Application of Digital Image Processing in a Deepfake Detection Network

Lucas Migliorin da Rosa[1], **Carlos Mauricio Serodio Figueiredo**[1]

[1]Escola Superior de Tecnologia – Universidade do Estado do Amazonas (EST - UEA)
69050-020 – Manaus – Am – Brazil

{lmdr.eng19,cfigueiredo}@uea.edu.br

Abstract. *The evolution of Generative Adversarial Networks (GANs) opens up a range of possibilities for malicious users to leverage this technology in order to extract information from other users and spoof their identities. Tools such as DeepFaceLab is an example of the use of these networks to create increasingly realistic Deepfakes, which makes it easier and easier to change people's faces in images or videos. The present work presents an evolution of models for detecting deepfakes through the application of digital image processing techniques and the evolution of literature models applying more current convolutional backbones. Such models are evaluated on datasets from the literature such as the Deepfake Detection Challenge and Faceforensics++.*

Resumo. *A evolução das redes Generative Adversarial Networks (GANs) abre um leque de possibilidades para que usuários mal intencionados aproveitem essa tecnologia a fim extrair informações de outros usuários e falsificando suas identidades. Ferramentas, como DeepFaceLab*[1] *é um exemplo da utilização dessas redes a fim de criar Deepfakes cada vez mais realísticos no qual permite que a troca dos rostos das pessoas em imagens ou vídeos seja cada vez mais fácil. O presente trabalho apresenta uma evolução de modelos de detecção de deepfakes por meio da aplicação de técnicas de processamento digital de imagens e evolução de modelos da literatura aplicando backbones convolucionais mais atuais. Tais modelos são avaliados em datasets da literatura como o Deepfake Detection Challenge e Faceforensics++.*

1. Introdução

O avanço do *Deep Learning* em Redes Adversárias Generativas (GANs) tornou-se comuns vídeos produzidos por outras pessoas no qual a face de uma pessoa é substituída por outra criando assim vídeos vídeos falsos com aspecto realístico. Essa tecnologia, conhecida como *Deepfake* está cada vez mais presente no cotidiano criando notícias falsas, difamando celebridades, líderes e até mesmo usuários comuns que não se têm conhecimento sobre esse tipo de tecnologia [Ismail et al. 2021, Lewis et al. 2020].

Diante do cenário da era digital no qual existem dispositivos que utilizam reconhecimento facial para desbloqueio de contas do próprio dispositivo ou até mesmo dentro de chamadas de vídeos com outros usuários, o *Deepfake* é visto como uma

[1]https://github.com/iperov/DeepFaceLab

ameaça. Usuários maliciosos com tecnologias apropriadas podem extrair informações falsificando sua identidade ferindo a segurança digital de outros usuários [Abbas et al. 2018, Lewis et al. 2020, Ismail et al. 2021].

Exemplos como o do trabalho realizado por [Perov et al. 2020], mostram um *framework* criado para facilitar o *Deepfakes* em vídeos. Essa facilidade permite que usuários comuns da internet sem conhecimentos muito avançados de GANs consigam por meio de uma ferramenta criar vídeos *fakes* customizados com graus de realismo. Alguns vídeos estão disponíveis no YouTube constastando o que essa ferramenta é capaz de fazer demonstrando o nível de realismo do *deepfake* aplicado.

Tendo em vista a necessidade de identificar vídeos *fakes* para melhorar a segurança dos usuários, os autores [Coccomini et al. 2022] desenvolveram um método computacional capaz de identificar esse tipo de vídeo. O presente trabalho visou continuar essa tarefa aumentando as métricas para identificação desses vídeos e melhorá-lo comparando os resultados obtidos no trabalho original com os novos após novas mudanças feitas bem como aplicação de diferentes pré-processamentos e mudanças no *backbone* da rede.

Os conjuntos de treino para a rede foram os mesmos, *Faceforensics++* [Rössler et al. 2019] no qual utilizou cinco tipos diferentes de aplicações de *Deepfakes*, ou seja, cinco métodos distintos e *Deepfake Detection Challenge* (DFDC) [Brian Dolhansky 2020] para predição presente nos conjuntos de treino, validação e teste.

O trabalho está organizado da seguinte forma: a seção 2 aponta os principais trabalhos relacionados com esse artigo, a seção 3 descreve o método proposto e as tecnologias utilizadas, seção 4 apresenta o sistema e sua arquitetura, a seção 5 descreve os experimentos e resultados e, por fim, a seção 6 apresenta as conclusões e possibilidade de trabalhos futuros.

2. Trabalhos relacionados

Nesta seção serão discutidos trabalhos relacionados que propõem soluções sobre detecção de *Deepfakes*, além disso, serão abordadas semelhanças e diferenças com o presente trabalho.

[Ismail et al. 2021] criaram uma rede chamada YOLO-InceptionResNetV2-XGBoost no qual seu objetivo é detectar *Deepfakes* em vídeos usando para treinamento os datasets *FaceForensics* e *Celeb DF*. Comparados com o presente artigo, os datasets usados para treinamento foram *Faceforensics++* e DFDC. Além disso, YOLO-InceptionResNetV2-XGBoost conta com o XGBoost no qual se mostrou promissor na detecção de vídeos *fakes* e em radiografias do tórax de pessoas com COVID-19. Essa decisão do uso do XGBoost, de acordo com os autores, serviu para evitar o *Overfitting*. A acurácia do modelo ficou em 90.62%, enquanto o presente trabalho conseguiu obter um resultado de 96.00%.

[Lewis et al. 2020] criaram uma rede híbrida no qual captura tanto características das imagens nos vídeos quanto o áudio no dataset DFDC no qual também foi utilizado pelo presente trabalho. Além disso, os autores utilizaram *transfer learning* juntamente com redes *Long Short Term Memorys* (LSTM) unindo as características extraídas dos áudios e imagens a fim de classificar um vídeo entre *real* ou *fake*. Outra característica importante foi a quantidade de frames usados para a detecção sendo 30 *frames* para os

autores enquanto o presente artigo utilizou de 15 *frames* obtendo 54% e 95% respectivamente nas medidas de acurácias para o dataset DFDC.

O trabalho desenvolvido por [Zhao et al. 2021] reformularam o método convencional de detecção de *deepfakes* através de um problema binário para um problema de *Fine-Grained Classification*. Essa transformação trás uma nova perspectiva para o problema, além disso os autores trouxeram nova função de *loss*, métodos de *augmentation* específicos para essa nova abordagem e uma nova arquitetura de rede. Os datasets usados foram *FaceForensics++*, *Celeb-DF* e DFDC, enquanto o presente trabalho optou por usar somente *FaceForensics++* e DFDC. Pra validação dos dados, os autores usaram *FaceForensics++* com acurácia de 97.60% enquanto o presente trabalho apresentou para esse dataset 88.00%, já para o DFDC a métrica utilizada foi o *logloss* no qual não aplicamos no presente artigo.

No trabalho realizado por [Guarnera et al. 2020] os autores a fim de detectar características nas imagens *fakes*, partiram da suposição que imagens que sofreram alterações em seus conteúdos através de modelos GANs, como GDWCT, STARGAN, ATTGAN, STYLEGAN e STYLEGAN2 deixam vestígios de suas manipulações. A partir disso, o método proposto por eles faz o encode das informações juntamente com a função *Expectation Maximization (EM)* para calcular a distância de imagens reais ou falsas. Comparado com o presente trabalho, optamos em usar redes Convolucionais ligadas à uma rede Transformer responsável por codificar as informações para a tomada de decisão através de um neurônio *Multi Layer Perceptron (MLP)*. Além disso, optamos por usar dois diferentes conjuntos de dados para treinamento.

Diante aos trabalhos citados, existem semelhanças na metodologia aplicada para a execução do presente trabalho. A escolha dos dados para treinamento, por exemplo, assemelha-se aos trabalhos de [Ismail et al. 2021] utilizando o dataset *FaceForensics++* e de [Lewis et al. 2020] usando o dataset DFDC, entretanto ainda que [Zhao et al. 2021] utilizem esses dois conjuntos de dados os autores reformularem a abordagem desse problema diferenciando-se como os dados são visto pela rede. Já os trabalhos de [Lewis et al. 2020, Ismail et al. 2021] possuem redes principais executando a função de extrator de características a fim de repassar essas características para a estrutura responsável pela predição. Todos também os trabalhos usaram entre 10 e 30 *frames* por vídeo a fim de predizer sua classe, real ou *fake*.

3. Método proposto

Nesta seção será apresentada a descrição do conjunto de dados, pré-processamento, ferramentas e arquitetura do algoritmo de *Machine Learning* utilizado neste trabalho. Foi empregada a abordagem de aprendizagem supervisionada para a classificação dos vídeos.

3.1. Base de dados

O DFDC[2] é um conjunto de dados criado pela empresa Facebook administrada pela *Meta* lançado em sua versão de pré-lançamento em 2019 e tendo sua versão final em 2020 disponível para *download* tanto no Kaggle quanto no próprio site deles em sub conjuntos de dados, ou seja, *datasets* com aproximadamente 2500 vídeos cada, entre eles reais e falsos.

[2]https://ai.facebook.com/datasets/dfdc/

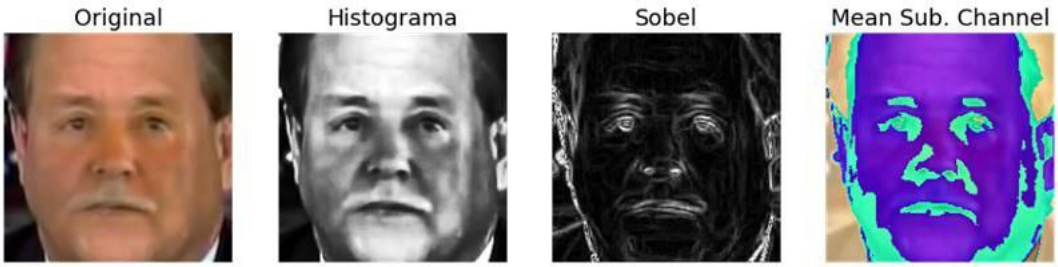


Figura 1. Exemplo de aplicação dos filtros escolhidos a partir do livro [Gonzalez and Woods 2009] a fim de melhorar a identificação de vídeos *fakes* e reais.

Para os conjuntos de treinamento, validação e teste foi usado a parte 49 disponibilizada no site contendo no total 3134 vídeos entre eles 2619 *fakes* e 515 reais com duração de 10 segundos e 29.06 ± 0.25 de *FPS* separando os dados de forma *holdout*, ou seja, 70%, 15% e 15% respectivamente. A escolha para a parte 49 se baseia em uma seleção aleatória dentre as outras possíveis. Além disso, o DFDC possui um conjunto específico para teste contendo 5000 vídeos com duração de 10 segundos e uma faixa de 28.30 ± 4.66 de *FPS*. Esse conjunto foi inteiramente usado para teste.

O conjunto de dados *Faceforensics++* lançado em 2019 contendo inicialmente 5 manipulações distintas de *deepfakse*, como Deepfakes, Face2Face, FaceSwap e NeuralTextures possui 1000 vídeos manipulados para cada método e em 2020 foi adicionado o método FaceShifter contemplando 6000 vídeos falsos e 1000 reais atualmente com duração de 16.72 ± 6.913 segundos e 27.63 ± 3.92 de *FPS*. A divisão dos vídeos para treino, validação e teste seguiu a proporção de 70%, 15% e 15% respectivamente. As instruções para download encontram-se no GitHub[3] dos autores.

Tanto o DFDC quanto o *Faceforensics++* foram usados neste trabalho por ser uma continuação do trabalho desenvolvido por [Coccomini et al. 2022] no qual utilizaram esses dados para realizar o treinamento, validação e teste do modelo.

3.2. Pré-processamento

Para que melhor seja identificado o *deepfake* na imagem, os autores [Coccomini et al. 2022] usaram um *Face detector* MTCNN [Zhang et al. 2016] para fazer o recorte do rosto das pessoas nos vídeos e em seguida separar em *frames* obtendo várias imagens que representem o vídeo. Somente esse pré-processamento foi utilizado pelos autores e no presente trabalho visamos testar outros tipos de técnicas que complementassem ao que foi originalmente usado.

A equalização de histograma, filtro de Sobel e *Mean Subtraction Channel* foram algumas das técnicas escolhidas para serem estudadas a partir do livro Processamento Digital de Imagens e aplicadas conforme o ilustrado na Figura 1 [Gonzalez and Woods 2009]. Buscávamos pré-processamentos que evidenciassem distorções nas imagens ocasionado pelo *deepfake* auxiliando o modelo nos processos de treinamento, validação e teste além de aumentar a generalização para identificar os diferentes modelos aplicados nos vídeos.

[3]https://github.com/ondyari/FaceForensics

Na entrada da rede, as imagens foram reduzidas para a dimensão de $(244x244x3)$ contendo 3 canais de cores e seus *pixels* foram mantidos em uma faixa de valores entre $[0, 255]$ na escala RGB.

3.3. Ferramentas

Foi utilizada a linguagem de programação *Python* na versão 3.7 em conjunto com a biblioteca de abstração de *Machine Learning Keras* na versão 2.7, aplicando o TensorFlow na versão 2.7 como *backend* de processamento.

Para o processamento de imagens e aplicação dos pré-processamentos foram usados a biblioteca *OpenCv* e *Numpy*.

3.4. Arquitetura do modelo de aprendizagem

O trabalho criado por [Coccomini et al. 2022] propõe dois modelos *Efficient ViT* e *Convolutional Cross ViT*, ambos possuem redes convolucionais CNN como *backbone* ligadas a um *transformer*.

Efficient ViT em seu trabalho original possui como extrator de características a rede *Efficient-Net B0* no qual tem como saída *chunks* de dimensão $7x7$ direcionadas como entrada para uma progressão linear ligada a um *transformer* responsável por fazer o *encoding* dessas informações. Por fim, para realizar a predição de um vídeo, um neurônio *Multi Layer Perceptron MLP* recebe essas informações e retorna um número a fim de compará-lo com um limiar estabelecido de 0.5, conforme o trabalho original, e em seguida classificá-lo como falso se acima do limiar, caso contrário em real.

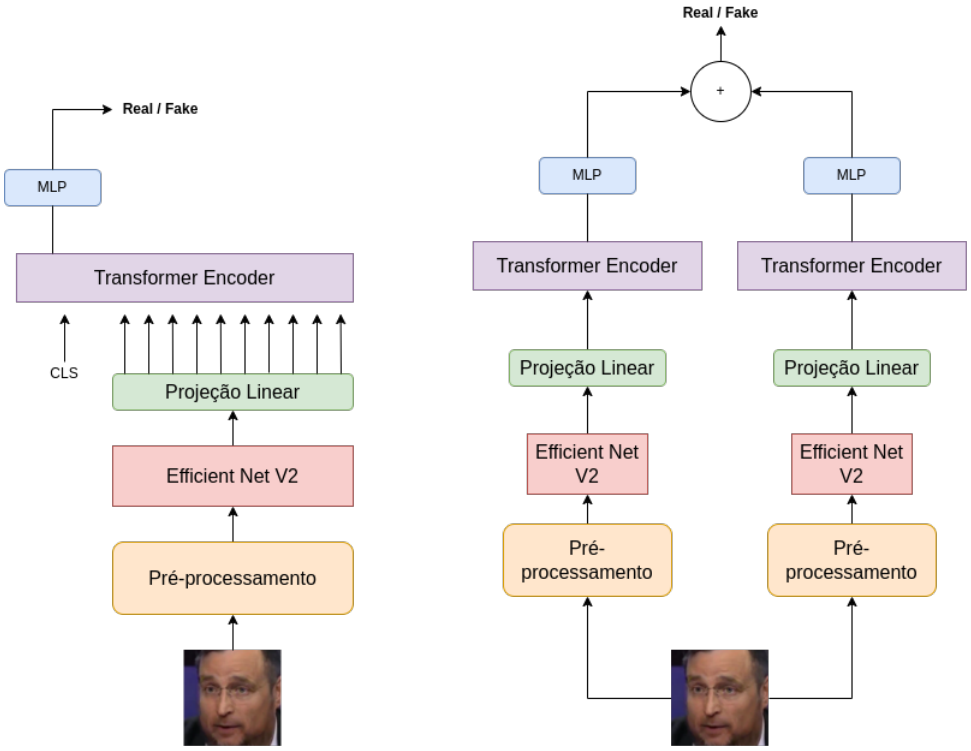
Já o modelo *Convolutional Cross ViT* atua com dois extratores de características iguais, sendo ele o *Efficient-Net B0*, porém o primeiro tem como saída *chunks* de dimensão $7x7$ e o segundo tem saída de *chunks* de dimensão $64x64$. Ambos passam por uma progressão linear que em seguida é direcionada ao *transformer* a fim de realizar o *encoder* dessas informações. Paralelamente, cada um possui um neurônio MLP cujas saídas são somadas, e então, passam pelo o limiar de classificação presente no modelo *Efficient ViT*

No presente trabalho fizemos alterações tanto no *backbone* buscando versões mais recentes do *Efficient Net*, como o *Efficient Net V2*, quanto em adicionar uma camada de pré-processamento antes da imagem sofrer a extração de característica pelo *Backbone*, como foi mencionado na seção 3.2 [Tan and Le 2021]. A visualização das redes *Efficient ViT* e *Convolutional Cross ViT* pode ser vista na Figura 2 no qual demonstra as novas melhorias propostas.

4. Experimentos e resultados

O processo de treinamento das duas redes foram executadas em uma GPU GTX2080 com capacidade de 8GB, uma CPU Intel i7, 16 GB de ram e um SSD de capacidade 1TB.

[Coccomini et al. 2022] em seu trabalho utilizaram a acurácia como métrica para ambos os *datasets* no conjunto de teste, entretanto somente para o DFDC utilizaram o F1 Score. Dessa forma, no presente trabalho buscamos avaliar o desempenho do modelo usando tais métricas e também em vídeos sem *deepfake* presentes no *Faceforensics++*, ou seja, imagens originais no qual não é reportado no trabalho original. Manteve-se um *threshold* de 0.5 a fim de determinar a veracidade dos vídeos para o problema em questão.



(a) Efficient ViT

(b) Convolutional Cross. Efficient ViT

Figura 2. Arquitetura da rede *Efficient Vit* e *Convolutional Cross Efficient Vit* com modificações do extrator de características e camada de pré-processamento proposta posicionada antes do *backbone*. Adaptado de [Coccomini et al. 2022]

4.1. Treino

O treinamento do modelo foi realizado a partir da base de dados citadas na seção 3.1 com *holdout* de 70% no *dataset Faceforensics++* e no DFDC estratificando-os a fim de manterem suas proporções entre vídeos *fakes* e reais, ou seja, os conjuntos além do treinamento mantiveram também a diferença proporcionalmente igual de vídeos reais e falsos. No trabalho original não há uma especificação clara se também foi realizado uma estratificação e o tempo de treinamento para cada experimento do presente artigo durou em média 48 horas.

No trabalho original desenvolvido por [Coccomini et al. 2022], o treinamento foi realizado usando o conjunto de treino completo do *dataset* DFDC, ou seja, as 50 partes disponíveis que totalizam 245 GB comprimidos e um subconjunto para treino do *Faceforensics++*. No presente trabalho, por haver limitações de tempo e de recursos computacionais optamos em usar somente uma parte, como citado na seção 3.1 a fim de obter resultados e comparações entre o trabalho original e os pré-processamentos nos vídeos citados na seção 3.2.

4.2. Validação

Durante a validação dos dados, seguiu-se a separação dos resultados para ambos os *datasets*, como no trabalho original no qual os resultados obtidos podem ser vistos nas tabelas

1 e 3. Além disso, foi adicionado no presente trabalho a coluna ”Original”a fim de avaliar o desempenho do modelo para os vídeos reais no conjunto de teste do *Faceforensics++*.

Os experimentos executados foram:

1. Mudança somente no *backbone* pela *Efficient Net V2*
2. *Efficient Net V2* e pré-processamento adicional histograma
3. *Efficient Net V2* e pré-processamento adicional sobel
4. *Efficient Net V2* e pré-processamento adicional *mean subtract channel* (MSC)

	Acurácia					
Modelo	FaceShifter	FaceSwap	NeuralTextures	Deepfakes	Original*	F1 Score
[Coccomini et al. 2022]	$0,7933$	$0,9176$	$0,4538$	$0,8733$	$0,7666$	$0,8454$
Experimento 1	$0,9066$	$0,9352$	$0,5769$	$0,9400$	**0,9266**	$0,9108$
Experimento 2	$0,9333$	$0,9705$	$0,6153$	$0,9333$	$0,8200$	**0,9114**
Experimento 3	**0,9800**	**0,9764**	**0,7153**	**0,9733**	$0,6133$	$0,8809$
Experimento 4	$0,7333$	$0,7823$	$0,4000$	0.9000	$0,8066$	$0,8120$

Tabela 1. Resultados obtidos na execução da lista de experimentos no *dataset Faceforensics++*. A coluna Original destina-se aos vídeos reais, ou seja, que não sofreram *deepfakes* e que não se faz presente em nosso *baseline*.

De acordo com a tabela 1, o experimento 3 indica melhoras na identificação das 4 redes usadas para geração de *deepfake*, porém quando comparada à identificação de vídeos reais há uma queda de $15,33\%$ na acurácia em relação ao trabalho original indicando um decaimento na generalização do modelo para esse tipo de vídeo. Ainda no contexto de vídeos reais, o experimento 1 obteve um aumento de 16% na acurácia e também na identificação das 4 redes de *deepfake*, entretanto usando a métrica f1 score o experimento 2 obtém um resultado de $0,9114$ com diferenças entre o trabalho original e o experimento 3 de $0,066$ e $0,006$, respectivamente.

Modelo	Acuracia (Média)
[Coccomini et al. 2022]	0.7609 ± 0.1628
Experimento 3	0.8570 ± 0.1405
Experimento 2	$\mathbf{0.8544 \pm 0.1298}$

Tabela 2. Comparação dos resultados pela acurácia média e desvio padrão.

A fim de melhor comparação, como pode ser visto na tabela 2, a média das acurácias desempenharam melhor no experimento 2 em que houve um desvio padrão menor dos três modelos apresentados ainda que a média do experimento 3 tenha uma diferença de $0,26\%$. Dessa forma, a aplicação do histograma e mudança da rede *backbone* para *Efficient Net V2* apresentam melhorias mais homogêneas no resultado no *dataset Faceforensics++* tanto nos vídeos *fake* quanto nos vídeos reais.

	Acurácia		
Model	Sem ruído	Com ruído*	F1-score
[Coccomini et al. 2022]	$0,9723$	$0,6662$	**0,7183**
Experimento 1	**0,9830**	**0,6680**	$0,6744$
Experimento 2	$0,9766$	$0,6580$	$0,6914$
Experimento 3	$0,9745$	$0,6406$	$0,6573$
Experimento 4	$0,9681$	$0,6156$	$0,6704$

Tabela 3. Resultados obtidos na execução dos experimentos no *dataset* DFDC. O conjunto com ruído pertence ao grupo de imagens que foram baixadas contendo um arquivo sinalizando qual tipo de ruído foi aplicado.

Para os vídeos presentes no DFDC, a tabela 3 mostra a acurácia e f1 score tanto em vídeos com ruído quanto sem ruído no conjunto de teste desse *dataset*. O próprio site que forneceu o dataset informa que esse conjunto recebeu alterações como também para cada vídeo há um arquivo *json* informando que tipo de ruído foi aplicado ou se foi aplicado. Não houveram mudanças em geral acima de 2% na acurácia para os vídeos sem ruído e o experimento 1 apresentou melhor acurácia do modelo com aumento de $1,1\%$ em relação ao trabalho original, entretanto observando o f1 score o trabalho original obtém melhor valor.

As tabelas 1 e 2 mostram que os experimentos realizados alcançaram diferentes melhoras no conjunto de dados *Faceforensics++* no qual sua melhor acurácia média foi de $85,44\% \pm 12,98\%$ com f1 score de $0,9114$. Essa análise teve como objetivo encontrar um experimento que melhor distribuísse melhorias tanto para a detecção de vídeos *fakes* quanto para os reais. Além disso, a tabela 3 mostrou que os pré-processamentos e a mudança do *backbone* da rede não alterou significavelmente os resultados do trabalho original, no qual há como hipótese que o treinamento foi executado somente com uma parte dos dados, como foi mencionado na seção 4.1 limitando a possibilidade de abstrações melhores para a rede devido a limitações computacionais e de tempo.

5. Conclusão e trabalhos futuros

O presente trabalho apresentou um estudo e seleção de técnicas de processamento digital de imagens com o objetivo de aplicar nos conjuntos de dados *Faceforensics++* e DFDC disponibilizados publicamente e buscando melhorias nos resultados apresentados no trabalho feito por [Coccomini et al. 2022] na identificação de vídeos *fakes* e reais. Além disso, buscou-se também a mudança do extrator de características para versões mais recentes, como o *Efficient Net V2* [Tan and Le 2021] com o propósito de compor as técnicas de pré-processamento e comparar seu impacto na rede com o trabalho original.

Os resultados adquiridos concluem que os pré-processamentos influenciam positivamente a predição do modelo para o problema de detectar vídeos reais ou não e a mudança do extrator de características impacta também positivamente. Ainda há também possibilidades de aplicar outras técnicas e estudar mais a fundo seus impactos uma vez que foram usados somente 3 técnicas distintas.

Para trabalhos futuros, pretende-se gradativamente aumentar a quantidade de dados para treinamento levando em consideração a estratificação dos dados e observando o

impacto que há sobre a rede. Além disso, pretende-se também estudar e aplicar diferentes tipos de *loss functions* em que não foi testado no trabalho original e também adicionar à validação métricas, como matriz confusão, revocação e precisão para uma investigação mais minuciosa.

Referências

Abbas, G., Humayoun, S. R., AlTarawneh, R., and Ebert, A. (2018). Simple shape-based touch behavioral biometrics authentication for smart mobiles. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, pages 1–3.

Brian Dolhansky, Joanna Bitton, B. P. J. L. R. H. M. W. C. C. F. (2020). The deepfake detection challenge dataset.

Coccomini, D. A., Messina, N., Gennaro, C., and Falchi, F. (2022). Combining efficientnet and vision transformers for video deepfake detection. In *Image Analysis and Processing–ICIAP 2022: 21st International Conference, Lecce, Italy, May 23–27, 2022, Proceedings, Part III*, pages 219–229. Springer.

Gonzalez, R. C. and Woods, R. E. (2009). *Processamento Digital de Imagens*, volume 3. Person.

Guarnera, L., Giudice, O., and Battiato, S. (2020). Deepfake detection by analyzing convolutional traces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 666–667.

Ismail, A., Elpeltagy, M., Zaki, M. S., and Eldahshan, K. (2021). A new deep learning-based methodology for video deepfake detection using XGBoost. *Sensors*, 21(16):5413.

Lewis, J. K., Toubal, I. E., Chen, H., Sandesera, V., Lomnitz, M., Hampel-Arias, Z., Prasad, C., and Palaniappan, K. (2020). Deepfake video detection based on spatial, spectral, and temporal inconsistencies using multimodal deep learning. In *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–9. IEEE.

Perov, I., Gao, D., Chervoniy, N., Liu, K., Marangonda, S., Umé, C., Dpfks, M., Facenheim, C. S., RP, L., Jiang, J., Zhang, S., Wu, P., Zhou, B., and Zhang, W. (2020). Deepfacelab: Integrated, flexible and extensible face-swapping framework.

Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., and Nießner, M. (2019). Faceforensics++: Learning to detect manipulated facial images. *CoRR*, abs/1901.08971.

Tan, M. and Le, Q. V. (2021). Efficientnetv2: Smaller models and faster training. *CoRR*, abs/2104.00298.

Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multi-task cascaded convolutional networks. *CoRR*, abs/1604.02878.

Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., and Yu, N. (2021). Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2185–2194.