Detectando *Deepfakes* em vídeos: Uma abordagem simples e eficiente utilizando Redes Neurais Residuais Profundas

Flavio de Barros Vidal[1], **Christian Cruvinel França**[1], **Carla M. C. Cavalcante Koike**[1]

[1]Department of Computer Science, University of Brasilia - Brazil

fbvidal@unb.br,christian.cruvinel.franca@gmail.com,ckoike@unb.br

Abstract. *The proliferation of deepfakes presents a significant challenge with negative societal impacts. This paper proposes an efficient approaching for the automatic detection of deepfakes using deep learning and residual neural networks (ResNets). A ResNet18 model was trained in a supervised approaching using the Deepfake Detection Challenge (DFDC) dataset. The methodology included image preprocessing and training strategies such as transfer learning and class weight adjustment. Results evaluation showed an accuracy of over 90% for image classification and above 92% for video classification, highlighting the effectiveness of the approach for real-world applications.*

Resumo. *A proliferação de deepfakes representa um desafio significativo com impactos negativos na sociedade. Este trabalho propõe uma abordagem eficiente para a detecção automática de deepfakes utilizando aprendizado profundo e redes neurais residuais (ResNets). Um modelo ResNet18 foi treinado de forma supervisionada com o conjunto de dados Deepfake Detection Challenge (DFDC). A metodologia incluiu pré-processamento de imagens e estratégias de treinamento como transferência de aprendizado e ajuste de pesos de classe. A avaliação demonstrou uma acurácia superior a 90% na classificação de imagens e acima de 92% na classificação de vídeos, indicando a eficácia da abordagem para aplicações reais.*

1. Introdução

A manipulação de vídeos por inteligência artificial, conhecida como *deepfake*, causa diversos impactos negativos, como a disseminação de desinformação, violação de privacidade e difamação. Diante disso, o desenvolvimento de métodos eficientes e robustos para detectar e combater *deepfake* em redes sociais e websites é crucial na era digital. Isso se deve ao potencial significativo dessas tecnologias para disseminar desinformação e comprometer a integridade das comunicações online. *Deepfakes*, conteúdos de áudio ou vídeo manipulados por IA generativa para parecerem autênticos, podem ser usados maliciosamente para enganar indivíduos, danificar reputações, influenciar eleições e facilitar fraudes financeiras. À medida que se tornam mais sofisticados, é fundamental que aplicações web incorporem tecnologias avançadas de detecção para mitigar seus efeitos, protegendo usuários e preservando a confiança nas plataformas digitais[Floridi 2018].

Uma das abordagens mais utilizadas e estudadas na literatura é o emprego de redes neurais residuais (ResNets) para extrair características discriminativas dos vídeos e classificá-los como autênticos ou falsificados [He et al. 2015]. Essa abordagem tem

demonstrado resultados satisfatórios em diversos domínios e cenários, superando outras técnicas baseadas em aprendizado profundo ou aprendizado clássico [Floridi 2018].

Este manuscrito está organizado como: A Seção 2 apresenta brevemente trabalhos relacionados com *deepfakes*, ResNets e sua aplicação para identificação de vídeos manipulados. A metodologia utilizada nesse trabalho é descrita na Seção 3, bem como detalhes da ResNet e dos experimentos realizados. Na Seção 4, os resultados são detalhados e comentados, enquanto a Seção 5 apresenta as conclusões do trabalho e ideias de trabalhos futuros.

2. Trabalhos Relacionados

O termo *deepfake* abrange diversas definições. Adotando a definição proposta por [Dolhansky et al. 2020], *deepfake* refere-se a vídeos nos quais um ou mais rostos foram substituídos por algoritmos de redes neurais. Rostos são de interesse particular devido ao amplo escopo de técnicas em visão computacional dedicadas à reconstrução e rastreamento facial [Zollhöfer et al. 2018]. Além disso, os rostos desempenham um papel crucial na comunicação humana, podendo enfatizar ou transmitir mensagens por si só [Frith 2009]. As técnicas atuais de manipulação facial podem ser categorizadas em manipulação de expressão e identidade [Rössler et al. 2019]. Um exemplo notável de manipulação de expressão é o método *Face2Face* [Thies et al. 2018], que permite a transferência de expressões faciais em tempo real. Outros trabalhos, como [Suwajanakorn et al. 2017], animam faces a partir de sequências de áudio. A manipulação de identidade facial, por outro lado, foca na troca de rostos, popularizada por aplicativos como Snapchat. *deepfakes* também realizam essa troca, mas empregam aprendizado profundo, exigindo treinamento para cada par de vídeos, um processo demorado [Rössler et al. 2019].

A produção de *deepfakes* geralmente se baseia em duas arquiteturas principais: **Redes Adversárias Generativas** (*GANs*) [Goodfellow et al. 2014] e **Autoencoders** [Hinton and Salakhutdinov 2006]. GANs envolvem uma competição entre uma rede geradora e uma discriminadora, visando gerar imagens realistas. Autoencoders, por sua vez, simplificam a dimensionalidade dos dados por meio de um *encoder* e reconstrução via *decoder*. E apesar do avanço na qualidade das técnicas de *deepfakes*, métodos de detecção também evoluíram. Abordagens simples, como detecção de artefatos visuais e inconsistências na pose de cabeça, provaram ser eficazes [Matern et al. 2019, Yang et al. 2018]. Técnicas mais avançadas, utilizando aprendizado profundo, são consideradas o estado da arte [Afchar et al. 2018, Sabir et al. 2019, Dang et al. 2020].

3. Metodologia Proposta

O fluxograma da Figura1 apresenta a estrutura básica da proposta de desenvolvimento, na qual será detalhada nas seções seguintes.

3.1. Elaboração da Base de Dados

A base de dados faz a utilização de um amplo conjunto de dados para o treinamento do modelo, chamada *Deepfake Detection Challenge (DFDC)* [Benpflaum et al. 2019] disponibilizada em dezembro de 2019. O repositório é um dos maiores conjuntos de dados público de vídeos com *deepfakes* com aproximadamente 100 mil vídeos originados de 3426 atores, todos anotados com as informações de vídeos com ou sem *deepfakes*.

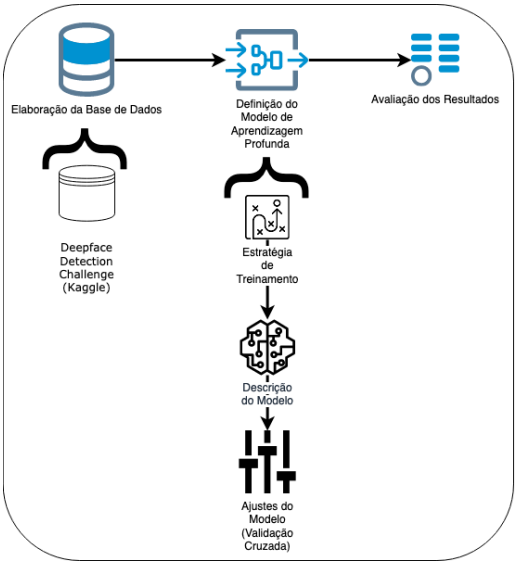


Figura 1. Fluxograma da metodologia proposta.

De maneira aleatória, foram selecionados 80% dos vídeos categorizados como falsos e 80% dos vídeos categorizados como verdadeiros para serem utilizados no treinamento, enquanto 20% dos vídeos categorizados como falsos e 20% dos vídeos categorizados como verdadeiros seriam utilizados para teste. A Tabela 1 apresenta a quantidade de imagens de rostos e cada classe obtidas para cada uma das partições *train* e *test*.

	train		**test**	
Classe	**FAKE**	**REAL**	**FAKE**	**REAL**
Qtd. de Imagens	787.990	164.950	174.662	36.601
% do Total de Imagens	67,7%	14,2%	15,0%	3,1%

Tabela 1. Quantidade de imagens geradas por classe.

3.2. Pré-processamento das Imagens

As imagens, antes de serem convertidas para tensores, foram redimensionadas para o tamanho de 224x224 pixels, para que as imagens possam ser armazenadas diretamente na memória de uma GPU e processadas de forma paralela durante uma iteração do treinamento do modelo. O redimensionamento da imagem foi realizado utilizando um recorte da região central da face. Inicialmente, a imagem é proporcionalmente redimensionada de forma que a menor dimensão dela se torne do tamanho da dimensão desejada. O recorte final é então realizado removendo trechos apenas da maior dimensão de forma que a imagem se torne quadrada.

Após o redimensionamento e o recorte das imagens, estas então são transformadas em tensores com valores no intervalo $[0, 1]$. Este processo é feito dividindo todos os pixels da imagem (valores no intervalo $[0, 255]$) de cada um dos 3 canais (RGB) pelo valor máximo de 255. Uma vez que a rede original foi treinada no conjunto de dados *ImageNet* [Russakovsky et al. 2015], é considerada uma boa prática normalizar as imagens da mesma forma como as imagens no conjunto de dados original foram normalizadas. Desta forma, após a obtenção dos tensores no intervalo de $[0, 1]$, as imagens em um mesmo lote de treinamento foram todas normalizadas utilizando as Equações 1, 2 e 3.

$$Pm_c = \frac{1}{224 \times 224} \sum_{i=1}^{224 \times 224} P_{c,i}, \tag{1}$$

$$Pstd_c = \sqrt{\frac{\sum_{i=1}^{224 \times 224} (P_{c,i} - Pm_c)^2}{224 \times 224}}, \tag{2}$$

$$Pn_{c,i} = \frac{Pn_{c,i} - Pm_c}{Pstd_c}, \tag{3}$$

onde Pm_c é a média dos valores do canal c de um tensor, $Pstd_c$ é o desvio padrão dos valores do canal c de um tensor e $Pn_{c,i}$ é o i-ésimo novo valor do canal c de um tensor. A Tabela 2 apresenta os valores originais de média e desvio padrão utilizados pela *ImageNet*.

Canal	**Vermelho**	**Verde**	**Azul**
Média (Pm_c)	0.485	0.456	0.406
Desvio Padrão ($Pstd_c$)	0.229	0.224	0.225

Tabela 2. Valores de média e desvio padrão originais na *ImageNet*.

3.3. Estratégia de Treinamento da ResNet

Ao final de cada época de treinamento, métricas de entropia cruzada são calculadas para os dados de treinamento e validação [Goodfellow et al. 2016]. Para determinar se um vídeo é *FAKE* ou *REAL*, o modelo classifica cada imagem individual do vídeo. Em seguida, a quantidade de imagens classificadas como falsas é comparada com uma fração (ρ) da quantidade de imagens classificadas como verdadeiras. A classe com a maior quantidade resultante define a classificação do vídeo. O valor de ρ ajusta o nível de conservadorismo da classificação: valores menores que 1.0 tornam a classificação como *FAKE* mais difícil, valores maiores facilitam essa classificação, e 1.0 atribui o mesmo peso a ambas as quantidades. A Figura2 ilustra esse processo.

A função de Perda (*Loss Function*) [Goodfellow et al. 2016] foi balanceada de forma a penalizar o modelo por errar exemplos mais raros do conjunto de dados. A cada época de treinamento completa realizada sobre o conjunto de dados de treinamento, as métricas de entropia cruzada [Goodfellow et al. 2016] foram obtidas tanto para o treinamento quanto para as imagens separadas para validação. Por se tratar de uma classificação binária (*FAKE* ou *REAL*), existem várias possíveis métricas de performance disponíveis para serem utilizadas.

As métricas da área sob a curva Característica de Operação do Receptor (*ROC*) [Fawcett 2006] e do Coeficiente de Correlação de Matthews (*MCC*) [Chicco and Jurman 2020] também são geradas, porém apenas para o conjunto de validação. O modelo foi treinado de forma que a métrica *MCC* fosse minimizada tanto quanto possível. Também foi utilizada uma política específica de treinamento apresentada em [Smith and Topin 2018] na taxa de aprendizado cíclica. A taxa de aprendizado máxima do modelo, considerando a política de taxa de aprendizado cíclica, foi encontrada utilizando o **Teste de faixa da taxa de aprendizado** proposto em [Smith 2017]. Para fins de otimização do processo de treinamento, optou-se em

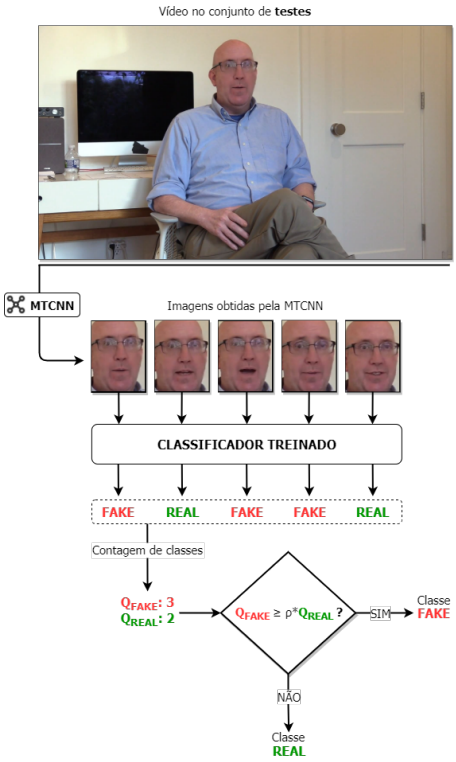


Figura 2. Processo de comparação final realizado para os vídeos do conjunto de testes.

utilizar representação numérica de ponto flutuante de precisão simples (16 bits). Esta técnica utilizada para reduzir o uso de memória das placas gráficas e memória principal das arquiteturas computacionais, sem comprometer sua acurácia.

Empregou-se também uma variante da estratégia de Transferência de Aprendizado (*Transfer Learning* [Goodfellow et al. 2016]), com a abordagem padrão de se substituir os pesos da última camada do modelo classificador, "congelar" [1] as camadas anteriores da rede neural enquanto treina a última camada por uma quantidade de épocas fixa e pré-definida e, após "descongelar" todas as camadas da rede, continuar o treinamento por mais uma quantidade fixa de épocas.

A arquitetura original da ResNet utilizada, apresenta a última camada composta por um perceptron multicamadas cujas saídas, após aplicada a função *softmax*, resultam nas probabilidades das 1000 classes do conjunto de dados *ImageNet*. Todavia, não é possível utilizá-la nessa estrutura original, uma vez que o problema deste trabalho consiste na classificação de apenas duas classes (*FAKE* e *REAL*). Os resultados são avaliados para todas as partições da validação cruzada estratificada. Para cada uma delas, as métricas de validação dos resultados são geradas e as médias e os desvios padrões são calculados, de forma que se obtenha a capacidade preditiva do modelo. Em seguida, o classificador é treinado em todas as imagens do diretório de treinamento *train* e as métricas finais para o modelo são obtidas para o conjunto de teste do diretório *test*.

[1] "Congelar" os parâmetros significa impedir que sejam atualizados pelo algoritmo de gradiente descendente durante o treinamento.

Para avaliar a qualidade do modelo na detecção de *deepfakes* em vídeos, as imagens do conjunto de teste foram segmentadas por título de vídeo e classificadas individualmente. As predições foram contabilizadas e a classificação final do vídeo foi determinada comparando a quantidade de imagens falsas com uma fração ρ das imagens verdadeiras. O coeficiente ρ ajusta a taxa de liberdade: $\rho < 1.0$ torna a decisão mais conservadora, enquanto $\rho > 1.0$ aumenta a flexibilidade. A Figura2 ilustra esse processo.

4. Resultados

Os resultados apresentados nesta seção, incluindo o treinamento e validação do modelo Resnet escolhido (ResNet18 [He et al. 2015]) com o conjunto de dados, foi realizado utilizando as configurações da Tabela 3 de hiperparâmetros durante o treinamento do modelo para todas as partições da validação-cruzada estratificada realizada.

	Valor
Épocas Congeladas	1
Épocas Descongeladas	10
Taxa de Aprendizado Máxima	0.01
Tamanho do Lote	352
Peso da Classe *FAKE*	1.00
Peso da Classe *REAL*	2.7581

Tabela 3. Hiperparâmetros utilizados para o treinamento da Resnet18.

Os pesos das classes foram ajustados na entropia cruzada para penalizar mais os erros na classe *REAL*, devido ao desbalanceamento do conjunto de dados. O peso da classe *REAL* foi calculado como a razão entre o número de exemplos da classe *FAKE* e *REAL*, dividida por 1.73 para maior estabilidade.

O treinamento utilizou a política de taxa de aprendizado de um ciclo, variando de 0.0004 a um valor máximo e reduzindo até 10^{-6}. Além disso, foram implementadas paradas antecipadas baseadas na métrica MCC: interrupção caso não houvesse melhora em 2 épocas e salvamento do modelo ao final de cada época com melhor MCC. As curvas ROC para todas as instâncias da validação cruzada e para o modelo final são apresentadas na Figura 3. A Tabela 4 exibe a média e o desvio padrão das métricas de acurácia, MCC, AUC-ROC, sensibilidade (*tvp*) e especificidade (*tvn*).

	Média	**Desvio Padrão**
Acurácia	96,49%	$\pm$ 0,05%
MCC	0,8782	$\pm$ 0,0018
Área sob a curva ROC	0,9914	$\pm$ 0,0002
tvp	97,75%	$\pm$ 0,09%
tvn	90,56%	$\pm$ 0,13%

Tabela 4. Tabela de médias e desvios padrões de algumas da métricas utilizadas na validação do modelo para as 5 partições da validação cruzada.

Observa-se que, para todas as métricas, o desvio padrão foi relativamente baixo. Isso significa que as partições obtidas pela validação cruzada estratificada são representativas, não existindo em nenhuma delas a forte presença de algum viés. Significa também

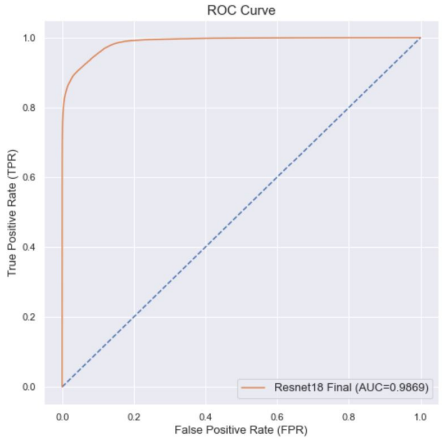


Figura 3. Curva ROC resultante do modelo obtido do conjunto de dados completo.

que o modelo foi capaz de generalizar o que aprendeu no treinamento de forma consistente durante todas as etapas. O modelo, em todos os casos da validação cruzada, foi capaz de manter um valor de MCC acima de 0,87, acertando em média 97,75% dos rostos falsos e 90,56% dos rostos reais. Isso significa que o modelo acerta, aproximadamente, 44 a cada 45 rostos falsos e 9 a cada 10 rostos reais. Além de serem valores relativamente bons, o modelo atingiu um certo resultado desejado de conservadorismo. O modelo acerta a maioria dos rostos falsos às custas de alguns rostos reais.

Observando a área sob a curva ROC (Figura 3) dos 5 modelos gerados pela validação cruzada, temos um valor superior a 0,99 em todos os casos. É importante ressaltar que um modelo com um bom valor para a área sob a curva ROC não necessariamente é um modelo confiante. Um modelo confiante atribui altos valores de probabilidade em suas predições.

Fatores como desfoque de movimento podem ter levado a classificações incorretas, pois o modelo não possui informação temporal para diferenciar distorções naturais de manipulações artificiais. Além disso, algumas técnicas de geração de *deepfakes*, como a MTCNN, podem falhar na substituição de todos os rostos em um vídeo, resultando em imagens sem alteração rotuladas como *FAKE*, pois a classe é definida pelo vídeo original. Também foram identificadas imagens incorretas entre os erros de classificação, incluindo áreas equivocadamente detectadas como rostos devido à imprecisão da MTCNN na extração facial. A tabela 5 abaixo indica as métricas obtidas pela etapa de treinamento no conjunto de dados completo.

	Obtido	**Esperado**	**Diferença**
Acurácia	95,85%	96,49%	-0,64%
MCC	0,8559	0,8782	-0,0223
Área sob a curva ROC	0,9885	0,9914	-0,003
tp_n	97,58%	97,75%	-0,17%
tn_n	87,89%	90,56%	-2,67%

Tabela 5. Tabela de resultados obtidos no conjunto de treinamento e teste final.

Observou-se uma leve degradação nas métricas em relação à validação cruzada.

Possíveis razões incluem:

- **Viés (*bias*)**: O conjunto DFDC contém algumas manipulações vocais além de *deepfakes*, e a amostragem aleatória pode ter concentrado mais desses casos no conjunto de testes. Como o modelo foi treinado apenas em imagens faciais, não possui informação sonora dos vídeos.
- **Hiperparâmetros**: O conjunto de treinamento completo é 25% maior que o usado na validação cruzada, mas os hiperparâmetros, como o tamanho de lote (352 imagens), não foram ajustados, o que pode ter impactado a convergência.

A curva ROC do modelo final (Figura 3) mantém o padrão dos modelos da validação cruzada, com uma leve perda de desempenho observada na compressão da curva. Para avaliar a confiança do modelo, a Figura 4 apresenta as estimativas de densidade por *kernel* das probabilidades atribuídas às classes *FAKE* e *REAL*.

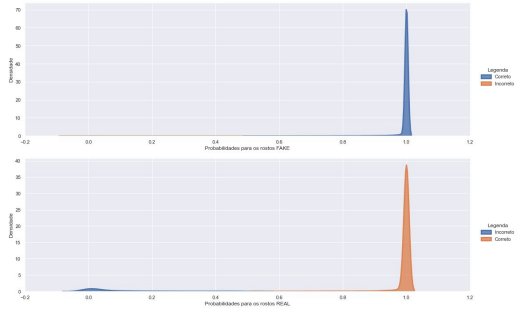


Figura 4. Estimativas de densidade por kernel gaussiano das distribuições das probabilidades resultantes do modelo para as classes *REAL* e *FAKE*.

Pela distribuição, é possível observar que o modelo é significativamente confiante em suas predições. O elevado pico no valor de probabilidade próximo de 1 indica que a maioria das imagens foram preditas com probabilidade quase máxima para suas respectivas classes corretas. Observa-se que, no caso das classes de rostos reais, existe um pequeno pico na região de 0. A Figura 5 abaixo apresenta os mesmos gráficos da Figura 4 anterior porém ampliada no intervalo de 0 a 1 no eixo de densidade.

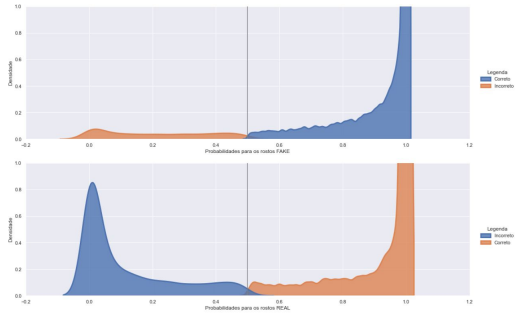


Figura 5. Estimativas de densidade por kernel gaussiano das distribuições das probabilidades resultantes do modelo para as classes *REAL* e *FAKE* com ampliação no intervalo de 0 a 1 no eixo de densidade.

Observa-se que, quando o modelo erra a classe *REAL*, uma parcela significativa dos erros é predita com extrema confiança para a classe oposta. Isso pode ser observado

pelo pico em 0 do gráfico inferior. Isso significa que existem imagens reais no conjunto de dados de teste que o modelo apresentou elevada confiança de que são falsas. Essas imagens podem estar no espectro daquelas discutidas anteriormente, isto é, imagens em desfoque de movimento ou imagens ruidosas (mãos, pescoços) que foram extraídas de um vídeo cuja classe era *REAL* e que, do ponto de vista de imagens estáticas, aparentam terem sofrido manipulações.

A predição por vídeo segue o processo descrito na Figura 2, onde a classe que um vídeo recebe é uma função da quantidade de imagens preditas falsas e reais. A Equação 4 abaixo explicita isso:

$$\text{Classe do vídeo} = \begin{cases} FAKE, & \text{se } Q_{FAKE} \geq \rho \times Q_{REAL} \\ REAL, & \text{caso contrário} \end{cases}, \tag{4}$$

onde Q_{FAKE} é a quantidade de imagens preditas *FAKE* e Q_{REAL} a quantidade de imagens preditas *REAL* para um vídeo específico.

Com as imagens agrupadas por vídeo, realizou-se a inferência do mesmo modelo obtido do conjunto de treinamento completo nos agrupamentos de imagens de cada vídeo, e para cada um deles a quantidade de cada classe obtida foi contabilizada. São ao total 17.606 vídeos resultantes no diretório *test*. A Figura 6 apresenta as matrizes de confusão original e normalizada resultantes desse processo utilizando $\rho = 1$.

(a) (b)

Figura 6. Matrizes de confusão (a) absoluta e (b) normalizada resultantes para a inferência do modelo nos agrupamentos de imagens de teste. Resultados para $\rho = 1$.

Temos que, dos 14.537 vídeos falsos, o modelo errou apenas 72, e dos 3.069 vídeos reais, o modelo errou 218, representando uma taxa de acerto de mais de 92% em ambos os casos. Era de se esperar um aumento nesse caso pois utilizando o processo descrito na Figura 2 estamos fornecendo um meio para que erros de imagens individuais possam ser suprimidos por uma taxa geral maior de acertos das outras imagens de um mesmo vídeo.

A fim de se entender o impacto do coeficiente ρ nessas métricas, o valor de ρ foi variado no intervalo $[0, 25, 4]$ com passadas de $0, 25$, resultando em uma avaliação para 16 diferentes valores de ρ. A Figura 7 a seguir apresenta o resultado desses 16 diferentes valores de ρ versus a taxa de falsa rejeição (*TFR*) e a taxa de falsa aceitação (*TFA*) de um mesmo modelo no conjunto de imagens agrupadas por vídeo do diretório *test*. Neste caso, a taxa de falsa rejeição simboliza a parcela de rostos reais que foram incorretamente

classificados como falsos e a taxa de falsa aceitação simboliza a parcela de rostos falsos que foram incorretamente classificados como reais.

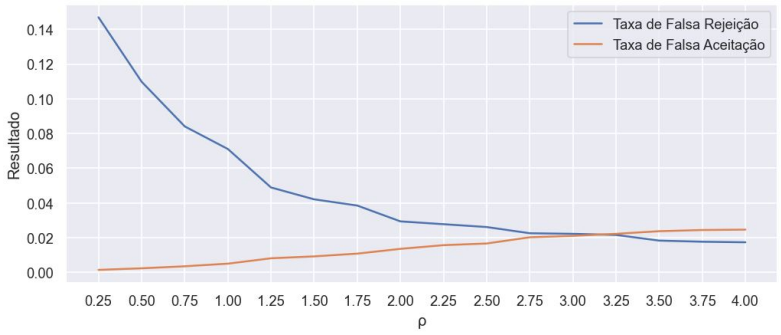
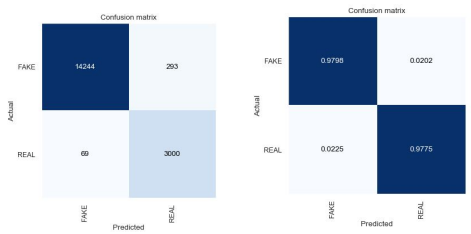


Figura 7. Diferentes valores de ρ versus as taxas de verdadeiro positivo e verdadeiro negativo.

Observa-se um comportamento interessante. Valores maiores de ρ não comprometem significativamente a taxa de falsa aceitação porém diminuem consideravelmente a taxa de falsa rejeição. Eventualmente, existe um valor para ρ que a taxa de falsa aceitação, inicialmente menor, supera o valor da taxa de falsa rejeição. Esse ponto de cruzamento é onde ocorre uma Taxa de Erro Igual ou EER (*Equal Error Rate*) do modelo. A Figura 8 a seguir apresenta as matrizes de confusão original e normalizada para o valor de $\rho = 2,75$ que, de acordo com a Figura 7, é visualmente um local onde as taxas de falsa aceitação e falsa rejeição estão próximas.



(a) (b)

Figura 8. Matrizes de confusão (a) absoluta e (b) normalizada resultantes para a inferência do modelo nos agrupamentos de imagens de teste. Resultados para $\rho = 2,75$.

Embora a proposta do modelo seja o conservadorismo, a perda ínfima de performance na taxa de verdadeiros positivos possa ser justificada pelo aumento de produtividade no acerto de vídeos reais com maior frequência pelo modelo. Em outras ocasiões o modelo talvez não necessite ser tão conservador e a melhor proposta passa a ser o melhor equilíbrio entre as partes. Para o valor de $\rho = 2,75$, temos um modelo que acerta ambas as classes com mais de 97% de chances.

Um valor de $\rho = 2,75$ significa que para um vídeo ser considerado *FAKE* é necessário que a quantidade de imagens preditas como *REAL* nele seja no mínimo $2,75$ vezes menor que a quantidade de imagens falsas. Para um vídeo de 10 segundos, com amostragem de 30 em 30 frames (aproximadamente de 1 em 1 segundo), serão aproximadamente 10 imagens por vídeo. Dessas 10 imagens, a faixa de valor possível para que o

vídeo seja considerado *REAL* é $Q_{REAL} \geq 3$, ou seja, 3 imagens ou mais preditas como *REAL* já são suficientes para que o vídeo seja considerado *REAL*. A rápida queda na taxa de falsa rejeição sem um forte prejuízo na taxa de falsa aceitação significa que, no geral, o peso por encontrar rostos reais é maior do que encontrar rostos falsos. Logo, encontrar rostos reais por si só é um forte indício de que o vídeo possa provavelmente ser real e não um *deepfake* (considerando 10 imagens por vídeo como é aproximadamente o caso para o conjunto de dados utilizado).

É importante destacar que para a utilização do método de averiguação utilizado em um vídeo exterior ao conjunto de dados é necessário que apenas o trecho do vídeo o qual se acredita ser um *deepfake* deve ser fornecido ao modelo. Uma vez que o processo de detecção é realizado em toda a extensão do vídeo, inserir trechos do vídeo que contenham rostos reais (supondo o conhecimento prévio) pode criar uma quantidade elevada de detecções de classes reais, o que pode levar a quantidade de detecções de classe *FAKE* ser negligenciada completamente.

5. Conclusões

Este trabalho explorou a detecção de *deepfakes* em vídeos com redes neurais convolucionais profundas, demonstrando ser uma solução viável para o problema e conjunto de dados utilizados.

A estratégia de treinamento baseada em transferência de aprendizado, com precisão e valores elevados para tamanho de lote e taxa de aprendizado, combinada com a política de um ciclo, mostrou-se eficiente. O modelo ResNet18 apresentou desempenho satisfatório, com acurácia média de $96,49\%$ e Coeficiente de Correlação de Matthews médio de $0,8782$, indicando bom equilíbrio entre verdadeiros positivos e negativos. A validação cruzada resultou em uma AUC-ROC média de $0,9914$, evidenciando excelente separação de classes.

Nos testes finais, a abordagem de considerar todas as imagens faciais de um vídeo para classificá-lo produziu ótimos resultados. O coeficiente $\rho \approx 2,75$ priorizou rostos preditos como reais, equilibrando as taxas de erro entre classes e atingindo mais de 97% de acerto por classe. Assim, a estratégia classifica corretamente cerca de 49 a cada 50 vídeos.

Referências

Afchar, D., Nozick, V., Yamagishi, J., and Echizen, I. (2018). Mesonet: a compact facial video forgery detection network.

Benpflaum, G, B., djdj, Kofman, I., Tester, J., JLElliott, Metherd, J., Elliott, J., Mozaic, Culliton, P., Dane, S., and Kim, W. (2019). Deepfake detection challenge. `https://kaggle.com/competitions/deepfake-detection-challenge`. Kaggle.

Chicco, D. and Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21.

Dang, H., Liu, F., Stehouwer, J., Liu, X., and Jain, A. (2020). On the detection of digital face manipulation.

Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., and Canton-Ferrer, C. (2020). The deepfake detection challenge dataset. *ArXiv*, abs/2006.07397.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874. ROC Analysis in Pattern Recognition.

Floridi, L. (2018). Artificial intelligence, deepfakes and a future of ectypes. *Philosophy and Technology*, 31.

Frith, C. (2009). Role of facial expressions in social interactions. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364:3453–8.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Matern, F., Riess, C., and Stamminger, M. (2019). Exploiting visual artifacts to expose deepfakes and face manipulations. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 83–92.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

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

Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., and Natarajan, P. (2019). Recurrent convolutional strategies for face manipulation detection in videos.

Smith, L. N. (2017). Cyclical learning rates for training neural networks.

Smith, L. N. and Topin, N. (2018). Super-convergence: Very fast training of neural networks using large learning rates.

Suwajanakorn, S., Seitz, S. M., and Kemelmacher-Shlizerman, I. (2017). Synthesizing obama: Learning lip sync from audio. *ACM Trans. Graph.*, 36(4).

Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., and Nießner, M. (2018). Face2face: Real-time face capture and reenactment of rgb videos. *Commun. ACM*, 62(1):96–104.

Yang, X., Li, Y., and Lyu, S. (2018). Exposing deep fakes using inconsistent head poses.

Zollhöfer, M., Thies, J., Garrido, P., Bradley, D., Beeler, T., Pérez, P., Stamminger, M., Nießner, M., and Theobalt, C. (2018). State of the art on monocular 3d face reconstruction, tracking, and applications. *Computer Graphics Forum*, 37.