Associando Redes Convolucionais e Características de Iluminação para Detectar Falsificações em Imagens

Thales Pomari†, Edmar Rezende*, Guilherme Ruppert*, Fernanda Balieiro† e Tiago Carvalho†
*CTI Renato Archer, Campinas-SP, Brasil 13069-901
†Instituto Federal de São Paulo (IFSP), Campinas-SP, Brasil 13069-901

***Abstract*—In the last years, fake news wave generated a lot of problems, and most of times, fake images are associated with these news. Looking for new ways to fight this problem back, this work presents a comparative study among different CNNs architecutres which are combined with illumination maps to perform image splicing detection. Investigating CNNs trained from scratch and also transfer learning process, this work presents results comparable to state-of-the-art methods.**

***Resumo*—Nos últimos anos, a onda de *fake news* trouxe grandes problemas para o mundo e, em boa parte das vezes, imagens falsas são associadas a tais notícias. Buscando por formas de combater este tipo de prática, este trabalho apresenta um estudo comparativo entre diferentes arquiteturas de CNNs associadas a características de iluminação, em uma metodologia aplicada ao problema de detecção de imagens falsas. Investigando CNNs treinadas a partir do zero e também o processo de *transfer learning*, este trabalho apresenta resultados comparados ao estado da arte.**

I. Introdução

Nos últimos anos, a onda de *fake news* trouxe grandes problemas para o mundo. Um exemplo ocorreu em fevereiro de 2018, quando um atentado na instituição de ensino *Marjory Stoneman Douglas High School* em Parklando, Florida-USA, resultou em dezessete pessoas mortas e outras dezessete feridas. Após este incidente, Emma Gonzalez, sobrevivente do ataque, realizou um discurso sobre o controle de armas que ganhou grande atenção da população mundial. Entretanto, poucas semanas após o ocorrido, diversas imagens falsas, como a mostrada na Figura 1, começaram a circular pelas redes sociais. Nelas, Emma aparece rasgando a constituição americana[1].

Fig. 1. Imagem falsa (à esquerda) e imagem verdadeira (à direita).

Acontecimentos como este trazem com efetividade uma grande desinformação para aqueles que não se preocupam com a fonte antes de absorver a notícia e, consequentemente, disseminam informações falsas podendo influenciar, de diferentes formas, outras pessoas.

Tais fatos ilustram o grande problema das chamadas *fake news*, e mostram o quão valiosos são os métodos capazes de trazer maior credibilidade para as notícias. Em especial, métodos que garantam a veracidade de imagens que possam vir a estampar capas de jornais ou revistas de grande alcance são extremamente necessários.

[1]https://www.telegraph.co.uk/news/2018/03/26/fake-images-parkland-shooting-survivor-emma-gonzalez-tearing/

No campo de Forense Digital, métodos baseados em redes convolucionais com a função de checar a veracidade de uma imagem são uma grande vertente, mostrando resultados expressivos em diferentes trabalhos [1], [2].

Redes neurais convolucionais (CNN) tiveram seu início com *LeCun et.al.* [3], [4], em um trabalho onde a principal ideia era criar um modo de detectar algarismos escritos a mão, classificando padrões de alta complexidade com um mínimo de pré-processamento.

Inspirado no trabalho de *Carvalho et.al.* [5], que utiliza características de iluminação e descritores de imagens para realizar a detecção de imagens falsificadas, e pelos recentes avanços nas CNNs, este trabalho realiza uma análise comparativa em três bases de dados públicas, entre diferentes arquiteturas de CNNs associadas a características de iluminação, quando aplicadas ao problema de detecção de falsificações em imagens.

De maneira geral, as três principais contribuições deste trabalho são: (1) uma avaliação comparativa entre diferentes arquiteturas de CNNs aplicadas ao problema de identificação de falsificações quando associadas a características de imagem; (2) uma avaliação comparativa entre diferentes arquiteturas de CNNs aplicadas ao problema de identificação de falsificações quando associadas a características de imagem e ao processo de *transfer learning*; (3) uma comparação entre as abordagens com e sem *transfer learning*.

O restante deste trabalho está organizado da seguinte forma: a Seção II descreve de maneira breve o conceito das arquiteturas de CNNs utilizadas neste trabalho. A Seção III descreve os principais resultados obtidos ao longo do trabalho. Por fim, a Seção IV apresenta as conclusões e direções para trabalhos futuros.

II. Método Proposto

O método proposto neste trabalho consiste em três etapas principais, conforme mostrado na Figura 2. A primeira etapa compreende o pré-processamento das imagens, que consiste em convertê-las para espaços de cores que buscam realçar características de iluminação (e expor inconsistências), seguido por um redimensionamento (de modo a adequar o tamanho da imagem para a entrada da etapa dois). A etapa dois consiste em utilizar uma arquitetura de rede convolucional para a extração das características das imagens. Por fim, a etapa três consiste na utilização das características extraídas na etapa dois para a classificação da imagem quanto a sua autenticidade. O principal objetivo desta seção é descrever as redes neurais que serão utilizadas e explicar a abordagem dos espaços de cores.

A. Espaços de Cores

Na etapa de pré-processamento, a proposta do método é converter as imagens para um espaço de cores no qual inconsistências na iluminação das imagens sejam realçadas. Além do espaço de cor RGB (no qual as imagens são originalmente representadas), são utilizados mais dois espaços de cor: o *Generalized Grayworld Estimates algoritmo* (GGE) [6] e o *Inverse Intensity Chromaticity Space* (IIC) [7].

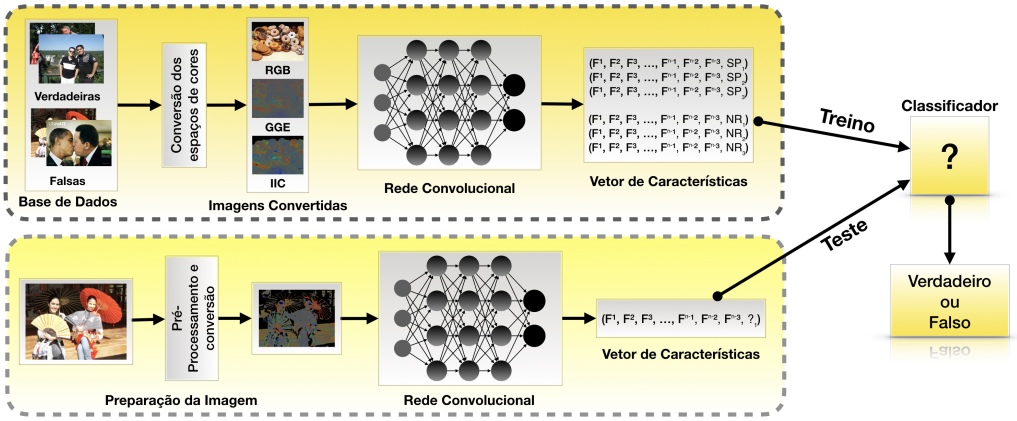

Fig. 2. Visão geral do método proposto.

Com esses três espaços de cores, será possível realçar alguns tipos de características que podem ser relevantes para a classificação da imagem, impulsionando a acurácia das redes convolucionais.

B. Redes Neurais Convolucionais

O trabalho proposto avalia a efetividade de diferentes arquiteturas de redes convolucionais para a tarefa de extração de características e classificação dos padrões. Esta seção descreve de maneira breve as características das principais arquiteturas avaliadas ao longo do trabalho.

1) VGG Net: A VGG Net, criada pelo Visual Geometry Group, consiste em uma versão melhorada da arquitetura conhecida como AlexNet [8], sendo suas principais variações a arquitetura VGG16 (com um total de 16 camadas) e a VGG19 (com um total de 19 camadas). As duas são compostas por seis blocos com diversas camadas, sendo os primeiros blocos compostos por uma combinação de camadas de convolução e o sexto sendo um bloco inteiramente conectado (FC). Na arquitetura da VGG16, o primeiro bloco de convolução possui duas camadas com 64 neurônios, os quais atuam como filtros convolucionais, de 3×3, sempre seguidos por uma camada de *max pooling* 2×2. Os demais blocos (2, 3, 4 e 5) seguem uma composição similar, mas aumentando o número de filtros por bloco (128, 256, 512, 512). O sexto bloco possui três camadas totalmente conectadas, em que as duas primeiras possuem 4096 neurônios e a terceira é uma camada de saída com 1000 neurônios com *softmax activation*. A diferença entre as arquiteturas VGG16 e VGG19 se resume a três camadas adicionais existentes na VGG19, sendo uma camada a mais no quarto, uma no quinto e uma no sexto bloco.

2) ResNet: Redes neurais residuais [9] são um tipo de rede neural convolucional desenvolvidas pelos pesquisadores da Microsoft Research. O principal aspecto que a torna diferente das demais arquiteturas de redes convolucionais é a aplicação do conceito de bloco residual, o qual usa atalhos entre as camadas, adicionando os valores inicias de entrada da camada à função ReLU [10] de saída, como mostrado na Equação 1:

$$y = F(x, Wi) + x. \tag{1}$$

sendo $F(x, Wi)$ a função que representa o mapa residual aprendido pelo bloco e x a imagem inicial.

3) Inception: A principal ideia por trás dos módulos da arquitetura da rede Inception [11] é o uso de diferentes tamanhos de filtros em um mesmo bloco, seguidos por uma camada de *pooling* ao fim do bloco, a qual visa concatenar a saída de todos os filtros no final. Diferente das demais, nesse tipo de abordagem, cada módulo pode ser comparado com uma pequena CNN sem o topo. Uma ideia interessante desta abordagem é o uso de uma camada 1×1 de convolução para reduzir o mapa de característica de entrada tornando convoluções de larga escala mais baratas em relação ao custo computacional necessário.

4) Xception: A arquitetura Xception é a combinação da ideia do módulo da Inception [11] e do conceito de *depthwise separable convolutions*. Dada uma imagem de entrada de $I_{w \times h \times d}$, onde w representa o comprimento da imagem, h representa a altura e d representa o número de canais, as *depthwise separable convolutions* realizam a etapa de convolução em dois passos (ao invés de um único como nas demais arquiteturas). No primeiro passo, é realizada a convolução de um *kernel* de $n \times n \times 1$, onde n representa o tamanho do kernel, com todos os canais de I individualmente. Logo após, no segundo passo da convolução, os resultados do passo 1 são combinados utilizando um novo kernel de $1 \times 1 \times d$. Isso melhora o desempenho do processamento das convoluções, possibilitando que a rede faça menos operações ao longo do treinamento.

5) DenseNet: DenseNets [12] são um tipo de arquitetura em que todas as camadas estão diretamente conectadas com as demais. De forma simplificada, o processo de funcionamento pode ser comparado com uma cascata, em que todas as saídas são adicionadas ao cálculo da função ReLU da próxima camada convolucional. O ato de se combinar todas as camadas acontece em uma camada de transição, que comprime o resultado de todas as camadas do bloco em uma única saída, a qual servirá de entrada para o próximo bloco.

Neste tipo de arquitetura existem duas variações, uma normal e uma que possui uma camada convolucional 1×1 que comprime o número de canais em cada camada de transição em 0.5. O bloco FC é composto por um filtro de 7×7 com *global average polling*, e originalmente uma função *softmax*.

6) MobileNet: Esta arquitetura foi proposta por pesquisadores do Google e foi desenvolvida para estar presente em aplicações móveis, tendo como principal objetivo a eficiência. Também é baseada no conceito de *depthwise separable convolutions*, o que torna a arquitetura mais leve, reduzindo a necessidade de um alto poder computacional.

C. Transfer Learning

Olivas et.al. [13] descreve o conceito de *transfer learning* como a reutilização dos pesos de uma rede treinada utilizando uma grande base de dados para um problema com escopo similar, como o reconhecimento de padrões em imagens. A etapa de ajuste dos pesos de uma rede neural é um processo com um grande custo computacional, portanto torna-se interessante a reutilização dos pesos.

D. Classificador

Para classificar os vetores de características extraídos da imagem utilizando as arquiteturas apresentadas na Seção II-B, utilizamos dois classificadores de padrões: *Softmax* [14] e *Support Vector Machine* (SVM) [15] com kernel linear.

III. Experimentos e Resultados

Para a validação do método proposto, foram utilizadas diferentes bases de dados públicas e diferentes formas de extração de características (treinando as arquiteturas do zero e utilizando os pesos treinados na base de dados ImageNet via o processo de *transfer learning*) como descrito nas seções a seguir.

A. Base de dados

Neste trabalho utilizamos três bases de dados públicas: a base DSO, DSI proposta por *Carvalho et.al.* [5] e a base de dados Columbia criada por Hsu e Chang [16].

A base de dados DSO é composta por 200 imagens, em que metade delas são originais (não adulteradas) e a outra metade são composições de imagens, todas elas com uma resolução de 2048×1536 pixeis. A base de dados DSI é composta por 50 imagens, sendo metade delas não adulteradas e metade proveniente de composições, todas baixadas da internet com diferentes resoluções. A base Columbia é composta por 60 imagens, sendo metade falsa e a outra metade verdadeira e sua resolução varia de 757×568 até 1152×768.

B. Ambiente de Testes

O ambiente em que os testes foram realizados é composto por um Intel®Xeon®E5-2620, 96 GB de memória RAM, duas placas de vídeo NVIDIA Titan XP, usando o sistema operacional Ubuntu 16.04 LTS. Todos os códigos foram desenvolvidos em Python 3.5, usando Keras 2.0.3[2], e TensorFlow 1.0.1[3].

C. Protocolo de validação e Métrica de Desempenho

A validação dos experimentos deste trabalho utilizou o protocolo de validação *5-fold cross validation* e a medida de acurácia como métrica de desempenho, sendo apresentados os resultados médios de cada cenário avaliado.

D. Cenário #1: Inicialização Aleatória dos Pesos

O primeiro conjunto de experimentos realizados avalia a performance do método proposto utilizando diferentes CNNs para realizar a extração de característica. Partindo de uma inicialização aleatória dos pesos, as redes para extração das características foram treinadas por 100 épocas para encontrar os pesos utilizados na extração. Para a classificação foi utilizado o classificador *softmax* com 2 classes. As tabelas I, II III apresentam a acurácia média para os espaços de cores IIC, GGE e RGB, respectivamente.

Para a base de dados DSO, a arquitetura Inception V3 associada ao espaço de cor IIC apresentou o melhor resultado de acurácia com $92,5\%$. Para a base de dados DSI, tanto a arquitetura Xception utilizando as imagens RGB (sem transformação) quanto a arquitetura Inception V3 em conjunto com o espaço IIC apresentaram o resultado de $86,0\%$. Por fim, a base de dados Columbia, teve o melhor resultado ao utilizarmos as arquiteturas InceptionResNetV2 e DenseNet com acurácia de $95,0\%$. As arquiteturas VGG não apresentaram resultados por não convergirem ao longo do treinamento.

[2]https://keras.io

[3]https://www.tensorflow.org

TABLE I

IIC			
Rede	**DSO**	**DSI**	**Columbia**
VGG16	-	-	-
VGG19	-	-	-
ResNet50	90,5%	84%	71,67%
InceptionV3	**92,5%**	82%	**80%**
InceptionResNetV2	90%	**86%**	78,33%
Xception	84,5%	82%	73,33%
DenseNet	90,5%	78%	70%
NASNetMobile	63%	68%	55%
MobileNet	91,5%	84%	**80%**

TABLE II

GGE			
Rede	**DSO**	**DSI**	**Columbia**
VGG16	-	-	-
VGG19	-	-	-
ResNet50	55,5%	50%	85%
InceptionV3	62,5%	46%	76,67%
InceptionResNetV2	66%	58%	85%
Xception	**92%**	**80%**	78,33%
DenseNet	58,5%	50%	**90%**
NASNetMobile	51,5%	54%	60%
MobileNet	61%	62%	86,67%

E. Experimentos 2: Inicialização dos Pesos por Transfer Learning

Nesta outra bateria de experimentos, não foi realizado o treinamento das CNNs para a extração das características. Ao invés de treinar cada uma das arquiteturas, utilizamos o processo de *transfer learning*, transferindo os pesos das arquiteturas previamente treinadas utilizando a base de dados do projeto ImageNet[4]. As tabelas IV, V VI apresentam a acurácia média para os espaços de cores IIC, GGE e RGB, respectivamente. Utilizando a abordagem *transfer learning*, sem o treinamento da rede, o melhor resultado para a base de dados DSO, foi obtido utilizando a arquitetura VGG19 associada ao espaço de cor IIC apresentou o melhor resultado de acurácia com 94%. Para a base de dados DSI, tanto a arquitetura VGG16 quanto a arquitetura InceptionV3 em conjunto com o espaço IIC apresentaram o resultado de $86,0\%$. Por fim, para a base de dados Columbia, o melhor resultado foi apresentado ao utilizarmos a arquitetura VGG16 em conjunto com as imagens RGB (sem transformação), sendo a acurácia de $82,0\%$.

IV. Conclusões e Trabalhos Futuros

Ao longo deste trabalho apresentamos um método para detectar falsificações do tipo *splicing* em imagens. Após converter uma imagem para um espaço de cores transformado que realça inconsistências de iluminação, são extraídas características da imagem utilizando uma CNN. Por fim as características são utilizadas para o treinamento de um classificador que determina a autenticidade da imagem.

Foram avaliados 3 tipos diferentes de espaços de cor, dois próprios para realçar característica de iluminação e o espaço de cores RGB, bem como diferentes arquiteturas de CNNs para a extração de características.

Avaliando o método proposto em 3 bases de dados públicas diferentes, os resultados obtidos foram de uma acurácia média de 94% para a base de dados DSO, 86% para a base de dados DSI e 95% para a base de dados Columbia, mostrando também a relevância da utilização de outros espaços de cores diferentes do espaço RGB.

[4]https://www.image-net.org/

TABLE III

RGB			
Rede	**DSO**	**DSI**	**Columbia**
VGG16	-	-	-
VGG19	-	-	-
ResNet50	50,5%	40%	85%
InceptionV3	54,5%	48%	90%
InceptionResNetV2	58%	44%	**95%**
Xception	**91%**	**86%**	75%
DenseNet	58,5%	56%	**95%**
NASNetMobile	54%	46%	58,33%
MobileNet	51,5%	40%	83,33%

TABLE IV

IIC			
Rede	**DSO**	**DSI**	**Columbia**
VGG16	93%	**86%**	71%
VGG19	**94%**	84%	74%
ResNet50	91%	80%	64%
InceptionV3	81%	**86%**	56%
InceptionResNetV2	86%	84%	59%
Xception	90%	84%	67%
DenseNet	90%	80%	**77%**
NasNetMobile	78%	74%	53%
MobileNet	92%	82%	74%

TABLE V

GGE			
Rede	**DSO**	**DSI**	**Columbia**
VGG16	52%	60%	**80%**
VGG19	59%	**62%**	76%
ResNet50	51%	48%	77%
InceptionV3	57%	56%	74%
InceptionResNetV2	49%	50%	63%
Xception	62%	60%	71%
DenseNet	**65%**	56%	81%
NasNetMobile	55%	58%	61%
MobileNet	56%	54%	73%

TABLE VI

RGB			
Rede	**DSO**	**DSI**	**Columbia**
VGG16	**57%**	62%	**82%**
VGG19	55%	**66%**	79%
ResNet50	55%	56%	71%
InceptionV3	56%	52%	69%
InceptionResNetV2	56%	50%	56%
Xception	49%	64%	69%
DenseNet	55%	60%	73%
NasNetMobile	57%	64%	59%
MobileNet	48%	**66%**	78%

Como direções para trabalhos futuros, podemos citar a combinação das características provenientes de diferentes espaços de cor, bem como a combinação de características extraídas por diferentes arquiteturas de CNNs, bem como a realização de testes em bases de dados contendo um maior número de imagens.

AGRADECIMENTOS

Gostaríamos de agradecer o apoio financeiro do projeto CAPES DeepEyes, a Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP (processos 2017/12631-6, 2018/00858-9), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (processo 423797/2016-6), e a NVIDIA pela doação das GPUs TITAN XP utilizadas nesta pesquisa.

REFERENCES

[1] E. R. de Rezende, G. C. Ruppert, A. Theóphilo, E. K. Tokuda, and T. Carvalho, "Exposing computer generated images by using deep convolutional neural networks," *Signal Processing: Image Communication*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596518303205

[2] T. Pomari, G. Ruppert, E. Rezende, A. Rocha, and T. Carvalho, "Image splicing detection through illumination inconsistencies and deep learning," in *IEEE International Conference on Image Processing*, 2018 - To appear.

[3] Y. L. Cun and Y. Bengio, "Word-level training of a handwritten word recognizer based on convolutional neural networks," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3 - Conference C: Signal Processing (Cat. No.94CH3440-5)*, vol. 2, Oct 1994, pp. 88–92 vol.2.

[4] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, *Object Recognition with Gradient-Based Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 319–345. [Online]. Available: https://doi.org/10.1007/3-540-46805-6_19

[5] T. J. d. Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. d. R. Rocha, "Exposing digital image forgeries by illumination color classification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1182–1194, July 2013.

[6] J. van de Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2207–2214, Sept 2007.

[7] C. Riess and E. Angelopoulou, "Scene illumination as an indicator of image manipulation," in *Information Hiding*, R. Böhme, P. W. L. Fong, and R. Safavi-Naini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 66–80.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[10] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. USA: Omnipress, 2010, pp. 807–814. [Online]. Available: http://dl.acm.org/citation.cfm?id=3104322.3104425

[11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[12] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[13] E. S. Olivas, J. D. M. Guerrero, M. M. Sober, J. R. M. Benedito, and A. J. S. Lopez, *Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques - 2 Volumes*. Hershey, PA: Information Science Reference - Imprint of: IGI Publishing, 2009.

[14] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*, F. F. Soulié and J. Hérault, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 227–236.

[15] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, July 1998.

[16] Y.-F. Hsu and S.-F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," in *International Conference on Multimedia and Expo*, 2006.