Detecção de Deepfakes: comparação de desempenho de modelos treinados com múltiplas bases de dados públicas

Eduardo Silva de Freitas
Departamento de Computação
CEFET-MG
Belo Horizonte, Brasil
cyber.eduardo@gmail.com

Rogério Gomes, Bruno Santos e Natália Batista
Departamento de Computação
CEFET-MG
Belo Horizonte, Brasil
{rogerio, bsantos, nataliabatista}@cefetmg.br

***Abstract*—Avanços na área de inteligência artificial têm permitido que conteúdos audiovisuais falsificados, conhecidos como *deepfakes*, sejam produzidos com alta qualidade visual. Esses vídeos, que apresentam pessoas agindo de forma supostamente real, podem representar uma ameaça à sociedade quando utilizados de forma maliciosa. Sendo assim, algoritmos de detecção se tornam necessários para que esse conteúdo possa ser detectado e moderado nos meios de difusão. Diversos modelos, propostos na literatura, são capazes de reconhecer características geracionais específicas das *deepfakes*. No entanto, falham quando submetidos à vídeos oriundos de técnicas de *Deepfake* posteriores a sua concepção. Sendo assim, este trabalho avalia técnicas de detecção, consideradas o estado da arte, como a arquitetura EfficientNet, e técnicas baseadas em redes neurais clássicas encontradas na literatura, como a ResNet-152. Para isso, modelos de detecção que implementam estas arquiteturas foram treinados por meio de diversas bases de amostras *deepfake* disponíveis publicamente. As bases selecionadas atendem a critérios baseados na qualidade visual, volume de vídeos e diversidade de métodos de produção empregados. Das soluções propostas, o modelo baseado na rede EfficientNet-B0 obteve as melhores métricas de teste quando avaliado por meio de *datasets* utilizados em competições, atingindo 80% de acurácia. Observou-se também que a estratégia de utilização de múltiplas bases de amostras foi a melhor abordagem para o problema, visto que os modelos treinados com somente uma base obtiveram um pior desempenho. A rede ResNet-152, treinada com múltiplas bases, apresentou bons resultados na tarefa de detecção de *deepfakes*, porém seu desempenho foi inferior ao alcançado pelo modelo baseado na arquitetura EfficientNet.**

***Index Terms*—Deepfake, Deep learning, Reconhecimento facial, Visão computacional, Redes convolucionais**

I. Introdução

A produção e disseminação de conteúdo audiovisual falsificado tem se popularizado nos últimos anos. Novas técnicas avançadas de inteligência artificial, como as redes adversariais generativas (*Generative Adversarial Networks* - GANs), têm contribuído para a criação, em alta qualidade, desse tipo de conteúdo, facilitando a sua disseminação nas redes sociais. Esse conteúdo é dito falsificado por ser criado a partir de uma superposição de imagens faciais de um indivíduo alvo aplicado à uma gravação de um outro indivíduo qualquer, simulando falas, expressões faciais e gestos que não foram de fato realizados pelo indivíduo alvo. Os resultados de diversas aplicações dessa técnica, conhecida como *Deepfake*[1] (do inglês, derivado da junção dos termos "*deep learning*" e "*fake*"), são vídeos que simulam celebridades, figuras políticas, ou qualquer pessoa que tenha sua imagem exposta na mídia, com a intenção clara de dar veracidade às informações veiculadas.

Fig. 1. Exemplos de aplicação da técnica de *Deepfake* em vídeos publicados no YouTube [1] [2]. As imagens à esquerda são capturas reais de artistas em cenas cinematográficas. As imagens à direita são *deepfakes*, resultantes de algoritmos treinados com amostras de imagens faciais de outro indivíduo para substituição da face na imagem original. Nestes exemplos, a ideia é substituir as faces dos artistas para reprodução de novas cenas.

Essa técnica, geralmente utilizada para fins de entretenimento, como mostrado na Figura 1, pode se transformar em uma ameaça à privacidade e à segurança nacional, quando aplicada como ferramenta de desinformação, bem como lesar a

[1]Neste trabalho, distinguem-se os termos *Deepfake*, com inicial maiúscula, e *deepfake*. A primeira refere-se a técnica de produção dos vídeos falsificados, enquanto o último refere-se ao vídeo resultante da técnica.

imagem de um indivíduo, principalmente devido à dificuldade de identificação de sua autenticidade. Há casos recentes, por exemplo, em que as *deepfakes* foram utilizadas como artifício para desestabilização política, como ocorrido nos Estados Unidos, no Gabão e na Malásia [3]. Estudos demográficos indicam, no entanto, que a maioria do conteúdo *deepfake* disponível na Internet está relacionado ao entretenimento adulto e que, em quase sua totalidade, o indivíduo alvo exposto no vídeo é do gênero feminino, sendo a maior parte formado por atrizes e musicistas [3].

Sendo assim, técnicas automáticas de detecção se tornam indispensáveis para o reconhecimento das *deepfakes*, evitando, dessa forma, que conteúdo danoso ou ilícito seja difundido. Este trabalho, portanto, tem como objetivos: avaliar o desempenho de redes neurais artificiais (RNA), consideradas estado da arte no tema, na detecção de vídeos alterados.

II. Trabalhos relacionados

A proliferação de *deepfakes* na Internet despertou o interesse das grandes corporações e dos pesquisadores na criação de técnicas computacionais que sejam capazes de identificá-las. Korshunov e Marcel [4] demonstram a vulnerabilidade de técnicas de reconhecimento facial baseadas em VGG [5] e Facenet [6]. A análise desses métodos é conduzida sobre uma base de amostras contendo 620 *deepfakes*, de produção autoral, geradas a partir de algoritmos de superposição facial, em vídeos oriundos da base de dados VidTIMIT [7]. A falha em distinguir imagens autênticas em ambos os modelos é evidenciada pelos altos valores de *false acceptance rate* (FAR)[2], sendo 85,62% para a VGG e 95,00% para a FaceNet.

Korshunov e Marcel [4] também estabelecem comparativos a partir do treinamento de outros sistemas de detecção, como uma *long short-term memory* (LSTM), que é treinada com uso de características audio-visuais focadas em identificar a sincronização labial em vídeos. Esta técnica obteve *equal error rate* (EER)[3] de 41,8%, indicando ser incapaz de identificar *deepfakes* adequadamente. Os demais modelos produzidos, treinados somente com características visuais, combinam as seguintes técnicas: *Principal Component Analysis* (PCA) [9], *Image Quality Assessment* (IQA) [10], *Support Vector Machine* (SVM) [11], e estratégias específicas da área de *presentation attack detection* [12]. Destes modelos, o classificador que combinava SVM e IQA apresentou o melhor desempenho, sendo capaz de detectar *deepfakes* com 8,97% de EER.

Nguyen et al. [13] apresentam uma extensa revisão dos modelos de detecção automática de *deepfakes* da literatura, com foco nos recursos técnicos utilizados, como classificadores, características extraídas das amostras, forma digital das amostras e *datasets* utilizados. Os autores identificam que um dos desafios para o desenvolvimento de novos modelos de detecção é a capacidade de criar métodos robustos, escaláveis, e com maior capacidade de generalização. Isto é fundamentado na observação de que cada modelo explora alguma fraqueza geracional oriunda dos algoritmos que produzem as *deepfakes*. Considerando que os vídeos gerados estão cada vez mais verossímeis, é necessário que os modelos de detecção possam sempre lidar com o que há de mais recente produzido por estes algoritmos.

Nguyen et al. [13] sugerem a criação de um *benchmark* que facilite o desenvolvimento das técnicas de detecção. Os trabalhos revisados utilizam somente fragmentos de *datasets*, levando em conta todo o acervo de *deepfakes* na literatura. A utilização de uma crescente base de amostras, contendo o que há de mais recente em relação às *deepfakes* produzidas, pode ser capaz de criar cenários mais realistas, impondo um desafio mais concreto para os modelos de detecção a serem desenvolvidos. Este fator é levado em consideração, na produção deste trabalho, para a seleção das múltiplas bases de amostras que serão adotadas para treinamento e avaliação dos modelos produzidos, de forma que seja simulado um contexto mais próximo ao estado da arte das *deepfakes*.

Algumas das pesquisas especializam-se na produção e disponibilização de *datasets* de larga escala contendo vídeos autênticos e *deepfakes*, de forma a auxiliar a pesquisa por modelos mais eficazes de detecção desse conteúdo. Exemplos desse tipo de trabalho são as bases de amostras Celeb-DF [14] e FaceForensics++ [15], que serão objetos de estudo neste trabalho. Destacam-se também na literatura os trabalhos que, além de fornecer novas amostras para o desenvolvimento dos modelos de detecção, estimulam a produção destes através de competições com incentivos monetários. São elas: a *Deepfake Detection Challenge* (DFDC)[4] [16], realizada pelo grupo *Facebook AI*, e a base *DeeperForensics* [17], utilizada no *DeeperForensics Challenge*[5] e organizada pela *European Conference on Computer Vision* (ECCV). Assim como as bases supracitadas, os *datasets* destas competições também serão utilizados neste trabalho.

A solução vencedora da competição *Deepfake Detection Challenge* [16], que ocorreu no primeiro semestre de 2020, foi de Selim Seferbekov [18], por meio de uma abordagem de classificação *frame* por *frame*. Para constituir as amostras de treinamento do seu classificador, o autor utiliza o detector facial *Multi-task Cascaded Convolutional Networks* (MTCNN) [19]. Esta ferramenta é capaz de produzir diversas imagens, contendo faces extraídas do *dataset* disponibilizado no desafio. Também são geradas amostras, caracterizadas por variações das faces extraídas, através de cortes aleatórios nas imagens ou na remoção de regiões específicas das faces, como olhos, nariz ou boca. O *encoder* utilizado no modelo de detecção é o EfficientNet [20], com sua variação B-7. As variações deste *encoder*, cuja nomenclatura varia de B-0 à B-7, são

[2]FAR: É a probabilidade que um sistema tem em validar, de forma incorreta, o acesso de um usuário em um sistema biométrico, indicando falsos positivos [8].

[3]EER: Valor que indica a proporção entre falsos positivos e falsos negativos de um sistema. Quanto menor o EER, maior a acurácia de um sistema biométrico [8].

[4]A base de amostras e o nome da competição compartilham a mesma sigla, DFDC, em diversas referências da literatura. Neste trabalho, a sigla será empregada como referência à base de amostras da competição *Deepfake Detection Challenge* (https://www.kaggle.com/c/deepfake-detection-challenge).

[5]https://competitions.codalab.org/competitions/25228

constituídas por redes neurais convolucionais (*convolutional neural network* - CNN) cuja profundidade, largura e dimensão de suas camadas são variadas, de forma composta, visando uma maior eficiência e acurácia. As variações B-3, B-4, B-5 e B-6, também foram utilizadas em outras submissões do autor, mas a B-7 obteve melhor resultado, com um *overall log loss* de 0,4279, na avaliação final da competição. O autor opta por um cálculo heurístico, a partir da proporção de *frames* falsificados em relação a quantidade de *frames* utilizados como entrada, para determinar se um vídeo é autêntico.

Os vencedores da competição *DeeperForensics Challenge*, concluída em dezembro de 2020, apresentaram uma solução baseada na abordagem adotada por Selim Seferbekov. Os autores Baoying Chen, Peiyu Zhuang e Sili Li [21], utilizaram o MTCNN para detectar a região facial em cada *frame* das amostras de vídeo. Diferentemente da competição *Deepfake Detection Challenge*, a utilização de amostras de *datasets* públicos era permitida nesta competição, além das presentes na base *DeeperForensics*, provida pelos organizadores da competição. Com o objetivo de aumentar a capacidade dos modelos produzidos de generalizar, novas amostras foram produzidas por meio de técnicas de visão computacional sobre as imagens originais, como variações no contraste, saturação e ruído. A solução implementa três modelos (EfficientNet-B0, EfficientNet-B1 e EfficientNet-B2) para classificação das amostras. A média das saídas dos três modelos é calculada para compor o *score* final. A solução proposta pelos autores venceu a competição com *loss* de 0,2674. O destaque dessa solução é a utilização de três modelos EfficientNet com a menor quantidade de parâmetros, obtendo um melhor desempenho em termos de eficiência e tempo de execução.

III. Metodologia

A metodologia adotada neste trabalho está segmentada em duas etapas, conforme apresentado na Figura 2. A primeira etapa, entitulada ”Obtenção de amostras”, descreve a seleção dos *datasets* de vídeos e *deepfakes*, que tem seu conteúdo submetido a uma extração de imagens faciais, gerando as amostras de treinamento. A segunda etapa, denominada ”Treinamento e classificação”, descreve a arquitetura e produção dos modelos de detecção, apresentando os critérios utilizados para classificação, além da definição das métricas adotadas para avaliação de desempenho.

A. Obtenção de amostras

A base de amostras utilizada neste trabalho é composta pela junção de *datasets* disponíveis publicamente para treinamento de modelos de detecção de *deepfakes*. A utilização desses *datasets* tem como princípio a construção de uma base de grande escala, constituída por amostras diversificadas, geradas por meio de diferentes gerações e tipos de algoritmos de *Deepfake*. Os *datasets* sumarizados com suas principais características são apresentados na Tabela I e foram selecionados pelos seguintes critérios: *a*) quantidade de vídeos *deepfakes* únicos, *i.e.*, oriundos de diferentes vídeos alvos, *b*) quantidade total de vídeos *c*) diversidade dos métodos utilizados para a produção das *deepfakes* e *d*) distinção das características visuais dos vídeos utilizados.

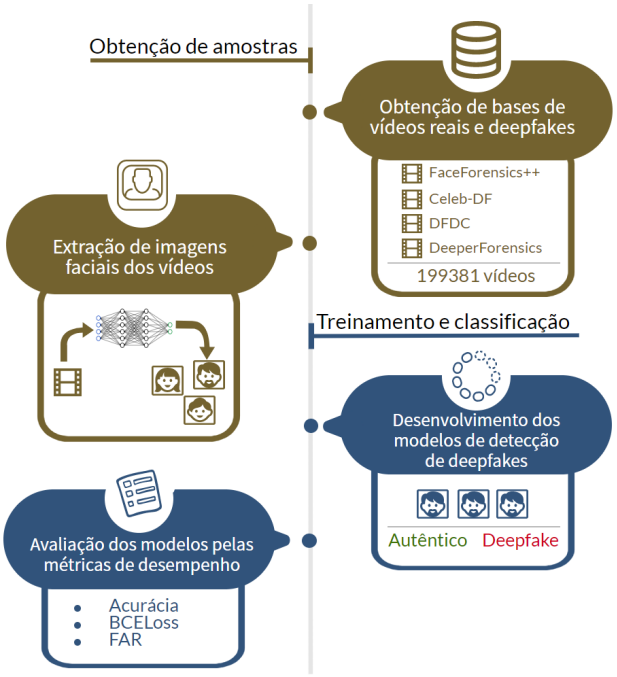


Fig. 2. Etapas da metodologia.

TABELA I
Datasets SELECIONADOS PARA TREINAMENTO DOS MODELOS.

Dataset	***Deepfakes***	**Total de vídeos**[a]	**Métodos**[b]
Celeb-DF [14]	5.639	6.229	1
FaceForensics++ [15]	4.000	5.000	4
DFDC [16]	104.500	128.154	8
DeeperForensics [17]	10.000	60.000	1

[a] Soma dos vídeos autênticos e das *deepfakes* presentes no *dataset*.
[b] Número de técnicas distintas de *Deepfake* usadas na criação da base.

Todos os *datasets* selecionados foram produzidos com diferentes técnicas de *Deepfake*. O FaceForensics++ e DFDC são produzidos por meio de múltiplas técnicas, incluíndo abordagens que utilizam redes neurais convolucionais, GANs e técnicas de computação gráfica. Na produção do *dataset* Celeb-DF, os autores optaram por ferramentas *open-source*, que são comumente utilizadas para produção de diversos vídeos que circulam atualmente na Internet. O método utilizado para produção do DeeperForensics, que é a base mais recente em relação à produção deste trabalho, é uma proposta autoral, que visa corrigir problemas que estão presentes em amostras dos demais *datasets*, como incompatibilidade facial e continuidade temporal, que afetam a qualidade visual. A diversidade de métodos empregados pelos trabalhos selecionados é um fator interessante para este trabalho, de forma que o modelo a ser proposto, quando treinado, possa generalizar amostras oriundas de diferentes técnicas de *Deepfake*.

No que se refere ao balanceamento das classes de amostras, sendo vídeos reais ou falsificados, os autores do DeeperForen-

sics indicam que sua base de amostras possui uma razão de cinco vídeos autênticos para cada vídeo falsificado. Os demais *datasets*, por sua vez, possuem um número elevado de *deepfakes* em relação ao total. Esta base se torna, então, o principal contribuinte de amostras autênticas para a composição da base utilizada neste trabalho. O *dataset* resultante, após coleção de todas as amostras de vídeos, é composto por 199.381 vídeos, sendo, aproximadamente, 38% autênticos e 62% *deepfakes*.

As amostras geradas a partir dos vídeos e utilizadas para treinamento dos modelos a serem propostos neste trabalho são imagens compostas pelo recorte de faces em sequências de *frames* dos vídeos, conforme mostrado na Figura 3. Essa abordagem é inspirada pela estratégia adotada nos trabalhos de Selim Seferbekov [18] e de Baoying Chen, Peiyu Zhuang e Sili Li [21], e se mostraram eficientes na geração dos segmentos de imagens a partir de vídeos.

De cada vídeo existente nas bases selecionadas são extraídas, no máximo, seis amostras. Esta limitação na quantidade de amostras por vídeo foi imposta pelo longo tempo de processamento da fase de extração devido ao elevado número de vídeos a serem processados. Cada vídeo é processado em conjuntos de cincos *frames* por iteração, produzindo pelo menos um recorte de face para cada *frame*. Para cada iteração, somente o recorte com maior probabilidade de apresentar uma face é adicionado ao *dataset* final, até que a quantidade máxima de amostras por vídeo seja atingida. Nenhuma métrica de qualidade foi utilizada para avaliar aspectos visuais da imagem e, portanto, é possível que sejam obtidos recortes de *frames* subsequentes com pouca variação, distorção, ou até mesmo sem uma face bem definida, como exemplificado na Figura 4. Contudo, a ferramenta de detecção facial foi configurada de forma a minimizar a probabilidade da ocorrência deste último tipo de falha.

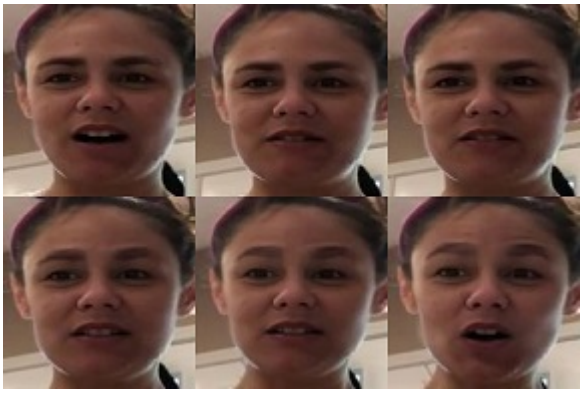


Fig. 3. Sequência de faces extraídas pelo MTCNN de uma amostra do DFDC.

A ferramenta utilizada para extração das imagens faciais dos vídeos é baseada no *Multi-task Cascaded Convolutional Networks* (MTCNN) de Zhang et al [19] e se encontra disponível na biblioteca pytorch[6]. A técnica MTCNN utiliza três redes convolucionais separadas para detecção facial. A primeira demarca uma caixa delimitadora em torno da face. A segunda refina as bordas dessa caixa e a última determina as fronteiras finais e os pontos de referência da face. A vantagem da implementação escolhida é a disponibilidade de modelos de reconhecimento facial pré-treinados. Um fator preponderante para escolha desta ferramenta, especificamente esta implementação em pytorch, é sua alta qualidade de detecção facial atrelada ao baixo tempo de processamento por vídeo. A ferramenta, neste trabalho, foi configurada com os seguintes parâmetros:

- *image_size=224*. A resolução da imagem produzida é de 224x224 *pixels*. Essa resolução é compatível com a entrada dos modelos a serem desenvolvidos;
- *selection_method='probability'*. O detector MTCNN priorizará a seleção de imagens que possuem a maior probabilidade de apresentar uma face. Outras configurações existentes de seleção envolvem somente a dimensão da face detectada. O fator importante, neste trabalho, é a existência da face na imagem, independente do tamanho;
- *keep_all=True*. Caso exista mais de uma face detectada em um mesmo *frame*, o MTCNN produzirá uma imagem para cada face. Dessa forma, as múltiplas faces presentes em um mesmo *frame* serão extraídas;
- *thresholds=[0.8, 0.9, 0.9]*. Este vetor de valores configura os limiares mínimos aceitáveis de probabilidade para cada uma das três fases de reconhecimento facial que compõe a técnica MTCNN. Os valores escolhidos neste trabalho são maiores que o padrão da ferramenta e foram escolhidos com o objetivo de reduzir o número de falsos positivos, *i.e.*, diminuir o número de imagens produzidas sem faces aparentes.

[6]Detector facial MTCNN, em pytorch, disponível no link https://github.com/timesler/facenet-pytorch

Fig. 4. Amostras extraídas de baixa qualidade. Os três recortes de imagens no topo apresentam distorção e recorte mal definido das faces. As três imagens abaixo, extraídas do mesmo vídeo, são recortes de *frames* subsequentes com pouca variação.

B. Treinamento e classificação

Os algoritmos selecionados para implementação neste trabalho foram escolhidos dentre os trabalhos encontrados na literatura e das técnicas propostas vencedoras de competições que abordam o tema de detecção de *deepfakes*. Foram selecionados a EfficientNet [20] e a ResNet-152 [22] para a

implementação de dois modelos distintos de detecção, no qual o primeiro se apresenta como o estado da arte e o segundo como uma abordagem clássica em problemas associados ao processamento de imagens.

A EfficientNet [20] é uma arquitetura, composta por redes neurais convolucionais, que alcança as melhores métricas de desempenho, quando comparada a técnicas similares, com uma quantidade inferior de parâmetros e um número menor de operações de ponto flutuante por segundo (*floating-point operations per second* - FLOPS). Ela é considerada, portanto, o estado da arte no que diz respeito a arquiteturas de redes neurais convolucionais (*Convolution Neural Networks* - CNNs). A EfficientNet se sobressai por possuir uma abordagem arquitetural que envolve a criação de CNNs cuja profundidade, largura e dimensão de suas camadas variam de forma composta, otimizando a acurácia e FLOPs de suas CNNs.

A partir de uma rede denominada EfficientNet B-0, tratada pelos autores como o modelo *baseline* da arquitetura EfficientNet, são produzidas variações de CNNs, rotuladas de B-1 até B-7, nas quais ocorrem o redimensionamento das camadas. Em cada variação, a profundidade da rede é alterada de forma proporcional com a largura e resolução de cada camada. A arquitetura da EfficientNet B-0 é segmentada em nove estágios, conforme especificações presentes na Tabela II. A arquitetura EfficientNet implementa camadas de convolução com a técnica *mobile inverted bottleneck* (referenciadas na literatura como MBConv), definida pela arquitetura MobileNetV2 de Sandler et al. [23]. Esta técnica implementa dois recursos: o afunilamento linear entre camadas (tradução livre do termo *linear bottleneck*) e conexões diretas entre as camadas de gargalo. O afunilamento linear surge do pressuposto que características não-lineares das camadas, como transformações não-lineares oriundas de funções de ativação (a *rectified linear activation function*, conhecida como ReLU, por exemplo), fazem com que a rede perca informações importantes. Logo, camadas *bottleneck* são inseridas para preservar e transmitir essa informação além destas transformações, mapeando os dados em um sub-espaço de menor dimensão. As conexões diretas entre essas camadas *bottleneck* surgem como atalhos para transmissão dessa informação, pressupondo que essas camadas contenham toda a informação pertinente ao modelo. Com isso, as conexões fornecem a capacidade de a rede acessar informações não modificadas pelos blocos de camadas de convolução.

De acordo com a literatura, a arquitetura ResNet-152 [22], também baseada em redes neurais convolucionais, tem apresentado bons resultados em tarefas de classificação de imagens em diferentes domínios de problemas, além de introduzir o conceito de conexão atalho (do inglês, *shortcut connection)*. Este tipo de conexão, realizado entre camadas não subsequentes, reduz significativamente o problema de dissipação do gradiente, em que os sinais de erro, propagados para as camadas iniciais, não são suficientes para o aprendizado da rede. Devido a esse avanço, redes mais profundas foram concebidas sem o comprometimento de sua aprendizagem,

TABELA II
ARQUITETURA DA REDE EFFICIENTNET B-0.

Estágio	Operações[a]	Resolução da entrada	Canais da saída	Camadas
1	Conv3x3	224 x 224	32	1
2	MBConv1, k3x3	112 x 112	16	1
3	MBConv6, k3x3	112 x 112	24	2
4	MBConv6, k5x5	56 x 56	40	2
5	MBConv6, k3x3	28 x 28	80	3
6	MBConv6, k5x5	14 x 14	112	3
7	MBConv6, k5x5	14 x 14	192	4
8	MBConv6, k3x3	7 x 7	320	1
9	Conv1x1 & *pooling* & FC	7 x 7	1280	1

[a] Nomenclatura das operações definidas conforme Tan e Le [20].

tais como a arquitetura de 152 camadas de profundidade especificada na Tabela III. Sendo assim, a ResNet-152, apesar da profundidade, é menos complexa que a arquitetura VGG [5], utilizada por Korshunov e Marcel [4], e será, portanto, utilizada neste trabalho, pela sua sua diversidade de aplicação em problemas de processamento de imagens.

TABELA III
ARQUITETURA DA REDE RESNET-152.

Estágio	Operações	Resolução da entrada	Canais da saída	Camadas
1	Conv 7x7[a]	224 x 224	64	1
2	Conv 3x3[a], *max pool*	112 x 112	64	1
3	Conv 1x1, 3x3, 1x1[b]	112 x 112	128	3
4	Conv 1x1, 3x3, 1x1[b]	56 x 56	256	8
5	Conv 1x1, 3x3, 1x1[b]	28 x 28	512	36
6	Conv 1x1, 3x3, 1x1[b]	14 x 14	1024	3
7	*Pooling* & FC	7 x 7	1024	1

[a] Convolução com 2 saltos (*stride*) no filtro.
[b] Bloco de convoluções sequenciais com filtros de dimensão variada.

O desenvolvimento prático deste trabalho foi realizado em Python, com auxílio de ferramentas e bibliotecas disponibilizadas para a linguagem. Para a implementação dos modelos propostos utilizou-se a biblioteca Keras [24], que fornece diversas ferramentas para o desenvolvimento de redes neurais profundas em Python e disponibiliza a implementação de diversas arquiteturas de redes convolucionais, conforme suas especificações da literatura, incluindo as redes EfficientNet e ResNet-152. Esta biblioteca também fornece redes pré-treinadas, oriundas da aprendizagem por transferência de outros domínios de problemas, *i.e.*, com os pesos das redes neurais já configurados. Contudo, como é de interesse deste trabalho obter novos modelos a partir do treinamento com diversas bases, as redes implementadas não utilizaram este recurso, e foram inicializadas com pesos aleatórios. Os modelos implementados seguiram as arquiteturas de referência das redes citadas, com a adição de uma nova camada na entrada das redes, para realização de alterações visuais no

input das imagens. Esta camada é responsável por realizar, sobre as imagens, operações aleatórias de rotação, translação e alteração de contraste. Desta forma, a variedade de dados é aumentada, impondo um maior desafio aos modelos na tarefa de generalização das amostras.

Para o treinamento dos modelos, as amostras de imagens foram particionadas em dados de treinamento, validação e teste. Os dados de teste utilizados são compostos, exclusivamente, pelas amostras oriundas dos vídeos utilizados nas competições *Deepfake Detection Challenge* e *DeeperForensics Challenge* para avaliação final das soluções dos competidores. A partição de dados em treinamento e validação foi feita de forma pseudo-aleatória utilizando estes dois datasets, excluindo-se os vídeos destinados para testes, além das amostras presentes nas demais bases de *deepfakes*. Procurou-se, no entanto, preservar o balanceamento de classes em cada partição resultante, formando uma distribuição de amostras com 80% para treinamento e 20% para validação.

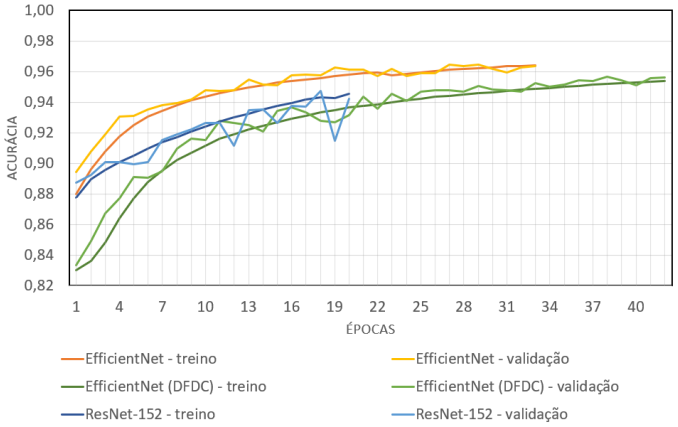
O número máximo de épocas de treinamento foi estabelecido em 50. Esta quantidade de épocas foi suficiente para convergência dos modelos, dado que o treinamento era interrompido devido à técnica de *early stopping* configurada para acompanhar a variação da acurácia de validação durante o treinamento. O *early stopping* foi configurado para interromper o processo caso a variação da acurácia de validação não sofresse aumento após quatro épocas seguidas.

As métricas utilizadas, para avaliação dos modelos de classificação durante treinamento e testes, são a acurácia, a entropia cruzada binária (do inglês, *binary cross entropy* - BCE, também conhecido como *logloss*), e a *false acceptance rate* (FAR). A métrica FAR mensura a quantidade de falsos positivos do classificador de acordo com a sensibilidade configurada, relacionando, assim, os erros e a sensibilidade do modelo.

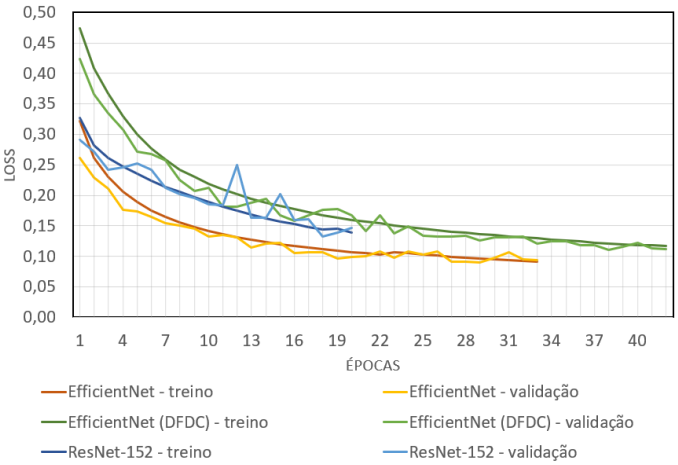
No processo de atualização dos pesos da rede optou-se pela utilização do otimizador Adam [25] com taxa de aprendizado constante e igual a 1×10^{-3}, em todas as épocas de treinamento.

IV. Resultados

Três modelos foram produzidos com os dados e as arquiteturas definidas. O primeiro modelo é a rede EfficientNet-B0 treinada com todas as amostras de imagens extraídas dos *datasets*. O segundo modelo implementa a mesma rede, porém treinado somente com amostras do *dataset* DFDC. Para distinguir esses dois modelos, o primeiro será nomeado conforme a sua arquitetura, EfficientNet, e o segundo será denominado EfficientNet-DFDC. O terceiro modelo é a rede ResNet-152, cujo treinamento utilizou todas as amostras produzidas dos *datasets* selecionados. Após treinamento, os três modelos foram submetidos à mesma rotina de teste com uso do *dataset* particionado para este fim. A acurácia e *loss* dos modelos, obtidas na fase de treinamento e validação, são apresentadas na Figura 5. As métricas obtidas nos testes, por sua vez, são mostradas na Tabela IV.



(a) Acurácia por época de treinamento.



(b) *Loss* por época de treinamento.

Fig. 5. Métricas obtidas durante treinamento dos modelos EfficientNet, EfficientNet-DFDC e ResNet-152.

É possível verificar na Figura 5a que a convergência da acurácia de treino e validação dos modelos atingiram valores superiores a 94%. Para os modelos EfficientNet e EfficientNet-DFDC, estes valores se estabilizaram após 30 épocas de treinamento. Em relação ao modelo ResNet-152, os valores de acurácia e de validação obtidos apresentam alta variação entre épocas quando comparado com os demais modelos, chegando a apresentar uma variação de aproximadamente 0,3 em relação

TABELA IV
PERFORMANCE DOS MODELOS DE DETECÇÃO DE *deepfakes* NA FASE DE TESTES SOBRE AMOSTRAS DE IMAGEM/VÍDEO DOS *datasets* DAS COMPETIÇÕES *Deepfake Detection Challenge* E *DeeperForensics Challenge*.

Modelo	**Acurácia**	***Loss***	***Loss* DFDC**[a]	***Loss* DF**[b]	**FAR (%)**
EfficientNet	0,808	0,591	0,843	0,082	0,235
EfficientNet DFDC [c]	0,708	0,903	0,911	0,359	0,200
ResNet-152	0,759	0,678	0,997	0,059	0,245

[a] Calculado na fase de testes com amostras da base DFDC.
[b] Calculado na fase de testes com amostras da base DeeperForensics.
[c] Modelo treinado somente com imagens de vídeos da base DFDC.

ao valor de treino, após a época 18. Esta alta variação mostra que o modelo estava tendendo ao *overfitting* do conjunto de dados. Sendo assim, e de forma a garantir a capacidade de generalização do modelo, os pesos utilizados no modelo ResNet-152, após sua fase de treinamento, foram os obtidos na época 18, durante a fase de treinamento. Os modelos baseados na arquitetura EfficientNet, por sua vez, mantiveram os pesos calculados na última época de treinamento. Na perspectiva da métrica de erro acumulado por época, apresentado na Figura 5b, os três modelos apresentaram efetiva capacidade de generalização, considerando a baixa variação, na ordem de 1×10^{-3}, nas últimas épocas treinadas e a não divergência da curva de erro de validação em relação às curvas de erro de treino.

As métricas obtidas na fase de testes dos modelos, apresentadas na Tabela IV, indicam a eficiência dos modelos de detecção de *deepfakes* obtidos neste trabalho quando submetidos às mesmas amostras utilizadas para classificação final das soluções propostas nas competições *Deepfake Detection Challenge* e *DeeperForensics Challenge*. De acordo com estas métricas, a rede EfficientNet, treinada com amostras de todos os *datasets* selecionados, é a que possui melhor desempenho entre os modelos propostos na identificação de conteúdo *deepfake*. A acurácia obtida por este modelo nos cenários de classificação das imagens individuais e de rótulos por vídeo, a partir do agrupamento das amostras de imagens, é superior aos demais modelos, além de apresentar menor erro. A *false acceptance rate* (FAR) calculadas para todos os modelos são similares, o que indica que todos possuem a mesma tendência em classificar erroneamente uma amostra como autêntica. Apesar da mesma tendência, a quantidade de falsos positivos indicados por esta métrica é considerada baixa. A acurácia de aproximadamente 80% na identificação de *deepfakes* apresentada pelo modelo EfficientNet e a consistência de suas métricas obtidas em treinamento, indicam que o modelo é satisfatório para identificação de vídeos *deepfakes*.

Em vista dos resultados obtidos, o modelo EfficientNet, que implementa a rede EfficientNet-B0 e treinado com amostras de diversos *datasets*, é considerado como a melhor solução proposta neste trabalho. Logo, verifica-se que a rede EfficientNet-B0, quando submetida a um treinamento com múltiplas bases de amostras, obtém uma maior capacidade de generalização. Isso pode ser atribuído a maior variabilidade de amostras existentes, que carregam diferentes informações inerentes às técnicas de *Deepfake* pela qual foram produzidas. A rede ResNet-152, quando comparada ao modelo EfficientNet-DFDC, reforça que a melhor estratégia para este problema é o treinamento com múltiplas bases de dados, considerando que esta arquitetura clássica obteve melhor desempenho quando comparada ao modelo que implementa o estado da arte, mas que foi treinado com apenas um *dataset*.

As competições supracitadas utilizaram a métrica de *log loss* para ranqueamento final das soluções dos competidores. Logo, foi realizado um comparativo, utilizando as mesmas bases de amostras utilizadas na avaliação final de cada competição, entre os modelos vencedores e os resultados obtidos nesse artigo. Na competição *DeeperForensics Challenge* (DF), a solução vencedora, proposta por Chen et al [21], obteve um *loss* de 0,2674 no *ranking* final. Os resultados obtidos nesse trabalho, por sua vez, mostraram que os modelos EfficientNet e ResNet-152 foram mais eficientes que a solução vencedora, apresentando, respectivamente, um *loss* de 0,082 e 0,059 (Tabela IV). Contudo, ao se comparar os resultados obtidos nesse trabalho, quando se utilizou a base da competição *Deepfake Detection Challenge* (DFDC), com a solução vencedora da competição proposta por Selim Seferbekov [18], que obteve um *loss* de 0,4279, observou-se que os modelos EfficientNet e ResNet-152 não apresentaram um bom desempenho, alcançando um *loss* de 0,843 e 0,997, respectivamente. Esta diferença de desempenho nos testes com os *datasets* de cada competição pode ser atribuída a diferentes características presentes nos vídeos que compõem estas bases. Na partição de testes da base DFDC, as amostras apresentam imagens ruidosas, com alta variação de contraste e foco, além de alguns vídeos apresentarem, de forma intermitente, recortes de faces extraídos de outros vídeos. É possível que o nível de alteração visual aplicado nas amostras de treinamento neste trabalho não tenha sido suficiente para que o modelo produzido seja capaz de distinguir imagens autênticas e *deepfakes* em vídeos com transformações visuais severas. Estes resultados indicam a sensibilidade do desempenho dos modelos de detecção à diferentes aplicações, reforçando-se a necessidade de produção de modelos ajustados para cada aplicação. Com isso, corrobora-se com a ideia proposta em Nguyen et al. [13] para composição de um *benchmark* que facilite o desenvolvimento e comparação de diferentes abordagens de detecção de *deepfakes*.

V. Considerações finais

Esse trabalho propôs a utilização de arquiteturas de redes convolucionais, consideradas como o estado da arte, bem como de arquiteturas clássicas encontradas na literatura, na detecção de conteúdo audiovisual falsificado ou *deepfakes*. Além disso, esse trabalho procurou mostrar que, além das características arquiteturais das redes serem relevantes para resolução desse tipo de problema, a utilização de *datasets* variados, produzidos por diversas técnicas de *Deepfake* distintas, pode contribuir na obtenção de modelos com maior capacidade de generalização. O desempenho dos modelos foi avaliado utilizando as métricas de acurácia, *loss* e FAR, após serem treinados com amostras de imagens faciais extraídas dos vídeos autênticos e *deepfakes* disponíveis nesses *datasets*.

No desenvolvimento desse trabalho foi levado em consideração o tempo de processamento dos algoritmos implementados, bem como sua capacidade computacional. Na etapa de produção de amostras, a quantidade de imagens faciais produzidas por vídeo foi reduzida de forma que todos os vídeos disponíveis nos *datasets* pudessem ser processados dentro de uma janela de tempo adequada. Em relação às arquiteturas definidas, o fator temporal também foi fundamental na determinação das redes a serem utilizadas no trabalho. Dessa forma, redes de menor complexidade e que

apresentavam bom desempenho em domínios de problemas similares em processamento de imagens se apresentaram como sendo as melhores opções.

Levando-se em consideração estes aspectos e os resultados encontrados, pode-se concluir que as redes produzidas foram capazes de realizar a detecção de *deepfakes* de forma eficiente. Da mesma forma, foi possível verificar que a rede EfficientNet mostrou-se, assim como nos trabalhos recentes de Selim Seferbekov [18] e Baoying Chen, Peiyu Zhuang e Sili Li [21], ser uma excelente opção arquitetural para este domínio de problema. Outra importante conclusão, foi constatar que o treinamento das redes com múltiplas bases de amostras de vídeos autênticos e *deepfakes* foi uma excelente estratégia para esta aplicação, por promover um aumento na capacidade dos modelos em detectar vídeos falsificados produzidos por distintas técnicas de *Deepfake*.

Ressalta-se que, neste trabalho, optou-se pela variação B-0 da arquitetura EfficientNet, que é vista como a menos complexa e de menor desempenho quando comparada com as demais variações existentes. Logo, é possível prever que a utilização de redes mais complexas dessa arquitetura poderiam apresentar um melhor desempenho. Sendo assim, como proposta de trabalho futuro, sugere-se considerar o treinamento de redes mais complexas, baseadas na arquitetura EfficientNet, por meio de múltiplas bases de amostras de vídeos *deepfakes*. Da mesma forma, seria interessante avaliar a utilização de múltiplas bases de amostras com técnicas de visão computacional, de forma a produzir amostras mais variadas e de melhor qualidade.

Por fim, esse trabalho concluiu que é possível produzir modelos de detecção automática de *deepfakes* com alta eficiência e que ferramentas automáticas de detecção podem ser úteis na detecção e moderação de conteúdos audiovisuais.

AGRADECIMENTOS

Os autores gostariam de agradecer ao CEFET-MG pelo suporte financeiro, sem o qual esse trabalho não teria sido possível.

REFERÊNCIAS

[1] Ctrl Shift Face, "Jim Carrey DeepFake [VFX Comparison]," 3 de Setembro 2019. [Online]. Available: https://youtu.be/JbzVhzNaTdI?t=17

[2] ——, "Freddie Mercury DeepFake [VFX Breakdown]," 25 de Agosto 2019. [Online]. Available: https://youtu.be/iwvF9orOnWI?t=68

[3] H. Ajder, G. Patrini, F. Cavalli, and L. Cullen, "The state of deepfakes: Landscape, threats, and impact," Setembro 2019. [Online]. Available: https://sensity.ai/mapping-the-deepfake-landscape/

[4] P. Korshunov and S. Marcel, "DeepFakes: a New Threat to Face Recognition? Assessment and Detection," *arXiv e-prints*, p. arXiv:1812.08685, Dec. 2018.

[5] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv e-prints*, p. arXiv:1409.1556, Sep. 2014.

[6] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[7] C. Sanderson, "The vidtimit database," IDIAP, Tech. Rep., 06 2004.

[8] J. Andress, *The Basics of Information Security*, 2nd ed. Elsevier, 2014.

[9] A. P. Engelbrecht, *Computational intelligence: an introduction*. John Wiley & Sons, 2007.

[10] J. Galbally and S. Marcel, "Face anti-spoofing based on general image quality assessment," in *2014 22nd international conference on pattern recognition*. IEEE, 2014, pp. 1173–1178.

[11] D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 746–761, 2015.

[12] A. Agarwal, R. Singh, M. Vatsa, and A. Noore, "Swapped! digital face presentation attack detection via weighted local magnitude pattern," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2017, pp. 659–665.

[13] T. T. Nguyen, C. M. Nguyen, D. Tien Nguyen, D. Thanh Nguyen, and S. Nahavandi, "Deep Learning for Deepfakes Creation and Detection: A Survey," *arXiv e-prints*, p. arXiv:1909.11573, Sep. 2019.

[14] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3207–3216.

[15] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1–11.

[16] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. Canton Ferrer, "The DeepFake Detection Challenge (DFDC) Dataset," *arXiv e-prints*, p. arXiv:2006.07397, Jun. 2020.

[17] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, "Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 2886–2895.

[18] S. Seferbekov. [Online]. Available: https://github.com/selimsef/dfdc_deepfake_challenge

[19] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[20] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[21] B. Chen, P. Zhuang, and S. Li. [Online]. Available: https://github.com/beibuwandeluori/DeeperForensicsChallengeSolution

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[24] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[25] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv e-prints*, p. arXiv:1412.6980, Dec. 2014.