Mitigando os Efeitos de GAN em Classificação de Imagens com CNN

Jackson Mallmann[1,2], Altair Santin[2], Alceu Britto[2], Roger Santos[2]

[1]Instituto Federal Catarinense – Jardim Maluche
88.354-300 - Brusque - SC

[2]Programa de Pós-Graduação em Informática (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR)
80.215-901 - Curitiba - PR

jackson.mallmann@ifc.edu.br, altair.santin@pucpr.br,
alceu@ppgia.pucpr.br, robson.roger@ppgia.pucpr.br

Abstract. *CNN (Convolutional Neural Network) has been frequently used for troubleshooting, generating a model that can predict an image class. In this work, the absence of integrity in CNN is verified by using a GAN (Generative Adversarial Network). For this purpose, we model an authenticity classifier based on the algorithm NB (Naive Bayes). When the proposed NB and the CNN models work together, 88.88% of accuracy was reached, while 89.88% of the fake images were identified and discarded. In the specific case of CNN, an accuracy of 85.06% was obtained with a confidence of 95%.*

Resumo. *A CNN (Convolutional Neural Network) tem sido frequentemente usada para solução de problemas, gerando um modelo que pode prever a classe da imagem. Neste trabalho, a ausência de integridade na CNN é verificada usando uma GAN (Generative Adversarial Network). Para isso, modelamos um classificador de autenticidade baseado no algoritmo NB (Naive Bayes). Quando os modelos NB e CNN propostos trabalham juntos, 88,88% de acerto foram alcançados. Em 89,88% dos casos as imagens fakes foram identificadas e descartadas. No caso específico da CNN, obteve-se uma precisão de 85,06% com uma confiança de 95%.*

1. Introdução

Os atuais trabalhos acadêmicos que visam a detecção de imagem PI (Pornografia Infantil) com frequência baseiam-se no uso de CNN (*Convolutional Neural Network*): rede que gera modelo via treinamento. O modelo tem o objetivo de realizar classificações: PI e NPI (Não Pornografia Infantil), por exemplo.

Entretanto, a CNN pode classificar uma imagem *fake* como se fosse imagem real, produzindo o mesmo resultado. Uma imagem *fake* é a imitação da imagem real. Por exemplo, uma imagem com características de PI pode ser classificada erroneamente por CNN desde que essa imagem sofra interferência de outra imagem gerando "pequenas alterações", i.e., se gere uma imagem *fake*.

A possibilidade de alteração do resultado da CNN destaca fragilidade, já que existem usuários mal-intencionados que podem tirar proveito da situação. Estudos

recentes da literatura, mostram ataques em CNN como: *poisoning attacks*, *privacy-aware learning* e *evasion attacks* [Norton e Qi 2017].

O contexto PI foi escolhido por sua importância e pela não existência de base de PI pública. Ressalta-se que este trabalho não utilizou imagens PI para treinamento da CNN. Para se ter ideia, a *SaferNet*[1], no ano de 2018 registrou 57.816 denúncias envolvendo PI, distribuídas em 23.627 URLs (*Uniform Resource Locator*).

No intuito de solucionar este problema, apresenta-se um classificador de autenticidade que deve ser utilizado em conjunto com uma CNN, assim quando uma imagem for submetida a classificação da CNN, primeiro deve-se verificar sua autenticidade.

O classificador de autenticidade se baseia no algoritmo NB (Naive Bayes), neste caso, e objetiva detectar imagens *fakes*, sendo treinado por imagens geradas por uma GAN (*Generative Adversarial Network*).

O artigo está organizado da seguinte forma. A Seção 2 descreve os conceitos para a detecção de imagem PI. A Seção 3 discute os trabalhos relacionados. Na Seção 4 apresenta-se a proposta. Na Seção 5 são abordados os aspectos técnicos que proporcionam a reprodutibilidade do trabalho, assim como a base de imagens formalizada, os resultados, e a discussão. Finalmente, na Seção 6 apresenta-se as conclusões.

2. Fundamentação

Nesta Seção aborda-se brevemente aspectos teóricos que são utilizados no trabalho: CNN, NB, GAN, e métricas de avaliação.

Existem várias técnicas de classificação que geram modelo via treinamento [Witten et al. 2016], sendo que no caso de CNN é usada uma grande quantidade de imagens [Ponti et al. 2017].

As CNNs existentes, de maior destaque, incluem as arquiteturas ResNet [He et al. 2015], InceptionV3 [Szegedy et al. 2016] e VGG-Net [Simonyan e Zisserman 2014], sendo as arquiteturas VGG-16 e VGG-19 as mais comuns desta última. Para ambos os casos, é necessário a inicialização de parâmetros de configuração, como por exemplo, número de épocas (*epochs*), taxa de aprendizado (*learning rate*), *decayr* e *momentaum* [Ponti et al. 2017].

Para se utilizar uma arquitetura de CNN existente pode-se optar pelo uso de *transfer-learning* [Oquab et al. 2014]. Neste caso, existe a possibilidade da utilização dos pesos de uma arquitetura previamente treinada, como por exemplo, usando os pesos do treinamento ImageNet que possui mais de 1.2 milhões de imagens dívidas em 1000 classes [Krizhevsky et al. 2012].

Outra técnica de classificação, usando *Machine Learning*, é o NB: classificador probabilístico baseado no teorema de *Bayes* [Good 1965] e [Sebastiani 2002]. O NB determina a classe de maior probabilidade para cada nova amostra, verificando as

[1] http://www.safernet.org.br/site/indicadores

chances da existência de determinados atributos encontrados no objeto sendo classificado.

Há também o classificador GAN que compreende um par de redes neurais em competição: um falsificador contra um especialista, onde o falsificador fica cada vez melhor em falsificar, e o especialista cada vez melhor em descobrir as falsificações. O falsificador, é conhecido como gerador de imagens parecidas, relativas ou ruidosas, e o especialista, é conhecido como discriminador por comparar as falsificações e as imagens reais, com o objetivo de geração de novas imagens e o treinamento das duas redes neurais [Goodfellow et al. 2014]. O potencial do uso de GAN é enorme, visto que está pode imitar qualquer distribuição de dados, i.e., gerar imagem *fake* [Creswell 2018].

Para todos os classificadores, durante e após seu treinamento realizam-se predições. Isto é feito em duas etapas, em que se utilizada uma probabilidade, para designar a confiança. Neste trabalho, quanto maior a probabilidade maior será a chance de uma imagem pertencer a classe correspondente.

A predição pode utilizar diferentes métricas, como por exemplo a ACC (acurácia) que é baseada na somatória da taxa de categorização (TP: *true positives* + TN: *true negatives*) correta e dividida pela somatória de todas as classificações. Esses valores podem ser retirados da *Confusion Matrix*. Assim, é possível realizar a comparação entre diferentes arquiteturas na solução de um mesmo problema.

3. Trabalhos Relacionados

A seguir são descritos trabalhos da literatura que se utilizam de técnicas para a detecção de imagens PI, GAN e NB.

[Polastro e Eleuterio 2010] apresentam trabalho em que desenvolvem o *software* NuDetective para realizar varreduras em computadores a fim de encontrar arquivos suspeitos. Para a detecção de imagem PI foi verificado o percentual de exposição de pele. Utilizaram 7.244 imagens em seus experimentos, sendo que os resultados apontam para ACC = 95%.

No uso de técnicas de reconhecimento visual para a identificação de material suspeito de PI foi utilizado descritor para representação das imagens e treinamento do classificador SVM (*Support Vector Machine*). Foi desenvolvido um sistema que apresentou ACC = 80% [Ulges e Stahl 2011]. De maneira análoga, [Sae-Bae et al. 2014] implementaram um sistema baseado no percentual de exposição de pele. Em experimentos com 105 imagens envolvendo crianças sem contexto sexual apresentou TP = 83% de detecção de imagens.

[Vitorino et al. 2016] apresentaram o uso do descritor SURF (*Speeded-UP Robust Features*), dicionários visuais e classificador SVM. Em experimentos com 4.998 imagens PI foi obtido ACC = 68,3%.

Por sua vez, [Yiallourou et al. 2017] produziram 88 imagens PI baseando-se em questionários. Posteriormente, treinam um modelo de regressão linear com estas imagens, sendo que obtiveram ACC = 76,14%.

[Vitorino et al. 2018] treinaram a CNN GoogLeNet com pesos da ImageNet [Krizhevsky et al. 2012] e a base Pornography-2k [Moreira et al. 2016], i.e, treinam a

CNN para detecção de imagem PA (Pornografia Adulto). Posteriormente, treinam a mesma CNN com os pesos obtidos na etapa inicial, entretanto, para a detecção de PI, obtendo ACC = 86,5%.

[Macedo et al. 2018] propuseram um mecanismo que utiliza duas CNNs. Primeiro, é analisado a existência de PA em imagem, e em caso positivo, é feita a localização da face do indivíduo na imagem para estimativa da idade. Em caso do indivíduo ser uma criança, concluem que imagem é PI. Os autores apresentam a metodologia e a elaboração de uma base PI não público denominado de *region-based annotated child pornography* (RCPD[2]) composto por 2.138 imagens PI. Os resultados da aplicação do mecanismo na base RCPD obteve ACC = 79,84%.

Os trabalhos relacionados apresentados podem ser divididos em 4 grupos. No primeiro, 2 peritos desenvolveram o *software* NuDetective [Polastro e Eleuterio 2010], similar ao trabalho de [Sae-Bae et al. 2014]. No segundo, apresentaram o uso de método de regressão linear [Yiallourou et al. 2017]. No terceiro, propõem-se uma abordagem a partir da extração das características discriminadoras de imagens, para no final ser empregado o classificador SVM [Vitorino et al. 2016], assim como [Ulges e Stahl 2011], que também utilizam descritor e SVM. E no quarto, utilizou-se de CNN [Vitorino et al. 2018] e [Macedo et al. 2018].

Outrora, existem trabalhos que fomentam ataques a redes neurais. A exemplo cita-se o trabalho em que se implementa um visualizador de imagens denominado de *Adversarial Playground*: faz-se a visualização de imagens reais e as mesmas imagens com modificações em *pixels*. Para modificações, os autores fizeram a inserção de *pixels* nas imagens reais. Para tal, eles propuseram alterações no algoritmo JSMA (*Jacobian Salliency Map Approach*) [Norton e Qi 2017]. Os autores concluíram que o tempo médio para a modificação das imagens reais foi reduzido em 1,5 vezes em relação ao uso do algoritmo original, o JSMA.

Ademais, verifica-se a existência de vários métodos para a classificação de imagens, entre eles, para a classificação de imagem pornográfica [Karamizadeh e Arabsorkhi 2018]. Por exemplo, em [McClelland e Marturana 2014] aplicou-se o algoritmo NB para a classificação de imagem PI em aplicação investigativa. Foi proporcionado ACC de 74,8%.

Desta forma, foram apresentados trabalhos que visam a detecção de imagem PI, modificação e visualização de imagens e aquele que se baseia no algoritmo NB para a detecção de imagem PI. Verificou-se que o uso de CNN para a detecção de imagem PI tem sido feito com frequência nos trabalhos atuais, ainda que os resultados apresentados por estes são aceitáveis. Por sua vez, a CNN pode classificar qualquer tipo de imagem, seja uma imagem real, ou imagem que tenha sido modificada no uso do algoritmo JSMA por exemplo. É possível o uso de NB para a classificação de imagem, ou mesmo que NB seja aplicado em conjunto com a CNN, visando fortalecer com segurança a classificação da CNN.

[2] http://www.patreo.dcc.ufmg.br/datasets/rcpd/

4. Método Proposto

A Figura 1 apresenta a visão geral da proposta (*overview*). Usamos um classificador com o algoritmo NB com validação cruzada (*Cross-validation*), que tem bom desempenho e baixo custo computacional. O NB pode calcular e prever probabilidades diferentes em grandes volumes de dados, com base em vários atributos, criando assim, uma base de treinamento para classificar imagens *fakes*, que no nosso trabalho fará o papel de classificador de autenticidade. Para treinamento do NB, a GAN receberá como entrada as imagens reais das duas classes (1C: classe 1 criança e CA: classe criança e adulto), e as imagens da base ImageNet [Krizhevsky et al. 2012] que serão consideradas neste caso como ruído na geração de imagens parecidas (modificadas).

A GAN que produzirá imagens *fakes,* na mesma quantia que as imagens reais, é a Cyclegan (*Cycle-Consistent Adversarial Networks*) [Zhu et al. 2017]. As imagens *fakes* são nominadas de teste *fake*. A GAN foi necessária, visto que é eficaz na geração de novas imagens com ruídos e características de outros objetos, não sendo necessário serem objetos de mesma espécie das imagens já utilizadas, como por exemplo, a criação da imagem de uma criança, com características de um animal, fruta, paisagem, etc.

Ao testar uma imagem *fake*, esta será descartada após execução do classificador de autenticidade. Em caso contrário a imagem será analisada pela CNN.

A CNN é gerada a partir do treinamento com imagens 1C e CA usando a técnica de *transfer-learning*, onde uma arquitetura de CNN pré-treinada aprende o modelo daquilo que é NPI, i.e., imagina-se que tudo o que não for 1C ou CA, será FP ou FN, exigindo a análise de um especialista para definir ser imagem é PI.

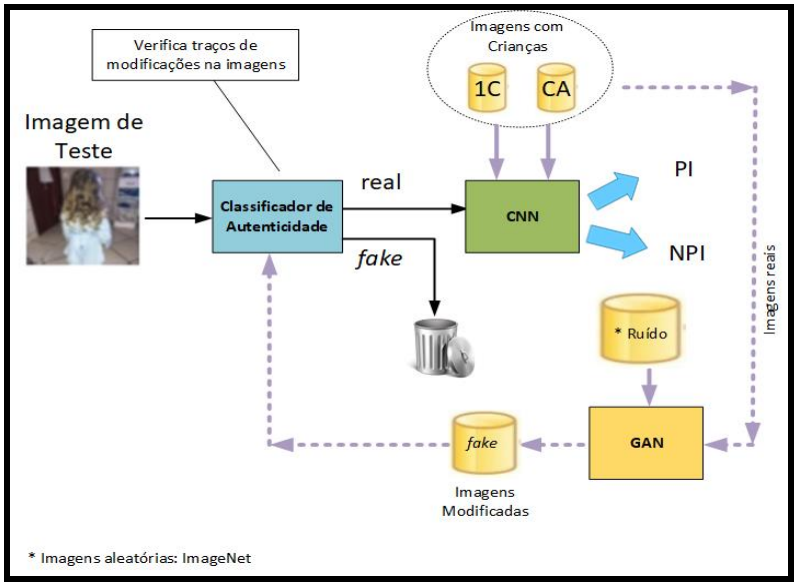

Figura 1. Visão Geral da Proposta

Para melhor entendimento do treinamento da CNN, a explicação foi dividida em duas etapas: a primeira, etapa de ajuste fino da rede pré-treinada (Figura 2) e a segunda etapa, predição (Figura 3).

Na primeira etapa é realizada a obtenção do modelo, a partir de uma grande quantidade de imagens divididas nas duas classes: 1C e CA. Submete-se a uma arquitetura de CNN os pesos de um treinamento prévio a obtenção de um modelo que

classifique 1C e CA. Optou-se em utilizar os pesos de *transfer-learning* da ImageNet [Krizhevsky et al. 2012]. Neste caso, todas as camadas da arquitetura utilizada foram mantidas e com seus pesos aleatoriamente inicializados, exceto a última camada, já que havia sido treinada pela ImageNet.

Para experimentos, aplicam-se as 4 arquiteturas de CNN de uso mais comum: InceptionV3, VGG16, VGG19 e ResNet para a comparação de seus resultados [Ponti et al. 2017]. Estas arquiteturas são treinadas, gerando-se assim, modelos treinados.

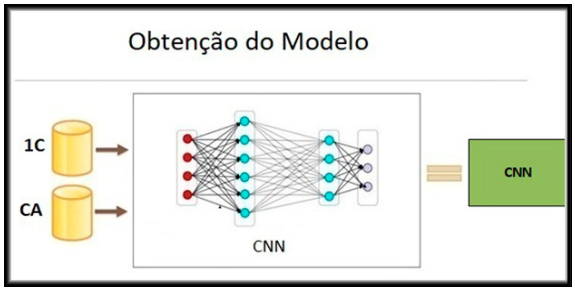

Figura 2. Etapa de Ajuste Fino da Rede Pré-treinada

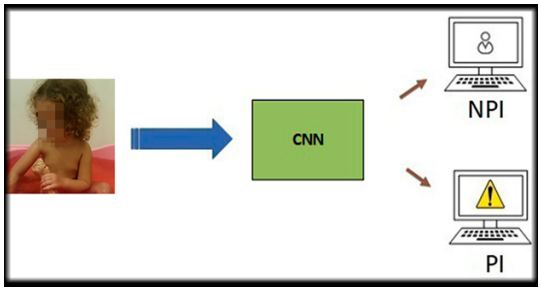

Figura 3. Predição

Na segunda etapa foi realizada a predição das imagens de teste no uso do modelo gerado pela CNN. Para validação da predição analisamos a ACC considerando-se um grau de confiança.

Optou-se pela utilização da confiança pelo fato da última camada do modelo CNN gerar a probabilidade de uma imagem pertencer a determinada classe, para tentar identificar se a classificação resultante está correta. Nos trabalhos relacionados isto não é comentado, isto significa que os autores usam grau de confiança implícito de 50% em sua maioria, chance igualitária, dado que são duas classes.

5. Estudo de Caso

A implementação foi dividida em duas etapas, treinamento da CNN e classificador de autenticidade. Na primeira, foi desenvolvido o *software* TrainPI, para treinamento da CNN, em linguagem de programação Python 3[3], com auxílio das bibliotecas TensorFlow[4] e Keras[5]. O *software* divide-se em 3 módulos: configuração, treinamento e predição.

No módulo de configuração são informados os endereços onde estão alocadas as imagens de treinamento, validação e predição. No módulo treinamento é realizado o *transfer-learning*. É neste módulo que é fornecida a arquitetura CNN a ser utilizada e os parâmetros de configuração da arquitetura: *epochs, learning rate, decayr* e *momentaum*. Foi estipulado o uso dos parâmetros: rede = (InceptionV3, VGG16, VGG19 ou ResNet), *epochs* = 20, *learning rate* = (0.0001, 0.001, 0.01, 0.007 ou 0.009), *decayr* = 1e-6, *momentaum* = 0.9. Estes parâmetros foram retirados da literatura e somente os melhores resultados do treinamento são apresentados na subseção 5.3; salientando que o

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

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

[5] https://keras.io/

treinamento foi realizado com uso de uma placa de vídeo GPU (*Graphics Processing Unit*) Titan-XP.

Durante o treinamento, e após cada *epoch* o modelo é salvo em disco. Ao final da execução foram obtidos 20 modelos, usados na predição, onde se indica qual dos 20 modelos de treinamento e arquitetura se deseja testar.

A segunda etapa compõe a implementação da proposta do trabalho com o auxílio das bibliotecas Scikit-learn[6]: classificador de autenticidade. Usando a GAN, extraimos as características para formação das imagens *fakes* utilizando uma pasta de teste com imagens reais 1C e outra pasta com imagens reais CA. Para o treino foram inseridas as imagens da base ImageNet [Krizhevsky et al. 2012] como ruído. Desta forma, imagens *fakes* com as características das imagens de ruído foram geradas. Após isso foram adicionadas as imagens *fakes* geradas como treino para o classificador de autenticidade. Para preparação dos arquivos de teste, foi utilizado o filtro ColorLayoutFilter que faz o processo de extração de características de cores MPEG7 das imagens. Dividiu-se cada imagem em 64 blocos e se calculou a cor média de cada bloco gerando um novo arquivo de dados. Este novo arquivo é classificado e treinado pelo algoritmo NB para identificação ou descarte de imagens *fakes* antes do envio para a CNN.

Na sequência é descrito a divisão da base de imagens formalizada, os resultados experimentais, assim como, as comparações entre os resultados obtidos com os resultados encontrados nos trabalhos relacionados, i.e., as discussões.

5.1 Base de Imagens

Para execução deste trabalho foi construída uma base de imagens composta por 2 classes: 1C e CA. Em todos os vídeos existe a presença da imagem de criança.

Figura 4. Exemplo de Imagens da Classe 1C

Figura 5. Exemplo de Imagens da Classe CA

Os *frames* (imagens) extraídos foram categorizados de acordo com a situação como filme, série, viagem, programa de auditório, novela, vídeo de criança, brincadeira, música, médico, dentista, dentista, esporte, educação e outros. Amostras dos *frames*, agora como imagens, foram pré-processados e todas as imagens contendo criança com

[6] https://scikit-learn.org

idade estimada inferior a 18 anos, estando sozinha, ou na companhia de um adulto, foram armazenadas em pastas conforme sua classe.

As imagens 1C e CA caracterizam-se por possuir uma criança. São imagens em que um humano consegue reconhecer a existência de uma criança. Exemplo de imagens 1C: criança deitada, de costas, pulando, brincando, cantando, dançando, estudando, tomando banho, ou mesmo imagem em que não aparece a face da criança, e sim, o corpo. Exemplo de imagens CA: criança sendo analisada por um médico, criança abraçando um adulto, criança no colo de um adulto etc. Na Figura 4 são mostrados 9 exemplos de imagens 1C, i.e., crianças com idades variadas e em várias situações: deitadas, escovando os dentes, sentada e de costas etc.

De maneira análoga na Figura 5, entretanto, com imagens CA são mostradas situações onde sempre existe no mínimo, parte do corpo de um adulto e uma criança. Por exemplo, um adulto banhando uma criança, uma criança no colo de um adulto e uma criança dormindo de costas para um adulto.

Na Figura 6 são mostradas imagens teste *fake* 1C e na Figura 7 imagens teste *fake* CA. As imagens *fakes* foram geradas pela GAN a partir das imagens de teste, lembrando que para cada imagem teste existe uma imagem *fake* correspondente.

Figura 6. Exemplo de Imagens *fakes* da Classe 1C

Figura 7. Exemplo de Imagens *fakes* da Classe CA

5.2 Divisão da Base de Imagens

Para experimentação dividiu-se a base de imagens em treinamento, validação e teste. Por questões de metodológicas os vídeos foram divididos na seguinte proporção: 60% para treinamento e os outros 40% para validação e teste.

O treinamento, assim como a validação e teste são formados por duas classes. Os vídeos que formam o treinamento da classe 1C são diferentes dos vídeos do treinamento da classe CA. Ainda, o treinamento recebeu imagens que não estão na base de validação e nem nos testes, sendo que as imagens de 40% dos vídeos foram adicionados para a validação, e aleatoriamente, foi retirado metade das imagens da validação e adicionada à base de teste.

A base possui 161.120 imagens: 96.676 de treinamento, 32.222 de validação e 32.222 de teste. Para efeitos de experimentos, foram geradas 32.222 imagens *fakes* usando GAN. Resumindo existem três bases de teste:

- **teste:** 32.222 imagens reais (1C e CA);
- **teste *fake*:** 32.222 imagens geradas pela GAN (1C e CA);

- **teste + teste *fake:*** imagens reais + imagens *fakes*. Totalizando 64.444 imagens (1C e CA).

5.3 Resultados Experimentais

Nos experimentos foi utilizado um computador com processor Intel i7-7700K CPU (*Central Processing Unit*) @ 4.20GHz, 16 GB, GPU Titan-XP e Windows 10. Inicialmente foram realizados experimentos apenas com a CNN: foram gerados 400 modelos (4 arquiteturas, 5 diferentes *learning rate* e 20 *epochs*), ou seja, para cada arquitetura foram gerados 100 modelos.

Na Tabela 1 foi registrado o resultado dos melhores modelos referente a cada arquitetura CNN baseando-se na métrica ACC. Neste momento o valor da confiança utilizado foi de 50%, já que a grande maioria dos trabalhos na literatura utilizam este valor. Registrou-se a ACC durante o treino, a ACC da classificação das imagens teste, a ACC da classificação das imagens *fakes* e ACC da classificação das imagens pertencentes ao teste e teste *fake*.

Tabela 1. Resultados – Treinamento e Teste – CNN

Arquiteturas	*Learning rate*	ACC treino	ACC teste	ACC teste *fake*	ACC teste + teste *fake*
InceptionV3	0,001	98,72%	**87,96%**	**84,26%**	**86,00%**
VGG16	0,0001	80,00%	75,00%	71,14%	73,12%
VGG19	0,007	76,91%	73,00%	68,00%	71,13%
ResNet	0,0001	98,30%	81,34%	77,84%	77,90%

Analisando os resultados apresentados na Tabela 1, verificou-se que de todas as arquiteturas experimentadas a que gerou melhores resultados foi a arquitetura **InceptionV3**, com *learning rate* **0,001** e predição das imagens do teste com 50% de confiança (ACC) = **87,96%.** Na Figura 8 são apresentados 2 gráficos do experimento que obteve o melhor resultado.

No gráfico da esquerda da Figura 8 plotou-se ACC x *epoch*, onde é perceptível a melhora da ACC conforme o aumento do número de *epochs* do treinamento. Verifica-se que próximo da *epoch* = 16 tem-se ACC alta, o que se caracteriza como ponto de interesse.

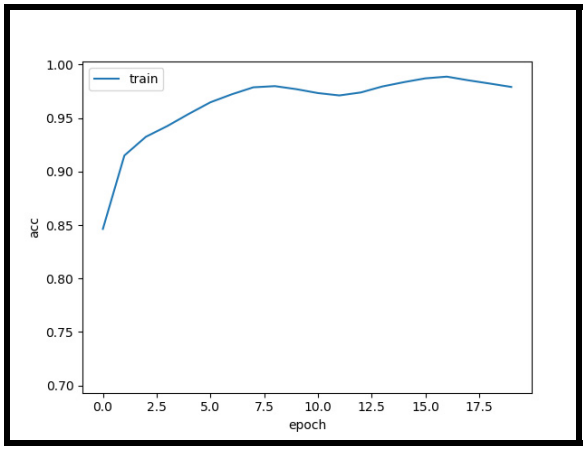

Figura 8. InceptionV3 – 0,001: *Epoch* X ACC

Após este experimento, realizou-se o experimento da predição das imagens de teste no com uso de diferentes graus de confiança: 20, 50, 80, 90 e 95, sendo os

resultados (ACC) apresentados na Figura 9. Observa-se que, quanto maior a confiança, menor é a ACC. Analisando este gráfico os autores concluem que mesmo para o maior grau de confiança (95%) o **ACC teste** é aceitável (85,06%).

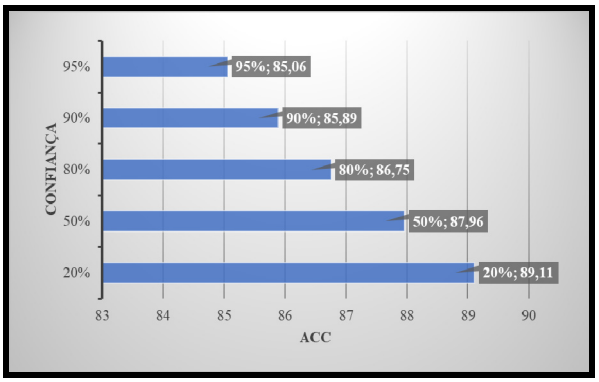

Figura 9. ACC x Confiança

Por sua vez, na Tabela 2 é apresentada a ACC em experimentos para análise da ACC do classificador de autenticidade.

Tabela 2. Resultados – Classificador de Autenticidade

	teste	teste *fake*	teste + teste *fake*
ACC	12,00%	**82,47%**	41,21%

Na Tabela 3 são apresentados os tempos de processamento para o classificador de autenticidade, CNN, classificador de autenticidade + CNN, usando 32.222 imagens no intuito de mostrar a performance da proposta. Salienta-se que os experimentos foram executados no uso de CPU para padronização do experimento.

Tabela 3. Resultados – Tempo de Processamento

Classificador de Autenticidade	CNN	Classificador de Autenticidade + CNN
0,32 seg.	1.063 seg.	1.052 seg.

Na Tabela 4 são mostrados os resultados da ACC na execução do conjunto: classificador de autenticidade e CNN. Objetivou-se verificar a ACC de todo o processo de classificação, i.e., desde a inserção de uma imagem teste no classificador de autenticidade, mesmo que a imagem não seja classificada como imagem *fake*, até a imagem ser submetida a CNN; o grau de confiança neste experimento foi de 50%.

Tabela 4. Resultados para ACC (Classificador de Autenticidade + CNN)

	teste	teste *fake*	teste + teste *fake*
ACC	**87,88%**	89,88%	**88,88%**

5.4 Discussão

Na Tabela 5 é apresentado um resumo parcial de trabalhos relacionados que envolvem a detecção de PI (Seção 3) e feito uma comparação. Este resumo serve para a comparação com os resultados proporcionados pela CNN.

Os trabalhos de [Vitorino et al. 2018] e [Macedo et al. 2018] fazem o uso de CNN e investigam imagem PI e são passíveis de comparação com os resultados da nossa proposta. Entretanto, [Vitorino et al. 2018] utiliza uma abordagem de duas etapas

para detecção de PI e obtiveram ACC=86,5% utilizando o classificador SVM, mas não fica claro qual o valor de confiança empregado, além do que, utilizaram uma base de imagens não pública e portanto não temos como refazer o experimento para fazer comparações mais específicas. Outro trabalho que apresenta bons resultados na detecção de PI é [Vitorino et al. 2016] com ACC=68,3%, embora não use CNN, e tenha a mesma limitação do anterior com relação a base.

Tabela 5. Comparações dos Trabalhos Relacionados

Solução	Formato	CNN	Imagens PI Públicas para Experimentação	Imagens NPI disponíveis	Resultados	Parceria Polícia ou Envolvimento de Policiais no Trabalho
[Macedo et al. 2018]	Imagem	Sim	Não	Não	ACC = 79,84%	Sim
[Vitorino et al. 2016]	Imagem	Não	Não	Não	ACC = 68,3%	Sim
[Vitorino et al. 2018]	Imagem	Sim	Não	Não	ACC = 86,5%	Sim
Software NuDetective [Polastro e Eleuterio 2010]	Vídeo e Imagem	Não	Não	Não	ACC = 95,0%	Sim

Os trabalhos que permitem comparação com o nosso é o trabalho do *software* NuDetective [Polastro e Eleuterio 2010]. Foi possível a obtenção de cópia do *software* NuDetective com os pesquisadores que o criaram. Os resultados estimados do uso do classificador de autenticidade e da CNN treinada, e do *software* NuDetective estão disponíveis na Tabela 6. Ambos foram executados tendo como entrada as imagens de teste e posteriormente, as imagens do teste *fake* + teste *fake* (nas imagens não existe qualquer evidência de pornografia).

A CNN (nossa proposta) atingiu 87,96% de ACC e os falsos positivos foram de 12,04% na análise do teste e 14,00% na análise do teste + teste *fake*. Diante deste resultado, nossa proposta apresentou apenas 12,04% das imagens do teste como suspeitas de PI, enquanto que no *software* NuDetective este percentual foi 51,00%. Obviamente o software NuDetective não foi concebido para evitar os efeitos de uma GAN, mas do jeito que está seria vítima fácil da mesma.

Tabela 6. Comparação da Literatura com a Nossa Proposta

Solução	ACC teste	ACC teste + teste *fake*
Software NuDetective [Polastro e Eleuterio 2010]	51,00%	48,32%
Classificador de autenticidade + CNN [Nossa proposta]	**12,04%**	**14,00%**

Na Tabela 1 foi mostrado que a ACC da classificação realizada pela CNN com imagens teste foi de 87,96% enquanto para teste + teste *fake* houve uma piora de quase 2%, ficando a ACC em 86%. Isto significa que a CNN classificou imagem *fake* como se fossem imagens reais, do contrário este valor deveria ser bem menor.

Por sua vez, o classificador de autenticidade obteve ótimo resultado nos experimentos (Tabela 2). Quando realizado experimento com as imagens teste *fake*, o classificador de autenticidade conseguiu identificar 82,47%. Neste caso, apenas 17,53% das imagens *fake* seriam enviadas a CNN o que pode ser considerado um ótimo resultado em relação ao que há na literatura.

Tendo em vista os resultados apresentados pode-se afirmar que computacionalmente se torna custoso o uso do classificador de autenticidade em conjunto com a CNN, porém com relação a autenticidade este custo se justifica. Por

exemplo, em experimento usando 10 imagens, registrou-se tempo total de 0,56 segundos (Tabela 3), obviamente este tempo pode ser diminuído com o uso de GPU, que não foi explorado neste artigo.

Por fim, foi na Tabela 4 foram mostrados os resultados deste trabalho, onde a CNN obteve ACC 86,00% na classificação de teste + teste *fake* (Tabela 1). Quando A CNN é executada em conjunto com o classificador de autenticidade obteve-se ACC de 88,88%.

6. Conclusões e Direções Futuras

Neste trabalho mostrou-se a ausência de integridade em CNN. Em nosso caso, conforme mostrado na Tabela 1, a CNN classifica as imagens teste com acurácia equivalente a 87,96%, teste *fake* com 84,26%, e teste + teste *fake* com 86,00%. A acurácia obtida mostra que as imagens *fakes* foram classificadas como se fossem reais, o que mostra um gap de segurança no uso de CNN para classificação de imagens.

Porém, o classificador de autenticidade auxilia na identificação de imagem *fake* antes de enviá-las para a CNN. Com isso, as chances de uma imagem *fake* ser classificada como se fosse real se torna menor, já que os resultados do classificador de autenticidade obtiveram ACC de 82,47% na classificação de imagem *fake*. Nos experimentos, detectou-se a maioria das imagens *fakes*, possibilitando que a CNN apenas classifique imagens reais, já que a CNN sozinha não consegue distinguir com clareza as imagens *fakes* das reais.

O uso de GAN pode ser computacionalmente mais custoso. Por isto, é proposto o uso do classificador de autenticidade para garantir maior segurança na seleção de imagens *fakes*, antes do envio da imagem para a CNN, garantindo maior acurácia na execução das classificações.

Os resultados experimentais também mostram desempenho superior a literatura: ACC de 85,06% com 95% de confiança, com confiança igual a 50% obteve-se uma taxa de erro igual a 12,04% (Tabela 6), número que se mostra inferior aos trabalhos relacionados.

A grande maioria dos trabalhos relacionados a PI não podem ser reproduzidos, nem tão pouco comparados, haja visto que as bases de PI utilizadas não são públicas e nem disponíveis sob requisição de outros pesquisadores.

Como trabalhos futuros, pretende-se dar continuidade na investigação da ausência de integridade da CNN, assim como demais redes neurais. Por sua vez, a maior dificuldade para realização deste trabalho foi a manipulação de grande quantidade de imagens para realização dos experimentos.

Agradecimentos

Os autores agradecem ao CNPq pelo apoio financeiro parcial ao projeto, processo: 430972/2018-0, e a NVIDIA *Corporation* pela doação de uma GPU Titan-XP usada nos experimentos. O estudante Jackson Mallmann agradece ao IFC (Instituto Federal Catarinense) e ao Capes pela bolsa concedida através do edital n° 231/2017.

Referências

Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A. A. (2018). "Generative Adversarial Networks". IEEE Signal Processing Magazine.

Good, I. J. (1965). “The Estimation of Probabilities: An Essay on Modern Bayesian Methods”, M.I.T. Press.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza M.; Xu, B.; Warde-Farley D.; Sherjil O.; Aaron C.; Yoshua B. (2014). "Generative Adversarial Nets." Neural Inf. Processing Systems (NIPS).

He, K.; Zhang, X.; Ren, S.; Sun, J. (2015). “Deep residual learning for image recognition”. CoRR, abs/1512.03385, 2015.

Karamizadeh, S. e Arabsorkhi, A. (2018). “Methods of Pornography Detection: Review”. In Proceedings of the 10th International Conference on Computer Modeling and Simulation (ICCMS 2018). ACM, New York, NY, USA, 33-38. DOI: https://doi.org/10.1145/3177457.3177484.

Krizhevsky, A.; Sutskever, I.; Hinton, G.E. (2012). “ImageNet classification with deep convolutional neural networks”, In Advances in neural information processing systems, pages 1097–1105.

McClelland, D.; Marturana, F. (2014). “A Digital Forensics Triage Methodology based on Feature Manipulation Techniques”, In: Proc. of the International Conference on Communications Workshops, pages. 676-681.

Macedo, J.; Costa, F.; Santos, J.A. dos. (2018). “A Benchmark Methodology for Child Pornography Detection”. 31st SIBGRAPI Conf. on Graphics, Patterns and Images (SIBGRAPI).

Moreira, D.; Avila, S.; Perez, M.; Moraes, D.; Testoni, V.; Valle, E.; Goldenstein, S.; Rocha, A. (2016). “Pornography classification: The hidden clues in video space-time”, Forensic Science Int. Vol. 268, p. 46-61.

Norton, A.P.; Qi, Y. (2017). “Adversarial-Playground: A Visualization Suite Showing How Adversarial Examples Fool Deep Learning”. IEEE Symposium on Visualization for Cyber Security (VizSec).

Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. (2014). “Learning and transferring mid-level image representations using convolutional neural networks” in Proc. of the IEEE conf. on computer vision and pattern recognition, pp. 1717–1724.

Polastro, M.C. e Eleuterio, P.M.S. (2010). "NuDetective: a Forensic Tool to Help Combat Child Pornography through Automatic Nudity Detection", Workshop on Database and Expert Systems Applications (DEXA).

Ponti, A. M.; Ribeiro, L.S.F.; Nazare, T.S.; Bui, T.; Collomosse, J. (2017). “Everything You Wanted to Know about Deep Learning for Computer Vision but Were Afraid to Ask”, 30th SIBGRAPI Conf. on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), Niterói, 2017, pp. 17-41.

Sae-Bae, N.; Sun, X.; Sencar, H.T.; Memon, N.D. (2014). “Towards automatic detection of child pornography”, In: The 2014 IEEE Int. Conf. on Image Processing (ICIP), pages 5332–5336.

Sebastiani, F. “Machine Learning in Automated Text Categorization”, ACM Computing Surveys, Vol. 34. No. 1, Março 2002, p.1-47.

Simonyan, k.; e Zisserman, A. (2014). “Very deep convolutional networks for large-scale image recognition,” CoRR, vol. abs/1409.1556, 2014.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. (2016). “Rethinking the inception architecture for computer vision”. In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pages 2818–2826, 2016.

Ulges, A.; e Stahl, A. (2011). “Automatic detection of child pornography using color visual words,” in Multimedia and Expo (ICME), 2011 IEEE Int. Conf. on. IEEE, 2011, pp. 1–6.

Vitorino, P.; Avila, S.; Rocha, A. (2016). “A Two-tier Image Representation Approach to Detecting Child Pornography”. Proc. of XII Workshop de Visão Computacional.

Vitorino, P.; Avila, S.; Perez, M.; Rocha, A. (2018). “Leveraging deep neural networks to fight child pornography in the age of social media”. Journal of Visual Communication and Image Representation (50).

Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. (2016). “Practical Machine Learning Tools and Techniques”, Editora Morgan Kaufmann. 4th Edition.

Yiallourou, E.; Demetriou, R.; Lanitis, A. (2017). "On the Detection of Images Containing Child-Pornographic Material". 24th Int. Conf. on Telecommunications (ICT), Limassol, 2017, pp. 1-5.

Zhu, Jun.; Park, T.; Tinghui, Z.; Alexei, A.. Efros (2017). “Unpaired Image-to Image Translation using Cycle-Consistent Adversarial Networks” IEEE Int. Conf. on Computer Vision (ICCV).