ANÁLISE COMPARATIVA DE MODELOS CNN PARA DETECÇÃO DE DEEPFAKES UTILIZANDO EFFICIENTNET E GRAD-CAM

Lucas Mairon Oliveira Camilo[1], Claudio Caueh Oliveira Xavier[2], Rosana C. B. Rego[2]

[1]*Universidade Federal Rural do Semi-Árido, Rio Grande do Norte, Brasil*
(lucasmairon2@gmail.com)
[2]*Departamento de Engenharias e Tecnologia, Universidade Federal Rural do Semi-Árido, Rio Grande do Norte, Brasil*

Resumo: Este trabalho apresenta um estudo comparativo entre uma CNN desenvolvida do zero e o modelo pré-treinado EfficientNetV2-S para a detecção de *deepfakes*. Utilizou-se um conjunto de dados de 20.000 imagens (StyleGAN3 e FFHQ). O modelo EfficientNetV2-S superou a CNN simples, atingindo 93% em todas as métricas de desempenho avaliadas. O modelo selecionado foi integrado ao mecanismo Grad-CAM para IA Explicável e disponibilizado em uma interface *web* por meio do Streamlit.

Palavras-chave: Visão Computacional; Redes Neurais; EfficientNetV2-S; Deepfakes; IA Explicável.

INTRODUÇÃO

Nos últimos anos, o campo da Visão Computacional avançou de forma acentuada, impulsionado pelo desenvolvimento das Redes Neurais Convolucionais (CNNs) (LECUN; BENGIO; HINTON, 2015). Essas arquiteturas consolidaram-se como referência para tarefas de classificação de imagens, superando abordagens baseadas em extração manual de características, em razão da capacidade de extração automática de representações hierárquicas.

O desenvolvimento e o treinamento de uma CNN do zero para uma tarefa específica, embora possa resultar em um modelo adaptado, apresenta desafios práticos. Modelos desenvolvidos do zero frequentemente exigem grandes volumes de dados rotulados e tempo computacional considerável para convergir. Em cenários de escassez de dados ou de limitação de *hardware*, o Aprendizado por Transferência (*Transfer Learning*) constitui uma alternativa, pois permite o reaproveitamento de representações aprendidas por redes treinadas em grandes bases de dados.

A família *EfficientNetV2* (TAN; LE, 2021) foi proposta com o objetivo de otimizar simultaneamente a acurácia, a eficiência computacional e a velocidade de treinamento em redes neurais convolucionais. Diferentemente das versões anteriores, a arquitetura emprega uma estratégia de escalonamento composto aprimorada e introduz os blocos convolucionais fundidos (*Fused-MBConv*), os quais reduzem o custo computacional nas camadas iniciais da rede. Nesse contexto, a variante *EfficientNetV2-S* destaca-se por apresentar capacidade de generalização com menor número de parâmetros e menor tempo de treinamento em comparação a arquiteturas convolucionais convencionais.

Este trabalho apresenta uma comparação entre uma arquitetura CNN desenvolvida do zero e o modelo pré-treinado *EfficientNetV2-S*. O estudo analisa métricas de desempenho, incluindo acurácia e F1-Score, na tarefa de detecção de rostos sintéticos (*Fake Faces*). Para os experimentos, foi utilizado um conjunto de dados com 20.000 imagens, composto por 10.000 faces sintéticas geradas pelo algoritmo StyleGAN3 (KARRAS et al., 2021) e 10.000 imagens reais do *dataset Flickr-Faces-HQ* (FFHQ) (KARRAS; LAINE; AILA, 2019). Além disso, o método de explicabilidade Grad-CAM (SELVARAJU et al., 2017) foi aplicado para analisar as regiões de maior relevância nas imagens, contribuindo para a compreensão do processo de decisão dos modelos.

O restante deste trabalho está organizado da seguinte forma: a Seção Material e Métodos detalha a metodologia, as arquiteturas utilizadas e o conjunto de dados; a Seção Resultados e Discussão discute os experimentos e resultados obtidos; e, por fim, a Seção Conclusão apresenta as considerações finais e propostas de trabalhos futuros.

TRABALHOS RELACIONADOS

O trabalho proposto por Rossler et al. (2019) apresenta o FaceForensics++, um *benchmark* para detecção de manipulações faciais. O estudo foca na criação e avaliação de uma base de dados a partir de vídeos, analisando diferentes tipos de manipulações em vídeos, diferenciando-se da presente proposta, que trata da identificação em imagens estáticas.

Em seu trabalho, Wang et al. (2020) investigam se imagens geradas por IA ainda apresentam padrões identificáveis por CNNs, confirmando que esse tipo de rede mantém capacidade para distinguir imagens reais de sintéticas com base em padrões extraídos.

Yu, Davis e Fritz (2019) apresenta uma análise sobre as ‘impressões digitais” deixadas por modelos GAN nas imagens por eles geradas, identificando que cada modelo

de geração deixa padrões próprios na imagem, o que permite não apenas distinguir imagens reais de sintéticas, mas também identificar o modelo gerador responsável por cada imagem.

Quanto ao Grad-CAM, Selvaraju et al. (2017) introduz esse método, uma técnica de IA explicável que utiliza gradientes das classes-alvo para identificar as regiões de maior relevância nas imagens. Esse método foi adotado para gerar a explicabilidade do modelo proposto neste trabalho.

O trabalho de Mansoor e Iliev (2025) analisa diferentes modelos, como InceptionResNetV2, DenseNet201, InceptionV3 e ResNet152V2, em um conjunto de dados gerado pelo StyleGAN, para identificação de *deepfakes*. O estudo também incorpora análise de explicabilidade da IA.

Deng, Suo e Li (2022) aplicam a arquitetura EfficientNetV2S para a identificação de vídeos falsificados por meio de técnicas de *deepfake*, explorando a arquitetura para identificar vídeos manipulados por características visuais. Os resultados indicam desempenho adequado da arquitetura na tarefa proposta.

Nguyen et al. (2022) apresentam uma revisão sobre técnicas de criação e detecção de *deepfakes*, abrangendo tanto os mecanismos de geração de conteúdos falsificados quanto os métodos para sua detecção.

MATERIAL E MÉTODOS

Neste trabalho, foram treinados e avaliados dois modelos de classificação de imagens: uma CNN desenvolvida do zero e uma adaptação da EfficientNetV2S, modelo disponibilizado pelo TensorFlow[1], adotando uma abordagem comparativa entre as duas arquiteturas.

Para visualização dos códigos realizados no treinamento dos modelos foi criado um repositório no GitHub, disponível em https://github.com/ClassNeuralNetwork/deepfake-identification, contendo a licença MIT.

Dados Utilizados Para o Treinamento

O *dataset* utilizado está disponível na plataforma Kaggle[2] em https://www.kaggle.com/datasets/troykueh/real-vs-fake-faces-stylegan3/data. O conjunto contém 20.000 amostras, distribuídas igualmente entre imagens de faces reais e faces sintéticas. Os dados são organizados em duas colunas: *filepath*, com o caminho para a imagem, e *label*, com a classe correspondente (real ou fake).

Pré-Processamento de Dados

O conjunto de dados foi particionado em três subconjuntos: treino (70%), validação (15%) e teste (15%), todos com *batch size* de 64. Foram aplicadas operações de *prefetch* nos subconjuntos de treino e validação, utilizando *buffer size* com *autotune*. Como as imagens já se encontravam no formato 256×256, nenhum redimensionamento foi necessário.

[1]https://www.tensorflow.org/
[2]https://www.kaggle.com/

Na CNN desenvolvida do zero, foi aplicada a normalização dos pixels, dividindo-os por 255. Esse procedimento não foi necessário para o modelo EfficientNetV2S, pois a própria arquitetura realiza o pré-processamento internamente.

Arquitetura do Modelo CNN Simples

A arquitetura da CNN, ilustrada na Figura 1, é composta por quatro camadas convolucionais com 32, 64, 128 e 256 filtros, respectivamente, todas com função de ativação ReLU. Após cada camada convolucional, são aplicados *BatchNormalization* e *MaxPooling*. A arquitetura é encerrada com uma camada *Flatten*, uma camada densa de 256 neurônios com ativação ReLU, uma camada *Dropout* de 50%, que desativa metade dos neurônios aleatoriamente durante o treinamento, e uma camada de saída com um neurônio e ativação sigmoid, adequada para classificação binária.

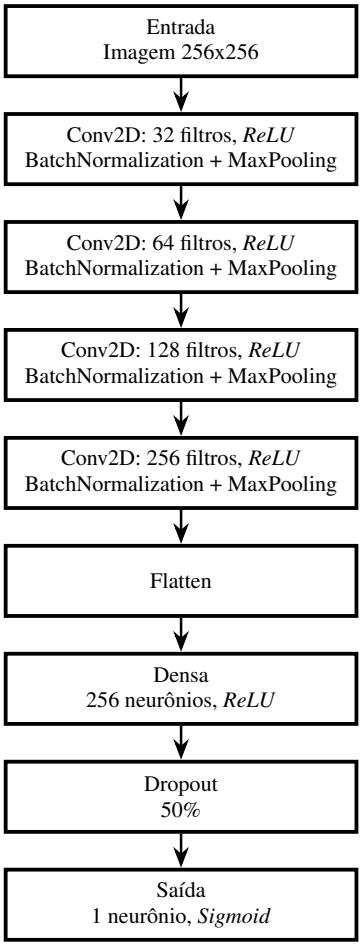

Figura 1: Arquitetura da CNN desenvolvida do zero.

Arquitetura do Modelo EfficientNetV2S

A arquitetura do modelo EfficientNetV2S, ilustrada na Figura 2, recebe como entrada imagens de dimensão 256×256. O bloco convolucional é formado pela própria EfficientNetV2S pré-treinada, seguida de uma camada GlobalAveragePooling2D para redução dimensional. A parte de classificação é composta por um perceptron

multicamadas com duas camadas densas com função de ativação ReLU e regularização L2 de 1×10^{-4}, cada uma seguida de um *Dropout* de 20%. A arquitetura é concluída com uma camada densa de um neurônio com ativação sigmoid, adequada para classificação binária.

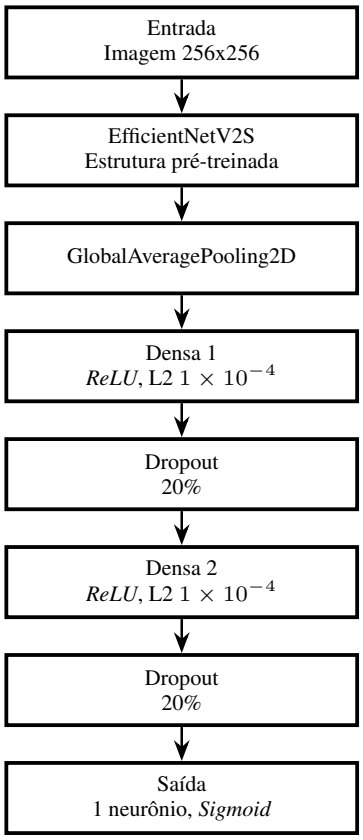

Figura 2: Arquitetura do modelo EfficientNetV2S.

Data Augmentation nos Modelos

Em ambos os modelos foi utilizado *data augmentation*, implementado como uma camada do modelo, responsável por gerar transformações das imagens de entrada, como rotação e *zoom*, durante o treinamento, com o objetivo de reduzir o *overfitting* e ampliar a capacidade de generalização dos modelos.

Callbacks Utilizados nos Modelos

Os dois modelos utilizam mecanismos de *callback* durante o treinamento: o *early stopping*, configurado para monitorar a função perda de validação e interromper o treinamento quando ela comece a aumentar, e o *checkpoint*, para salvar os pesos do modelo no estado de menor perda de validação.

Compilação e Treinamento dos Modelos

Os dois modelos foram compilados com a função perda *binary crossentropy*, adequada para saídas binárias, e otimizador Adam com taxa de aprendizado de 1×10^{-3}. A CNN desenvolvida do zero foi treinada por até 200 épocas, enquanto o modelo EfficientNetV2S foi treinado por até 100 épocas. Em ambos os casos, o treinamento utilizou o subconjunto de validação para monitoramento do desempenho a cada época.

RESULTADOS E DISCUSSÃO

Para o tipo de problema tratado, foram utilizadas as métricas de Acurácia, Precisão, Recall e F1-Score, além da matriz de confusão e da curva de perda. Esta seção apresenta os resultados dos dois modelos e uma análise comparativa entre eles.

Resultados de Treino do Modelo CNN simples

Conforme a Tabela 1, as métricas do modelo foram: acurácia de 0,86, precisão de 0,90, recall de 0,81 e F1-Score de 0,85. Os resultados indicam que o modelo obteve desempenho abaixo do observado para o EfficientNetV2S, sugerindo margem para ajustes na arquitetura ou no processo de treinamento.

Tabela 1: Métricas do modelo CNN simples.

Acurácia	Precisão	Recall	F1-Score
0.8630	0.9006	0.8160	0.8562

Para ilustrar o desempenho durante o treinamento, foram gerados gráficos de acurácia e de função perda, apresentados na Figura 3. As curvas indicam oscilações, possivelmente associadas ao volume de dados utilizado. Observa-se também sinais de *overfitting* a partir da época 60, indicando dificuldade de generalização do modelo.

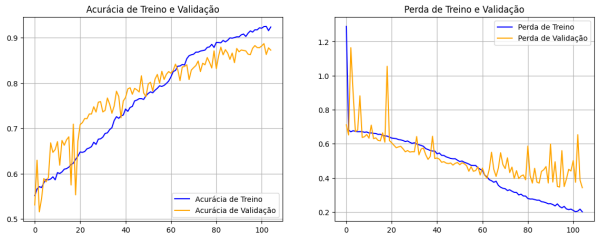

Figura 3: Gráfico de Acurácia e Função Perda do Modelo CNN simples.

A Figura 4 apresenta a matriz de confusão do modelo. Foram registrados 135 falsos positivos e 276 falsos negativos, indicando que a taxa de erro é mais elevada na classificação de faces sintéticas.

Resultados de Treino do Modelo EfficientNetV2S

Em relação às métricas do modelo EfficientNetV2S, exibidas na Tabela 2, observa-se que os valores permanecem uniformes em 93% para todas as métricas, indicando que, mesmo diante de oscilações na curva de perda, o treinamento produziu resultados consistentes.

Tabela 2: Métricas do modelo EfficientNetV2S.

Acurácia	Precisão	Recall	F1-Score
0.93	0.93	0.93	0.93

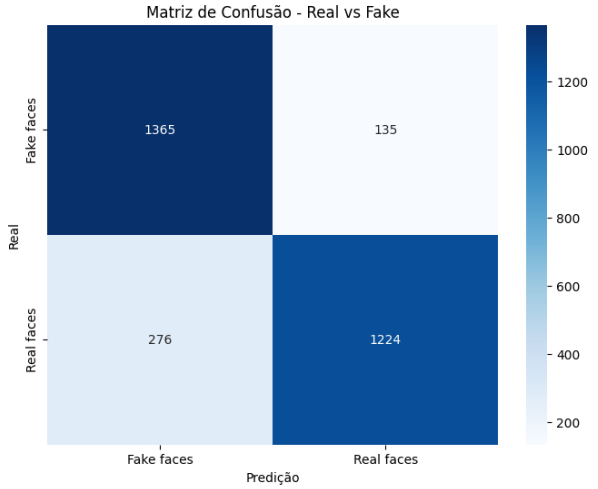

Figura 4: Matriz de Confusão do Modelo CNN simples.

Quanto ao desempenho durante o treinamento, a Figura 5 apresenta as curvas de acurácia e de função perda. Observam-se oscilações nas curvas, possivelmente associadas ao volume de dados fornecidos à rede. Também é possível verificar um sinal de *overfitting* na curva de perda de validação, o que pode indicar que o modelo atingiu o limite de aprendizado possível com essa arquitetura e conjunto de dados.

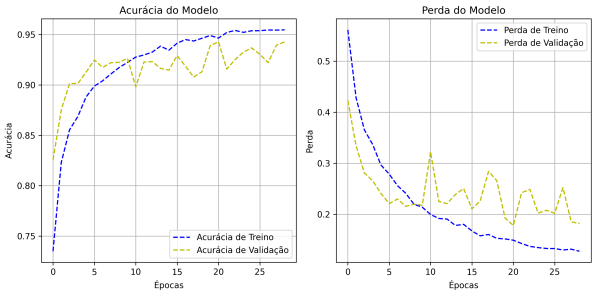

Figura 5: Gráfico de Acurácia e Função Perda do Modelo EfficientNetV2S.

A Figura 6 apresenta a matriz de confusão do modelo, na qual foram registrados 101 falsos positivos e 109 falsos negativos, indicando desempenho superior ao da CNN na classificação de *deepfakes*.

Análise Comparativa dos modelos

Com base nos resultados obtidos em ambos os modelos, a análise comparativa indica que o modelo EfficientNetV2S obteve desempenho superior: além de apresentar valores de métricas superiores em todos os indicadores, registrou menor número de erros no subconjunto de validação.

Selecionado o modelo EfficientNetV2S, foi gerado o relatório de IA explicável para análise do processo de decisão do modelo, e desenvolvida uma aplicação no Streamlit para classificação de imagens fornecidas pelo usuário.

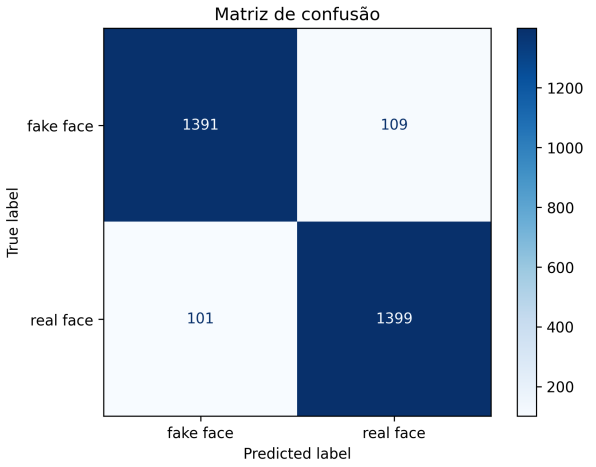

Figura 6: Matriz de Confusão do Modelo EfficientNetV2S.

IA Explicável

Para a geração da IA explicável, foi utilizado o mecanismo Grad-CAM (*Gradient-weighted Class Activation Mapping*), que utiliza o cálculo do gradiente para identificar as regiões da imagem com maior relevância para a decisão do modelo(SELVARAJU et al., 2017).

A Figura 7 apresenta o resultado do Grad-CAM, exibindo as imagens originais, os mapas de calor correspondentes e as imagens sobrepostas ao mapa. A análise indica que o modelo baseia suas decisões em regiões da face, como olhos, cabelos e boca, e, em alguns casos, no plano de fundo da imagem.

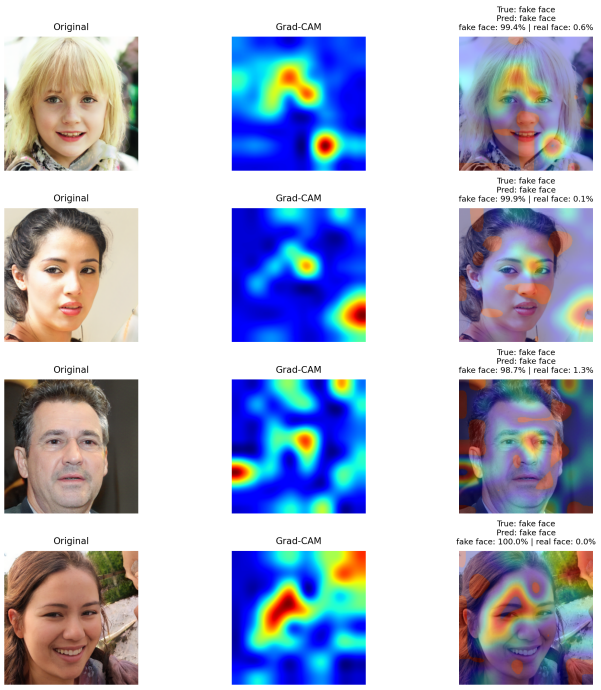

Figura 7: Relatório de IA explicável do modelo EfficientNetV2S.

Aplicação web

O Streamlit[3] foi utilizado para disponibilizar uma interface *web* de interação com o modelo, permitindo que o usuário submeta uma imagem e obtenha a classificação correspondente. A Figura 8 apresenta a interface desenvolvida, na qual o usuário pode submeter imagens a partir de arquivos ou por meio da câmera do dispositivo.

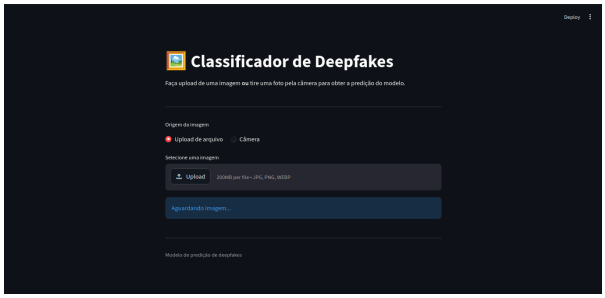

Figura 8: Tela do App Realizado com Streamlit.

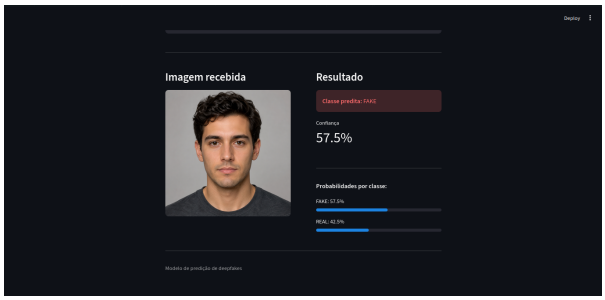

Figura 9: Teste do Modelo no Streamlit Com Imagem de Pessoa Fictícia.

Ao fornecer uma imagem, o sistema retorna a classificação da amostra, a confiança do modelo e as probabilidades por classe (real ou fake), conforme demonstrado na Figura 9.

CONCLUSÃO

Este estudo demonstrou a eficácia do Aprendizado por Transferência (*Transfer Learning*) na detecção de rostos sintéticos. A análise comparativa evidenciou que o modelo baseado na arquitetura EfficientNetV2-S superou a CNN desenvolvida do zero em todas as métricas analisadas, atingindo 93% em acurácia, precisão, recall e F1-Score.

Embora ambos os modelos tenham apresentado oscilações nas curvas de treinamento e sinais de *overfitting*, possivelmente decorrentes do volume do conjunto de dados, o modelo pré-treinado registrou menor taxa de erros, com redução na ocorrência de falsos positivos e falsos negativos. A integração do mecanismo Grad-CAM permitiu verificar que o modelo baseia suas decisões em regiões da face, como olhos, cabelos e boca, e, em alguns casos, no plano de fundo da imagem. O desenvolvimento da interface em Streamlit possibilitou a aplicação do classificador em um contexto de uso direto.

[3]https://streamlit.io/

Como proposta de trabalhos futuros, indica-se a ampliação do conjunto de dados com variações adicionais de imagens e o ajuste de hiperparâmetros para mitigar os efeitos de *overfitting* observados na curva de perda de validação.

USO DE INTELIGÊNCIA ARTIFICIAL

Em conformidade com as diretrizes de integridade científica e boas práticas acadêmicas recomendadas pela CAPES, os autores declaram que utilizaram ferramentas de inteligência artificial, incluindo ChatGPT e Gemini, exclusivamente para auxiliar na melhoria da linguagem e legibilidade deste manuscrito. O uso dessas ferramentas foi realizado sob supervisão humana, e os autores revisaram e editaram cuidadosamente o conteúdo, assumindo total responsabilidade pelo texto final publicado.

REFERÊNCIAS

DENG, L.; SUO, H.; LI, D. Deepfake video detection based on efficientnet-v2 network. *Computational Intelligence and Neuroscience*, Wiley Online Library, v. 2022, n. 1, p. 3441549, 2022.

KARRAS, T. et al. Alias-free generative adversarial networks. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2021. v. 34, p. 852–863.

KARRAS, T.; LAINE, S.; AILA, T. A style-based generator architecture for generative adversarial networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 4401–4410.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.

MANSOOR, N.; ILIEV, A. I. Explainable ai for deepfake detection. *Applied Sciences*, MDPI, v. 15, n. 2, p. 725, 2025.

NGUYEN, T. T. et al. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, Elsevier, v. 223, p. 103525, 2022.

ROSSLER, A. et al. Faceforensics++: Learning to detect manipulated facial images. In: *Proceedings of the IEEE/CVF international conference on computer vision*. [S.l.: s.n.], 2019. p. 1–11.

SELVARAJU, R. R. et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2017. p. 618–626.

TAN, M.; LE, Q. Efficientnetv2: Smaller models and faster training. In: PMLR. *International Conference on Machine Learning*. [S.l.], 2021. p. 10096–10106.

WANG, S.-Y. et al. Cnn-generated images are surprisingly easy to spot... for now. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2020. p. 8695–8704.

YU, N.; DAVIS, L. S.; FRITZ, M. Attributing fake images to gans: Learning and analyzing gan fingerprints. In: *Proceedings of the IEEE/CVF international conference on computer vision*. [S.l.: s.n.], 2019. p. 7556–7566.