNN. Cada imagem facial é dividida em 9 *patches*, cada um passando por uma rede PatchNet, e então uma decisão final para a face (real ou sintética) é obtida pela função de integração. Fonte: Elaborada pelo autor.

uma imagem inteira de 96×96 pixels) tornando-a ainda mais compacta, ou seja, exigindo menos dados para a mesma tarefa do que as outras CNNs. Por exemplo, a CNN proposta por Atoum et al. (2017) opera sobre vários *patches* faciais de 96×96 pixels obtidos aleatoriamente das imagens faciais, e a VGG-Face usada por Li et al. (2016) trabalha com imagens faciais com 224×224 pixels.

A quantidade total de operações de multiplicação requeridas por cada rede é a soma de todas as operações de multiplicação em cada uma de suas camadas convolucionais e totalmente conectadas. Nas camadas convolucionais, a quantidade de operações de multiplicação é calculada pela Equação 11.1:

$$N_{mult} = n_{in} \cdot s_k^2 \cdot p_k^2 \cdot n_{out} \tag{11.1}$$

onde n_{in} e n_{out} correspondem ao número de *feature maps* de entrada e saída da camada de convolução, respectivamente, s_k é a dimensão de *kernel* convolucional e p_k é o número de todas

as posições (na direção do eixo das abscissas) que o *kernel* pode ser aplicado aos *feature maps* de entrada considerando seu passo (*stride*). Para as camadas totalmente conectadas, o termo p_k não está presente e s_k corresponde à dimensão dos *feature maps* de entrada.

Como exemplo, para uma única parte da arquitetura da wCNN, trabalhando em *patches* faciais de 32×32 pixels, a quantidade de operações de multiplicação necessárias em sua primeira camada convolucional para o *forward pass* de um *patch* é $3 \cdot 5^2 \cdot 28^2 \cdot 20 = 1.176.000$ já que a rede trabalha com imagens de entrada RGB (3 canais), *kernels* de 5×5, *stride* de 1 e sem *padding*, permitindo que os *kernels* sejam aplicados nos *patches* de entrada originais em $28 \cdot 28$ posições. Neste sentido, toda a wCNN requer, em sua primeira camada de convolução, $9 \cdot 1.176.000 = 10.584.000$ operações de multiplicação.

A CNN de Atoum et al. (2017), que trabalha com dois *patches* obtidos aleatoriamente de cada imagem facial, precisa realizar $2 \cdot 6 \cdot 5^2 \cdot 96^2 \cdot 50 = 138.240.000$ operações de multiplicação, apenas em sua primeira camada de convolução, uma vez que trabalha com 6 canais nas imagens de entrada (espaços de cores HSV e YCbCr), *kernels* de convolução de tamanho 5×5, *patches* com tamanho de 96×96 (e *padding*) e 50 *feature maps* de saída.

A Tabela 11.1 mostra a quantidade de operações de multiplicação para um *forward pass* de uma imagem facial (ou *patch*) em cada camada (de convolução ou completamente conectada) das CNNs comparadas: wCNN, CNN baseada em *patches* aleatórios de Atoum et al. (2017) e a *Fine-Tuned* VGG-Face utilizada por Li et al. (2016). Para fins de comparação, também calculou-se a quantidade de operações de multiplicação exigidas por uma única PatchNet para executar um *forward pass* em uma imagem inteira, com 96×96 pixels (mesmas dimensões da entrada de wCNN ao considerar todos os *patches* juntos). Já que a VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015) é bastante profunda, apenas informou-se sua quantidade aproximada de operações de multiplicação de acordo com Canziani, Paszke e Culurciello (2016).

Tabela 11.1: Quantidade de operações de multiplicação requeridas no *forward pass* de cada imagem facial pela wCNN, PatchNet, CNN baseada em *patches* aleatórios de Atoum et al. (2017), e VGG-face (após *fine-tuning*) de Li et al. (2016). Fonte: Elaborada pelo autor.

Camada	**wCNN** Imagem 96×96	**PatchNet** Imagem 96×96	***Random Patches* CNN** *Patches* 96×96	***Fine-Tuned* VGG-Face** Imagem 224×224
1	10.584.000	12.696.000	138.240.000	16 camadas
2	22.500.000	44.100.000	207.360.000	
3	10.473.750	20.528.550	155.520.000	
4	16.758	1.862	77.760.000	
5	-	-	32.400.000	
6			4.500.000	
7			800.000	
8			1.600	
Total de Operações	**43.574.508**	**77.326.412**	**616.581.600**	**31.000.000.000**

Como mostrado na Tabela 11.1, a wCNN requer apenas 43.574.508 operações de multiplicação no *forward pass* de uma imagem de teste completa para classificação, sendo muito mais eficiente do que todas as outras CNNs avaliadas. Também é importante notar que o método baseado na VGG-Face (LI et al., 2016) requer uma enorme quantidade de operações de multiplicação para classificar uma imagem inteira, assim como a CNN baseada em *patches* proposta por Atoum et al. (2017), que, para o *forward pass* dos 2 *patches* faciais, necessita de 616.581.600 multiplicações. No modelo wCNN, especialmente por as partes da rede neural não compartilharem parâmetros entre si, o *forward pass* de uma face inteira (previamente dividida em nove *patches*) é muito eficiente, sendo ainda mais eficiente do que quando se trabalha com uma única PatchNet sobre toda a imagem facial.

Como observação, apesar da wCNN ser maior em largura do que uma única PatchNet, quando a PatchNet opera sobre imagens faciais completas, ela passa a apresentar muito mais parâmetros (cerca de 20.557.000 pesos) do que a wCNN trabalhando em 9 pequenos *patches* (cerca de 1.192.000 parâmetros por parte da rede neural), especialmente devido ao tamanho dos *feature maps* de entrada de sua primeira camada completamente conectada, de tamanho 21×21 para a PatchNet trabalhando em faces completas e 5×5 para a wCNN, indicando mais uma vez a eficiência do modelo proposto (redução de parâmetros na arquitetura neural).

11.2.2 Análise da Acurácia

A wCNN proposta foi avaliada em duas bases de imagens de *spoofing* bem referenciadas, a base Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012) e a CASIA FASD (*Chinese Academy of Sciences - Institute of Automation - Face Antispoofing Database*) (ZHANG et al., 2012), ambas contendo vídeos no espaço de cores RGB de faces reais e sintéticas (estáticas e dinâmicas), impressas ou apresentadas à câmera de captura usando *displays* digitais de diferentes dispositivos móveis (bases apresentadas na Seção 5.3).

Para cada base, inicialmente, comparou-se o desempenho da rede neural PatchNet (trabalhando nas imagens faciais completas) com a wCNN proposta, a fim de avaliar a relevância das características locais para detecção de *spoofing* facial quando comparadas com características globais (aprendidas sobre toda a face). Construiu-se curvas ROC (*Receiver Operating Characteristics*) para a wCNN e a PatchNet, ambas classificando as faces em todos os quadros dos vídeos de teste da base Replay-Attack e, consequentemente, classificando os vídeos com base na maioria dos votos dos seus quadros.

Para ambas as redes neurais, os mesmos parâmetros de inicialização e treinamento foram utilizados. Os pesos dos *kernels* de convolução foram inicializados com valores inversamente

proporcionais ao tamanho dos *feature maps* de entrada e saída de cada camada, enquanto que os *biases* dos neurônios foram inicializados em zero. As redes neurais foram treinadas por 2.000 iterações utilizando o *framework* Caffe (JIA et al., 2014) por meio da abordagem de Gradiente Descendente Estocástico (*Sthochastic Gradiente Descent* - SGD) (HINTON, 2012) com os seguintes parâmetros: 64 imagens por *batch* (normalizadas - valores no intervalo [0;1]), taxa inicial de aprendizado de $0,01$, *momentum* de 0,9 e *weight decay* de $0,004$. A política de decaimento da taxa de aprendizado foi novamente "inv".

Para a base de imagens Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012), segmentou-se as faces nos quadros dos vídeos usando os pontos de referência fornecidos pelos autores da própria base (obtendo-se imagens faciais de 96×96 pixels) e treinou-se as arquiteturas sobre elas. Variando o limiar de aceitação das faces como reais no intervalo $[0;1]$, classificou-se cada vídeo de teste com base na maioria das classificações de seus quadros e gerou-se curvas ROCs para as PatchNet e a wCNN. A Figura 11.4 mostra os resultados obtidos. Como pode-se observar, a wCNN obteve uma curva ROC melhor que o modelo PatchNet isolado, treinado em imagens faciais completas.

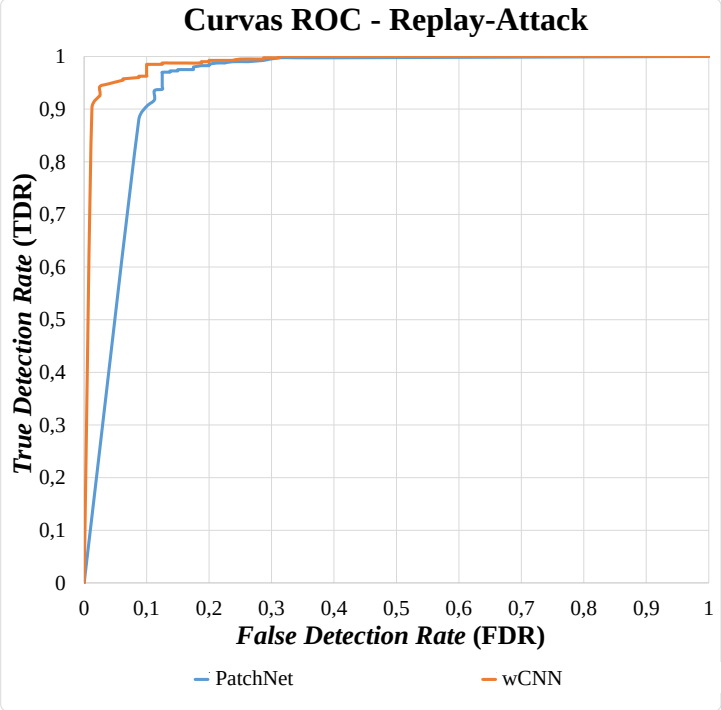


Figura 11.4: Curvas ROC para a wCNN e a PatchNet (trabalhando sobre as faces completas) para a base Replay-Attack. Quanto mais acima estiver a curva, melhor o método. Fonte: Elaborada pelo autor.

Realizou-se o mesmo experimento na base de imagens CASIA (ZHANG et al., 2012). Para segmentar as imagens faciais, neste caso, nos frames dos vídeos, utilizou-se um algoritmo ba-

seado no trabalho de Viola e Jones (2001), fixando também as imagens faciais em 96×96 pixels (e os mesmo 9 *patches* de 32×32 pixels para a wCNN). Como se pode observar na Figura 11.5, a arquitetura CNN estendida proposta, wCNN, obteve novamente uma melhor curva ROC do que a PatchNet trabalhando em imagens faciais completas.

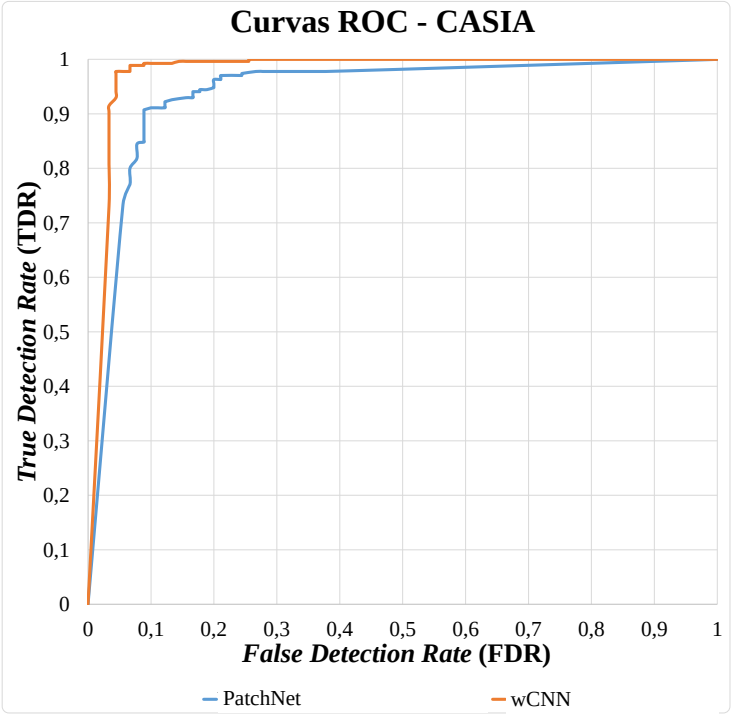


Figura 11.5: Curvas ROC para a wCNN e a PatchNet (trabalhando sobre as faces completas) na base de imagens CASIA. Fonte: Elaborada pelo autor.

A fim de comparar a abordagem proposta, isto é, a wCNN, com métodos estado-da-arte de anti-*spoofing* facial sobre a base Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012), a maioria deles empregando CNN bastante profundas ou métodos complexos, incluindo os apresentados na Seção 11.2.1, seguiu-se os protocolos propostos pelos próprios autores da base. Depois de treinar a CNN ampliada em largura nos vídeos de treinamento, utilizou-se os vídeos do conjunto de validação para encontrar o limiar de aceitação ótimo da função de integração da wCNN (para o esquema de maioria de votos), o qual permitia obter o mesmo valor de *False Detection Rate* (FDR) e *False Non-Detection Rate* (FNDR) para os vídeos de validação, ponto de operação do sistema conhecido como *Equal Error Rate* (EER). Fixou-se então o limiar e classificou-se os vídeos de teste calculando a taxa de *Half-Total Error Rate* (HTER) do sistema, a média entre a FDR e a FNDR dos vídeos de teste (dado o limiar fixado), a qual é a métrica final para a base Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012). A Tabela 11.2 mostra as taxas de EER e HTER da wCNN proposta, da PatchNet operando sobre faces completas e de outras abordagens estado-da-arte: *Fine-Tuned* VGG-Face (LI et al., 2016), *Combined Deep Part*

CNN (LI et al., 2016), *Random Patches* CNN (ATOUM et al., 2017), *Color* LBP (BOULKENAFET; KOMULAINEN; HADID, 2015) e CSURF (BOULKENAFET; KOMULAINEN; HADID, 2017).

Tabela 11.2: Resultados (%) em termos de EER no conjunto de validação e de HTER no conjunto de teste para a base Replay-Attack da arquitetura proposta (wCNN) e outras abordagens estado-da-arte. Quanto menores os valores, melhor o método. Os melhores valores estão destacados. Fonte: Elaborada pelo autor.

Método	EER	HTER
Fine-Tuned VGG-Face	8,40	4,30
PatchNet	7,28	10,37
wCNN	4,41	4,50
Combined Deep Part CNN	2,90	6,10
Random Patches CNN	2,50	**1,25**
Color LBP	0,40	2,90
CSURF	**0,10**	2,20

Como pode-se observar, os resultados obtidos pela wCNN foram melhores que os do método baseado na profunda VGG-Face (LI et al., 2016). Também é importante notar que, em plataforma de *hardware* restrito, contendo uma única GPU Nvidia GeForce GTX 980, com 4 GB de memória dedicada, não foi possível trabalhar satisfatoriamente com os métodos comparados devido a suas arquiteturas profundas e computacionalmente caras (algumas detalhadas em termos de eficiência na Seção 11.2.1). É importante mencionar ainda que, apesar de trabalhar com *patches*, como eles são selecionados de regiões aleatórias de cada face, o modelo de Atoum et al. (2017) também tende a aprender características faciais globais de *spoofing*, diferentemente de nossa abordagem. Além da complexidade da própria CNN, no trabalho de Atoum et al. (2017), os autores ainda combinam outra rede neural profunda capaz de estimar a profundidade facial a partir de imagens 2D para tentar melhorar ainda mais seus resultados. No entanto, não consideramos esse modelo complexo, pois é ainda menos apropriado para ambientes com restrições de *hardware*.

Em relação à base de imagens CASIA (ZHANG et al., 2012), por ter apenas conjuntos de treinamento e teste, calculamos o EER apenas para os vídeos de teste, como na literatura, após o treinamento do modelo wCNN no conjunto de treinamento, variando o limiar do função de integração no intervalo $[0;1]$. A Tabela 11.3 mostra os resultados obtidos neste conjunto de dados. Mais uma vez, obtivemos melhores resultados do que outros métodos de última geração que usam redes neurais muito complexas e profundas para a detecção de *spoofing* como: *Color* LBP (BOULKENAFET; KOMULAINEN; HADID, 2015), *Fine-Tuned* VGG-Face (LI et al., 2016), LSTM-CNN (XU; LI; DENG, 2015), *Common* CNN (YANG; LEI; LI, 2014), *Combined Deep Part* CNN (LI et al., 2016) e *Random Patches* CNN (ATOUM et al., 2017).

Tabela 11.3: Resultados (%) na base CASIA da CNN proposta (wCNN) e de outros métodos estado-da-arte. Os melhores valores estão destacados. Fonte: Elaborada pelo autor.

Método	EER
PatchNet	9,25
Color LBP	6,20
Fine-Tuned VGG-Face	5,20
LSTM-CNN	5,17
Common CNN	4,92
Combined Deep Part CNN	4,50
Random Patches CNN	**4,44**
wCNN	**4,44**

11.3 Conclusões

Neste capítulo, propõe-se uma arquitetura de CNN eficiente ampliada em largura ao invés de profundidade, denominada wCNN, para detecção robusta de *spoofing* facial. Além de apresentar resultados compatíveis com os de CNNs de última geração, bastante profundas, que sequer podem ser treinadas e avaliadas em GPU limitada, a arquitetura proposta economiza processamento e tempo no treinamento e teste, sendo bastante adequada para ambientes com restrições significativas de *hardware*.

Os resultados precisos da wCNN nas bases de imagens Replay-Attack e CASIA mostram que o aumento em largura da PatchNet, além de poupar recursos computacionais e tempo, representa uma boa alternativa para detecção de *spoofing* facial, inclusive dado que cada parte da rede neural pode ser treinada separadamente, em cada região facial, permitindo ainda que o modelo aprenda características locais de *spoofing*, as quais mostraram-se importantes para a detecção de ataques, melhorando também o desempenho do modelo proposto quando comparado a uma arquitetura CNN similar treinada em imagens de face completas (PatchNet) ou com arquiteturas trabalhando em *patches* aleatórios das faces.

Capítulo 12

DETECÇÃO DE ATAQUES E APRENDIZADO DE CARACTERÍSTICAS LOCAIS PROFUNDAS

Abordagens estado-da-arte, baseadas em Redes Neurais de Convolução (CNNs), apresentam bons resultados na detecção de *spoofing* facial. No entanto, como mencionado no Capítulo 11, esses métodos não consideram a importância do aprendizado de características locais profundas de cada região facial, embora se saiba, com base nos métodos de detecção e reconhecimento facial, que cada região facial apresenta diferentes aspectos visuais, que poderiam ser explorados para detecção de faces sintéticas, a fim de tornar o processo mais acurado. Neste capítulo propõe-se uma nova arquitetura de CNN atenta a características locais das faces para a detecção de ataques, treinada em duas etapas para tal tarefa. Inicialmente, cada parte da rede neural aprende características de uma dada região facial. Depois, todo o modelo é refinado nas imagens faciais completas. Os resultados mostram que essa etapa de pré-treinamento permite que a CNN aprenda diferentes pistas locais de *spoofing*, melhorando o desempenho e a velocidade de convergência do modelo, tornando o processo de treinamento mais eficiente e acurado.

Este capítulo refere-se ao conteúdo do artigo **"On the Learning of Deep Local Features for Robust Face Spoofing Detection"**, publicado e apresentado na *Conference on Graphics, Patterns and Images* (SIBGRAPI) em 2018, e a novos resultados que fazem parte de uma versão ampliada do referido artigo, em fase de submissão a periódico qualificado.

12.1 Abordagem Proposta

A informação espacial é extremamente importante em tarefas que envolvem faces, como a detecção facial (VIOLA; JONES, 2001) e o reconhecimento facial (TURK; PENTLAND, 1991; CHI-

ACHIA et al., 2014). Os diferentes padrões visuais de cada região facial codificam informações ricas e discriminativas necessárias para distinguir uma face de outros objetos, e também de outras faces. Em relação à detecção de *spoofing*, alguns trabalhos baseados em características *handcrafted* mencionaram que diferentes pistas de *spoofing* podem ser extraídas de diferentes regiões faciais (CHINGOVSKA; ANJOS; MARCEL, 2012; AKHTAR; FORESTI, 2016).

Recentemente, as arquiteturas de Aprendizado em Profundidade surgiram como boas alternativas para resolver problemas complexos e alcançaram resultados de ponta em muitas tarefas devido ao seu grande poder de abstração e robustez, trabalhando com características de alto nível, autoaprendidas a partir dos dados de treinamento (LECUN; BENGIO; HINTON, 2015; CANZIANI; PASZKE; CULURCIELLO, 2016). Entre as arquiteturas de aprendizagem profunda propostas, as Redes Neurais por Convolução (CNN) (LECUN et al., 1998) apareceram como uma das classes mais importantes de redes neurais profundas capazes de lidar com imagens digitais com grande desempenho.

Como já mencionado na Seção 4.2, alguns métodos estado-da-arte baseados em CNNs foram recentemente propostos para detecção de faces sintéticas (YANG; LEI; LI, 2014; XU; LI; DENG, 2015; LI et al., 2016; ATOUM et al., 2017; JOURABLOO; LIU; LIU, 2018). No entanto, nenhum deles leva em consideração os diferentes aspectos visuais de cada região facial e, consequentemente, os diferentes traços locais de *spoofing* que poderiam ser aprendidos pelas redes neurais para melhorar seus desempenhos. Todos os métodos propostos funcionam sobre imagens faciais completas, de forma holística, ou com *patches* aleatórios, isto é, o treinamento das redes neurais se dá com amostras extraídas de regiões aleatórias das faces, todas juntas. Isso pode degradar o desempenho do algoritmo de treinamento, já que o método de *backpropagation* pode acabar sendo distraído pelas diferentes informações visuais dos elementos faciais presentes nas diferentes regiões da face, em vez de efetivamente aprender as diferenças entre faces reais e sintéticas em cada região, com aspectos visuais semelhantes, diferindo apenas pelos traços de *spoofing*.

Neste contexto, neste capítulo propõe-se uma nova arquitetura de CNN treinada em duas etapas para um melhor desempenho na detecção de *spoofing* facial (além de eficiência na convergência do modelo no treinamento): (i) a fase de pré-treinamento local, na qual cada parte do modelo é treinada em cada região facial principal, aprendendo características profundas locais para a detecção de ataques e inicialização de todo o modelo em uma melhor posição no espaço de hiperparâmetros (a rede aprende a detectar múltiplas e diferentes pistas de *spoofing* de todas as regiões faciais); (ii) a fase global de ajuste fino, na qual todo o modelo é refinado com base nos pesos aprendidos independentemente por suas partes e em imagens faciais reais e

sintéticas inteiras, a fim de melhorar sua generalização. Os resultados obtidos nas bases Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012) e CASIA FASD (ZHANG et al., 2012), mostram que a etapa de pré-treinamento melhora a acurácia do modelo final e acelera sua velocidade de convergência durante o treinamento, aumentando sua eficiência. A abordagem proposta obteve resultados compatíveis com os dos métodos estado-da-arte mesmo trabalhando com uma arquitetura de CNN compacta.

12.1.1 Informações Espaciais e Regiões Faciais

A relação espacial entre os elementos e regiões faciais nas imagens codifica informações ricas que podem ser usadas para distinguir uma face do plano de fundo, de outros objetos ou mesmo de outras faces (TURK; PENTLAND, 1991; VIOLA; JONES, 2001). Os primeiros trabalhos de detecção e reconhecimento automatizado de faces já usavam esse tipo de informação, apresentando bons resultados e eficiência.

Em relação à detecção facial, os primeiros trabalhos de Viola e Jones (2001) usaram características de Haar para detectar a presença de faces em imagens digitais. Em suma, aplica-se, a cada área de uma determinada imagem, um classificador em cascata que verifica, hierarquicamente, se as principais características faciais estão presentes nesta área. As características de Haar capturaram contrastes entre regiões vizinhas na imagem (no caso, regiões claras e escuras vizinhas da face humana). A Figura 12.1 mostra duas características de Haar e suas correspondências com as regiões das faces humanas. Os retângulos pretos indicam que regiões mais escuras são esperadas, enquanto os retângulos brancos indicam que regiões mais claras são esperadas em uma determinada área. A característica mostrada ao meio foca em regiões mais escuras acima de regiões claras, típicas dos olhos (especialmente devido às sobrancelhas). A característica à direita procura o contraste do nariz (testa) e dos olhos em faces humanas.

Figura 12.1: Esquerda: face detectada com base nas características de Haar empregadas por Viola e Jones (2001). Centro e direita: exemplos de características de Haar. Fonte: Bradski (2000).

Com base no trabalho de Viola e Jones (2001), que permitiu a detecção automatizada de faces em tempo razoável para aplicações reais, muitos trabalhos foram posteriormente propostos que também exploraram os contrastes existentes nas regiões vizinhas da face (MITA; KANEKO;

HORI, 2005; MATHIAS et al., 2014; MA; BAI, 2016).

No contexto do reconhecimento facial, o primeiro método eficaz para cenários reais, dado o processo de alta complexidade de análise da face humana, foi proposto por Turk e Pentland (1991), baseado na Análise dos Componentes Principais (*Principal Component Analysis* - PCA) (PEARSON, 1901; HOTELLING, 1933), que pode ser usada para encontrar os autovetores mais discriminativos e que melhor descrevem a variância do conjunto de dados em análise (imagens faciais, neste caso), e reduzir a dimensionalidade do problema. Dada a similaridade de tais autovetores (quando representados como imagens 2D) com imagens faciais, Turk e Pentland os chamaram de autofaces (TURK; PENTLAND, 1991; DUSENBERRY, 2015). Como mostrado na Figura 12.2, é possível identificar os elementos e regiões faciais (e sua relação espacial) nas autofaces, indicando que esse tipo de informação é importante para diferenciar faces de pessoas distintas.

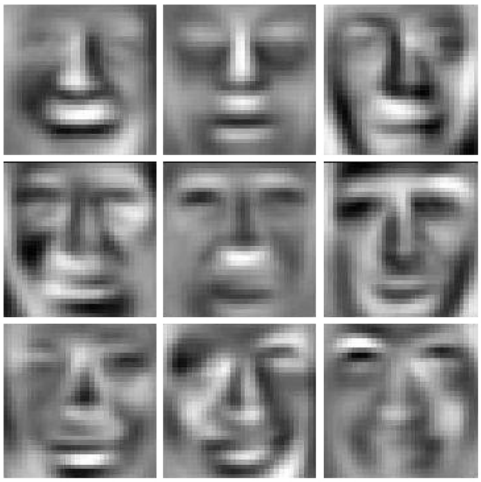


Figura 12.2: Autofaces, isto é, os autovetores mais discriminativos para descrever faces humanas de uma base de imagens faciais. Como pode ser visto, suas aparências lembram faces humanas e as diferentes regiões faciais são muito evidentes. Fonte: Dusenberry (2015).

Trabalhos baseados em outras transformações para reduzir a dimensionalidade do espaço de imagens faciais, como os baseados na Análise Discriminante Linear (*Linear Discriminant Analysis* - LDA) (FISHER, 1936), também costumam obter, como base do novo sistemas de coordenadas, vetores que lembram faces humanas quando vistos como imagens 2D, com as diferentes regiões faciais neles bastante salientes. As arquiteturas baseadas em CNNs para reconhecimento de faces, que aprendem as características mais discriminativas para representação facial a partir dos conjuntos de dados de treinamento, também capturam informações espaciais e relações entre elementos e regiões faciais, apresentando pesos de conexões entre neurônios que atuam como detectores dos elementos faciais (olhos, nariz, boca, etc.) e de seus posicionamentos (PARKHI; VEDALDI; ZISSERMAN, 2015).

Pesquisas da Psicologia mostram que os seres humanos têm uma habilidade extrema para detectar faces, com mais rapidez que qualquer outro objeto, e também destacam a importância da informação espacial e o posicionamento de cada região e elemento facial para a detecção e o reconhecimento de faces (LEWIS; ELLIS, 2003; PURCELL; STEWART, 1986). Purcell e Stewart (1986), por exemplo, constataram que o tempo requerido por um grupo de pessoas para identificar um estímulo visual como uma face era menor quando eram apresentadas imagens faciais normais do que quando apresentava-se faces com partes fora de lugar (a região da boca acima dos olhos, por exemplo).

Apesar de tudo isto, até então, nenhum trabalho da literatura investigou o uso de características locais profundas, aprendidas de cada região facial (com seu aspecto visual particular), para melhorar o desempenho, em termos de acurácia e eficiência, das redes neurais em profundidade na detecção de *spoofing* facial.

Embora haja esta falta de atenção em relação ao uso de características locais profundas para a detecção de *spoofing* facial na literatura, Krizhevsky (2009) demonstrou (trabalhando com redes baseadas em RBMs), em outras tarefas de classificação de imagens, que o uso de regiões locais (e fixas) das mesmas (informações visuais locais), em uma etapa inicial do treinamento do modelo de aprendizagem profunda, tende a melhorar seu desempenho, evitando também ficar preso em mínimos locais no espaço de busca de hiperparâmetros. Ba et al. (2016) também sugeriram o uso de *patches* faciais para inicializar modelos profundos aplicados ao reconhecimento facial baseados em estudos da Neurociência. O trabalho proposto por Santos, Souza e Marana (2017) também usa este passo inicial de treinamento baseado em *patches* fixos das imagems para melhorar a classificação visual de veículos nelas presentes.

12.1.2 lsCNN

Inspirado na abordagem de Krizhevsky (2009), neste trabalho propõe-se uma nova arquitetura de CNN para detecção de *spoofing* facial, denominada de lsCNN (*Locally Specialized* CNN), com um novo algoritmo de treinamento para um aprendizado mais efetivo das características de *spoofing* profundas locais, baseado em duas etapas: (i) fase de pré-treinamento local, na qual cada parte do modelo é treinada em cada região facial principal (pré-definida e fixada), aprendendo características locais profundas para a detecção de ataques e permitindo inicializar todo o modelo em uma melhor posição no espaço de busca de hiperparâmetros; e (ii) a fase de *fine-tuning* global, na qual todo o modelo é ajustado com base nos pesos aprendidos independentemente por suas partes nas regiões faciais e na apresentação de novas faces (reais e sintéticas) completas à rede neural, a fim de melhorar sua generalização.

Basicamente, a lsCNN apresenta 4 camadas convolucionais e de *pooling* (*CONV1/POOL1* a *CONV4/POOL4*) na parte inferior, com cada operação de convolução imediatamente seguida por uma operação de *Batch Normalization* (BN) e retificação de sinal (ReLU). A BN, como explanado no Capítulo 3, serve para normalizar os *feature maps* de saída das camadas convolucionais, melhorando o aprendizado (IOFFE; SZEGEDY, 2015). A função de retificação, em cada neurônio, atua como função de ativação, eliminando valores negativos nos *feature maps* resultantes da convolução e também acelerando o treinamento. No topo da rede, há uma camada totalmente conectada (*FC1*), seguida também por operações de BN e ReLU, bem como uma camada de *Dropout* (*DROP1*). Finalmente, há uma camada *softmax* com dois neurônios para classificar as faces em reais ou sintéticas. A Tabela 12.1 apresenta a arquitetura da lsCNN em termos de suas camadas, ou seja, tamanho de *kernels*, *strides*, tamanhos de *feature maps* de entrada e saída.

Tabela 12.1: Arquitetura proposta da lsCNN. A entrada da CNN são imagens faciais RGB (3 canais) com 96×96 **pixels: estruturas de tamanho** $3 \times (96 \times 96)$**. Fonte: Elaborada pelo autor.**

Camada	Tamanho do *Kernel*	*Stride*	Entrada	Saída
CONV1	3×3	1	$3 \times (96 \times 96)$	$27 \times (94 \times 94)$
POOL1	2×2	2	$27 \times (94 \times 94)$	$27 \times (47 \times 47)$
CONV2	3×3	1	$27 \times (47 \times 47)$	$36 \times (45 \times 45)$
POOL2	2×2	2	$36 \times (45 \times 45)$	$36 \times (23 \times 23)$
CONV3	3×3	1	$36 \times (23 \times 23)$	$45 \times (21 \times 21)$
POOL3	2×2	2	$45 \times (21 \times 21)$	$45 \times (11 \times 11)$
CONV4	3×3	1	$45 \times (11 \times 11)$	$54 \times (9 \times 9)$
POOL4	2×2	2	$54 \times (9 \times 9)$	$54 \times (5 \times 5)$
FC1	—	—	$54 \times (5 \times 5)$	$1 \times (450)$
DROP1	—	—	—	—
Softmax	—	—	$1 \times (450)$	$1 \times (2)$

Como mostrado na Tabela 12.1, a lsCNN espera imagens faciais de 3 canais (no espaço de cores RGB) como entrada. Embora outros espaços de cores permitam lidar com mais precisão com questões de iluminação, a fim de aproximar o modelo ao funcionamento do sistema visual humano (que capta apenas ondas de luz vermelhas, verdes e azuis) e sua percepção em condições naturais, bem como pelo fato de a maioria das câmeras digitais capturarem imagens no modo RGB, optou-se por essa representação em relação a outros modelos de cores.

12.1.2.1 Pré-Treinamento Local

Para inicializar todo o modelo da lsCNN em uma melhor posição no espaço de busca de hiperparâmetros e torná-la especializada em características de *spoofing* locais, em cada região

facial, divide-se cada face de treinamento em 9 principais regiões (*patches*), como mostrado na Figura 11.2, no Capítulo 11, regiões também adotadas para o reconhecimento facial (JAIN; ROSS; NANDAKUMAR, 2011).

Depois disso, também divide-se a arquitetura lsCNN em 9 CNNs independentes menores, chamadas de PatchNets para simplificar (com arquiteturas diferentes das PatchNets do Capítulo 11), apresentando, cada uma delas, um nono do tamanho do modelo original e sendo treinadas em cada uma das 9 regiões faciais principais consideradas, de $p1$ a $p9$. Cada PatchNet tem como entrada *patches* RGB com 32×32 pixels de uma região específica das faces de treinamento. A Tabela 12.2 mostra a arquitetura de cada PatchNet e a Figura 12.3 ilustra o processo de treinamento das 9 instâncias desta rede neural menor nas regiões faciais de uma determinada imagem. Como se pode observar, no topo de cada PatchNet estão 2 neurônios *softmax*, já que elas também são treinados para classificar seus respectivos *patches* em reais ou sintéticos.

Tabela 12.2: Arquitetura de cada CNN menor (PatchNet), parte da lsCNN, treinada em cada região facial, de $p1$ to $p9$ (*patches* com 32×32 pixels, também no espaço de cores RGB). Fonte: Elaborada pelo autor.

Camada	Tamanho do *Kernel*	*Stride*	Entrada	Saída
CONV1	3×3	1	$3 \times (32 \times 32)$	$3 \times (30 \times 30)$
POOL1	2×2	2	$3 \times (30 \times 30)$	$3 \times (15 \times 15)$
CONV2	3×3	1	$3 \times (15 \times 15)$	$4 \times (13 \times 13)$
POOL2	2×2	2	$4 \times (13 \times 13)$	$4 \times (7 \times 7)$
CONV3	3×3	1	$4 \times (7 \times 7)$	$5 \times (5 \times 5)$
POOL3	2×2	2	$5 \times (5 \times 5)$	$5 \times (3 \times 3)$
CONV4	3×3	1	$5 \times (3 \times 3)$	$6 \times (1 \times 1)$
POOL4	2×2	2	$6 \times (1 \times 1)$	$6 \times (1 \times 1)$
FC1	—	—	$6 \times (1 \times 1)$	$1 \times (50)$
DROP1	—	—	—	—
Softmax	—	—	$1 \times (50)$	$1 \times (2)$

12.1.2.2 *Fine-Tuning* Global

Depois de treinar as 9 redes neurais menores em suas respectivas regiões faciais, seus pesos e *biases* são usados para inicializar as partes da lsCNN para uma etapa de *fine-tuning* do modelo maior nas imagens faciais completas de treinamento, isto é, com 96×96 pixels, a fim de melhorar sua capacidade de generalização. Como mostrado na Figura 12.4, cada rede menor inicializa os pesos das conexões e *biases* de uma parte (um nono) do modelo lsCNN, do lado esquerdo (superior) para o lado direito (inferior). Os pesos da primeira PatchNet, por exemplo, inicializam as conexões entre os neurônios mais à esquerda do modelo lsCNN, responsáveis pelos primeiros *feature maps* de cada camada da rede ($FM1$ a $FM3$, no caso da primeira camada, e

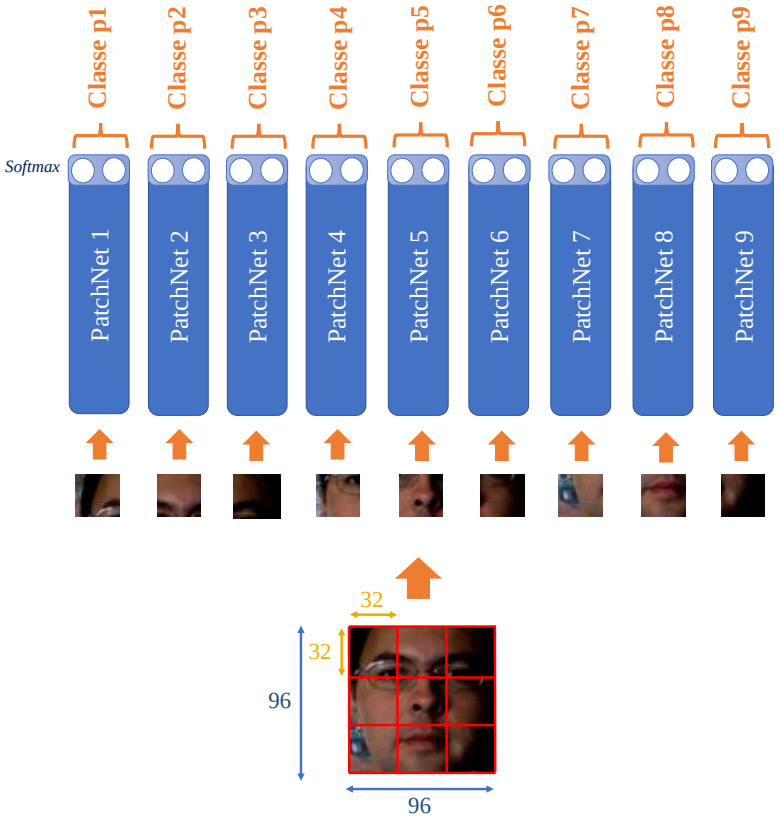


Figura 12.3: Ilustração do processo de pré-treinamento local da lsCNN. Dada uma imagem facial, ela é dividida em suas 9 regiões principais, de $p1$ a $p9$, e 9 instâncias de uma CNN menor (PatchNet) são treinadas em cada uma delas. Fonte: Elaborada pelo autor.

$FM1$ a $FM4$, na segunda camada), e assim por diante - similar ao trabalho de Krizhevsky (2009) com RBMs. As conexões da lsCNN entre os neurônios de diferentes partes são inicializadas em zero.

Os pesos das duas camadas totalmente conectadas no topo são inicializados aleatoriamente a partir de uma distribuição normal, a fim de melhorar ainda mais a generalização do modelo, como feito por Krizhevsky (2009). Seus *biases* são inicializados em zero. Na Figura 12.4, para simplificar, em cada parte da lsCNN, apenas as conexões de um neurônio em um determinado *feature map* para os neurônios da camada anterior são mostradas, bem como as conexões dos neurônios selecionados na primeira parte da lsCNN para seus *receptive fields* nas outras partes da rede neural. No entanto, a lsCNN apresenta todas as conexões de uma CNN tradicional.

Após a inicialização, as mesmas imagens faciais de treinamento (que foram divididas em *patches* na etapa anterior) são usadas para ajustar os pesos de todo o modelo lsCNN, permitindo também que ele detecte algumas características globais ou mais genéricas de faces completas, que não foram aprendidas localmente na etapa de pré-treinamento.

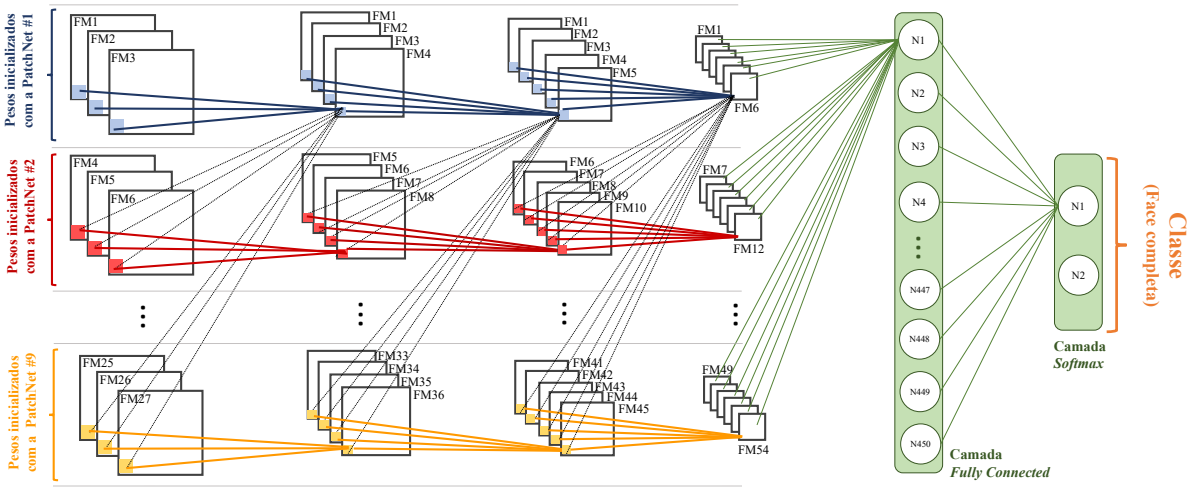


Figura 12.4: Inicialização do modelo lsCNN com base nos pesos das 9 PatchNets. As linhas coloridas mais grossas representam 3×3 conexões e são inicializadas com os pesos aprendidos por cada PatchNet. A primeira PatchNet, por exemplo, inicializa os pesos entre os primeiros neurônios (primeiros *feature maps* - FM) nas camadas da lsCNN. As linhas finas pretas pontilhadas também indicam 3×3 conexões, mas inicializadas em zero, e as linhas finas verdes são inicializadas com valores aleatórios de uma distribuição normal com média zero e desvio-padrão de $0,01$). As linhas horizontais cinza tracejadas estão representadas apenas para uma melhor visualização do processo de inicialização. Fonte: Elaborada pelo autor.

12.2 Experimento na Base NUAA

A base de imagens NUAA (*Nanjing University of Aeronautics and Astronautics*) (TAN et al., 2010), como descrita na Seção 5.3, contém imagens faciais em tons de cinza (já segmentadas e normalizadas) obtidas de faces reais e sintéticas: 3.491 imagens para treinamento e 9.123 imagens para teste. Realizou-se um experimento inicial neste conjunto menor de dados e, para isso, foi necessário reduzir a profundidade do modelo lsCNN, eliminando a terceira e quarta camadas convolucionais e de *pooling* devido ao menor tamanho das faces de entrada (64×64 pixels - *patches* de entrada com apenas 21×21 pixels). Dada essa redução em profundidade, para esse experimento aumentou-se a largura da lsCNN original: a primeira e segunda camadas convolucionais apresentaram 90 e 135 *feature maps* de saída, respectivamente. A camada totalmente conectada apresentou 1.350 neurônios e, seguindo-se LeCun et al. (1998), 5×5 *kernels* (com passo de 2 pixels) foram usados nas convoluções, dada a arquitetura com menos camadas. A primeira camada convolucional da lsCNN e das PatchNets tinha como entrada, respectivamente, $1 \times (64 \times 64)$ e $1 \times (21 \times 21)$ *feature maps* (trabalhando com imagens em escala de cinza).

O modelo lsCNN foi dividido em 9 partes normalmente e inicializou-se todos os pesos das PatchNets com base em valores aleatórios de uma distribuição normal com média zero e desvio-padrão de $0,0001$. Também normalizou-se as imagens faciais de entrada (antes de dividi-las em

seus 9 *patches*) subtraindo o valor médio do conjunto de treinamento e dividindo os valores dos pixels por 128, para garantir que a maioria deles ficasse no intervalo $[-1;1]$. Os *biases* dos neurônios das PatchNets foram todos inicializados em zero.

Como otimizador, empregou-se o método Adam (KINGMA; BA, 2015), com os seguintes parâmetros: 64 imagens de treinamento por *batch*, *learning rate* de $0,0001$, primeiro *momentum* de $0,9$ e segundo *momentum* de $0,999$. Primeiramente, as 9 PatchNets foram treinadas sobre seus respectivos *patches* por 2.000 iterações usando o *framework* Caffe (JIA et al., 2014). Depois, o modelo lsCNN foi inicializado com os pesos aprendidos nas PatchNets e treinado por mais 2.000, com o mesmo método otimizador e parâmetros, mas sobre as faces completas do conjunto de treinamento. Para comparação de desempenho, a fim de verificar se o pré-treinamento realmente surtia efeito na *performance* da lsCNN, também avaliou-se uma CNN com a mesma arquitetura da lsCNN, mas tradicionalmente treinada, ou seja, todos os seus pesos inicializados com valores aleatórios extraídos de uma distribuição normal com média zero e desvio-padrão de $0,0001$ (*biases* inicializados em zero) e treinada nas faces completas do conjunto de treinamento também por 2.000 iterações (seu ponto de convergência), com o mesmo método otimizador e parâmetros.

O objetivo deste experimento inicial foi, a princípio, verificar a melhoria no desempenho da lsCNN em comparação com uma CNN de mesma arquitetura, mas treinada tradicionalmente, considerando a mesma quantidade de iterações de treinamento (ambas treinadas por 2.000 iterações). A Figura 12.5 mostra as curvas ROC (*Receiver Operating Characteristics*) da lsCNN e da CNN tradicionalmente treinada em faces inteiras, aprendendo características globais. Como pode-se observar, a abordagem proposta apresentou uma curva ROC muito superior à curva da CNN tradicionalmente treinada. Com relação ao *Equal Error Rate* (EER), a lsCNN e a CNN treinada tradicionalmente obtiveram, respectivamente, 14,10% e 23,11%. Ou seja, a abordagem proposta foi novamente superior, segundo este novo critério, à CNN tradicionalmente treinada.

12.3 Experimentos nas Bases Replay-Attack e CASIA

A fim de permitir uma análise mais aprofundada da lsCNN, realizou-se experimentos maiores nas bases de imagens Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012) e CASIA FASD (ZHANG et al., 2012), ambas apresentadas na Seção 5.3. Desta vez, detectou-se e segmentou-se as faces nos quadros dos vídeos de ambos os conjuntos de dados usando a robusta rede neural MTCNN (*Multi-Task Convolutional Neural Network*) (ZHANG et al., 2016), CNN eficiente

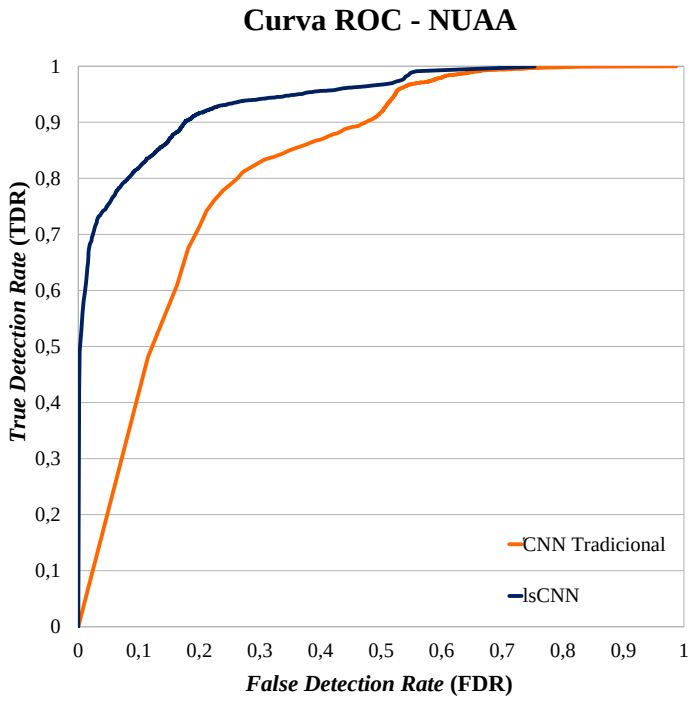


Figura 12.5: Curvas ROC da lsCNN e de uma CNN com mesma arquitetura, mas tradicionalmente treinada, isto é, treinada nas faces em tons de cinza da base NUAA, sem a etapa de pré-treinamento local. Quanto mais acima a curva, melhor o método. Fonte: Elaborada pelo autor.

treinada para detectar faces em imagens em múltiplas escalas, posições, poses, etc., para uma segmentação acurada das regiões faciais. Com base nos pontos de referência dos olhos de uma dada face, retornados como saída pela MTCNN, aplicou-se uma transformação de escala na respectiva imagem para normalizar a distância entre os dois olhos para 60 pixels (usando um algoritmo de redimensionamento de imagens baseado em interpolação a partir dos valores dos pixels mais próximos). Depois de detectar e normalizar a face em cada quadro, segmentou-se a região facial com base na posição dos olhos e capturando toda a face com uma janela de tamanho fixo de 96×96 pixels (no espaço de cores RGB). A Figura 12.6 ilustra algumas faces segmentadas com base neste algoritmo. Percebe-se a captura apenas da região facial, sem plano fundo, uma vez que o intuito era analisar as pistas de *spoofing* nas diferentes partes faciais. Nos experimentos em ambos os conjuntos de dados, para classificar um vídeo, considerou-se, a princípio, o esquema de maioria dos votos das faces em seus quadros. Quadros sem face detectadas pela MTCNN eram descartados.

Ao contrário do experimento com a base NUAA, nos experimentos com os conjuntos de dados Replay-Attack e CASIA, considerou-se a arquitetura original da lsCNN dadas as imagens faciais maiores obtidas (96×96 pixels). Depois de segmentar as faces de todos os quadros de todos os vídeos de treinamento, foi realizado um processo de aumento de base em ambas as bases de imagens. Em cada um delas, inicialmente e para cada imagem facial de treinamento,

Figura 12.6: Imagens faciais segmentadas pela MTCNN da base Replay-Attack (primeira linha) e CASIA (segunda linha). Percebe-se que não há região de fundo para melhor análise das regiões faciais. Fonte: Elaborada pelo autor.

gerou-se duas novas versões da mesma aumentando ou diminuindo os valores dos canais R, G e B em 50 unidades. Isso foi feito para forçar a rede a não confiar no brilho para a detecção de *spoofing* (não foram aplicadas técnicas para atenuar as sombras nas faces, pois elas são importantes para distinguir faces reais de faces falsas 2D).

Para cada uma das três versões de cada imagem facial de treinamento original, também foram aplicadas transformações de ruído ou borramento em três níveis cada (com baixas magnitudes para não afetar muito as imagens), para fazer com que a rede neural também aprendesse características de *spoofing* mais suaves e não confiasse somente em ruídos. A operação de borramento foi aplicada em três níveis (usando um filtro gaussiano de dimensão 2×2 e com desvios-padrão de $0,1$, $0,5$ e $1,0$), assim como o ruído gaussiano (com desvios-padrão de $0,0005$, $0,00075$ e $0,001$). Tais transformações foram aplicadas isoladamente, gerando, para cada uma das três imagens iniciais de uma dada face de treinamento, 6 representações dela. Nesse sentido, aumentou-se o conjunto de imagens faciais de treinamento 19 vezes (imagens originais e $3 \cdot 6 = 18$ imagens transformadas).

Para o conjunto de dados Replay-Attack, obteve-se 1.766.031 imagens faciais de treinamento e, para o conjunto de dados CASIA, 852.568 imagens. Novamente, inicializou-se os pesos das PacthNets menores com base em valores aleatórios de uma distribuição normal de média zero e desvio-padrão de $0,0001$ e normalizou-se cada canal das imagens faciais de entrada subtraindo o valor médio dele e dividindo todos os valores da imagem por 128, para garantir que a maioria deles pertencesse ao intervalo $[-1;1]$. Os *biases* dos neurônios das PatchNets foram todos inicializados em zero. Como otimizador, também empregou-se o método Adam (KINGMA; BA, 2015) em ambos os experimentos, com os mesmos parâmetros: 64 imagens de treinamento por *batch*, *learning rate* de $0,0001$, primeiro *momentum* de $0,9$ e segundo *momentum* de $0,999$.

Em ambos os experimentos, treinou-se as 9 PatchNets por 5.000 iterações nos *patches* faciais usando o *framework* Caffe (JIA et al., 2014) e inicializou-se todo o modelo lsCNN. Em seguida, treinou-se a lsCNN, também com o otimizador Adam e os mesmos parâmetros de treinamento, por mais 100.000 iterações. Para o conjunto de dados Replay-Attack, o melhor modelo da lsCNN foi obtido (tomando como base o conjunto de vídeos de validação) na **iteração 53.600**. Para a CNN com a mesma arquitetura da lsCNN, mas inicializada com valores aleatórios extraídos de uma distribuição normal com média zero e desvio-padrão de $0,0001$ (*biases* também inicializados em zero) e tradicionalmente treinada em imagens faciais completas (por 100.000 iterações, com o mesmo otimizador e iguais parâmetros que a lsCNN), o melhor modelo foi obtido apenas na **iteração 74.200**, ou seja, muito depois. Os resultados da abordagem proposta e dos métodos do estado-da-arte são apresentados na Tabela 12.3. Apresenta-se o EER (*Equal Error Rate*) na base de validação e o HTER (*Half-Total Error Rate*) na base de teste, após fixar o limiar de aceitação das faces no sistema (no limiar responsável pelo EER). Por simplicidade, denotamos a CNN tradicionalmente treinada com a mesma arquitetura da lsCNN como "CNN Tradicional". Dentre os métodos comparados encontram-se: *Efficient* VGG-Face (SOUZA et al., 2017), *Handcrafted Patches* (AKHTAR; FORESTI, 2016), *Whole Fine-Tuned* VGG-Face (LUCENA et al., 2017), *Fine-Tuned* VGG-Face (LI et al., 2016), *Combined Deep Part* CNN (LI et al., 2016), *Random Patches* CNN (ATOUM et al., 2017), MobileNet-v1 (HOWARD et al., 2017) e *Color* LBP (BOULKENAFET; KOMULAINEN; HADID, 2015).

Tabela 12.3: Resultados (%) na base Replay-Attack: *Equal Error Rate* (EER) no conjunto de validação e *Half-Total Error Rate* (HTER) nos vídeos de teste. Os melhores valores estão destacados. Fonte: Elaborada pelo autor.

Método	EER	HTER
Efficient VGG-Face	—	16,62
Handcrafted Patches	—	5,0
Whole Fine-Tuned VGG-Face	—	**1,20**
Fine-Tuned VGG-Face	8,40	4,30
Combined Deep Part CNN	2,90	6,10
Random Patches CNN	2,50	**1,25**
MobileNet-v1	1,67	3,13
Color LBP	0,40	2,90
CNN Tradicional	**0,33**	1,75
lsCNN	**0,33**	2,50

Como se pode observar, além de obter o menor EER, a lsCNN apresentou um ótimo HTER, muito menor do que valores obtidos por métodos computacionalmente caros, como o de Li et al. (2016), que trabalha com CNN extremamente complexa e profunda, a VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015). Apesar de ter obtido um resultado de HTER mais elevado do que

o obtido pela rede neural treinada tradicionalmente, a lsCNN obteve os resultados apresentados mais rapidamente (em uma iteração muito anterior do treinamento), como mencionado.

Vale observar também que a lsCNN obteve melhores resultados que a importante CNN denominada MobileNet-v1 (HOWARD et al., 2017), rede neural projetada para obter eficiência e poder ser executada em plataformas com maior restrição de *hardware*. O treinamento e a convergência da MobileNet também foram bem mais lentos que os da lsCNN, o resultado obtido de EER e HTER só foi encontrado na iteração 89.000. A rede MobilNet foi treinada, de 0 a 100.000 iterações, utilizando o *framework* Caffe (JIA et al., 2014), com base no código disponibilizado por Yang (2018).

Com relação ao experimento na base CASIA, o melhor modelo para a lsCNN foi obtido na **iteração 9.800**, enquanto que o melhor modelo para a CNN de mesma arquitetura tradicionalmente treinada foi obtido na **iteração 80.900**. A fim de comparar os desempenhos de tais métodos com as abordagens de última geração, medimos o EER, uma vez que esta base de imagens apresenta somente conjunto de dados de teste (não há conjunto de validação pré-definido). A Tabela 12.4 mostra os resultados. Os métodos comparados foram: MobileNet-v1 (HOWARD et al., 2017), *Fine-tuned* VGG-Face (LI et al., 2016), LSTM-CNN (XU; LI; DENG, 2015), *Common* CNN (YANG; LEI; LI, 2014), *Handcrafted Patches* (AKHTAR; FORESTI, 2016), *Combined Deep Part* CNN (LI et al., 2016) e *Random Patches* CNN (ATOUM et al., 2017).

Tabela 12.4: Resultados (%) na base CASIA da rede proposta lsCNN e outros método estado-da-arte. Os melhores valores estão destacados. Fonte: Elaborada pelo autor.

Método	EER
MobileNet-v1	7,78
Fine-tuned VGG-Face	5,20
LSTM-CNN	5,17
Common CNN	4,92
Handcrafted Patches	4,65
Combined Deep Part CNN	4,50
Random Patches CNN	**4,44**
CNN Tradicional	**4,44**
lsCNN	**4,44**

Como pode-se observar, a lsCNN obteve o menor EER no conjunto de dados CASIA, bem como a CNN tradicionalmente treinada e o trabalho de Atoum et al. (2017), resultado melhor que o de importantes abordagens da literatura, computacionalmente caras. Além disto, quando comparada com a CNN treinada tradicionalmente, o treinamento da lsCNN foi muito mais rápido (a lsCNN obteve seu melhor desempenho na iteração 9.800 frente a iteração 80.900 da CNN tradicionalmente treinada).

Como complemento, em termos de curvas ROC (de onde se obtiveram os valores de EER dos métodos avaliados neste trabalho), comparando-se as curvas da lsCNN e da CNN tradicionalmente treinada nas bases Replay-Attack e CASIA, pode-se notar melhor desempenho da arquitetura proposta em relação à CNN tradicional na base Replay-Attack e desempenhos próximos no caso da base CASIA, embora tenham obtido mesmos EERs nos dois casos (ambas CNNs com $0,33\%$ de EER na Replay-Attack e $4,44\%$ na CASIA) . A Figura 12.7 mostra as curvas das duas abordagens nas duas bases propostas.

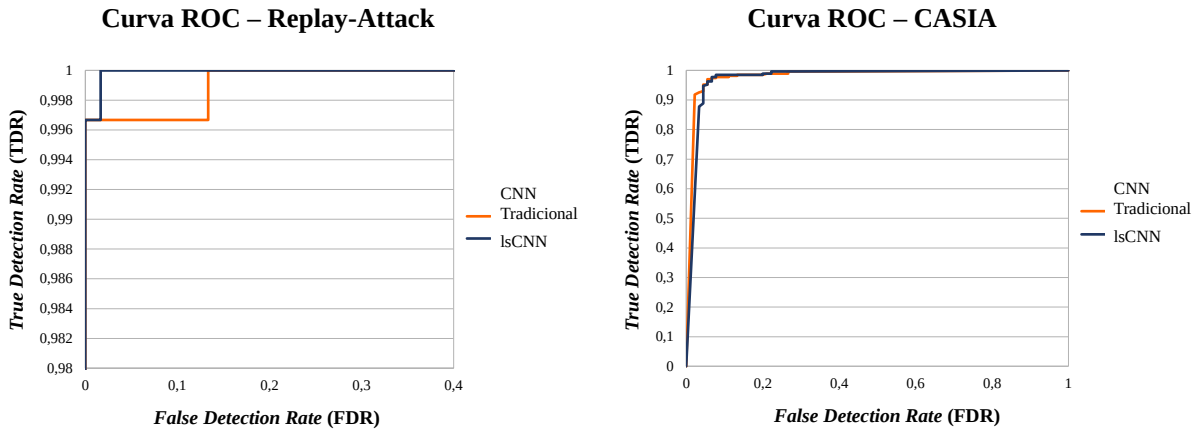


Figura 12.7: Curvas ROC para a lsCNN e a CNN tradicionalmente treinada sobre as bases Replay-Attack (à esquerda - sobre os vídeos do conjunto de validação) e a CASIA (à direita - sobre o conjunto de teste). As curvas da lsCNN, no geral, são ligeiramente superiores. Fonte: Elaborada pelo autor.

Uma outra tendência na avaliação de métodos de detecção de *spoofing* são os testes interbases, isto é, treina-se o método em uma base e testa-se em outra a fim de verificar sua capacidade de generalização uma vez que, geralmente, as imagens das bases diferem bastante em termos de iluminação, sensor de aquisição, ruídos, etc. Avaliou-se também a arquitetura lsCNN sendo treinada na Replay-Attack e testada na base CASIA e vice-versa. Do mesmo modo avaliou-se o desempenho da CNN tradicionalmente treinada (com a mesma arquitetura compacta da lsCNN) nos testes interbases. No caso do treino sobre a base Replay-Attack, treinou-se o modelo sobre o conjunto de vídeos de treinamento (na verdade tomou-se o melhor modelo obtido no teste intrabase), estimou-se o EER e o limiar ótimo para o sistema com base nos vídeos de validação e testou-se (encontrando o HTER) no conjunto de testes da base CASIA. Quando treinando sobre a base CASIA, utilizou-se os vídeos de treinamento para o aprendizado das redes (novamente tomou-se o melhor modelo do teste intrabase), os vídeos de teste para encontrar o EER e o limiar ótimo e o conjunto de vídeos de teste da base Replay-Attack para encontrar o HTER e testar o modelo. A Tabela 12.5 mostra os resultados de HTER obtidos pela lsCNN e pela CNN tradicio-

nalmente treinada, bem como de métodos estado-da-arte: LBP-TOP (PEREIRA et al., 2013), LBP (PEREIRA et al., 2013), *Common* CNN (YANG; LEI; LI, 2014), *Noise-Based* CNN (JOURABLOO; LIU; LIU, 2018), *Color* LBP (BOULKENAFET; KOMULAINEN; HADID, 2015) e *Depth and Blood Flow* CNN+LSTM (LIU; JOURABLOO; LIU, 2018).

Tabela 12.5: Resultados (%) em termos de HTER de testes interbases (considerando as bases Replay-Attack e CASIA) da lsCNN, da CNN tradicionalmente treinada e de outros método estado-da-arte. Os três melhores valores em cada caso estão destacados. Fonte: Elaborada pelo autor.

Método	Replay → CASIA	CASIA → Replay
LBP-TOP	60,6	49,7
LBP	57,6	55,9
CNN Tradicional	55,4	44,3
lsCNN	54,1	**32,8**
Common CNN	45,5	48,5
Noise-Based CNN	**41,1**	**28,5**
Color LBP	**39,6**	47,0
Depth and Blood Flow CNN+LSTM	**28,4**	**27,6**

Pode-se notar um expressivo ganho de desempenho da lsCNN sobre a CNN tradicionalmente treinada, indicando que o aprendizado de características profundas locais ajuda no processo de generalização da rede e a torna mais robusta a variações nas faces (iluminação, ruídos, resolução, etc.). Embora a lsCNN não tenha obtido os valores ótimos, ficou relativamente próxima do método proposto por Liu, Jourabloo e Liu (2018) no segundo teste (treino na CASIA e avaliação da Replay), sendo que para obter tal resultado só é necessário fazer *forward passes* das imagens faciais na arquitetura compacta da abordagem proposta. O trabalho de Liu, Jourabloo e Liu (2018), por sua vez, utiliza múltiplas redes neurais profundas, incluindo redes temporais para estimar a profundidade das faces bem como o batimento cardíaco a partir de informações visuais das mesmas, sendo bastante custoso.

12.3.1 Análise dos Pesos Sinápticos

Além das boas taxas de acerto e da convergência mais rápida durante o treinamento, outro fator que comprova que as informações profundas locais são bastante importantes no processo de detecção de *spoofing* são os próprios pesos aprendidos pela lsCNN, responsáveis pela detecção de características da faces a fim de classificá-las em reais ou sintéticas. A Figura 12.8 ilustra a magnitude dos pesos dos *kernels* de convolução (pesos sinápticos) entre a primeira e segunda camadas da lsCNN do melhor modelo para a base Replay-Attack após o *fine-tuning* global (iteração de treinamento 53.600). Quanto mais escuro o pixel na imagem mais próximo de zero e quanto mais claro, maior a magnitude do peso. Pode-se perceber que mesmo após a

execução do *fine-tuning* global, a rede manteve os pesos dos *kernels* herdados de cada PatchNet (na diagonal principal) mais elevados em comparação aos pesos entre partes distintas da rede, o que mostra que a rede continuou a se valer fortemente da detecção de características locais, mesmo após a etapa de generalização do treinamento.

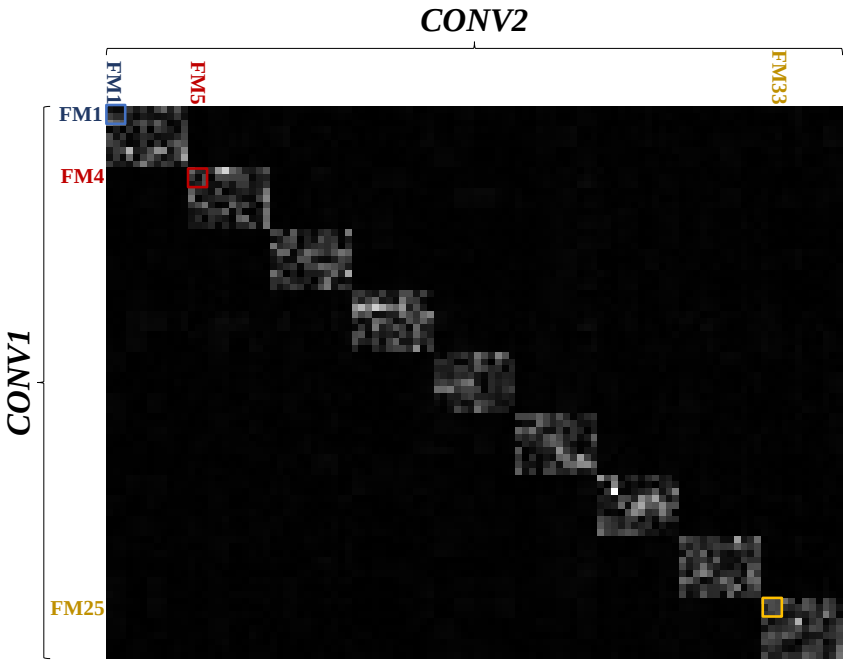
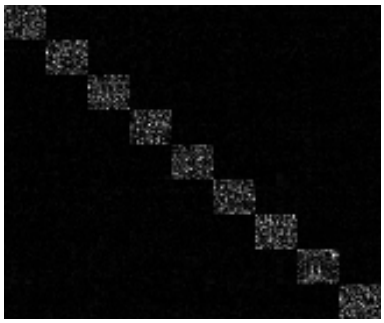


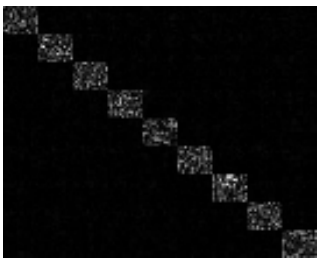
Figura 12.8: Pesos dos *kernels* 3×3 entre as camadas *CONV1* e *CONV2* da rede lsCNN para a base Replay-Attack. O quadrado azul indica, por exemplo, os 3×3 pesos do *kernel* entre o *feature map* de saída FM1 da camada *CONV1* e o *feature map* FM1 da camada *CONV2* (entre os neurônios da rede). Quanto mais claro o pixel, maior a magnitude do peso correspondente. Fonte: Elaborada pelo autor.

Vale observar que a preservação de grandes magnitudes dos pesos herdados das PatchNets não ocorre somente entre a primeira e segunda camada da lsCNN, a Figura 12.9 mostra também os pesos entre a segunda e terceira camadas e entre a terceira e quarta camadas convolucionais. Os pesos herdados das PatchNets (na diagonal principal) continuam com maiores magnitudes que os pesos entre partes distintas do modelo lsCNN. São exibidos os pesos entre as camadas convolucionais da CNN treinada tradicionalmente também. Percebe-se uma grande aleatoriedade neste caso.

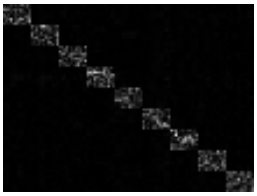
O mesmo ocorre para a rede lsCNN treinada sobre a base CASIA (melhor modelo na iteração 9.800) e a rede de mesma arquitetura mas treinada tradicionalmente (melhor modelo na iteração 80.900). Os pesos herdados das PatchNets continuam com magnitudes ligeiramente mais elevadas na lsCNN, ao contrário da rede tradicionalmente treinada, que apresenta pesos mais aleatórios, como mostra a Figura 12.10. Vale notar que para a base CASIA, a rede preservou menos os pesos aprendidos localmente, talvez por isto o desempenho da lsCNN tenha sido



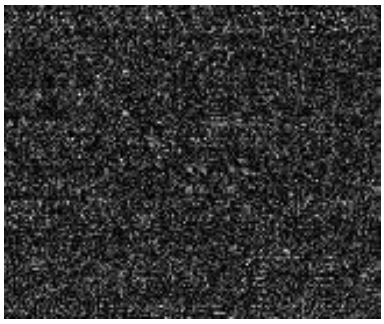
Pesos: *CONV1 - CONV2*



Pesos: *CONV2 – CONV3*

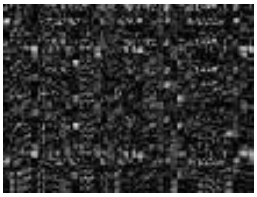


Pesos: *CONV3 – CONV4*



Pesos: *CONV1 - CONV2*

Pesos: *CONV2 – CONV3*



Pesos: *CONV3 – CONV4*

Figura 12.9: Pesos dos *kernels* entre as camadas *CONV1-CONV2*, *CONV2-CONV3* e *CONV3-CONV4* da rede lsCNN (ao topo) e da rede treinada tradicionalmente (na parte inferior) para a base Replay-Attack. Quanto mais claro o pixel, maior a magnitude do peso correspondente. Fonte: Elaborada pelo autor.

degradado e mais próximo ao da rede treinada de maneira tradicional.

Como reflexo dos pesos aprendidos, pode-se verificar também a natureza dos *feature maps* gerados pelas redes lsCNN e pela CNN tradicionalmente treinada a fim de identificar possíveis disparidades. A Figura 12.11 mostra os *feature maps* gerados pela lsCNN e pela CNN tradicionalmente treinada, dadas imagens de entrada da base Replay-Attack e CASIA. Pode-se observar que a lsCNN gera *feature maps* mais suaves, com as regiões faciais bem definidas, enquanto que os *feature maps* gerados pela CNN tradicionalmente treinada apresentam muitos ruídos (regiões de alta frequência) capturando detalhes finos demais das imagens de entrada, possivelmente dado o caráter mais randômico de seus pesos sinápticos. O pior desempenho de tal modelo (em termos de acurácia e tempo de convergência) pode ser explicado por este fato, isto é, por a rede basear sua detecção de ataques em detalhes ínfimos das imagens dados seus pesos randomizados.

Pesos: *CONV1 - CONV2*

Pesos: *CONV2 – CONV3*

Pesos: *CONV3 – CONV4*

Pesos: *CONV1 - CONV2*

Pesos: *CONV2 – CONV3*

Pesos: *CONV3 – CONV4*

Figura 12.10: Pesos dos *kernels* entre as camadas *CONV1-CONV2*, *CONV2-CONV3* e *CONV3-CONV4* da rede lsCNN (ao topo) e da rede treinada tradicionalmente (na parte inferior). Ambas as arquiteturas treinadas sobre a base CASIA. Quanto mais claro o pixel, maior a magnitude do peso correspondente. Fonte: Elaborada pelo autor.

12.3.2 Análise Estatística

Como apresentado, em ambos os experimentos, isto é, na base Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012) e na base CASIA (ZHANG et al., 2012), a lsCNN apresentou uma convergência muito mais rápida, tornando o treinamento mais eficiente, e com resultados estado-da-arte. Uma vez que o resultado da CNN pode depender dos valores de sua inicialização, da ordem das imagens de treinamento, dentre outros fatores, a fim de comprovar a superioridade da abordagem proposta sobre a CNN similar porém de treinamento tradicional, de forma estatística, isto é, indepedendente de possíveis variações dos parâmetros de treinamento, repetiu-se o treino da lsCNN e da CNN tradicional 5 vezes em cada base de imagens a fim de estimar, a cada iteração, parâmetros de acurácia, *loss* e os valores de EER e HTER nas imagens (vídeos) de validação e teste das bases, de forma a provar estatisticamente que o treinamento proposto da lsCNN, isto é, considerando informações profundas locais da face, é realmente superior ao treinamento tradicional de CNNs (sem considerar informações locais), ou seja, os resultados

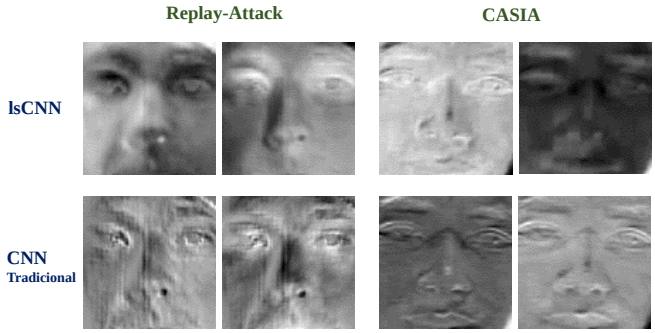


Figura 12.11: *Feature maps* gerados pela lsCNN (primeira linha) e pela CNN tradicionalmente treinada (segunda linha) na camada *CONV1* a partir das mesmas imagens da base Replay-Attack (dois *feature maps* mais à esquerda da linha) e da base CASIA (dois *feature maps* mais à direita da linha). Fonte: Elaborada pelo autor.

apresentados anteriormente não foram frutos de um caso isolado, mas o treinamento proposto realmente traz ganhos às CNNs para detecção de *spoofing* facial.

Deste modo, treinou-se 5 lsCNNs, da mesma forma que no experimento inicial, bem como 5 CNNs com mesma arquiteturas, mas da maneira tradicional (sem considerar informações locais), para cada base de imagens, de 0 a 100.000 iterações, utilizando o *framework* Caffe (JIA et al., 2014), em todos os casos. A cada treinamento, novos valores de inicialização (extraídos das curvas normais) bem como novas ordens de apresentação das imagens de treinamento às redes eram tomadas pelo próprio *framework* Caffe. A cada 50 iterações de treinamento media-se a acurácia da rede sob análise (classificando todas as imagens faciais do conjunto de validação, no caso da base Replay, e do conjunto de teste, no caso da base CASIA - percentual de imagens isoladas corretamente classificadas, não considerando classificação de vídeos) e o valor de *loss*, que representa o "custo" das classificações tomadas pela rede para as imagens de validação ou teste ("custo" das classificações incorretas). Dado o maior esforço computacional de cálculo, a cada 1.000 iterações também media-se o EER considerando as classificações dos vídeos das bases (baseadas nas votações dos seus quadros), bem como o HTER (para a base Replay-Attack), a fim de valiar a melhora de desempenho real das redes durante o treinamento.

Para cada base, após realizar as medidas de acurácia e *loss* para as 5 CNNs tradicionais durante as 100.000 iterações de treinamento, obteve-se valores médios para tal arquitetura. O mesmo se fez para as 5 lsCNNs treinadas. A Figura 12.12 mostra a evolução da acurácia média das 5 CNNs tradicionalmente treinadas e das 5 lsCNNs avaliadas para cada base de imagens. Pode-se perceber que a acurácia média das lsCNNs cresce muito mais rápido e já no início do treinamento do que a das CNNs tradicionais, em ambos os casos, inclusive ficando em patamares superiores nas iterações finais do treinamento.

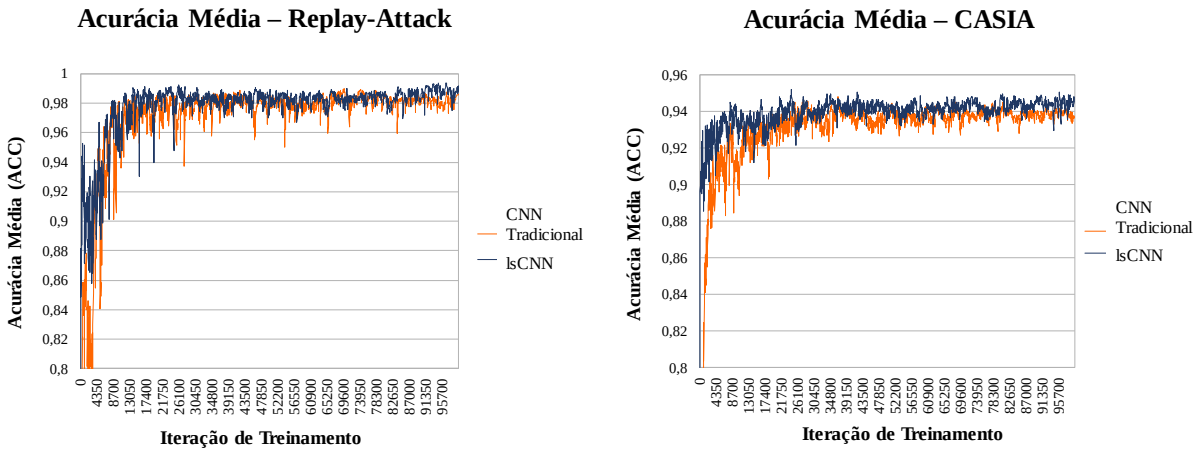


Figura 12.12: Acurácia média das 5 lsCNNs e das 5 CNNs tradicionalmente treinadas por iteração de treinamento, de 0 a 100.000, nas bases Replay-Attack e CASIA. Quanto mais alta a acurácia, melhor. Fonte: Elaborada pelo autor.

Por sua vez, mediu-se o *loss* médio a cada 50 iterações das 5 lsCNNs e as 5 CNNs tradicionalmente treinadas, em cada experimento (na base Replay-Attack ou CASIA). A Figura 12.13 mostra os resultados. Pode-se perceber que nas primeiras iterações, em ambos os casos, o *loss* médio das lsCNNs é maior dada a recém-inicialização dos pesos das PatchNets, o que e é esperado (KRIZHEVSKY, 2009), entretanto depois os valores médios de *loss* caem acentuadamente e mais rapidamente do que no caso das 5 CNNs tradicionais, permanecendo em patamares mais baixos, o que também indica melhores desempenhos das lsCNNs.

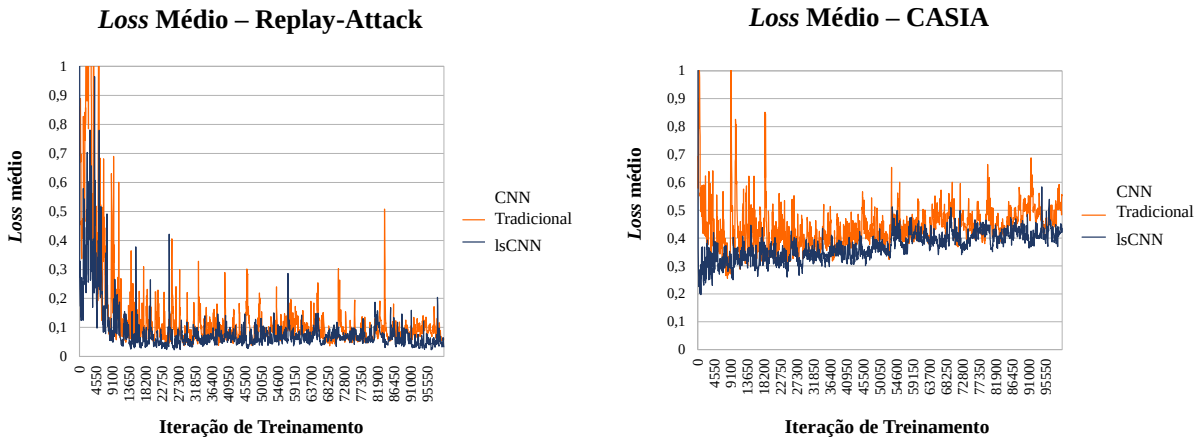


Figura 12.13: *Loss* médio das 5 lsCNNs e das 5 CNNs tradicionalmente treinadas, por iteração de treinamento, de 0 a 100.000, em cada base (Replay-Attack e CASIA). Quanto mais baixo o *loss*, melhor. Fonte: Elaborada pelo autor.

Em relação ao EER (e HTER para a base Replay-Attack), percebe-se que a queda nos

valores médios é bem mais rápida nas 5 lsCNNs do que nas CNNs treinadas normalmente, isto é, sem considerar informações faciais profundas locais. A Figura 12.14 ilustra o desempenho médio das 5 lsCNNs e das 5 CNNs comuns em relação ao EER e HTER (médios) na base Replay-Attack e ao EER médio na base CASIA.

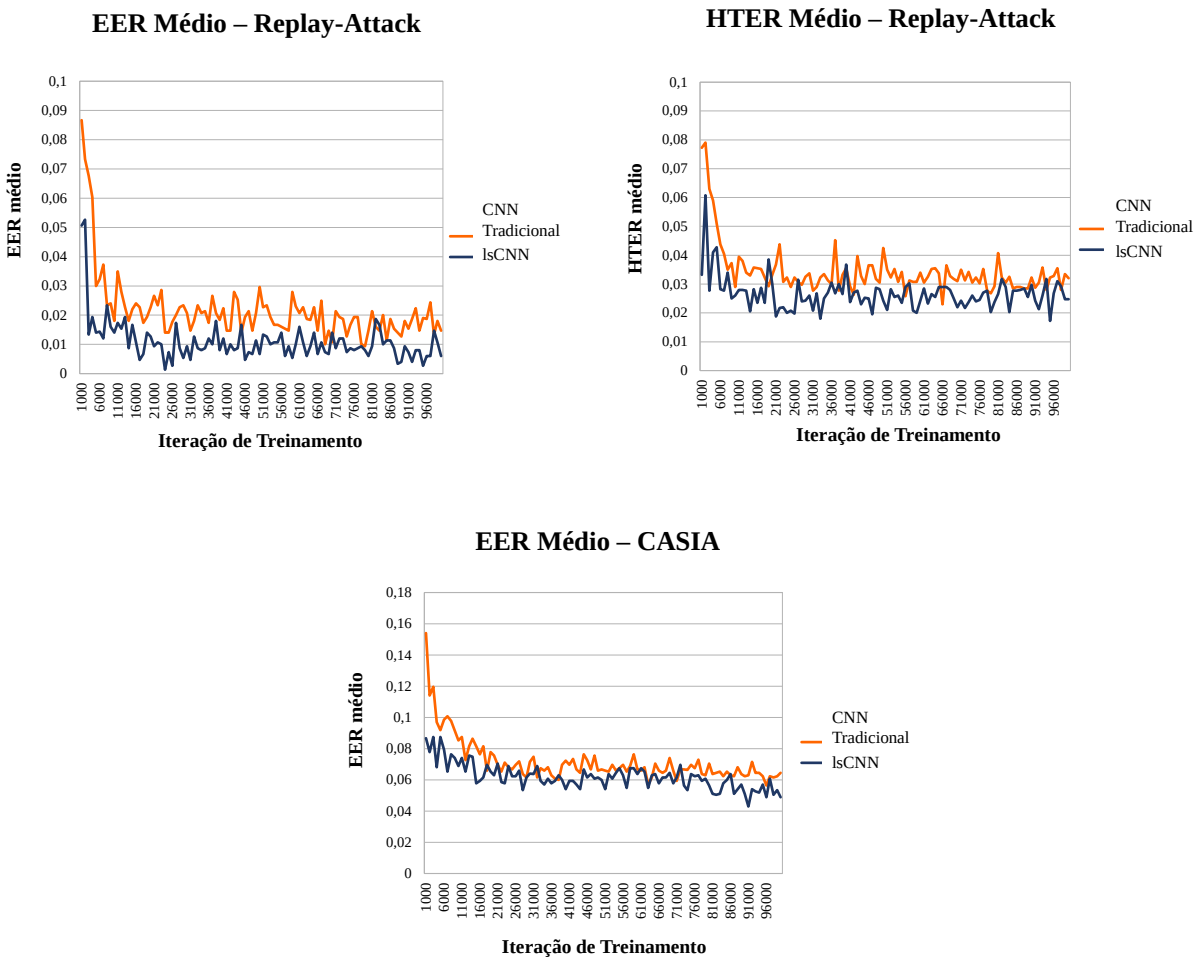


Figura 12.14: EER médio no conjunto de vídeos de validação e respectivo HTER médio no conjunto de vídeos de teste da base de imagens Replay-Attack bem como EER médio no conjunto de teste da base CASIA para as 5 lsCNNs avaliadas e as 5 CNNs tradicionais. Quanto menores os valores, melhor. Fonte: Elaborada pelo autor.

Pode-se perceber que em todos os gráficos, as curvas referentes aos valores médios das lsCNNs se apresentaram melhores durante quase todo o treinamento, já com expressivo ganho de desempenho no início do treino, o que comprova que o treinamento proposto com a inicialização da lsCNN com base em pesos aprendidos localmente nas faces melhora não só a eficiência da rede durante o aprendizado mas também suas taxas de acerto. Apesar de no experimento inicial sobre a base Replay-Attack a lsCNN ter obtido HTER de $2,50\%$ enquanto que a CNN tradicional obteve $1,75\%$, na média, os HTERs da lsCNNs são melhores (menores) que

das CNNs tradicionais.

Vale mencionar ainda que a iteração de convergência média, isto é, a média das iterações de melhor resultado, das 5 lsCNNs foi de **43.600** para a base Replay-Attack e **46.000** para a base CASIA, enquanto que a iteração de convergência média para as CNNs tradicionais foi de **51.300** na base Replay-Attack e **41.800** na base CASIA. Isto demonstra que, considerando as duas bases, em média, as lsCNNs também convergiram mais rapidamente do que as CNNs tradicionais, sendo mais eficientes no treinamento (aprendendo mais rápido).

Apesar da realização de apenas 5 treinamentos de cada arquitetura (devido ao custo necessário para tal feito) em cada base (Replay-Attack e CASIA), a fim de verificar a significância estatística da melhora de desempenho (em termos de acurácia) das lsCNNs em relação às CNNs tradicionais, para cada treinamento realizado selecionou-se o valor de acurácia máximo da respectiva rede neural (lsCNN ou CNN tradicional). Dados os 5 valores de acurácia máximos das lsCNNs e das CNNs tradicionais na base Replay-Attack e na base CASIA, calculou-se o *p-value* para cada base de imagens, a partir do teste *t* (assumindo-se a normalidade e homocedasticidade das distribuições de valores). Neste sentido, encontrou-se um ***p-value*=0,49%** para a base Replay-Attack e um ***p-value*=2,58%** para a base CASIA, valores bastante baixos (menos de 5% - percentual tido como referência de significância estatística na literatura), comprovando a superioridade da lsCNN em termos de acurácia, estatisticamente, nos repetidos experimentos realizados.

12.3.3 *Patches* Aleatórios

A fim de mostrar que as características locais profundas das faces são realmente as responsáveis pelo melhor desempenho da lsCNN em termos de eficiência no treinamento bem como nas melhores medidas (médias) de acurácia, *loss*, EER e HTER, e não apenas o fato da referida rede ser inicializada com pesos já otimizados por 5.000 iterações de treinamento (nas PatchNets), podendo-se alegar que o melhor resultado deve-se apenas a esta inicialização não-aleatória do modelo, em um ponto melhor do espaço de busca de hiperparâmetros (ao invés de posições aleatórias), treinamos mais 5 vezes a arquitetura lsCNN, porém, desta vez, misturando todos *patches* faciais de forma que cada PatcheNet fosse treinada por 5.000 iterações para inicializar a lsCNN, porém recebendo informações visuais de todas as partes das faces (ou seja não aprendendo características específicas de cada região facial), aleatoriamente, a fim de mostrar que mesmo com o pré-treinamento por 5.000 iterações, por não aprender características locais das faces, a rede não teria tão bom desempenho como no treinamento original da lsCNN.

A Figura 12.15 ilustra as taxas de EER e HTER médias (sobre as bases Replay-Attack e CA-

SIA) para a lsCNN tradicional, a lsCNN treinada sobre *patches* aleatórios e a CNN de mesma arquitetura tradicionalmente treinada. Pode-se observar que, embora haja uma melhora de desempenho da CNN tradicionalmente treinada para a lsCNN treinada sobre *patches* aleatórios, esta ainda apresenta desempenho pior que a lsCNN original, capaz de aprender características locais da face, demonstrando a importância do aprendizado de tais padrões locais.

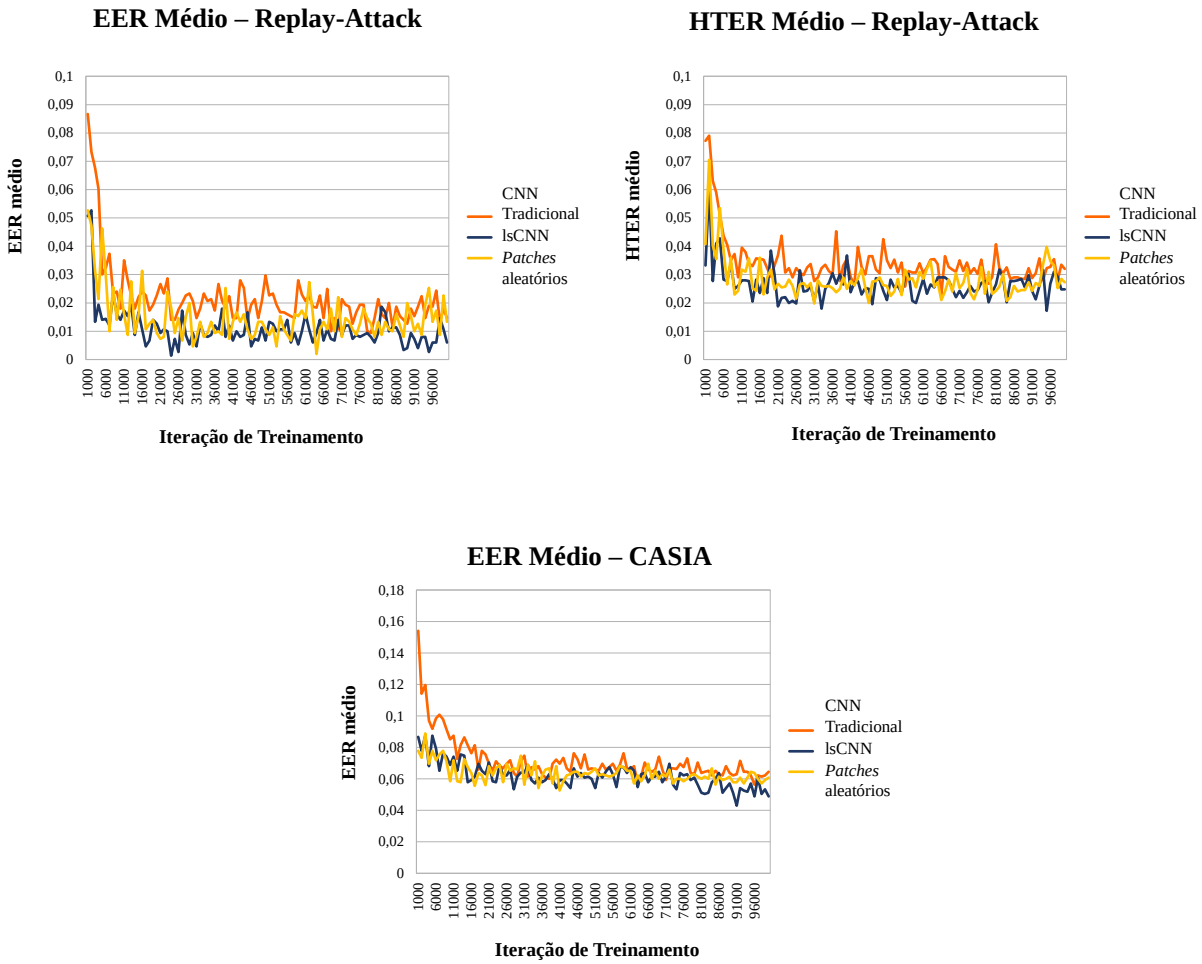


Figura 12.15: HTER e EER médio nas bases Replay-Attack e CASIA da lsCNN, da CNN tradicionalmente treinada e da lsCNN treinada sobre *patches* aleatórios. Quanto menores os valores, melhor. A lsCNN original apresentou o melhor desempenho. Fonte: Elaborada pelo autor.

12.3.4 Votação e Amostragem de *Frames*

A arquitetura lsCNN proposta apresenta apenas quatro camadas convolucionais, sendo bastante compacta. Em média o *forward pass* de uma imagem na MTCNN para detecção facial e da face segmentada na lsCNN toma apenas cerca de 0,2 segundos (quando sobre uma placa gráfica Nvidia Titan Xp). A fim de verificar a robustez da lsCNN em relação à amostragem

de *frames* para classificação dos vídeos bem como tornar o processo compatível com sistemas de tempo real (sendo que os vídeos das bases da literatura apresentam cerca de 25 *frames* por segundo), realizou-se experimentos amostrando os quadros dos vídeos da base Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012) e CASIA (ZHANG et al., 2012), cada vez mais espaçadamente, a fim de verificar se o desempenho da lsCNN se preservava em comparação com seu desempenho original (sem amostrar *frames*). Avaliou-se o desempenho dos melhores modelos da lsCNN e da CNN tradicional obtidos na Seção 12.3 para cada base. A Tabela 12.6 mostra os resultados originalmente obtidos na base Replay-Attack e CASIA (sem amostragem de quadros) e os resultados com a amostragem dos *frames* dos vídeos de validação e teste. Conforme pode-se observar, não houve grande perda de *performance* da lsCNN e da CNN tradicional, em comparação com os resultados originais da Seção 12.3, mesmo ao se amostrar um entre 16 quadros dos vídeos. Isto mostra a viabilidade da arquitetura proposta também em ambientes de tempo real.

Tabela 12.6: Resultados (%) nas bases Replay-Attack e CASIA com a amostragem de *frames* dos vídeos. O valor $1/n$ indica a amostragem de um quadro a cada n quadros consecutivos. Os resultados originais das redes neurais (sem amostragem) também constam para fins de comparação. Fonte: Elaborada pelo autor.

Método	Replay (EER)	Replay (HTER)	CASIA (EER)
CNN Tradicional (original)	0,33	1,75	4,44
lsCNN (original)	0,33	2,50	4,44
CNN Tradicional (1/2 *frames*)	0,33	1,63	4,44
lsCNN (1/2 *frames*)	0,33	2,50	5,18
CNN Tradicional (1/4 *frames*)	0,33	1,50	4,44
lsCNN (1/4 *frames*)	0,33	2,38	4,81
CNN Tradicional (1/8 *frames*)	0,33	1,50	5,19
lsCNN (1/8 *frames*)	0,33	2,50	5,56
CNN Tradicional (1/16 *frames*)	0,33	1,63	4,81
lsCNN (1/16 *frames*)	0,33	2,63	5,56

12.3.5 Espaços de Cores HSV e YCbCr

Ainda tratando-se dos experimentos com as bases Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012) e CASIA (ZHANG et al., 2012), como muitos métodos estado-da-arte trabalham com imagens faciais no espaço HSV e YCbCr ao invés do RGB, avaliamos a lsCNN também sobre estes espaços de cores, em complemento aos experimentos no espaço RGB. Os experimentos foram repetidos utilizando-se versões convertidas das imagens faciais. Como a MTCNN só detecta faces em imagens RGB, fazia-se a detecção, segmentação e normalização facial com as imagens em RGB, para o aumento de base aplicava-se as transformadas também em RGB,

e só então convertia-se as imagens para o um dos espaços de cores mencionados, HSV ou YCbCr. Os resultados em termos de EER e HTER para a base Replay-Attack e para a base CASIA encontram-se na Tabela 12.7.

Tabela 12.7: Resultados (%) da lsCNN e da CNN tradicionalmente treinada sobre imagens no espaço RGB (original), HSV e YCbCr para comparação. Os melhores valores em cada espaço de cores estão destacados. Fonte: Elaborada pelo autor.

Método	Replay (EER)	Replay (HTER)	CASIA (EER)
CNN Tradicional (original - RGB)	**0,33**	**1,75**	**4,44**
lsCNN (original - RGB)	**0,33**	2,50	**4,44**
CNN Tradicional (HSV)	**0,00**	2,63	6,67
lsCNN (HSV)	0,66	**2,25**	**6,30**
CNN Tradicional (YCbCr)	**0,00**	3,38	4,81
lsCNN (YCbCr)	**0,00**	**2,88**	**4,44**

Conforme pode-se perceber na Tabela 12.7, não houve muito ganho de desempenho nos espaços de cores HSV e YCbCr em relação aos resultados originais obtidos na Seção 12.3 (faces em RGB). Entretanto, vale ressaltar que, nos espaços HSV e YCbCr, a lsCNN apresentou resultados ainda melhores que a CNN tradicionalmente treinada, além de ter preservado uma convergência mais rápida (dezenas de iterações antes em ambos os espaços de cores), indicando que o pré-treinamento por *patches* também é válido e importante nestes outros espaços de cores e não só para faces em RGB.

12.3.6 Análise Temporal

Nos testes realizados sobre as bases Replay-Attack (CHINGOVSKA; ANJOS; MARCEL, 2012) e CASIA (ZHANG et al., 2012), como dito, classificou-se sempre os vídeos com base na votação de todos seus *frames*. Até então nenhuma característica temporal foi aprendida a fim de detectar ataques. Neste sentido, nesta subseção, arquiteturas de redes que trabalham com informações temporais são avaliadas. Repetiu-se os experimentos com a lsCNN na base Replay-Attack e CASIA mas trabalhando-se em duas frentes: (i) com a C3D (TRAN et al., 2015), CNN que efetua operações de convolução e *pooling* não só espaciais mas também temporais (entre *frames* consecutivos dos vídeos); e (ii) acoplando-se uma rede LSTM (HOCHREITER; SCHMIDHUBER, 1997) no topo da lsCNN (entre a camada completamente conectada e a *softmax*) a fim de capturar informações temporais profundas.

12.3.6.1 lsCNN e C3D

A arquitetura da lsCNN foi adaptada para se trabalhar com informações temporais seguindo a arquitetura da C3D, isto é, seus *kernels* de convolução e *pooling* máximo, que apresentavam tamanho 3×3 e 2×2, respectivamente, passaram a apresentar uma terceira dimensão a fim de trabalhar com múltiplos *frames* por vez, resultando em *kernels* de dimensões $3 \times 3 \times 3$ e $2 \times 2 \times 2$, respectivamente. O restante da arquitetura da lsCNN foi preservado (número de camadas, quantidade de *feature maps*, os *strides* dos *kernels*, etc.).

Realizou-se experimentos considerando clipes (sequências de quadros) de 32, 64 e 128 *frames* dos vídeos com a arquitetura proposta, denominada lsC3D. Para treinar tal arquitetura, utilizou-se o mesmo algoritmo da lsCNN tradicional: treinou-se cada parte da lsC3D em uma sequência de *patches* de uma mesma região facial e depois os pesos dos *kernels* tridimensionais locais serviram da base para inicialização do modelo maior, que sofria então um ajuste fino sobre clipes de imagens faciais completas.

Comparou-se o desempenho da lsC3D e de CNN de igual arquitetura, mas tradicionalmente treinada, ao aumentar o tamanho dos clipes, de 32 a 128 *frames*, como dito. Para classificar um vídeo baseou-se na classificação de todos seus clipes (maioria de seus votos). A inicialização dos modelos foi feita da mesma maneira: pesos das PatchNets e do modelo tradicionalmente treinado extraídos de distribuição gaussiana com média zero e desvio-padrão $0,0001$ e *biases* em zero. O mesmo otimizador (Adam) e mesmos parâmetros de treinamento foram empregados no treinamento da lsC3D e da CNN 3D tradicional. A Tabela 12.8 ilustra os resultados obtidos em cada caso para a base Replay-Attack bem como os resultados originais da lsCNN e CNN (2D) tradicionalmente treinada (da Seção 12.3).

Tabela 12.8: Resultados (%) da lsC3D e da CNN 3D tradicionalmente treinada na base Replay-Attack (e da lsCNN e CNN de mesma arquitetura originais). Melhores valores destacados. Fonte: Elaborada pelo autor.

Método	EER	HTER
CNN Tradicional (original)	0,33	1,75
lsCNN (original)	0,33	2,50
CNN 3D Tradicional (32 *frames*)	1,67	1,50
lsC3D (32 *frames*)	**0,00**	**1,00**
CNN 3D Tradicional (64 *frames*)	3,33	2,13
lsC3D (64 *frames*)	1,67	2,13
CNN 3D Tradicional (128 *frames*)	0,67	1,38
lsC3D (128 *frames*)	1,67	3,38

Conforme pode-se observar, os resultados (trabalhando-se com 32 *frames*) foram bem me-

lhores para a lsC3D do que quando não se trabalhando com a extração de informações temporais (somente baseado na votação de *frames* - modelo lsCNN). O pré-treinamento em *patches* também melhorou os resultados em se trabalhando com redes convolucionais espaço-temporais (exceto no caso de 128 *frames*, cujo modelo de CNN 3D treinado tradicionalmente já havia obtido resultados muito bons - 0,67% e 1,38% de EER e HTER, respectivamente). Vale observar que, dado o maior esforço computacional, em cada caso, treinou-se a lsC3D e a respectiva CNN 3D tradicional por apenas 50.000 iterações. No caso da base Replay-Attack, isoladamente, as lsC3Ds demoraram cerca de 15.000 iterações a mais para convergir (obtenção do melhor modelo), em cada caso, do que as CNNs 3D tradicionalmente treinadas.

Em se tratanto da base CASIA, a Tabela 12.9 apresenta os resultados obtidos. Observa-se que, neste caso, o pré-treinamento local não ajudou muito no desempenho da rede neural espaço-temporal (resultados ligeiramente piores para a lsC3D do que para a CNN 3D tradicionalmente treinada) e as informações temporais não melhoraram o desempenho da lsCNN original (os resultados foram piores do que os apresentados na Seção 12.3), porém merece ser mencionado que as convergências das lsC3Ds foram, em geral, mais rápidas (cerca de 10.000 iterações antes da convergência das respectivas CNNs 3D tradicionalmente treinadas). Como dito, treinou-se as redes neurais até a iteração 50.000 apenas, devido ao esforço computacional envolvido.

Vale observar que tanto no experimento com a base Replay-Attack quanto com a base CASIA, apenas um treinamento por tipo de CNN foi realizado - lsC3D e CNN 3D tradicionalmente treinada. Caso repetidos, na média de vários experimentos, os resultados de taxas de erros tendem a ser melhores para a lsC3D bem como a convergência mais rápida. É importante ressaltar também que a piora no desempenho, na base CASIA, ao se trabalhar com informações temporais, pode ter se dado devido à existência de vídeos muito curtos, alguns com menos de 128 *frames*, motivo pelo qual também não foi possível avaliar as CNNs trabalhando-se com 128 quadros (alguns vídeos de teste seriam excluídos do processo por serem muito curtos, tornando a comparação com os demais experimentos injusta).

12.3.6.2 lsCNN e LSTM

A fim de avaliar o aprendizado de características temporais utilizando redes neurais recorrentes, mais especificamente redes neurais LSTM, e verificar a possível melhora de desempenho da lsCNN com a incorporação de tais redes, acoplou-se uma unidade LSTM (com saída de 100 posições) entre a camada completamente conectada da lsCNN e a camada *softmax*, como mostra a Figura 12.16. Treinou-se cada PatchNet também com uma unidade LSTM (com cerca

Tabela 12.9: Resultados (%) da lsC3D e da CNN 3D tradicionalmente treinada na base CASIA (e da lsCNN e CNN de mesma arquitetura originais). Melhores resultados destacados. Fonte: Elaborada pelo autor.

Método	EER
CNN Tradicional (original)	**4,44**
lsCNN (original)	**4,44**
CNN 3D Tradicional (32 *frames*)	7,45
lsC3D (32 *frames*)	8,55
CNN 3D Tradicional (64 *frames*)	7,34
lsC3D (64 *frames*)	10,30

de um nono do tamanho da LSTM associada à lsCNN) sobre sequências de cada região facial e, após o treinamento local, utilizou-se os pesos das PatchNets para inicializar a lsCNN, até a camada $CONV4/POOL4$, como de costume. Os pesos da LSTM associada à lsCNN bem como os pesos das camadas completamente conectadas foram todos randomicamente inicializados baseados em uma distribuição normal de média zero e desvio-padrão $0,01$ (para fins de generalização da arquitetura), ou seja, não se inicializou a LSTM da lsCNN com os pesos aprendidos localmente pelas LSTMs das PatchNets.

Avaliou-se a arquitetura, denominada de lsLSTM para fins de simplicidade, sobre sequências de *frames* de comprimento 32 e 64 quadros. Devido ao alto custo computacional e ao fato de alguns vídeos da base CASIA sequer apresentarem 128 *frames*, não se avaliou os resultados usando sequências deste comprimento (como feito para a C3D). A Tabela 12.10 apresenta os resultados obtidos sobre as bases Replay-Attack e CASIA pela lsLSTM e pela CNN+LSTM (mesma arquitetura da lsLSTM) tradicionalmente treinada.

Tabela 12.10: Resultados (%) da lsLSTM e de rede de mesma arquitetura tradicionalmente treinada nas bases Replay-Attack e CASIA (e da lsCNN e CNN de mesma arquitetura originais). Melhores resultados destacados. Fonte: Elaborada pelo autor.

Método	Replay (EER)	Replay (HTER)	CASIA (EER)
CNN Tradicional (original)	**0,33**	**1,75**	**4,44**
lsCNN (original)	**0,33**	2,50	**4,44**
CNN+LSTM Tradicional (32 *frames*)	6,67	5,25	7,41
lsLSTM (32 *frames*)	3,33	2,50	8,14
CNN+LSTM Tradicional (64 *frames*)	15,67	18,50	21,85
lsLSTM (64 *frames*)	15,67	15,88	37,91

Conforme pode-se observar a inicialização por *patches* melhorou o desempenho da arquitetura lsLSTM, em geral, em relação à CNN+LSTM tradicional, porém, diferentemente da C3D, não houve ganhos em relação ao desempenho da lsCNN original. Assim como no trabalho

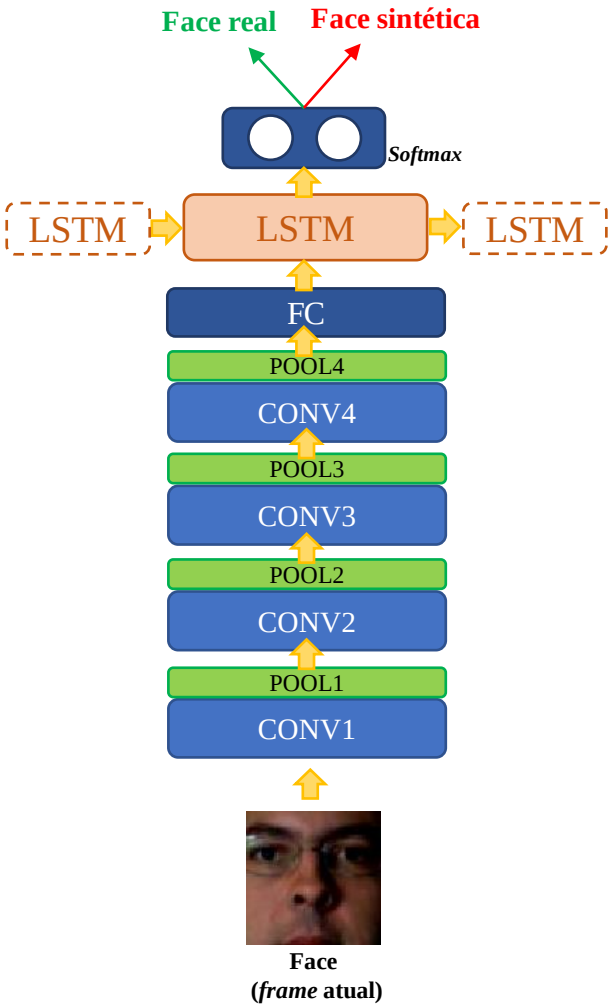


Figura 12.16: Arquitetura proposta com a LSTM entre a camada FC (*Fully Connected*) e a *softmax*. A cada *frame* a LSTM recebe informações da lsCNN (da face atual) bem como de seu estado anterior a fim de gerar uma saída para a camada *softmax*. O estado da LSTM atual alimenta a análise futura do próximo *frame*. Fonte: Elaborada pelo autor.

de Tran et al. (2015), a C3D se mostrou melhor para capturar informações temporais, até devido à duração relativamente limitada dos vídeos das bases da literatura. Como observação, as iterações de convergência entre os modelos CNN+LSTM tradicionalmente treinados e as redes lsLSTM foram bastante próximas neste caso. Vale notar ainda que, ao aumentar muito o tamanho das sequências de *frames* (64 quadros), a rede LSTM no topo da arquitetura lsLSTM acaba perdendo sua capacidade de armazenar informações passadas e de generalização, deteriorando bastante os resultados.

12.4 Conclusões

Embora os métodos de detecção e reconhecimento de facial considerem as diferentes regiões da face humana para tais tarefas, nenhuma técnica até então havia usado características profundas locais de *spoofing* para a detecção de ataques, como proposto. Resultados experimentais mostram um aumento na acurácia da CNN (na maioria dos casos), além de melhor eficiência no treinamento, quando inicializada com base em uma etapa local de pré-treinamento (nas principais regiões faciais). A lsCNN obteve resultados compatíveis com o estado-da-arte nos conjuntos de dados avaliados com um modelo bastante compacto, sendo também muito mais eficiente do que as CNNs tradicionais da literatura, como a VGG-Face (PARKHI; VEDALDI; ZISSERMAN, 2015), que é bastante usada na detecção de *spoofing* facial. Informações temporais também melhoraram (no caso da lsC3D trabalhando com 32 *frames*) os resultados da lsCNN, porém o uso de sequências de quadros muito longas não trouxe benefícios.

A abordagem de treinamento proposta baseada em características locais pode ainda ser aplicada para o treinamento de outros modelos de CNN, incluindo arquiteturas maiores, a fim de melhorar ainda mais seus desempenhos na detecção de *spoofing*, bem como suas eficiências durante o treinamento.

Capítulo 13

CONCLUSÃO

Pelo presente trabalho pôde-se observar que os ataques de *spoofing* facial, ou ainda ataques de apresentação facial, são tema atual e importante tanto para a comunidade científica quanto para a industrial. Sem métodos de contramedida os sistemas de reconhecimento facial, em geral, podem ser burlados, gerando sérios prejuízos às instituições e às pessoas.

A literatura apresenta uma série de métodos para detecção de *spoofing* facial, com taxas de acurácia cada vez mais elevadas, com destaque para os métodos basedos em Aprendizado em Profundidade, que vêm ganhando espaço em tal aplicação dado o bom desempenho das redes neurais multicamadas. Entretanto, conforme apresentou-se, tais métodos se valem de arquiteturas de redes neurais bastante profundas e complexas para conseguir tais taxas de acerto. Neste sentido, os trabalhos baseados em Aprendizado de Máquina em Profundidade, por não considerarem o quesito eficiência na avaliação de desempenho, na maioria das vezes, sequer podem ser avaliados em ambiente computacional com *hardware* mais restrito, quanto mais em plataformas e dispositivos móveis, tendência nos dias atuais.

Com base em tudo isto, neste trabalho investigou-se métodos de Aprendizado em Profundidade capazes de obter taxas de acerto compatíveis com o estado-da-arte, mas garantindo também eficiência computacional e, conforme mostrado nesta tese, tais abordagens são possíveis. Mesmo utilizando abordagens multicamadas, quer seja baseadas em RBMs ou CNNs, técnicas de otimização de tais arquiteturas, seja na disposição das camadas e neurônios ou nos algoritmos de treinamento, podem ser empregadas de forma a obter soluções multicamadas eficientes e acuradas na detecção de *spoofing* facial.

Os capítulos que apresentam arquiteturas baseadas nas RBMs para detecção de *spoofing* facial, como a RBM discriminativa e a DDRBM, proposta nesta tese, mostram que pode-se tornar tais redes, por natureza generativas, em bons e eficientes classificadores, podendo ser treinados

e avaliados rapidamente sobre CPUs (não necessitando de processamento paralelo em placas gráficas). Mostrou-se também neste trabalho que as RBMs podem ainda ser transformadas em bons extratores de características e redutores de dimensionalidade, capazes de aprender as melhores características de amostras a elas apresentadas, e associadas a classificadores SVM para a detecção de *spoofing* facial, por exemplo, atuando de forma mais acurada que outras técnicas tradicionais e bem referenciadas como a PCA (*Principal Component Analysis*).

Pelos experimentos realizados, pôde-se constatar também que, apesar dos bons resultados das redes basedas em RBMs na detecção de *spoofing* facial, as CNNs ainda tendem a apresentar acurácias mais elevadas, dadas suas operações de convolução e amostragem que naturalmente tratam sinais 2D (caso das imagens faciais), apesar de serem, em geral, mais custosas (exigindo GPUs para treinamento e teste). Neste sentido, são propostas nesta tese, redes neurais convolucionais eficientes baseadas em textura, a LBPnet e a n-LBPnet, mostrando-se que características convolucionais de textura (mesmo quando se empregando CNNs compactas para extraí-las) permitem melhores resultados que características de textura *handcrafted* na detecção de *spoofing* facial. A abordagem de Transferência de Aprendizado proposta mostra também que pode-se reutilizar as características aprendidas por arquiteturas complexas (em problemas de domínios similares) de forma a não ser necessário retreinar os modelos e assim empregá-los diretamente no problema sendo tratado a mais baixo custo computacional. A expansão da arquitetura de uma CNN compacta em largura ao invés de profundidade (wCNN), proposta nesta tese, também mostrou reduzir o esforço computacional necessário para o processamento das faces e preservar resultados estado-da-arte na detecção de faces sintéticas.

Especial destaque merece a abordagem proposta baseada no uso de características locais profundas da face no treinamento de CNNs (apresentada no Capítulo 12), a qual mostra que tais informações são importantes no contexto de detecção de ataques de apresentação facial e podem, além de permitir obter-se os melhores resultados dentre as abordagens propostas, melhores até que o estado-da-arte em alguns casos, acelerar a convergência dos modelos em seus treinamentos, tornando-os mais eficientes. A rede lsCNN proposta, mesmo com poucas camadas convolucionais, mostrou-se mais acurada e eficiente inclusive que a MobileNet, rede proposta para ambientes de *hardware* restrito, também avaliada neste trabalho pela primeira vez no contexto de detecção de *spoofing* facial. O uso de informações temporais, isto é, a combinação da lsCNN e da C3D, trabalhando-se com blocos de 32 *frames* por vez dos vídeos, melhorou ainda mais a acurácia do modelo baseado em *patches* locais proposto na base Replay-Attack, permitindo-se obter resultados quase ótimos.

Com base em tudo isto, pode-se concluir que é possível projetar arquiteturas de Apren-

dizado de Máquina em Profundidade acuradas e também eficientes, como as apresentadas, valendo-se de técnicas de otimização, disponibilizando-as para uso em arquiteturas computacionais menos robustas. Além disto, a eficiência traz o ganho de tempo, tornando o processo de detecção de ataques mais rápido e adequado a sistemas de tempo real, não retardando a resposta do reconhecimento aos indivíduos sob identificação, e melhorando a experiência dos usuários.

13.1 Contribuições da Tese

Em suma, dentre as principais contribuições desta tese pode-se destacar:

i) Avaliação das redes baseadas nas Máquinas de Boltzmann Restritas (RBM - *Restricted Boltzmann Machines*) no contexto da deteção de *spoofing* facial, estudos até então não realizados na literatura;

ii) Demonstração do melhor desempenho das RBMs na representação e redução de dimensionalidade de vetores de características que técnicas consolidadas em tal tarefa, como a PCA (*Principal Component Analysis*), para a detecção de *spoofing* facial;

iii) Proposta de arquiteturas discriminativas baseadas em RBMs, como a DDRBM (*Deep Discriminative Restricted Boltzmann Machines*), para a detecção de *spoofing* facial, demonstrando a vantagem do uso de arquiteturas profundas;

iv) Avaliação de Redes Neurais de Convolução (CNN - *Convolutional Neural Networks*) compactas na detecção de *spoofing* facial, como a LBPnet e a n-LBPnet, bem como da eficiente rede neural MobileNet-v1, até então não avaliada neste contexto;

v) Demonstração da superioridade de características convolucionais de textura baseadas no LBP (*Local Binary Pattern*) em relação a características de textura LBP *handcrafted* e a características de textura (também baseadas no LBP) extraídas pelas redes de RBMs, constatando a superioridade das CNNs em relação às RBMs na detecção de *spoofing* facial 2D;

vi) Proposta do uso da Transferência de Aprendizado "off the shelf" para detecção de *spoofing* facial a partir de CNN pré-treinada em domínio similar (a VGG-Face), tornando possível trabalhar com tal CNN profunda em ambiente com restrições de *hardware*;

vii) Proposta de otimizações na arquiteturas das CNNs, como sua expansão em largura, a fim de reduzir o custo computacional do processamento das imagens faciais mantendo resultados compatíveis com o estado-da-arte;

viii) Proposta de um novo algoritmo para treinamento de CNNs baseado em informações profundas locais das faces para a detecção de *spoofing* facial, acelerando a convergência das redes neurais bem como melhorando seus resultados;

ix) Avaliação de redes neurais temporais na tarefa de deteção de *spoofing* sobre a arquitetura de CNN proposta com algoritmo de treinamento em duas etapas (atualizando pesos de conexões temporais também com base em informações locais);

x) Demonstração de que a técnica de otimização baseada no novo algoritmo de treinamento proposto, que se vale de informações locais, permite obter melhores resultados do que as demais técnicas de otimização para CNNs propostas nesta tese (melhorando ainda mais seu desempenho ao agregar-se informações temporais - rede neural C3D).

13.2 Trabalhos Futuros

Além de avaliar o algoritmo de treinamento proposto no Capítulo 12 em outras arquiteturas de CNN, incluindo a GoogleNet (SZEGEDY et al., 2014), a Inception-v3 (SZEGEDY et al., 2015) e a ResNet (HE et al., 2015), pretende-se trabalhar com imagens com mais de 3 canais (combinação de diferentes espaços de cores em diferentes bandas) como entrada para as redes neurais profundas (após conseguir maior poder computacional - novas placas gráficas mais potentes). Representações de textura das imagens faciais também podem ser usadas para compor as bandas das imagens de entrada das redes profundas.

Pretende-se ainda avaliar as abordagens propostas em novas bases de imagens para detecção de *spoofing* facial, recentemente propostas na literatura, como a SiW (*Spoof in the Wild*) (LIU; JOURABLOO; LIU, 2018), bem como trabalhar com características profundas baseadas na versão do LBP para imagens coloridas.

13.3 Publicações

Durante o doutorado foram publicados os seguintes artigos, relacionados a esta tese:

1) **"Deep Texture Features for Robust Face Spoofing Detection"**, G. B. Souza, D. F. S. Santos, R. G. Pires, A. N. Marana e J. P. Papa, Anais do *IEEE International Symposium on Circuits and Systems* (ISCAS), p. 1–2, 2017 (*abstract*) - **qualis A1**;

2) **"Deep Texture Features for Robust Face Spoofing Detection"**, G. B. Souza, D. F. S. Santos, R. G. Pires, A. N. Marana e J. P. Papa, *IEEE Transactions on Circuits and Systems II: Express Briefs*, v. 64, n. 12, p. 1397–1401, 2017 - **qualis A2**;

3) **"Efficient Transfer Learning for Robust Face Spoofing Detection"**, G. B. Souza, D. F. S. Santos, R. G. Pires, A. N. Marana e J. P. Papa, Anais do *Iberoamerican Congress on Pattern Recognition* (CIARP), p. 643–651, 2017 - **qualis B1**;

4) **"Detecção de Spoofing Facial: Uma abordagem baseada nas Máquinas de Boltzmann Restritas"**, G. B. Souza, A. N. Marana e J. P. Papa, *Revista Eletrônica Paulista de Matemática*, v. 10, p. 158-166, 2017 - **qualis B4** (área Interdisciplinar);

5) **"A Restricted Boltzmann Machine-Based Approach for Robust Dimensionality Reduction"**, G. B. Souza, D. F. S. Santos, R. G. Pires, A. N. Marana e J. P. Papa, Anais do *Workshop de Visão Computacional* (WVC), p. 138–143, 2017 - **qualis B5**;

6) **"Detecção de Ataques a Sistemas de Reconhecimento Facial: uma Abordagem Baseada nas Máquinas de Boltzmann Restritas"**, G. B. Souza, A. N. Marana e J. P. Papa, Anais do *Encontro Regional de Matemática Aplicada e Computacional* (ERMAC), 2017;

7) **"On the Learning of Deep Local Features for Robust Face Spoofing Detection"**, G. B. Souza, J. P. Papa e A. N. Marana, Anais da *Conference on Graphics, Patterns and Images* (SIBGRAPI), 2018 - **qualis B1**;

8) **"Efficient Width-Extended Convolutional Neural Network for Robust Face Spoofing Detection"**, G. B. Souza, D. F. S. Santos, R. G. Pires, J. P. Papa e A. N. Marana, Anais da *Brazilian Conference on Intelligent Systems* (BRACIS), p. 230–235, 2018 - **qualis B2**;

9) **"Deep Discriminative Restricted Boltzmann Machine (DDRBM) for Robust Face Spoofing Detection"**, G. B. Souza, J. P. Papa e A. N. Marana, *Progress in Human-Computer Interaction*, v. 1, n. 3, p. 1–8, 2018;

10) **"On the Importance of Deep Local Features for Face Presentation Attack Detection"**, G. B. Souza, J. P. Papa e A. N. Marana, (em submissão);

11) **"Deep Boltzmann Machines for Robust Fingerprint Spoofing Attack Detection"**, G. B. Souza, D. F. S. Santos, R. G. Pires, A. N. Marana e J. P. Papa, Anais da *International Joint Conference on Neural Networks* (IJCNN), p. 1863–1870, 2017 - **qualis A1**;

12) **"Deep Features Extraction for Robust Fingerprint Spoofing Attack Detection"**, G. B. Souza, D. F. S. Santos, R. G. Pires, A. N. Marana e J. P. Papa, *Journal of Artificial Intelligence and Soft Computing Research*, v. 9, n. 1, p. 41–49, 2018;

Os artigos de 1 a 10 são os apresentados nos Capítulos de 6 a 12, versando sobre detecção de *spoofing* em sistemas de reconhecimento facial. Os artigos 11 e 12 são parte integrante dos resultados desta tese e tratam da detecção de *spoofing* facial em sistemas biométricos baseados em impressões digitais.

Também foram publicados os seguintes artigos, frutos de atividades extraordinárias realizadas durante o doutorado:

1) **"Shape Analysis Using Multiscale Hough Transform Statistics"**, L. A. Ramos, G. B. Souza e A. N. Marana, Anais do *Iberoamerican Congress on Pattern Recognition* (CIARP), p. 452–459, 2015 - **qualis B1**;

2) **"A Graph-Based Approach for Contextual Image Segmentation"**, G. B. Souza, G. M. Alves, A. L. M. Levada, P. E. Cruvinel e A. N. Marana, Anais da *Conference on Graphics, Patterns and Images* (SIBGRAPI), 2016 - **qualis B1**;

3) **"A New Approach for Plant Phenotyping and Image Segmentation Based on Contextual Information"**, G. M. Alves, P. E. Cruvinel, G. B. Souza, A . N. Marana e A. L. M. Levada, Anais da *II Latin-American Conference on Plant Phenotyping and Phenomics for Plant Breeding*, 2017;

4) **"A Deep Boltzmann Machine-Based Approach for Robust Image Denoising"**, R. G. Pires, D. F. S. Santos, G. B. Souza, A. N. Marana, A. L. M. Levada e J. P. Papa, Anais do *Iberoamerican Congress on Pattern Recognition* (CIARP), p. 525–533, 2017 - **qualis B1**;

5) **"A Robust Restricted Boltzmann Machine for Binary Image Denoising"**, R. G. Pires, D. F. S. Santos, L. A. M. Pereira, G. B. Souza, A. L. M. Levada e J. P. Papa, Anais da *Conference on Graphics, Patterns and Images* (SIBGRAPI), 2017 - **qualis B1**;

6) **"A 2D Deep Boltzmann Machine for Robust and Fast Vehicle Classification"**, D. F. S. Santos, G. B. Souza e A. N. Marana, Anais da *Conference on Graphics, Patterns and Images* (SIBGRAPI), 2017 - **qualis B1**;

7) **"Introduction to Deep Learning - Theory and Practice"**, G. B. Souza, J. P. Papa e A. N. Marana, Anais da *Brazilian Conference on Intelligent Systems* (BRACIS), p. 1–2, 2018 (*abstract*) - **qualis B2**;

8) **"Cross-Domain Deep Face Matching for Real Banking Security Systems"**, J. S. Oliveira, G. B. Souza, A. R. Rocha, F. E. Deus e A. N. Marana, (em submissão).

O artigo 1 se refere a uma extensão do trabalho do mestrado. Os artigos 2 e 3 tratam-se da segmentação de imagens baseada em grafos e foram confeccionados como parte dos trabalho de disciplina do doutorado e publicados em congressos. Os artigos de 4 a 6 versam de trabalhos da aplicação de Aprendizado de Máquina em Profundidade em outros problemas que não detecção de *spoofing*. O artigo 7 trata-se de um tutorial selecionado para apresentação e ministrado aos conferencistas da *Brazilian Conference on Intelligent System* (BRACIS), realizada na IBM *Research*, em São Paulo/SP, em 2018. O último artigo versa sobre o uso de Aprendizado de Máquina em Profundidade, mas para o Reconhecimento Facial.

13.4 Premiações

Em relação às premiações e honrarias recebidas decorrentes dos trabalhos do doutorado até o momento, pode-se citar:

2016

- **2º lugar** - Concurso de Reconhecimento Facial (utilizando Aprendizado de Máquina em Profundidade) na *International Summer School for Advanced Studies on Biometrics for Secure Authentication*, realizada pela Universidade de Sassari e pela *International Association of Pattern Recognition* (IAPR), em Algueiro (Itália);

- **Melhor Trabalho** - I *Workshop* do Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos (UFSCar);

2017

- ***IEEE CIS Student Travel Grant Award*** - Pelo artigo apresentado na *International Joint Conference on Neural Networks* de 2017, honraria conferida pela *IEEE Computational Intelligence Society*;

2018

- **Tutorial "*Deep Learning - Theory and Practice*"** - Selecionado entre as duas melhores propostas e apresentado na *Brazilian Conference on Intelligent Systems* (BRACIS) 2018.

REFERÊNCIAS

ACKLEY, D. H.; HINTON, G. E.; SEJNOWSKI, T. J. A learning algorithm for Boltzmann Machines. *Cognitive Science*, v. 9, p. 147–169, 1985.

AKHTAR, Z.; FORESTI, G. L. Face spoof attack recognition using discriminative image patches. *Journal of Electrical and Computer Engineering*, v. 16, 2016.

ALECRIM, E. *Inteligência artificial do Google pode fazer buscas por objetos em vídeos*. 2016. Disponível em: <https://tecnoblog.net/210439/google-cloud-video-intelligence/>.

ALOTAIBI, A.; MAHMOOD, A. Deep face liveness detection based on nonlinear diffusion using convolution neural network. *Signal, Image and Video Processing*, Springer, v. 11, n. 4, p. 713–720, 2017.

ANJOS, A.; MARCEL, S. Counter-measures to photo attacks in face recognition: a public database and a baseline. In: *Anais da International Joint Conference on Biometrics*. Estados Unidos: IEEE, 2011. p. 1–7.

ARASHLOO, S. R.; KITTLER, J.; CHRISTMAS, W. Face spoofing detection based on multiple descriptor fusion using multiscale dynamic binarized statistical image features. *IEEE Transactions on Information Forensics and Security*, v. 10, n. 11, p. 2396–2407, 2015.

ATOUM, Y. et al. Face anti-spoofing using patch and depth-based CNNs. In: *Anais da International Joint Conference on Biometrics*. Estados Unidos: IEEE, 2017. p. 319–328.

BA, J. et al. Using fast weights to attend to the recent past. *CoRR*, abs/1610.06258, 2016. Disponível em: <http://arxiv.org/abs/1610.06258>.

BAO, W. et al. A liveness detection method for face recognition based on optical flow field. In: *Anais da International Conference on Image Analysis and Signal Processing*. Estados Unidos: IEEE, 2009. p. 233–236.

BAUM, L. E.; EGON, J. A. An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology. *Bulletin of the American Mathematical Society*, v. 73, p. 360–363, 1967.

BAUM, L. E.; PETRIE, T. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, v. 37, n. 6, p. 1554–1563, 1966.

BAY, H.; TUYTELAARS, T.; GOOL, L. SURF: Speeded up robust features. In: *Anais da European Conference on Computer Vision*. [S.l.: s.n.], 2006. p. 404–417.

BBC. *Facebook desenvolve ferramenta que pode te reconhecer mesmo sem mostrar o rosto*. 2015. Disponível em: <https://www.bbc.com/portuguese/noticias/2015/06/150622_facebook_rosto_mdb>.

BENGIO, Y. Deep learning of representations: Looking forward. *Statistical Language and Speech Processing*, v. 7978, p. 1–37, 2013.

BERGSTRA, J.; YAMINS, D.; COX, D. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In: *Anais da International Conference on Machine Learning*. Estados Unidos: JMLR, 2013.

BOULKENAFET, Z.; KOMULAINEN, J.; HADID, A. Face anti-spoofing based on color texture analysis. In: *Anais da International Conference on Image Processing*. [S.l.]: IEEE, 2015. p. 2636–2640.

BOULKENAFET, Z.; KOMULAINEN, J.; HADID, A. Face antispoofing using speeded-up robust features and Fisher vector encoding. *IEEE Signal Processing Letters*, v. 24, n. 2, p. 141–145, 2017.

BOULKENAFET, Z.; KOMULAINEN, J.; HADID, A. On the generalization of color texture-based face anti-spoofing. *Image and Vision Computing*, v. 77, p. 1 – 9, 2018.

BRADSKI, G. The OpenCV library. *Journal of Software Tools*, 2000.

BREITE, S. *Authentication by selfie - Will Mastercard bring a smile to the payments world? How secure is it and how has the market responded?* 2016. Disponível em: <https://www.globalbankingandfinance.com/authentication-by-selfie-will-mastercard-bring-a-smile-to-the-payments-world-how-secure-is-it-and-how-has-the-market-responded/>.

CANZIANI, A.; PASZKE, A.; CULURCIELLO, E. An analysis of deep neural network models for practical applications. *CoRR*, abs/1605.07678, 2016. Disponível em: <http://arxiv.org/abs/1605.07678>.

CHANG, C.; LIN, C. LIBSVM: A library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, v. 2, p. 1–27, 2011.

CHELLAPILLA, K.; PURI, S.; SIMARD, P. High performance convolutional neural networks for document processing. In: *Anais da International Workshop on Frontiers in Handwriting Recognition*. França: CCSD, 2006.

CHEN, W.; ER, M. J.; WU, S. Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, v. 36, n. 2, 2006.

CHIACHIA, G. et al. Learning person-specific representations from faces in the wild. *IEEE Transactions on Information Forensics and Security*, v. 9, n. 12, p. 2089–2099, 2014.

CHINGOVSKA, I.; ANJOS, A.; MARCEL, S. On the effectiveness of local binary patterns in face anti-spoofing. In: *Anais da International Conference of the Biometrics Special Interest Group*. Estados Unidos: IEEE, 2012. p. 1–7.

CHOI, J. Y.; PLATANIOTIS, K. N.; RO, Y. M. Using colour local binary pattern features for face recognition. In: *Anais da International Conference on Image Processing*. [S.l.]: IEEE, 2010. p. 4541–4544. ISSN 2381-8549.

CHUNG, M. G. et al. Automatic video segmentation based on spatio-temporal features. *Korea Telecom*, v. 1, n. 4, p. 4–14, 1999.

COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, v. 19, p. 297–301, 1965.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, 1995.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: *Anais da IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.]: IEEE, 2005. p. 886–893.

DAUGMAN, J. G. Theory of communication. *Journal of the Optical Society of America A*, v. 2, n. 7, p. 1160–1169, 1985.

DEB, D.; BEST-ROWDEN, L.; JAIN, A. K. Face recognition performance under aging. In: *Anais da IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.]: IEEE, 2017. p. 548–556.

DEB, D. et al. Face recognition: Primates in the wild. *CoRR*, abs/1804.08790, 2018. Disponível em: <http://arxiv.org/abs/1804.08790>.

DEMARTINI, M. *Mastercard lança aplicativo para você pagar compras com selfies*. 2016. Disponível em: <http://exame.abril.com.br/tecnologia/mastercard-lanca-aplicativo-para-voce-pagar-compras-com-selfies/>.

DENG, L.; DONG, Y. Deep learning: methods and applications. *Foundations and Trends in Signal Processing*, v. 7, n. 3–4, p. 197–387, 2014.

DUSENBERRY, M. *On eigenfaces: creating ghost-like images from a set of faces*. 2015. Disponível em: <https://mikedusenberry.com/on-eigenfaces>.

FECHNER, G. *Elements of Psychophysics*. Nova Iorque: Holt, Rinehart and Winston, 1966.

FISCHER, A.; IGEL, C. An introduction to Restricted Boltzmann Machines. In: *Anais da Iberoamerican Congress on Pattern Recognition*. Estados Unidos: Springer-Verlag, 2012. p. 14–36.

FISHER, R. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, v. 7, p. 179–188, 1936.

FOSSE, G.; BAPTISTA, P. P. *Pesquisa Febraban de Tecnologia Bancária 2018*. [S.l.], 2018.

GABOR, D. Theory of communication. *Journal of IEEE*, v. 93, p. 429–457, 1946.

GALBALLY, J.; FIERREZ, J.; GARCIA, J. O. Vulnerabilities in biometric systems: attacks and recent advances in liveness detection. *Database*, v. 1, n. 3, p. 1–8, 2007.

GALBALLY, J. et al. *Biometric Spoofing: A JRC Case Study in 3D Face Recognition*. [S.l.], 2014.

GEMAN, S.; GEMAN, D. Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 6, n. 6, p. 721–741, 1984.

GERS, F. A.; SCHRAUDOLPH, N. N.; SCHMIDHUBER, J. Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, v. 3, p. 115–143, 2003.

GRAVES, A. *Supervised Sequence Labelling with Recurrent Neural Networks*. Estados Unidos: Springer, 2012.

HAN, H.; OTTO, C.; JAIN, A. K. Age estimation from face images: Human vs. machine performance. In: *Anais da International Conference on Biometrics*. Estados Unidos: IEEE, 2013.

HAN, H. et al. Demographic estimation from face images: human vs. machine performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 37, n. 6, p. 1148–1161, 2015.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEIN, I. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 3, n. 6, p. 610–621, Nov 1973. ISSN 0018-9472.

HE, K. et al. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <http://arxiv.org/abs/1512.03385>.

HIGA, P. *O sistema de câmeras do governo chinês consegue encontrar uma pessoa em 7 minutos*. 2018. Disponível em: <https://tecnoblog.net/230001/sistema-cameras-vigilancia-china/>.

HINTON, G. E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, v. 14, n. 8, p. 1771–1800, 2002.

HINTON, G. E. A practical guide to training Restricted Boltzmann Machines. In: MONTAVON, G.; ORR, G. B.; MÜLLER, K. R. (Ed.). *Neural Networks: Tricks of the trade*. Estados Unidos: Springer, 2012.

HINTON, G. E.; OSINDERO, S.; TEH, Y. W. A fast learning algorithm for Deep Belief Nets. *Neural Computation*, v. 18, n. 7, p. 1527–1554, 2006.

HINTON, G. E.; SEJNOWSKI, T. J. Optimal perceptual inference. In: *Anais da IEEE Conference on Computer Vision and Pattern Recognition*. Estados Unidos: IEEE, 1983. p. 448–453.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, v. 9, n. 8, p. 1735–1780, 1997.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. In: *Anais da National Academy of Sciences of the USA*. [S.l.: s.n.], 1982. v. 79, n. 8, p. 2554–2558.

HOTELLING, H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, v. 24, p. 417–441, 1933.

HOWARD, A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. Disponível em: <http://arxiv.org/abs/1704.04861>.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. Disponível em: <http://arxiv.org/abs/1502.03167>.

ISO; IEC. *ISO/IEC 30107 - Presentation Attack Detection*. Estados Unidos, 2016.

JAIN, A.; FLYNN, P.; ROSS, A. *Handbook of Biometrics*. Estados Unidos: Springer, 2008.

JAIN, A. K.; BOLLE, R.; PANKANTI, S. *Biometrics - Personal Identification in Network Society*. Estados Unidos: Springer, 2006.

JAIN, A. K. et al. Biometrics: a grand challenge. In: *Anais da International Conference on Pattern Recognition*. Estados Unidos: IEEE, 2004. v. 2, p. 935–942.

JAIN, A. K.; ROSS, A.; PRABHAKAR, S. An introduction to biometric recognition. *Special Issue on Image- and Video-Based Biometrics*, v. 14, p. 4–20, 2004.

JAIN, A. K.; ROSS, A. A.; NANDAKUMAR, K. *Introduction to Biometrics*. Estados Unidos: Springer, 2011.

JIA, Y. et al. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, abs/1408.5093, 2014. Disponível em: <http://arxiv.org/abs/1408.5093>.

JOURABLOO, A.; LIU, Y.; LIU, X. Face de-spoofing: Anti-spoofing via noise modeling. In: *Anais da European Conference on Computer Vision*. Munich, Germany: [s.n.], 2018.

JUSTI, A. *PF e MP identificam funcionários suspeitos de fraude em portos do PR*. 2014. Disponível em: <http://g1.globo.com/pr/parana/noticia/2014/02/pf-e-mp-identificam-funcionarios-suspeitos-de-fraude-em-portos-do-pr.html>.

KANNALA, J.; RAHTU, E. BSIF: Binarized statistical image features. In: *Anais da International Conference on Pattern Recognition*. [S.l.: s.n.], 2012. p. 1363–1366.

KIM, W.; SUH, S.; HAN, J. Face liveness detection from a single image via diffusion speed model. *IEEE Transactions on Image Processing*, v. 24, n. 8, p. 2456–2465, 2015.

KINGMA, D.; BA, J. Adam: a method for stochastic optimization. In: *Anais da International Conference for Learning Representations*. [S.l.: s.n.], 2015.

KOLLREIDER, K.; FRONTHALER, H.; BIGUN, J. Non-intrusive liveness detection by face images. *Image and Vision Computing*, v. 27, n. 3, p. 233–244, 2009.

KRIZHEVSKY, A. *Learning Multiple Layers of Features from Tiny Images*. [S.l.], 2009.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, p. 1106–1114, 2012.

LAROCHELLE, H.; BENGIO, Y. Classification using discriminative Restricted Boltzmann Machines. In: *Anais da International Conference on Machine Learning*. [S.l.: s.n.], 2008. p. 536–543.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, p. 436–444, 2015.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Anais da IEEE*, v. 86, n. 11, p. 2278–2324, 1998.

LEITE, P. C.; PAES, C. *Caso dos dedos de silicone completa 4 meses e médicos ainda são ouvidos*. 2013. Disponível em: <http://exame.abril.com.br/tecnologia/mastercard-lanca-aplicativo-para-voce-pagar-compras-com-selfies/>.

LEWIS, M.; ELLIS, H. How we detect a face: a survey of psychological evidence. *International Journal of Imaging Systems and Technology - Facial Image Processing, Analysis, and Synthesis*, v. 13, n. 1, p. 3–7, 2003.

LI, J. Eye blink detection based on multiple Gabor response waves. In: *Anais da International Conference on Machine Learning and Cybernetics*. Estados Unidos: IEEE, 2008. p. 2852–2856.

LI, J. et al. Live face detection based on the analysis of Fourier spectra. *Biometric Technology for Human Identification*, v. 5404, p. 296–303, 2004.

LI, L. et al. An original face anti-spoofing approach using partial convolutional neural network. In: *Anais da International Conference on Image Processing Theory, Tools and Applications*. [S.l.: s.n.], 2016. p. 1–6.

LIU, S. et al. 3D mask face anti-spoofing with remote photoplethysmography. In: *Anais da European Conference on Computer Vision*. [S.l.: s.n.], 2016. p. 85–100.

LIU, Y.; JOURABLOO, A.; LIU, X. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In: *Anais da International Conference on Computer Vision and Pattern Recognition*. Estados Unidos: IEEE, 2018.

LOWE, D. Object recognition from local scale-invariant features. In: *Anais da International Conference on Computer Vision*. [S.l.: s.n.], 1999. p. 1150–1157.

LUCENA, O. et al. Transfer learning using convolutional neural networks for face anti-spoofing. In: *Anais da International Conference on Image Analysis and Recognition*. [S.l.: s.n.], 2017. p. 27–34.

MA, S.; BAI, L. A face detection algorithm based on Adaboost and new Haar-like feature. In: *Anais da International Conference on Software Engineering and Service Science*. Estados Unidos: IEEE, 2016.

MAATTA, J.; HADID, A.; PIETIKAINEN, M. Face spoofing detection from single images using micro-texture analysis. In: *Anais da International Joint Conference on Biometrics*. Estados Unidos: IEEE, 2011. p. 1–7.

MAATTA, J.; HADID, A.; PIETIKAINEN, M. Face spoofing detection from single images using texture and local shape analysis. *Biometrics (IET)*, v. 1, n. 1, p. 3–10, 2012.

MACKAY, D. J. C. *Information Theory, Inference and Learning Algorithms*. Reino Unido: Cambridge University Press, 2003.

MANJUNATH, B. S.; MA, W. Y. Texture features for browsing and retrieval of image data. *IEEE Transanctions on Pattern Analysis and Machine Intelligence*, v. 18, n. 8, p. 837–842, 1996.

MARCEL, S.; NIXON, M. S.; LI, S. Z. *Handbook of Biometric Anti-Spoofing: Trusted Biometrics under Spoofing Attacks*. Londres: Springer, 2014.

MATHIAS, M. et al. Face detection without bells and whistles. In: *Anais da European Conference on Computer Vision*. [S.l.: s.n.], 2014. p. 720–735.

MENOTTI, D. et al. Deep representations for iris, face, and fingerprint spoofing detection. *IEEE Transactions on Information Forensics and Security*, v. 10, n. 4, p. 864–879, 2015.

MITA, T.; KANEKO, T.; HORI, O. Joint Haar-like features for face detection. In: *Anais da IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2005.

MONTAVON, G.; ORR, G. B.; MÜLLER, K. R. *Neural networks: Tricks of the Trade*. 2. ed. Estados Unidos: Springer, 2012.

NAIR, V.; HINTON, G. E. Implicit mixtures of Restricted Boltzmann Machines. *Advances in Neural Information Processing Systems*, v. 21, p. 1145–1152, 2014.

NOWARA, E. M.; SABHARWAL, A.; VEERARAGHAVAN, A. Ppgsecure: Biometric presentation attack detection using photopletysmograms. In: *Anais da International Conference on Automatic Face and Gesture Recognition*. [S.l.]: IEEE, 2017. p. 56–62.

OJALA, T.; PIETIKAINEN, M.; HARWOOD, D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In: *Anais da International Conference on Pattern Recognition*. Estados Unidos: IEEE, 1994. v. 1, p. 582–585.

OJALA, T.; PIETIKAINEN, M.; HARWOOD, D. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, v. 29, n. 1, p. 51–59, 1996.

OJALA, T.; PIETIKAINEN, M.; MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 7, p. 971–987, 2002.

OJANSIVU, V.; HEIKKILA, J. Blur insensitive texture classification using Local Phase Quantization. In: *Anais da International Conference on Image and Signal Processing*. Estados Unidos: IEEE, 2008.

OLAH, C. *Understanding LSTM Networks*. 2015. Disponível em: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

PAN, G. et al. Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In: *Anais da International Conference on Computer Vision*. Estados Unidos: IEEE, 2007. p. 1–8.

PARKHI, O. M.; VEDALDI, A.; ZISSERMAN, A. Deep face recognition. In: *Anais da British Machine Vision Conference*. Swansea: [s.n.], 2015.

PARVEEN, S. et al. Face liveness detection using Dynamic Local Ternary Pattern (DLTP). *Computer*, v. 5, n. 2, 2016.

PATEL, K. et al. Live face video vs. spoof face video: Use of Moiré patterns to detect replay video attacks. In: *Anais da International Conference on Biometrics*. Estados Unidos: IEEE, 2015.

PATEL, K.; HAN, H.; JAIN, A. K. Cross-database face antispoofing with robust feature representation. In: *Anais da Chinese Conference on Biometric Recognition*. Chengdu: [s.n.], 2016. p. 611–619.

PATEL, K.; HAN, H.; JAIN, A. K. Secure face unlock: Spoof detection on smartphones. *IEEE Transactions on Information Forensics and Security*, v. 11, n. 10, p. 2268–2283, 2016.

PEARSON, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, v. 2, n. 6, p. 559–572, 1901.

PEIXOTO, B.; MICHELASSI, C.; ROCHA, A. Face liveness detection under bad illumination conditions. In: *Anais da International Conference on Image Processing*. Estados Unidos: IEEE, 2011. p. 3557–3560.

PEREIRA, T. F. *A Comparative Study of Countermeasures to Detect Spoofing Attacks in Face Authentication Systems*. Dissertao (Mestrado) — Universidade Estadual de Campinas, Campinas, 2013.

PEREIRA, T. F. et al. LBP-TOP based countermeasure against facial spoofing attacks. In: *Anais da Asian Conference on Computer Vision - Workshop on Computer Vision With Local Binary Pattern Variants*. Daejeon: [s.n.], 2012.

PEREIRA, T. F. et al. Can face anti-spoofing countermeasures work in a real world scenario? In: *Anais da International Conference on Biometrics*. [S.l.: s.n.], 2013.

PERRONNIN, F.; SANCHEZ, J.; MENSINK, T. Improving the Fisher kernel for large-scale image classification. In: *Anais da European Conference on Computer Vision*. [S.l.: s.n.], 2010. p. 143156.

PINTO, A. S. et al. Video-based face spoofing detection through visual rhythm analysis. In: *Anais da Conference on Graphics, Patterns and Images*. Ouro Preto: [s.n.], 2012. p. 221–228.

PINTO, N. et al. A high-throughput screening approach to discovering good forms of biologically-inspired visual representation. *PLoS Computational Biology*, v. 5, n. 11, p. 1–12, 2009.

PRABHU, R. *CNN Architectures - LeNet, AlexNet, VGG, GoogLeNet and ResNet*. 2018. Disponível em: <https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848>.

PURCELL, G.; STEWART, A. The face-detection effect. *Bulletin of Psychonomic Society*, v. 24, p. 118–120, 1986.

PYMNTS. *Apples Face ID Hacked By Vietnamese Researchers*. 2017. Disponível em: <https://www.pymnts.com/news/retail/2017/cyber-mondays-positive-ecommerce-data/>.

RATHA, N.; CONNELL, J.; BOLLE, R. An analysis of minutiae matching strength. In: *Anais da International Conference on Audio- and Video-Based Biometric Person Authentication*. Suécia: [s.n.], 2001. p. 223–228.

REN, S. et al. Face alignment at 3000 fps via regressing local binary features. In: *Anais da Conference on Computer Vision and Pattern Recognition*. [S.l.]: IEEE, 2014. p. 1685–1692.

RUMELHART, D. E.; HINTON, G.; WILLIANS, R. Learning internal representations by error propagation. In: RUMELHART, D. E.; MCCLELLAND, J. L. (Ed.). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge: MIT Press, 1986. v. 1, p. 318–362.

RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, v. 115, n. 3, p. 211–252, 2015.

SALAKHUTDINOV, R. *Software*. 2015. Disponível em: <http://www.cs.toronto.edu/ rsalakhu/code.html>.

SALAKHUTDINOV, R.; HINTON, G. E. Deep Boltzmann Machines. In: *Anais da International Conference on Artificial Intelligence and Statistics*. Estados Unidos: JMLR, 2009.

SALAKHUTDINOV, R.; HINTON, G. E. An efficient learning procedure for Deep Boltzmann Machines. *Neural Computation*, v. 24, n. 8, p. 1967–2006, 2012.

SANTOS, D. F. S.; SOUZA, G. B.; MARANA, A. N. A 2D Deep Boltzmann Machine for robust and fast vehicle classification. In: *Anais da Conference on Graphics, Patterns and Images*. Niterói: [s.n.], 2017. p. 155–162.

SCHMID, P. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, v. 656, p. 5–28, 2010.

SCHWARTZ, M. J. *Apple iPhone 6 Touch ID Hacked*. 2014. Disponível em: <https://www.bankinfosecurity.com/apple-iphone-6-touchid-hacked-a-7348>.

SCHWARTZ, W. R. et al. Human detection using Partial Least Squares analysis. In: *Anais da International Conference on Computer Vision*. [S.l.]: IEEE, 2009. p. 24–31.

SCHWARTZ, W. R.; ROCHA, A.; PEDRINI, H. Face spoofing detection through Partial Least Squares and Low-Level Descriptors. In: *International Joint Conference on Biometrics*. Estados Unidos: IEEE, 2011.

SCHWARTZ, W. R.; SILVA, R. D. da; PEDRINI, H. A novel feature descriptor based on the Shearlet transform. In: *Anais da International Conference on Image Processing*. [S.l.]: IEEE, 2011.

SILVA, M. V.; MARANA, A. N.; PAULINO, A. A. On the importance of using high resolution images, third level features and sequence of images for fingerprint spoof detection. In: *Anais da International Conference on Acoustics, Speech and Signal Processing*. Estados Unidos: IEEE, 2015. p. 1807–1811.

SIMARD, P.; STEINKRAUS, D.; PLATT, J. Best practices for Convolutional Neural Networks applied to visual document analysis. In: *Anais da International Conference on Document Analysis and Recognition*. Estados Unidos: IEEE, 2003. p. 958–963.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: *Anais da International Conference on Learning Representations*. [S.l.: s.n.], 2015.

SIROVICH, L.; KIRBY, M. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, v. 4, n. 3, p. 519–524, 1987.

SMOLENSKY, P. Information processing in dynamical systems: Foundations of harmony theory. In: RUMELHART, D. E.; MCCLELLAND, J. L. (Ed.). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge: MIT Press, 1986. v. 1, p. 194–281.

SOUZA, G. B. et al. Efficient transfer learning for robust face spoofing detection. In: *Anais do Iberoamerican Congress on Pattern Recognition*. [S.l.: s.n.], 2017.

STAUFFER, C.; GRIMSON, W. Adaptive background mixture models for real-time tracking. In: *Anais da Conference on Computer Vision and Pattern Recognition*. [S.l.]: IEEE, 1999.

SUN, Z.; SUN, L.; LI, Q. Investigation in spatial-temporal domain for face spoof detection. In: *Anais da International Conference on Acoustics, Speech and Signal Processing*. [S.l.]: IEEE, 2018. p. 1538–1542.

SZEGEDY, C. et al. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. Disponível em: <http://arxiv.org/abs/1409.4842>.

SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. Disponível em: <http://arxiv.org/abs/1512.00567>.

TAN, X. et al. Face liveness detection from a single image with sparse low rank bilinear discriminative model. In: *Anais da European Conference on Computer Vision*. Grécia: [s.n.], 2010. p. 504–517.

TAN, X.; TRIGGS, B. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, v. 19, n. 6, p. 1635–1650, 2010.

TANG, Y.; SALAKHUTDINOV, R.; HINTON, G. E. Robust Boltzmann Machines for recognition and denoising. In: *Anais da Conference on Computer Vision and Pattern Recognition*. Estados Unidos: IEEE, 2012.

TIRUNAGARI, S. et al. Detection of face spoofing using visual dynamics. *IEEE Transactions on Information Forensics and Security*, v. 10, n. 4, p. 762–777, 2015.

TRAN, D. et al. Learning spatiotemporal features with 3D convolutional networks. In: *Anais da International Conference on Computer Vision*. [S.l.]: IEEE, 2015.

TRONCI, R. et al. Fusion of multiple clues for photo-attack detection in face recognition systems. In: *Anais da International Joint Conference on Biometrics*. Estados Unidos: IEEE, 2011. p. 1–6.

TURK, M.; PENTLAND, A. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, MIT Press, Estados Unidos, v. 3, n. 1, p. 71–86, 1991.

UOL. *Reconhecimento facial do Facebook te ajuda a rastrear fotos perdidas por aí*. 2018. Disponível em: <https://noticias.uol.com.br/tecnologia/noticias/redacao/2018/02/27/reconhecimento-facial-do-facebook-ganha-novos-recursos.htm>.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: *Anais da Conference on Computer Vision and Pattern Recognition*. Estados Unidos: IEEE, 2001.

WANG, Z. et al. Face aging with identity-preserved conditional generative adversarial networks. In: *Anais da Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018.

WEN, D.; HAN, H.; JAIN, A. K. Face spoof detection with image distortion analysis. *IEEE Transactions on Information Forensic and Security*, v. 10, n. 4, p. 746–761, 2015.

WOLD, H. Partial Least Squares. *Encyclopedia of Statistical Sciences*, v. 6, p. 581–591, 1985.

XU, Z.; LI, S.; DENG, W. Learning temporal features using LSTM-CNN architecture for face anti-spoofing. In: *Anais da Asian Conference on Pattern Recognition*. Malásia: [s.n.], 2015.

YAN, J. et al. Face liveness detection by exploring multiple scenic clues. In: *Anais da International Conference on Control Automation Robotics Vision*. [S.l.: s.n.], 2012. p. 188–193.

YANG, J.; LEI, Z.; LI, S. Z. Learn Convolutional Neural Network for face anti-spoofing. *CoRR*, abs/1408.5601, 2014. Disponível em: <http://arxiv.org/abs/1408.5601>.

YANG, J. et al. Face liveness detection with component dependent descriptor. In: *Anais da International Conference on Biometrics*. [S.l.: s.n.], 2013.

YANG, S. *MobileNet*. 2018. Disponível em: <https://github.com/shicai/MobileNet-Caffe>.

ZHANG, K. et al. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, v. 23, n. 10, p. 1499–1503, 2016.

ZHANG, Z. et al. A face antispoofing database with diverse attacks. In: *Anais da International Conference on Biometrics*. Estados Unidos: IEEE, 2012.

ZHAO, G.; PIETIKAINEN, M. Dynamic texture recognition using Local Binary Patterns with an application to facial expressions. *IEEE Transanctions on Pattern Analysis and Machine Intelligence*, v. 29, n. 6, p. 915–928, 2007.