



**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E
INOVAÇÃO**

**PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA
PIBIC**

RELATÓRIO FINAL

**MECANISMO PARA IDENTIFICAÇÃO DE IMAGEM SUBMETIDA A
GAN (*GENERATIVE NEURAL NETWORK*)**

**CURITIBA
2021**

Vinicius Clemente da Silva Quintino

Bacharelado em CiberSegurança

Escola Politécnica

**MECANISMO PARA IDENTIFICAÇÃO DE IMAGEM SUBMETIDA A GAN
(*GENERATIVE NEURAL NETWORK*)**

Relatório Parcial apresentado à Pontifícia Universidade Católica do Paraná, Pró Reitoria de Pesquisa, Pós-Graduação e Inovação, Programa Institucional de Bolsas de Iniciação Científica, sob orientação do Prof. Dr. **Altair Olivo Santin** e Doutorando Jackson Mallmann.

LISTA DE FIGURAS

Figura 1 – Modelo Cyclegan	9
Figura 2 - Fluxograma do projeto.....	11
Figura 3 – Imagens geradas a partir da GAN	13
Figura 4 – Histograma das Amostras.....	13
Figura 6 – Imagens utilizadas na segunda Comparação	14
Figura 7 - Resultado base - img_2_pred – Interseciton e Correlation.....	16
Figura 8 - Resultado base - img_2_pred – chi-square e bhattacharyya.....	17
Figura 9 - Resultado base_pred - img_2_pred – correlation e intersection.....	17
Figura 10 - Resultado base_pred - img_2_pred – chi-square e bhattacharyya	18

LISTA DE TABELAS

Tabela 1 - Resultado entre Histograma 14

Tabela 2 - Acurácia obtida sem Noise2Noise 15

Tabela 3 - Acurácia obtida utilizando Noise2Noise 15

SUMÁRIO

1	INTRODUÇÃO	6
2	OBJETIVO(S).....	7
3	REVISÃO DE LITERATURA.....	8
4	MATERIAIS E MÉTODO.....	10
5	RESULTADOS.....	12
6	DISCUSSÃO DOS RESULTADOS	18
7	CONSIDERAÇÕES FINAIS	20
8	ETAPAS FUTURAS	21
9	OUTRAS ATIVIDADES REALIZADAS.....	21

RESUMO

Introdução: A GAN, conhecida como (*Generative Adversarial Network*) possui uma excelente capacidade de produzir imagens falsas através de imagens reais, usando duas funções: *Generative*, que produz imagens idênticas a imagens verídicas e um *Discriminative*, que objetiva o encontro de variações resultante do *Generative*. Utilizou-se o modelo CycleGAN que por sua vez não necessita reconhecer dados de entradas, além de aprender a mapear características em comum entre uma imagem de entrada e uma de saída, para alcançar resultados satisfatórios, no entanto o resultado contém diversas imperfeições (ruídos). Os ruídos são um grande problema nas análises entre imagens, pois impossibilita obter uma acurácia significativa, todavia utilizou-se de um modelo Noise2Noise baseado em regressão em CNN (*Convolutional Neural Network*) que possui o intuito de remover tais ruídos, tornando em imagens mais nítidas. **Objetivo:** Busca-se criar um mecanismo que identifique imagem produzida por GAN, a fim de mitigar os efeitos negativos proporcionado por este modelo, a exemplo *Deep Fakes*. **Materiais e Método:** O estudo propôs identificar imagem submetida a GAN, para tanto utilizou-se do modelo CycleGAN para obtenção das imagens, como também o modelo Noise2Noise para remoção dos ruídos contidos nas imagens, bem como de uma ferramenta chamada OpenCV que dispõe de diversos modelos para análise de imagens, pelo qual usufruiu-se do Histograma representando a distribuição dos *pixels* através de um gráfico. **Resultados:** Os resultados mostraram-se promissores principalmente nas imagens sem ruídos, no entanto não foi possível obter uma acurácia elevada. Em certos momentos pode-se averiguar algo em torno de 10% para imagens sem ruídos sobre imagens contendo ruídos. **Considerações Finais:** Conclui-se que o estudo realizado atingiu o objetivo. No entanto, as análises realizadas não alcançaram um resultado exímio, todavia este trabalho contribui para o desenvolvimento de ferramentas voltadas a segurança computacional, como também ao que se diz respeito a detecção de anomalias em imagens através dos métodos aqui citados.

Palavras-chave: *Generative Adversarial Network*, *Deep Fake*, Noise2Noise, *Convolutional Neural Network*.

1 INTRODUÇÃO

As GANS (*Generative Adversarial Nets*) foram introduzidas por Goodfellow et al. em 2014, o qual propôs um sistema que utiliza dois modelos simultâneos, sendo um capaz de gerar resultados semelhantes a dados verídicos chamado de *Generative* e outro *Discriminative*, o qual é apto a detectar variações proporcionadas através do *Generative*.

Segundo Goodfellow (2014, p.1), “O modelo gerador pode ser considerado análogo a uma equipe de falsificadores, tentando produzir moeda falsa e usá-la sem detecção, enquanto o modelo *Discriminative* é análogo à polícia, tentando detectar a moeda falsa”. Ambos os modelos aprendem novas formas de aprimorar, através de um processo adversário, onde um busca maximizar a probabilidade de acertos e o outro a minimizar a taxa de erros, ao ponto em que se torna indistinguível um dado falso de um verdadeiro (GOODFELLOW, 2014).

Outro termo emerge através da utilização de GANS: *DeepFake* (“*deep learning*” e “*fake*”). Nguyen et al. 2019 explica através do exemplo: “técnica que pode sobrepor imagens de rosto de uma pessoa alvo a um vídeo de uma fonte para criar um vídeo da pessoa alvo fazendo ou dizendo coisas a pessoa fonte faz”.

DeepFake necessita de um grande volume de imagens, vídeos para o seu treinamento. Utilizam dados disponíveis na internet, ou seja, de figuras públicas, líderes políticos e famosos, tornando-os alvos fáceis para tal ação. Consequentemente, o uso indevido pode gerar conflitos entre países, grupos religiosos, ameaçando a segurança não apenas de figuras públicas, mas da sociedade (NGUYEN, 2019).

2 OBJETIVO(S)

Criar mecanismo que identifique imagem produzida por GAN.

3 REVISÃO DE LITERATURA

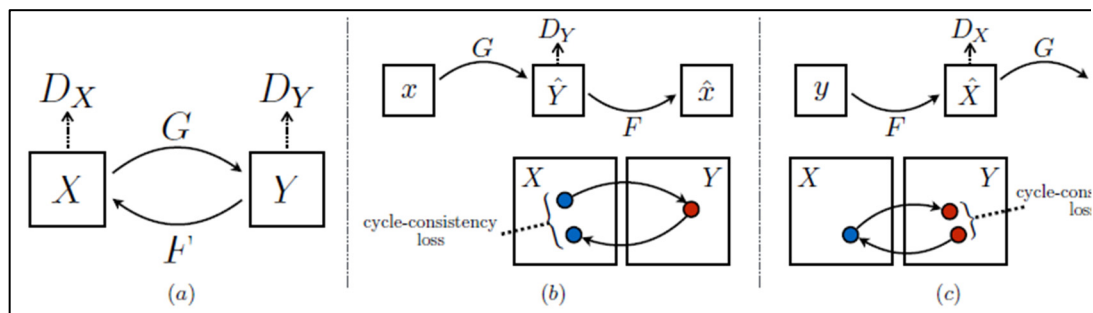
A proposta do projeto está direcionada para a identificação de imagens produzidas por GAN. Sabe-se que existem várias redes que implementam GAN, como a CycleGAN e a Pix2pix (EFROS, 2018). Além do mais, o potencial do uso de GAN é enorme, visto que podem imitar qualquer distribuição de dados, i.e., gerar um subconjunto de imagens.

Por exemplo, a Cyclegan se destaca dentre diversos modelos de GAN, pois não necessita conhecer dados de entrada e saída emparelhado, visto que obter estes dados emparelhados pode ser muito custoso, já que existem poucos conjuntos de dados para determinados fins, e difícil judicialmente, em sua maioria se exige autoria sobre tais imagens (ZHU, 2017).

Contudo, este modelo aprende mapear uma imagem de entrada com uma imagem de saída, a medida proposta prevê que ambos os domínios pertencem ao mesmo mundo pressuposto, com representações distintas, todavia sendo fundamental conhecer este relacionamento. Ao introduzir uma coleção de pares de exemplo ao domínio X e uma coleção de exemplo a Y , após várias exposições de treinamentos, o resultado se dá ($G: X \rightarrow Y$), de tal forma que ($\hat{y} = G(x)$, $x \in X$) seja indistinguível de $y \in Y$, ou seja, as amostras resultantes de G se tornam pertencentes a Y .

Os processos padrões da GAN frequentemente geram colapso, onde o dado de entrada torna-se o mesmo da saída, assim sendo necessário o uso de duas funções: a primeira faz a tradução e a segunda desfaz. Também se utiliza do ciclo de consistência de perda, o qual compara a acurácia da geração de imagens, isto é, quão próximo está a imagem treinada de uma real, até que chegue a um ponto onde não se possa identificar a qual domínio pertença a imagem treinada, ilustrado na Figura 1.

Figura 1 – Modelo Cyclegan



Fonte: ZHU; 2017

Há diversas dificuldades físicas nos dispositivos de gravações. As imagens estão propensas a ruídos ao acaso no processo de construção das imagens. Entende-se como ruído alguma deformidade durante a análise e extração de informações (GOYAL et al., 2020, p. 1).

É indispensável a utilização de ferramentas que possam reconstruir imagens, visto que as imagens podem ser capturadas em qualquer local. As imagens têm se tornado uma revolução, todavia tais capturas são feitas rotineiramente em lugares impróprios, com baixa luminosidade, e principalmente, contendo ruídos (GOYAL et al., 2020, p. 1).

Lehtinen et al. (2018) contribui na área de restauração de imagens corrompidas têm procurado evitar modelos convencionais, buscando aprender mapear observações de imagens corrompidas para sinais limpos, apenas olhando as amostras incompletas, não sendo necessário utilizar modelos de probabilidade estatística ou alguma imagem anterior, aprendendo diretamente com o treinamento.

Ademais, o trabalho de Lehtinen et al. (2018) só é possível utilizando um modelo de regressão baseado em (CNN) *Convolutional Neural Network*, com inúmeros pares de entradas (\hat{x}_i ; y_i) a partir de imagens corrompidas (\hat{x}_i), como também de exemplos limpos (y_i), tudo perante uma função de perda (L), que pode ser visto pela Equação 1.

$$\text{Argmin} \sum L(f(\hat{x}^i); y_i). \text{ (Equação 1)}$$

Diversos métodos foram desenvolvidos a fim de remover ruídos de imagens e chegar a sua forma original. Em comum, o ruído é eliminado através da obtenção da média, realizado por um modelo de remoção de ruídos, como exemplo, o modelo de suavização gaussiano (BUADES et al., 2005, p. 2).

Também existe o método *Noise*: composto por uma imagem U e um operador de remoção D_h , o qual depende de um parâmetro H . Desta forma, define-se como ruído a diferença entre a imagem, como mostrado na Equação 2.

$$u - D_h u. \text{ (Equação 2)}$$

4 MATERIAIS E MÉTODO

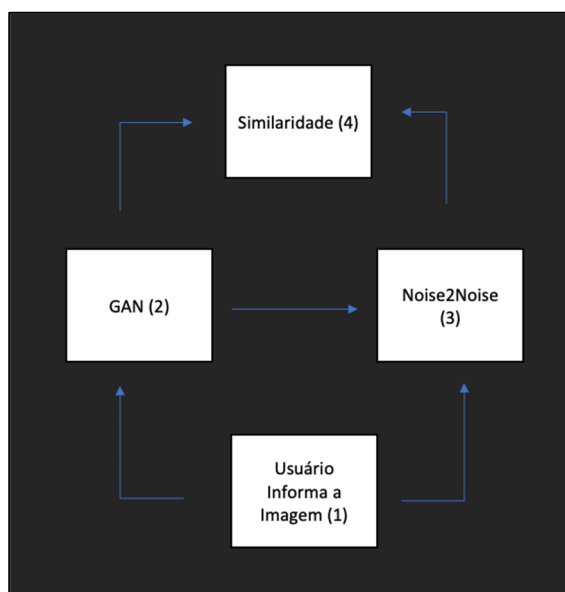
O objetivo é criar mecanismo que identifique imagem produzida por GAN. A Figura 2 ilustra o fluxograma do projeto. O mecanismo considera que um usuário informa uma imagem de teste (1), o qual será representado neste projeto de “imagem real”, que futuramente é processada pela GAN (2), a qual gera uma nova imagem (imagem falsa).

Por sua vez, a imagem falsa é processada pelo algoritmo Noise2Noise (3), onde tem-se o objetivo de retirar os ruídos, resultando em uma nova imagem (imagem *denoise*). Em posse das três imagens (real, falsa e *denoise*) faz-se o cálculo de similaridade (4).

Aplicou-se a CycleGAN (ZHU, 2017) para a obtenção das imagens falsas, da mesma maneira, utilizou-se do algoritmo Noise2Noise (LEHTINEN et al., 2018) para remoção de ruídos da imagem falsa. E para cálculo da similaridade recorreu-se a uma ferramenta que calcula a similaridade de imagens. Utilizou-se do histograma, mecanismo disposto pela biblioteca do Opencv¹.

¹ <https://opencv.org/>

Figura 2 - Fluxograma do projeto



Fonte: O autor (2021)

O histograma representa um gráfico que faz a distribuição dos *pixels* mais escuros (próximos de 0) para os mais claros de uma imagem (próximos de 255). Deste modo, pode-se comparar histogramas das imagens (real, falsa e *denoise*). Possibilita-se averiguar se há padrões ou peculiaridades diante do gráfico comparativo de duas imagens, como também sobre a imagem real, pelo qual corroboram para uma análise mais assertiva.

Procurou-se utilizar de diferentes métodos para remoção dos ruídos, sendo possível identificar qual alcançará o melhor resultado para o modelo em questão (CycleGAN). Assim, valeu-se dos métodos *Gaussian* e *Poisson* sobre as imagens. Ambos geram como resultado imagens contendo poucos ruídos. Uma vez feito a remoção de ruídos, busca-se identificar qual método se saiu melhor, para prosseguir para a etapa (4).

5 RESULTADOS

Foi utilizado um projeto disposto no GitHub² para o treinamento da GAN. O projeto contempla uma variedade de modelos, além de diversos materiais de apoio, bem como uma expressiva base de dados, facilitando a melhor execução e aprendizado.

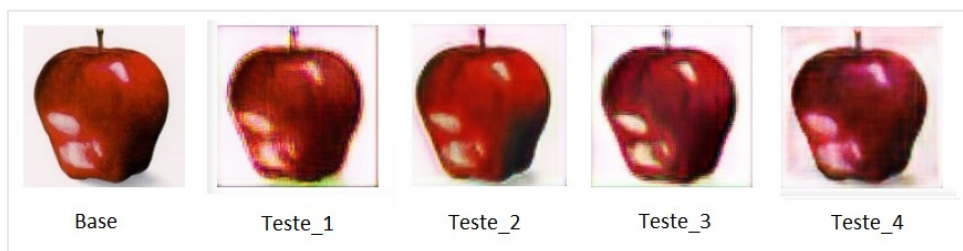
Diante de um cenário em que se preza o distanciamento social, não foi possível utilizar o laboratório da PUC, no entanto, optou-se por realizar a parte prática em um ambiente remoto usando uma VM (*Virtual Machine*) com alto poder computacional, contando com um processador i7 da Intel e uma GPU Tesla M60 major. Tais recursos foram disponibilizados em um servidor do PPGIA (Programa de Pós-Graduação em Informática).

Utilizou-se a Cyclegan para geração das imagens falsas, sendo necessário fazer algumas modificações no código. Para o treinamento foi proposto a base de dados contendo maçãs e laranjas. Também foi alterado a quantidade de imagens para 60 e o número de *epochs* foi aumentando gradualmente, para analisar quais modificações ocorreriam à medida que foi exposto a base.

Para iniciar o experimento foi selecionado cinco dentre centenas de imagens produzidas. Em um primeiro momento foi aplicado um algoritmo para gerar o histograma das imagens selecionadas, a fim de examinar em quais intervalos se concentra a maior distribuição de *pixel* e sua tonalidade.

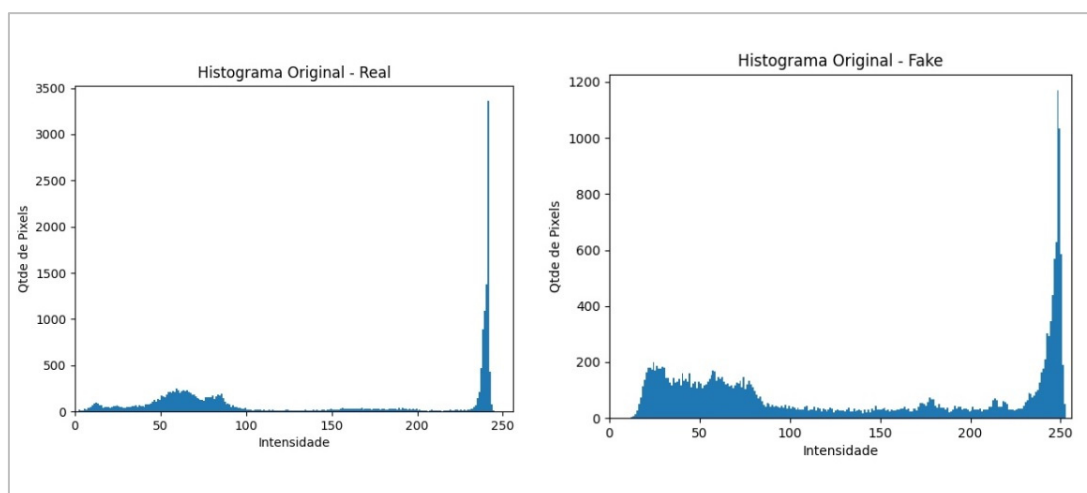
² <https://github.com/eriklindernoren/Keras-GAN>

Figura 3 – Imagens geradas a partir da GAN



Fonte: O autor (2021).

Figura 4 – Histograma das Amostras



Fonte: O autor (2021).

O Opencv fornece vários métodos integrados para comparação de histogramas. Nos métodos de *Correlation* e *Intersection*, quanto maior o resultado, mais preciso é a correspondência. Outrossim, para *chi-square* e *Bhattacharyya*, o valor resultante mais baixo representa o contrário.

Na Figura 3, a imagem descrita como base indica a imagem real (a qual será testada com todas as outras), já as demais apresentadas como falsas foram geradas pela GAN. A imagem teste_1 foi obtida com 10 *epochs* de treinamento, teste_2 com 50 *epochs*, teste_3 com 100 *epochs* e teste_4 com 150 *epochs*.

A análise foi feita utilizando métodos de comparação de histogramas anteriormente citados, onde a primeira imagem corresponde a real (qual será

comparada com as demais imagens). Na Figura 5 visualiza-se a comparação entre a imagem real (base) e as imagens geradas (falsas).

Tabela 1 - Resultado entre Histograma

Método	Base – Base	Base – Teste_1	Base – Teste_2	Base – Teste_3	Base – Teste_4
<i>Correlation</i>	1.0	0.112	0.269	0.305	0.081
<i>Intersection</i>	6.796	3.578	3.881	4.360	2.067
<i>Chi-Square</i>	0.0	7074.870	16618.880	15520.810	4753.137
<i>Bhattacharrya</i>	0.0	0.741	0.736	0.754	0.845

Fonte: O autor (2021)

Após a obtenção dos resultados com o histograma, optou-se por utilizar o algoritmo Noise2Noise o qual busca remover os ruídos de uma imagem. Compara-se agora as imagens falsas com as *denoise*, sendo que na Figura 6 são apresentados exemplos de imagens *denoise*.

Figura 5 – Imagens utilizadas na segunda Comparação



Fonte: O autor (2021)

A Noise2Noise possui diversos métodos de remoção de ruídos. Foi utilizado os métodos *Gaussian* e *Poisson*. Em seguida foram fornecidas as imagens Base (real) e

Img_2_orig para fazer a retirada dos ruídos. Gerou-se as imagens Base_pred e Img_2_pred ambas suavizadas, ou seja, sem ruídos.

Outra vez foi realizado a comparação entre imagens com o histograma. A Figura 7 indica os valores adquiridos ao fim deste processo. Vale ressaltar que estes valores são de imagens contendo ruídos.

Tabela 2 - Acurácia obtida sem Noise2Noise

Método	Base - Base	Base – Img_2_orig
Correlation	1.0	0.275
Intersection	6.968	3.835
Chi-Square	0.0	10098.40
Bhattacharyya	0.0	0.731

Fonte: O autor (2021)

Ao equiparar duas imagens iguais o valor resultante servirá como referência para as próximas análises, ou seja, quanto mais próximo for do valor base maior a acurácia. Ao comparar uma imagem real e uma imagem falsa (gerada por uma GAN) os valores tenderão a ser diferentes. Por exemplo, as imagens Base – Img_2_orig representados na Figura 7.

Tabela 3 - Acurácia obtida utilizando Noise2Noise

Método	Base_pred – Base_pred	Base_pred – Img_2_orig
Correlation	1.0	0.166
Intersection	64.790	14.817
Chi-Square	0.0	439.789
Bhattacharyya	0.0	0.792

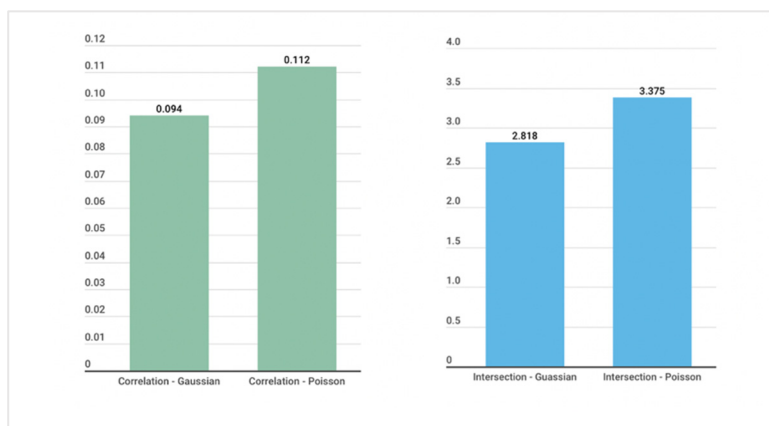
Fonte: O autor (2021)

De forma similar, comparou-se Base_pred – Base_pred e obteve os valores base, em seguida foi realizado a comparação entre as imagens Base_pred e Img_2_orig, apresentados na Figura 8.

Esta análise se da em duas etapas diferentes, a primeira etapa que iremos chamar de (Etapa 1) compara imagens tendo como modelo de referência uma imagem contendo ruídos (Base), os resultados representados através das Figuras 7 e Figura 8, de maneira idêntica a segunda etapa (Etapa 2) se utiliza de uma referência sem ruídos (Base_pred), disposto nas Figuras 9 e Figura 10.

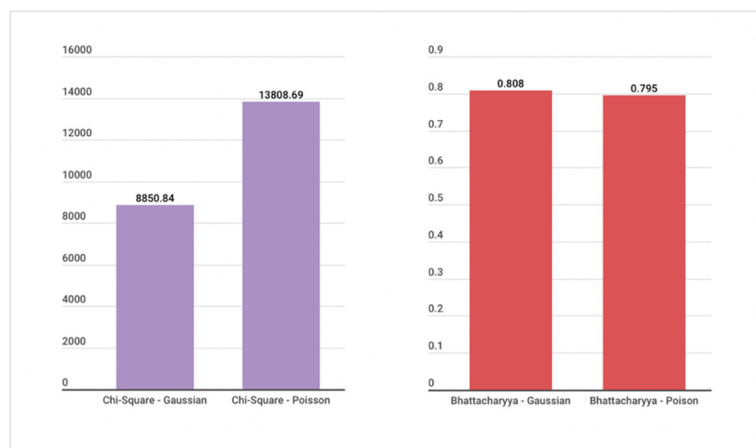
Na Figura 7 apresenta-se a apuração das demais comparações, de modo que foi utilizado dois métodos como forma de remoção de ruídos, *Gaussian* e *Poisson*, além dos métodos do histograma para aferir os valores *Correlation* e *Intersection*.

Figura 6 - Resultado base - img_2_pred – Interseciton e Correlation



Fonte: O autor (2021)

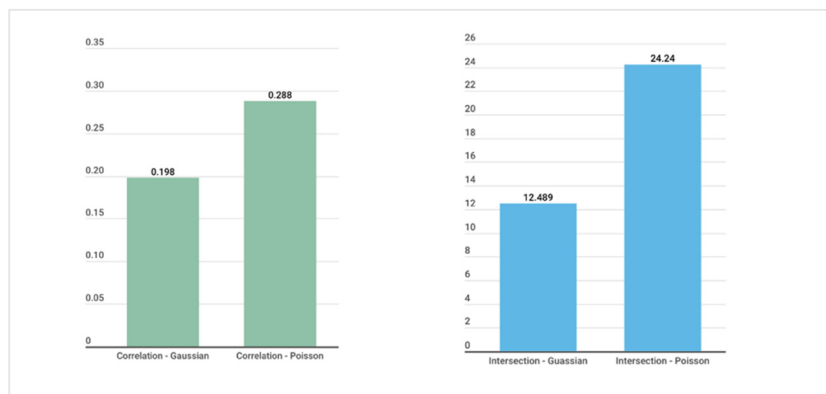
Figura 7 - Resultado base - img_2_pred – *chi-square* e *bhattacharyya*



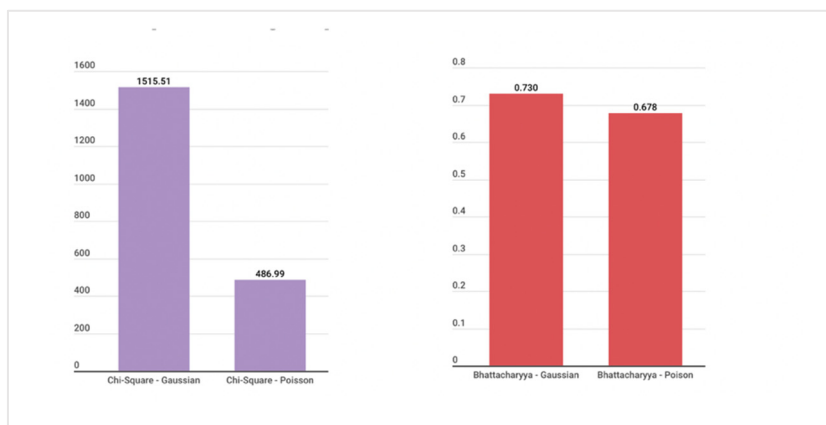
Fonte: O autor (2021)

Na Figura 9 apresenta-se um gráfico com valores obtidos na comparação da imagem Base – Img_2_pred, onde Base é a imagem real e Img_2_pred a imagem gerada pela GAN sem ruídos (*denoise*).

Figura 8 - Resultado base_pred - img_2_pred – *correlation* e *intersection*



Fonte: O autor (2021)

Figura 9 - Resultado base_pred - img_2_pred – *chi-square* e *bhattacharyya*

Fonte: O autor (2021)

Nesta etapa foi computado os resultados entre Base_pred e Img_2_pred, onde Base_pred representa a imagem original, no entanto retirado os ruídos e img_2_pred a imagem falsa sem ruídos.

6 DISCUSSÃO DOS RESULTADOS

No tocante a Cyclegan analisou-se que aumentando a quantidade de *epochs* houve uma melhora visual na geração das imagens falsas. Contudo, em torno de 100 *epochs* houve uma regressão visual das amostras. Não se trata de uma função exponencial. Em outras palavras, se dobrarmos o valor de *epochs*, irá dobrar a acurácia. Percebe-se que existe um limite para que se possa alcançar um resultado eficaz.

O primeiro experimento foi realizado com a base em relação a si mesma. Era de se esperar um ótimo resultado, visto que não havia diferença entre as imagens. Outrora, as outras imagens não possuíam uma nitidez adequada nem uma tonalidade correta (comparando com a real). Não se esperava um resultado satisfatório.

O valor de *Correlation* aumentou gradualmente conforme o número de *epochs* aumenta, porém no teste_4 este valor diminuiu. Observou-se também que o valor da *Intersection* aumentou até o teste_3, e novamente no teste_4 voltou a diminuir. Já os

outros métodos, *Chi-Square* e *Bhattacharyya* não passou de 0.7, que é um valor muito abaixo do esperado, todavia no geral notou-se que os valores melhoraram conforme o número de *epochs* aumentou.

Ao averiguar os resultados obtidos referente a Tabela 1, computado de uma imagem (Base) original contra uma imagem falsa (img_2_orig) contendo ruídos, observa-se que foi possível atingir uma acurácia sobre os seguintes métodos: *Correlation* 27,5%, *Intersection* 55,03%, *Bhattacharyya* 26,9%.

Analogamente, os resultados da Tabela 2 corresponde a imagem real, todavia sem a presença de ruídos (*denoise* ou Base_pred) e a imagem falsa contendo ruídos (img_2_orig), contudo os valores sobre a (Base_pred) são diferentes de (Base), visto que foi removido os ruídos, segue a acurácia alcançada sobre os métodos: *Correlation* 16,6%, *Intersection* 22,8% e *Bhattacharyya* 20,8% em comparação com os valores de referência.

Nota-se que houve uma diferença significativa entre (img_2_orig), (Base) bem como a imagem real sem ruídos (Base_pred), ao analisar o método *Correlation* foi aferido uma diferença de 10,6%, da mesma forma o método *Bhattacharyya* encontra-se com 6,1% de ganho, pode-se dizer que ao remover os ruídos (Base_pred) e ao comparar com outra imagem com ruídos, não se tem uma acurácia satisfatória.

A Figura 9 e a Figura 10 contém os valores entre Base e img_2_pred, ou seja, entre a imagem real com ruídos e a imagem falsa sem ruídos, ademais utilizando os métodos do Noise2Noise (*Gaussian*, *Poisson*) e os métodos do histograma (*Correlation*, *Intersection* etc.).

O método *Correlation Poisson* atingiu uma acurácia um pouco maior sobre o método *Gaussian*, algo em torno de 0,018%, da mesma maneira o método *Intersection Poisson* obteve 0,55% acima. Contudo ao equiparar com os valores da Base – img_2_orig ainda permanecem á baixo, o método *Chi-Square Gaussian* atingiu um valor superior em relação ao *Poisson*, como também sobre o *Chi-Square* da Tabela 1. Já o *Bhattacharyya Poisson* atingiu 0,013% superior, similarmemente se manteve 0,064% inferior a Tabela 1 Base e img_2_orig.

A Figura 11 e a Figura 12 mostra a análise Base_pred e img_2_pred, utilizando Noise2Noise, logo foram retirados os ruídos, todos os métodos *Poisson* foram mais promissores que o *Gaussian*. O método *Correlation Poisson* sobressaiu 0,09% a frente do *Correlation Gaussian*. O *Intersection Poisson* teve um ganho de 11,76% sobre o *Intersection Gaussian*.

Os resultados adquiridos ao interferir na reconstrução da imagem contendo ruídos se manteve uma acurácia superior, ou seja, se compararmos a imagem Base_pred com img_2_orig, assim como Base_pred e img_2_pred obteve os seguintes valores: *Correlation* (img_2_pred contra img_2_orig) se saiu com 0,172% acima, *Intersection* (img_2_pred contra img_2_orig) 9,42% mais alto.

7 CONSIDERAÇÕES FINAIS

Conclui-se que o estudo realizado atingiu o objetivo: identificar imagens submetidas a GANs. No entanto, foi realizado com apenas um modelo de GAN, a CycleGAN, pelo qual se obteve amostras excepcionais. De maneira similar, utilizou-se de um algoritmo (Noise2Noise) para a remoção de ruídos em imagens. Para atingir um resultado ainda melhor, com base nos resultados discutidos, o presente estudo abordou as características resultantes de uma imagem buscando aprender a criar uma imagem cada vez mais próxima de uma imagem real.

Analizou-se que as melhores amostras foram obtidas em torno de 50 *epochs* de treinamento. A partir disto sofreram distorções regressando o resultado visual, constatou-se também uma característica peculiar nas imagens submetidas a CycleGAN.

A presença de muitos ruídos, assim possibilitando distingui-las de imagens reais, para tanto foi utilizado o mecanismo Noise2Noise para remover boa parte dos ruídos, sendo o método *Poisson* alcançando melhor desempenho, notou-se uma melhora visual como também nos valores analisados com o Histograma.

Indubitavelmente o estudo foi capaz de provar quanto melhor o resultado obtido pela CycleGAN mais promissor será a amostra, logo removendo os ruídos pode-se

chegar ao ponto de não ser possível distinguir entre uma imagem real e uma imagem falsa, no quesito visual, apenas constatando o histograma para diferenciá-las.

Entretanto não foi possível alcançar uma acurácia excelente. O melhor resultado obtido se encontra nas Figuras 9 e 10, em torno de 30% no método *Correlation*, 28% *Intersection* e 40,5%. Vale ressaltar que tais valores são referentes ao modelo CycleGAN.

8 ETAPAS FUTURAS

Pretende-se aumentar o número de *epochs* e de amostras para 100 a 200, a fim de que se possa analisar não apenas 4 imagens, mas com centenas. Por outro lado, busca-se o viés de comparação entre Noise2Noise e NoiseOpenCV. Ademais, utilizar outras GANs para que se possa comparar os resultados com a CycleGAN, a exemplo, Pix2Pix entre outras.

Não foi possível realizar os experimentos com diferentes GANs, para uma continuidade do projeto poderia ser feito utilizando a Pix2Pix ou outra GAN para comparar com os resultados obtidos através da CycleGAN.

Outrossim, pode-se utilizar outros métodos do Histograma comparando os resultados, como também métodos de Noise2Noise, além de desenvolver aplicações para segurança computacional que detecte imagem que tenha sido produzida por GAN.

9 OUTRAS ATIVIDADES REALIZADAS

Reuniões nas quartas-feiras em um grupo fechado no Teamns para os alunos dedicados ao PIBIC, como também mestrado e doutorado em conjuntos dos orientadores. Este era o momento para relatar o que foi realizado durante a semana, novas ideias eram sugeridas assim como novos objetivos a serem realizadas para o próximo encontro.

Além das reuniões semanais, eventualmente me reunia com o meu orientador para tirar algumas dúvidas e alinhar os objetivos do projeto, da mesma forma com o suporte para incluir a GPU e corrigir erros.

REFERÊNCIAS

BUADES, Antoni; COLL, Bartolomeu; MOREL, Jean-Michel. **Non-Local Means Denoising**. p.208-212. July. 2017. Disponível em: http://dx.doi.org/10.5201/apol.2011.bcm_nlm Acesso em: 06 Julh. 2021.

EFROS, Alexei; ISOLA, Phillip; ZHOU, Thinghui; ZHU, Jun-Yan. **Image-to-Image Translation with Conditional Adversarial Networks**. V.3. p.1-17. Nov. 2018. Disponível em: <https://arxiv.org/abs/1611.07004>. Acesso em: 5 fev. 2021.

GOODFELLOW, Ian J; POUGET-ABADIE, Jean; MIRZA, Mehdi; XUM, Bing; WARDEFARLEY, David; OZAI, Sherjil; COURVILLE, Aaron; BENGIO, Yoshua. Generative Adversarial Nets. **Cornell University**. arXiv. v.1406.2661. p.1-9. June 2014. Disponível em: <https://arxiv.org/abs/1406.2661>. Acesso em: 2 fev. 2021.

GOYAL, Bhawna; DOGRA, Ayush; AGRAWAL, Sunil; SOHI, B.S; SHARMA, Apoorav. Image denoising review: From classical to state-of-the-art approaches. **ScienceDirect Elsevier**. V.55. p.220-244. March 2020.

IMAGE denoising, **OPENCV**. 2021. Disponível em: https://docs.opencv.org/3.4/d5/d69/tutorial_py_non_local_means.html. Acesso em: 28 Jane. 2021.

KASPHYAP, Madhav Mahesh; TAMBWEKAR, Anuj; MANOHARA, Krishnamoorthy; NATARAJAN, S. Speech Denoising without Clean Training Data: a Noise2Noise Approach. **Department of Computer Science and Engineering, PES University**. p.1-5. Apr 2018.

LEHTINEN, Jaakko; MUNKBERG, Jacob; HASSELGREN, Jon; LAINE, Samuli; KARRAS, Tero; AITALLA, Miika; AILA, Timo. Noise2Noise: Learning Image Restoration without Clean Data. **Cornell University**. p.1-12. Aug 2018. Disponível em: <https://arxiv.org/abs/1803.04189>. Acesso em: 06 Julh. 2021.

NGUYEN, Thanh Thi; NGUYEN, Quoc Viet Hung; NGUYEN, Coung M; NGUYEN, Dung; NGUYEN, Duc Thanh; NAHAVANDI, Saeid. Deep Learning for Deepfakes Creation and Detection. **Computing Research Repository (CoRR)**. v.1909.11573. p.1-12. 25 sep. 2019. Disponível em: <https://arxiv.org/abs/1909.11573>. Acesso em: 2 fev. 2021.

ODENA, Augustus, Christopher Olah, Jonathon Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. **International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017**. v.1. p.1-12. 20 june, 2017.

ZHANG, Fan; FAN, Linwei; FAN, Hui; ZHANG, Caiming. Brief review of image denoising techniques. **Visual Computing for Industry, Biomedicine, and Art**. p.1-12. July 2019.

ZHU, Jun-Yan; PARK, Taesung; ISOLA, Phillip; EFROS, Alexei A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. **Computer Vision Foundation (CVF)**. arXiv. v.1. p.2223-2232.