IDENTIFICAÇÃO, CLASSIFICAÇÃO E ANÁLISE DE DESEMPENHO DAS IMAGEM REAIS E GERADAS POR IA ATRAVÉS DE 3 REDES NEURAIS CONVOLUCIONAIS

Marcio Francisco da costa [1]

RESUMO

Este artigo trata da identificação, classificação e avaliação de desempenho de imagens reais e geradas por Inteligência Artificial, com ênfase na integridade e autenticidade. Redes neurais convolucionais foram utilizadas para treinar modelos visando alta precisão na análise dos dados do dataset. As imagens foram tratadas, selecionadas e obtidas da plataforma Kaggle. Diversas bibliotecas do Python foram empregadas no treinamento, sendo fundamentais para alcançar os resultados esperados. A aplicação dessas tecnologias melhorou significativamente a precisão na distinção entre imagens autênticas e geradas por IA, garantindo uma análise mais confiável e robusta.

Palavras-chave: Redes Neurais; Python; Kaggle.

ABSTRACT

This article deals with the identification, classification and performance evaluation of real and Artificial Intelligence-generated images, with an emphasis on integrity and authenticity. Convolutional neural networks were used to train models aiming for high accuracy in analyzing the dataset data. The images were processed, selected and obtained from the Kaggle platform. Several Python libraries were used in the training, being fundamental to achieving the expected results. The application of these technologies has significantly improved the accuracy in distinguishing between authentic and AI-generated images, ensuring a more reliable and robust analysis.

Keywords: Neural Networks; Python; Kaggle.

1 INTRODUÇÃO

No processamento digital uma imagem pode ser definida como uma função bidimensional f(x,y) onde as coordenadas espaciais do plano são x,y, com essas coordenadas podemos analisar os níveis de cinza de uma imagem ou em outras palavras a intensidade. Observe que uma imagem digital é composta por números finitos de elementos com sua localidade e valor específicos, que são chamados de pictóricos, elementos de imagens, pel e pixel, são termos utilizados para representar os elementos de uma imagem digital. Um dos elementos mais avançados dos nossos sentidos é a nossa visão, tanto que se as imagens exercem um papel importante na percepção humana. Mas se for feita uma análise de uma imagem utilizando a visão humana, pode-se observar algumas limitações no visual do

[1] Discente do Curso de Ciência da Computação da Universidade La Salle- Unilasalle, matriculada na disciplina TCC. E-mail: marcio.costa0023@unilasalle.edu.br , sob a orientação da Prof. Aline Duarte Riva. E-mail:aline.riva@unilasalle.edu.br, Data de entrega: julho de 2024.

espectro eletromagnético (EM), por outro lado alguns aparelhos de processamento cobrem todo o espectro EM, variando de ondas gama a ondas de rádio eles são capazes de trabalhar com imagens geradas por fontes que os humanos não associam muito bem, tais como ultrassom, microscopia eletrônica e imagens geradas por computador. **(Gonzalez e Woods, 2010)**

O processamento digital de imagens (PDI) é uma empreitada complexa, composta por uma série de tarefas entrelaçadas. Tudo começa com a captura de uma imagem, que geralmente reflete a luz na superfície dos objetos, realizada por um sistema de aquisição. Após a captura, por meio de um processo de digitalização, a imagem precisa ser convertida em uma forma adequada para ser manipulada por computadores. A classificação de imagens é amplamente beneficiada pelo Aprendizado de Máquina, especialmente em sua aplicação. A disponibilidade e qualidade dos exemplos rotulados estão diretamente ligadas ao desempenho dos classificadores. Geralmente, quanto mais exemplos rotulados temos, maior é a capacidade dos classificadores em prever corretamente. No entanto, na prática, é comum enfrentarmos dificuldades em obter grandes conjuntos de dados rotulados, o que pode ser caro e trabalhoso. Nesses casos, recorremos à abordagem de Data Augmentation (DA), que é essencialmente uma técnica que amplia a quantidade e diversidade dos dados disponíveis em um conjunto de dados. Em termos simples, ele cria novas variações dos dados existentes. Por exemplo, se temos imagens como dados de entrada, o Data Augmentation pode gerar novas imagens ao aplicar pequenas alterações, como rotações, espelhamentos, ampliações e diminuições, mudanças de contraste e assim por diante. Isso é útil porque aumenta a quantidade de dados de treinamento disponíveis para os modelos de aprendizado de máquina, sem a necessidade de coletar manualmente uma grande quantidade de novos dados. Em essência, o Data Augmentation nos ajuda a aproveitar ao máximo os dados que já temos, tornando-os mais diversos e, portanto, melhorando o desempenho dos modelos de aprendizado.**(ERCEMAPI, ano 2020).**

Como objetivo geral tem-se utilizado uma ferramenta de código aberto para realizar uma análise de um sistema de detecção de imagens. Nesta análise, será utilizado um conjunto de dados contidos em um dataset específico, onde essas imagens serão submetidas a um processo de tratamento. Através do treinamento da Inteligência Artificial, busca-se identificar, entre as imagens, quais delas são realmente autênticas e quais são geradas por uma IA. Utiliza-se a arquitetura ResNet, uma Rede Neural Convolucional (ConvNet - Convolutional Neural Network - CNN), para avaliar a eficácia e precisão do sistema na distinção entre imagens autênticas e sintéticas.

Como objetivos específicos este projeto visa realizar uma análise do sistema de detecção e classificação de imagens capaz de distinguir entre imagens reais e imagens geradas por IA. A análise será conduzida utilizando um conjunto de dados de imagens obtidas de um dataset específico, com o propósito de estabelecer parâmetros para a identificação e distinção entre conteúdos autênticos e sintéticos. Busca-se entender a precisão e a eficácia na identificação e classificação das imagens, utilizando Redes Neurais específicas, com foco na crescente necessidade de garantir a autenticidade dos dados visuais.

Utilizando técnicas de aprendizagem profunda, especialmente a arquitetura de rede neural convolucional ResNet, a análise proporciona uma visão detalhada das características das imagens, permitindo a diferenciação entre elas. Além disso, o estudo aborda a Aprendizagem Residual, uma abordagem que simplifica o treinamento de redes neurais profundas. A análise incluirá a avaliação de duas

arquiteturas de rede, uma versão simplificada das redes VGG e uma rede residual com conexões de atalho, com o objetivo de compreender melhor o processo de treinamento e a precisão na distinção entre imagens geradas por IA e imagens reais. Especificamente, serão analisadas três redes da arquitetura ResNet.

2 REFERENCIAL TEÓRICO

Nos últimos anos, temos presenciado um rápido avanço na geração de imagens sintéticas por meio da Inteligência Artificial (IA). Com a crescente capacidade de detectar fotos produzidas por IA, surge uma necessidade urgente de garantir a autenticidade dos dados associados a cada imagem. Diante desses desafios, torna-se imperativo desenvolver um sistema capaz de detectar e preservar a integridade e autenticidade dos dados de todas as fotos e imagens geradas pela IA.

A Escola de Comunicação e Artes da Universidade de São Paulo (USP) levanta uma reflexão interessante sobre o uso da IA na geração de imagens. Embora essa tecnologia possa ser uma fonte poderosa de criatividade, também apresenta desafios significativos. Por um lado, facilita a manipulação de obras de arte e a criação de vídeos e imagens extremamente realistas, inclusive com a capacidade de substituir rostos por expressões faciais sincronizadas, como observado nas deep fakes. Por outro lado, essa facilidade levanta questões sobre a autenticidade e confiabilidade das imagens produzidas, pois podem ser manipuladas de maneira tão convincente que se torna difícil distinguir o que é real do que é falso. Essa dualidade entre a capacidade de criar e a preocupação com a autenticidade torna a discussão em torno do uso da IA na geração de imagens ainda mais complexa e relevante.**(ESCOLA DE COMUNICAÇÃO E ARTE USP ano 2023)**

Explorando a relevância do tema mencionado, conduziu-se um estudo para avaliar a viabilidade de desenvolver um algoritmo utilizando a linguagem de programação Python, conhecida por sua alta versatilidade e ampla gama de bibliotecas disponíveis. O objetivo desse algoritmo era criar um sistema capaz de distinguir entre imagens geradas por IA e imagens reais. A motivação para essa distinção reside no fato de que imagens processadas por um computador, mas não por uma IA, são consideradas como imagens reais. Para alcançar esse objetivo, adotou-se um método de aprendizagem profunda baseado na arquitetura de rede neural convolucional ResNet. Essa abordagem permite uma análise mais precisa das características das imagens, possibilitando assim a diferenciação entre imagens geradas por IA e imagens autênticas.

Na Aprendizagem Residual, a ideia é simplificar o treinamento de redes neurais profundas usando funções residuais. Em vez de tentar que cada camada se aproxime diretamente do resultado desejado, as camadas são incentivadas a se aproximar da diferença entre o resultado desejado e a entrada original. Isso ajuda a resolver o problema de degradação, onde redes mais profundas têm desempenho inferior. Com as funções residuais, os ajustes nos pesos das camadas são mais diretos, facilitando o treinamento. Para o conjunto de dados ImageNet, foram testadas duas arquiteturas de rede. A primeira é uma versão simplificada das redes VGG, com camadas convolucionais predominantemente utilizando filtros de 3x3 e seguindo regras de design simples. Esta rede possui 34 camadas ponderadas, com uma complexidade computacional muito menor em comparação com as redes VGG originais. A segunda arquitetura é uma rede residual, baseada na versão

simplificada mencionada anteriormente. Esta rede inclui conexões de atalho, que permitem à rede se transformar em uma versão residual. Quando as dimensões de entrada e saída são iguais, são usados atalhos de identidade diretos. Quando as dimensões mudam, há duas opções: usar atalhos de identidade ou atalhos de projeção para combinar dimensões. Ambas as opções são aplicadas com uma passada de 2 quando os atalhos atravessam mapas de características de diferentes tamanhos.

Os algoritmos supervisionados lidam com dados em que os atributos estão relacionados e visam prever quais atributos influenciam a variável independente. Por outro lado, os algoritmos não supervisionados exploram os dados em busca de relacionamentos, agrupamentos ou distribuição sem a necessidade de variáveis independentes previamente definidas.

Neste projeto, vamos explorar algumas bibliotecas essenciais da linguagem Python, que desempenham um papel fundamental na análise de dados e aprendizado de máquina. Uma dessas bibliotecas é o Pandas, que é amplamente reconhecida como uma ferramenta essencial para manipulação de dados em Python, especialmente no campo da ciência de dados. O Pandas oferece uma interface amigável e consistente para trabalhar com conjuntos de dados de diferentes fontes, como planilhas e arquivos de texto. Ele simplifica tarefas como carregamento, limpeza, manipulação e análise de dados, permitindo operações avançadas, como alinhamento, mesclagem e agregação de informações. Além disso, o Pandas funciona como uma camada de abstração sobre outra biblioteca importante, o NumPy. O NumPy é uma biblioteca poderosa que facilita cálculos científicos e manipulação de dados em Python. Ele oferece recursos para análise numérica, operações matriciais e álgebra linear, entre outras funcionalidades. Ao utilizar o NumPy em conjunto com o Pandas, podemos abstrair a implementação de procedimentos matemáticos complexos, tornando mais fácil e eficiente a manipulação e análise de grandes conjuntos de dados. Essas duas bibliotecas são fundamentais para qualquer projeto de ciência de dados em Python, fornecendo as ferramentas necessárias para explorar, analisar e extrair insights valiosos de dados de forma eficaz e eficiente.

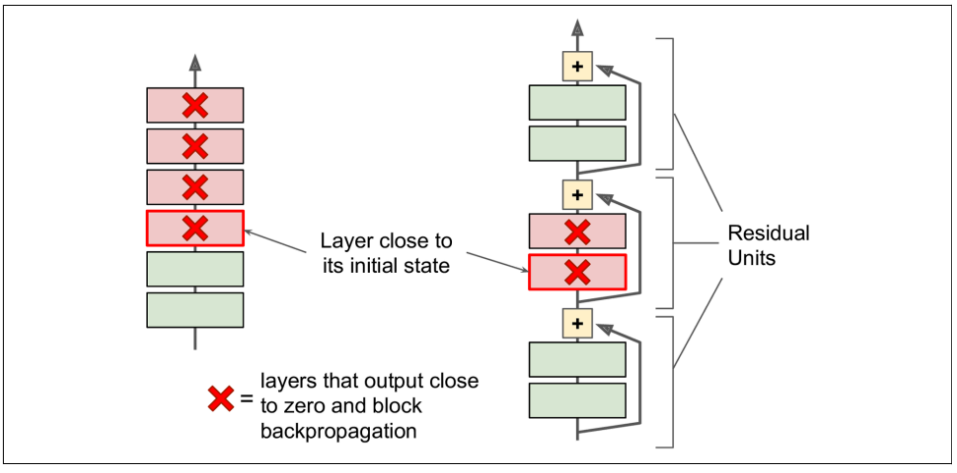
O PyTorch é uma biblioteca de código aberto em Python para aprendizado de máquina, especialmente para implementações de aprendizado profundo, como visão computacional e processamento de linguagem natural, para quem já está familiarizado com Python, oferecendo recursos como suporte à programação orientada a objetos e gráficos de computação dinâmica. O Pandas, como mencionado anteriormente, compartilha semelhanças com o NumPy, mas se destaca por sua estrutura de dados principal, o DataFrame. O DataFrame do Pandas é uma estrutura bidimensional que pode conter uma variedade de dados. Ele oferece recursos poderosos para lidar com dados tabulares, permitindo ao usuário manipular e analisar os dados de forma eficiente. Além disso, o Pandas fornece uma série de funções especializadas para calcular estatísticas e realizar operações de transformação nos dados, facilitando a análise e exploração dos conjuntos de dados. O Fastai é outra biblioteca de aprendizado profundo que oferece componentes de alto nível para obter resultados de forma rápida e fácil, enquanto também fornece componentes de baixo nível para construir abordagens personalizadas. Ele se destaca por sua arquitetura em camadas, que simplifica o uso de técnicas comuns de aprendizado profundo e processamento de dados. Com o Fastai, os usuários podem escrever menos código e obter resultados de maneira mais eficiente, aproveitando a flexibilidade do Python e do PyTorch. A biblioteca é projetada para ser de fácil acesso e produtiva, mas

também permite personalização, mantendo uma estrutura organizada que facilita a extensão e a modificação. **(Howard & Gugger, ano 2020)**

A biblioteca Matplotlib, assim como o Pandas e o NumPy, é apoiada pela NumFOCUS, uma organização sem fins lucrativos que promove a acessibilidade e a reprodução de computação científica e técnica.

Ao importar o módulo PyPlot do Matplotlib, podemos utilizar uma série de instruções para criar visualizações gráficas, como gráficos de linhas, de barras, de dispersão, entre outros. Essas visualizações são úteis para representar dados de forma clara e compreensível, auxiliando na análise e na comunicação de resultados em projetos de ciência de dados e análise exploratória. Com a ascensão da Rede Residual (ResNet) no desafio ILSVRC 2015 foi um marco na evolução da visão computacional. Desenvolvida por Kaiming He e sua equipe, a ResNet alcançou uma taxa de erro nos top-5 de menos de 3,6%, utilizando uma arquitetura de rede neural convolucional (CNN) profunda, com 152 camadas. O segredo dessa conquista está nas chamadas conexões de atalho. Essas conexões introduzem uma dinâmica única no processo de treinamento da rede neural. Ao adicionar a entrada original à saída da rede, ela é desafiada a modelar não apenas a função alvo \(h(x) \), mas também a diferença entre a saída esperada e a entrada original, \(f(x) = h(x) - x \). Esse fenômeno é chamado de aprendizagem residual e revela-se uma ferramenta poderosa na facilitação do treinamento de redes profundas. O cerne da eficácia da aprendizagem residual reside na sua capacidade de direcionar a rede para aprender apenas os detalhes essenciais da transformação desejada, em vez de tentar capturar todas as nuances da função \(h(x) \). Isso é particularmente útil na mitigação do problema do desaparecimento do gradiente, que historicamente limitava a capacidade de treinar redes profundas. Em termos práticos, as conexões saltadas oferecem uma solução elegante para o desafio de treinar redes profundas, permitindo a construção de arquiteturas cada vez mais complexas e eficazes. Ao proporcionar trajetos diretos para o fluxo de informação durante o treinamento, essas conexões ajudam a evitar a degradação do desempenho à medida que a profundidade da rede aumenta. Assim, a aprendizagem residual emerge como um paradigma fundamental no desenvolvimento de arquiteturas de redes neurais profundas, impulsionando avanços significativos na visão computacional e em campos relacionados. **(Geron Aurélien, ano 2019)**

Figura. 1 Apresenta a arquitetura da Rede neural profunda regular (esquerda) e rede residual profunda (direita)



Fonte: Geron Aurélien pág 377

Na Figura 1 está sendo ilustrando o problema de camadas "mortas" em redes neurais profundas e como as Unidades Residuais (Residual Units) ajudam a mitigar esse problema. As camadas com o "X" vermelho representam camadas que produzem saídas próximas de zero. Essas camadas "mortas" podem bloquear o fluxo de gradientes durante o processo de retropropagação, dificultando o treinamento eficaz da rede neural. Quando muitas camadas produzem saídas próximas de zero, o gradiente de erro não consegue retropropagar de maneira eficiente, resultando em camadas que permanecem em seu estado inicial e não aprendem. As Unidades Residuais são apresentadas como uma solução para esse problema. Cada unidade residual adiciona um caminho de atalho (skip connection) que permite que o gradiente flua diretamente para camadas anteriores. Mesmo se uma camada produzir uma saída próxima de zero, o caminho de atalho ajuda a manter a retropropagação eficiente, permitindo que a rede aprenda de maneira mais eficaz. As Unidades Residuais permitem que a rede aprenda funções de identidade mais facilmente, ajudando a evitar a degradação do desempenho à medida que a rede se torna mais profunda. **(Geron Aurélien, ano 2019)**

Para entender melhor como as imagens reais são identificadas e classificadas em comparação com aquelas geradas por Inteligência Artificial (IA), vamos detalhar alguns códigos e exemplos coletados através da plataforma Kaggle, usando um dataset específico. Através de imagens, vamos mostrar comparações entre diferentes redes neurais convolucionais (CNNs) que foram testadas em ciclos de treinamento. Primeiramente, trataremos as camadas adicionais do modelo por cinco épocas usando o método fit_one_cycle. Este é um método recomendado para treinar modelos sem realizar um ajuste fino (fine-tuning). Em essência, o que o fit_one_cycle faz é começar o treinamento com uma taxa de aprendizado baixa, aumentá-la gradualmente durante a primeira parte do treinamento e depois diminuí-la.

Já essa função learn.unfreeze() é usada para "descongelar" todas as camadas de um modelo de rede neural treinado. Quando treinamos modelos de deep learning, geralmente congelamos as camadas iniciais para preservar os pesos aprendidos durante um pré-treinamento em um grande conjunto de dados. Ao descongelar as camadas, permitimos que todas as camadas do modelo sejam ajustadas durante o treinamento.

Logo após, o descongelamento será utilizado a função learn.lr_find() do Fastai que executa a técnica de "learning rate finder" (encontrador de taxa de aprendizado). Essa técnica ajuda a encontrar a melhor taxa de aprendizado para o treinamento de um modelo, ela testa várias taxas de aprendizado em uma escala logarítmica e calcula a perda para cada uma delas.

Com o parâmetro suggest_funcs que aceita uma lista de funções que sugerem diferentes pontos ótimos na curva de perda versus taxa de aprendizado, nas quais são determinadas por:

- **Minimum**: Sugere a taxa de aprendizado correspondente à menor perda.
- **Steep** : Sugere a taxa de aprendizado onde a curva de perda é mais íngreme.
- **Valley** : Sugere a taxa de aprendizado no vale da curva de perda.
- **Slide** : Sugere a taxa de aprendizado onde a curva de perda começa a deslizar (flattening out). **(HOWARD e GUGGER, ano 2020; PASZKE, Adam; GROSS, Sam; MASSA**, **ano 2019**, **SMITH; ano 2017**).

3 DESENVOLVIMENTO

O projeto tem como objetivo a análise de um sistema voltado para a detecção e classificação de imagens, abrangendo tanto aquelas geradas por inteligência artificial (IA) quanto as reais, seja pintura ou editada por tratamento de luz e distorção de cores. Este sistema será composto por algoritmos implementados em linguagem de programação Python, os quais farão uso de diversas bibliotecas para realizar a detecção e classificação das imagens provenientes de um dataset. Dentre essas bibliotecas incluem, pandas, fastai, matplotlib.pyplot, numpy, opencv2, torch, sklearn entre outros, nos quais alguns deles serão utilizados para gerar gráficos de treinamentos, bem como demonstrações dos processos de tratamento dos dados coletados no conjunto desses. Uma funcionalidade crucial desse sistema será sua capacidade de identificar e distinguir as imagens que serão processadas e diagnosticadas pela IA, contribuindo assim para uma análise mais eficaz e precisa dos dados.**(KRICHEVSKY, SUTSKEVER, HINTON; ano 2017)**.

No campo da ciência de dados e aprendizado de máquina, é rotineiro lidar com dados armazenados em bancos de dados relacionais ou em arquivos de texto disponíveis online. Em um cenário ideal, os dados usados em algoritmos de aprendizado de máquina deveriam ser completos, do mesmo tipo e seguir uma distribuição normal. Porém, as coleções de dados nem sempre estão organizadas dessa maneira. Por isso, é necessário realizar uma etapa de pré-processamento, que envolve tratar os dados, como remover valores faltantes ou substituí-los por valores mais significativos, como média, mediana ou moda, além de padronizar a unidade de medida. Isso permite que os dados sejam melhor utilizados em algoritmos de machine learning.

É comum encontrar amostras de dados armazenadas de forma incorreta, antigas, pequenas demais ou até mesmo grandes demais. Grandes conjuntos de dados podem não ser garantia de qualidade, já que bancos de dados extensos, com milhões de registros, podem ter passado por modificações e atualizações ao longo do tempo, o que pode resultar em campos desatualizados ou dados truncados.

Após o pré-processamento dos dados, a escolha do algoritmo é a fase mais crucial e determinante para o sucesso do projeto de aprendizado de máquina. Essa etapa, também chamada de modelagem preditiva, envolve a seleção de um modelo estatístico capaz de prever resultados com base nos dados disponíveis. Ao explorar os dados, é fundamental escolher um modelo matemático que possa identificar as relações entre os atributos dos dados. Em geral, os algoritmos de aprendizado de máquina são divididos em duas categorias principais: supervisionados e não supervisionados.

Entre as etapas dos processos, vamos utilizar o dataset que contém as imagens reais e imagens geradas por IA, organizadas em pastas para facilitar o carregamento e a classificação. Logo após serão usados o `ImageDataLoaders` do Fastai para carregar as imagens, e aplicar transformações apropriadas e dividir os dados em conjuntos de treino e validação. Utilizará o método `fit_one_cycle`para treinar as camadas adicionais por cinco épocas.

Para garantir uma análise mais completa, vamos comparar diferentes arquiteturas de redes neurais convolucionais, tais como ResNet34, ResNet101, e ResNet152. Este fluxo de trabalho detalhado permite comparar diferentes modelos

de CNN, avaliando qual arquitetura oferece o melhor desempenho na tarefa de identificar e classificar imagens reais versus geradas por IA. Nestas avaliações incluirão métricas como acurácia, matriz de confusão e análise de perdas, proporcionando uma visão clara do desempenho de cada rede **(Howard e Gugger; ano 2020.)**

O modelo apresentado da rede neural convolucional é complexa, projetada para processar imagens de entrada de 224x224 pixels com 3 canais (RGB) e um batch size de 64. A arquitetura do modelo inicia-se com uma camada convolucional que aplica 64 filtros de tamanho 7x7 e um stride de 2, resultando em uma saída de dimensões 64x64x112x112. Em seguida, um MaxPooling com um tamanho de 3x3 e um stride de 2 reduz a dimensão para 64x64x56x56.

Este modelo inclui diversos blocos residuais, cada um composto por múltiplas camadas convolucionais (Conv2d), normalização (BatchNorm2d) e ReLU, mantendo as dimensões através de conexões de atalho (skip connections). Conforme a rede se aprofunda, a profundidade e as dimensões dos mapas de características aumentam progressivamente de 64 até 2048 filtros.

Após as camadas convolucionais, o modelo utiliza AdaptiveAvgPool2d e AdaptiveMaxPool2d para reduzir a dimensão espacial para 1x1, resultando em uma saída de tamanho 64x2048x1x1. A saída do pooling é então achatada para formar um vetor de 64 x 4096. Para regularização e normalização, são usadas BatchNorm1d e Dropout.

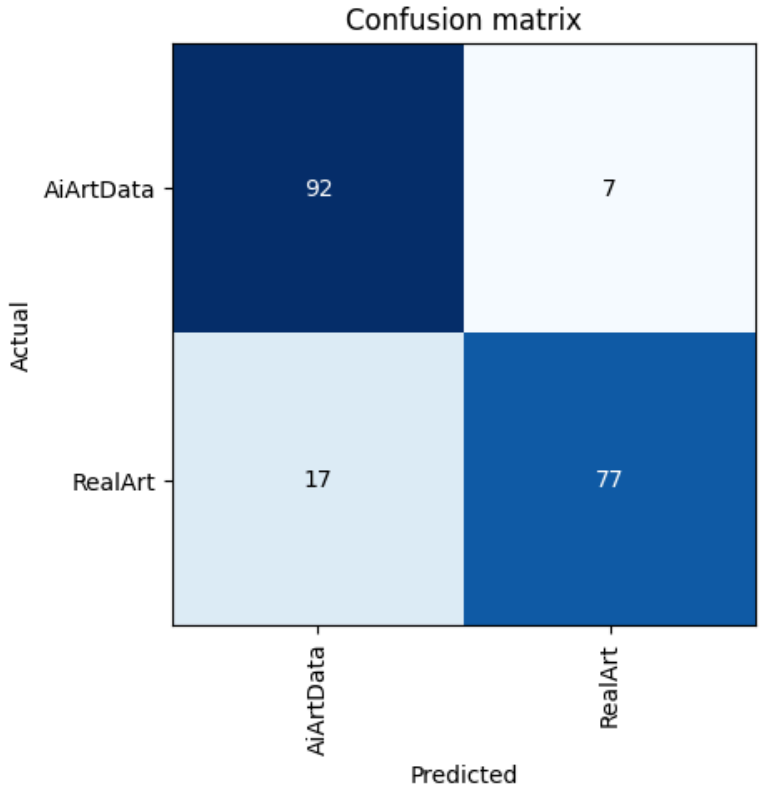
As camadas lineares do modelo incluem uma camada com 2.097.152 parâmetros, que transforma a entrada para uma dimensão de 64 x 512, seguida de uma camada final que produz a saída de 64 x 2, correspondente às classes do problema de classificação.

O modelo possui um total de 44.607.552 parâmetros, dos quais 2.212.736 são treináveis e 42.394.816 são congelados. O congelamento até o grupo de parâmetros 2 sugere que as camadas iniciais são utilizadas como extratoras de características fixas.

Para a otimização, o modelo utiliza o método Adam, um método eficiente baseado em gradiente. A função de perda utilizada é a CrossEntropyLoss, que achata as saídas.

Durante o treinamento, são utilizados callbacks para monitorar e registrar o progresso, transformar dados e visualizar resultados. Os callbacks empregados incluem TrainEvalCallback, CastToTensor, Recorder e ProgressCallback. **(GOODFELLOW; BENGIO; COURVILLE; ano 2017, Kaiming; Xiangyu; Shaoqing; ano 2016.)**

Figura 2 - Matriz de confusão ResNet34



Fonte: Autor

Como pode ser observado na matriz de confusão apresentada na Figura 2, essa ferramenta é utilizada para avaliar a performance de um modelo de classificação das duas categorias em questão: imagens criadas por inteligência artificial (AiArtData) e imagens criadas por humanos (RealArt).

A matriz de confusão revela os seguintes dados: 92 instâncias de AiArtData foram corretamente classificadas (Verdadeiro Positivo - TP), 17 instâncias de RealArt foram incorretamente classificadas como AiArtData (Falso Positivo - FP), 7 instâncias de AiArtData foram erroneamente classificadas como RealArt (Falso Negativo - FN) e 77 instâncias de RealArt foram corretamente identificadas (Verdadeiro Negativo - TN).

Com base nessa matriz, podemos calcular as seguintes métricas para avaliar a eficácia do modelo: A precisão, que representa a proporção de previsões corretas entre todas as feitas para uma classe, foi de 84,4% para as imagens geradas por IA, indicando que 84,4% das amostras identificadas como geradas por IA estavam corretas.

O recall, que mede a proporção de verdadeiros positivos entre todos os casos que realmente pertencem à classe, foi de 92,9% para as imagens geradas por IA, significando que 92,9% das amostras reais foram corretamente identificadas pelo modelo.

O F1-Score, média harmônica da precisão e do recall, proporcionando um balanço entre os dois, foi de 88,4% para as imagens geradas por IA, mostrando que

o modelo mantém um equilíbrio adequado entre precisão e recall. A acurácia, que é a proporção de todas as previsões corretas em relação ao total, atingiu 87,6%, indicando que a maioria das previsões estava correta.

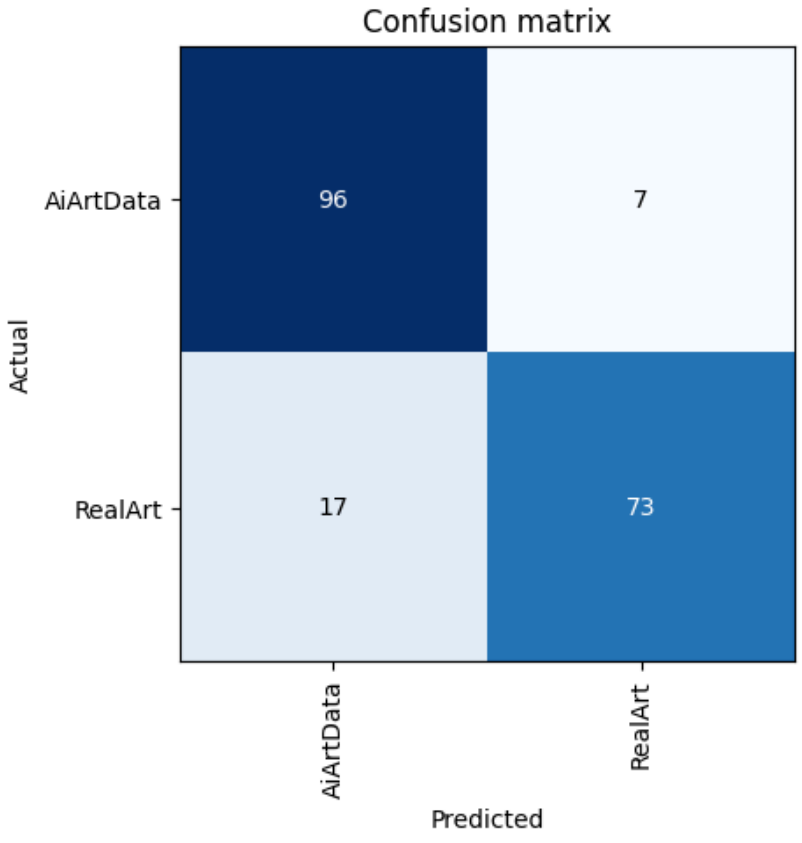
O suporte, que indica o número de ocorrências reais de cada classe no conjunto de dados, foi de 99 instâncias para as imagens geradas por IA e de 94 para RealArt.

A média macro, que é a média aritmética das métricas calculadas para cada classe sem considerar o número de instâncias de cada classe, foi de 88,6% para precisão, recall e F1-Score, indicando que o modelo mantém um desempenho equilibrado para ambas as classes.

A média ponderada, que considera o suporte de cada classe, proporcionando uma média ponderada das métricas, também foi de 88,6% para precisão, recall e F1-Score, refletindo o bom desempenho geral do modelo.

Esses resultados demonstram um excelente desempenho do modelo na diferenciação entre as imagens geradas por IA e as reais. Com uma taxa de acerto de 87,6%, o modelo é capaz de identificar corretamente a maioria das amostras. As métricas de precisão, recall e F1-Score para ambas as categorias - geradas por IA e reais - são bastante similares, mostrando um equilíbrio na capacidade do modelo de fazer previsões precisas para ambas as classes e de minimizar os erros de classificação.

Figura 3 - Matriz de confusão ResNet101



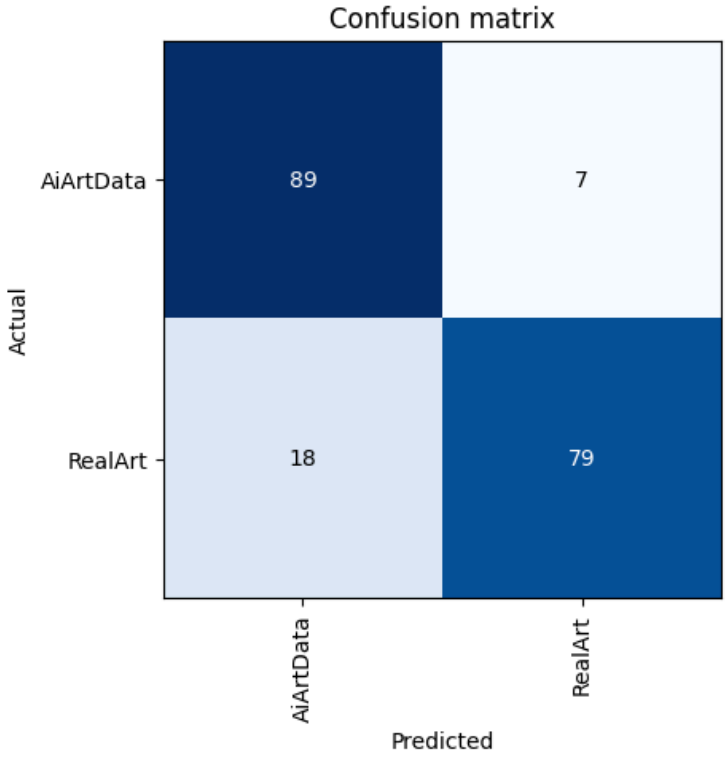
Fonte: Autor

Na matriz de confusão apresentada na Figura 3, observamos um panorama da performance do modelo de classificação em comparação com a Figura 2. Houve uma notável melhoria na identificação correta de imagens AiArtData, com um aumento nos verdadeiros positivos e uma redução nos falsos negativos. Isso indica uma maior precisão do modelo ao distinguir imagens geradas por inteligência artificial. Por outro lado, ocorreu um aumento nos falsos positivos e uma diminuição nos verdadeiros positivos para RealArt, sugerindo uma leve queda na precisão do modelo ao classificar imagens criadas por humanos.

Especificamente, o modelo na categoria AiArtData identificou corretamente 96 instâncias (verdadeiros positivos) e errou em 7 (falsos negativos). Para RealArt, acertou 73 instâncias (verdadeiros negativos) e errou em 17 (falsos positivos).

Com base nestes resultados, calculamos métricas essenciais de desempenho do modelo. A precisão para AiArtData foi de 84,9%, indicando que a maioria das predições para essa classe estava correta. O recall para AiArtData foi de 93,2%, mostrando que o modelo identificou a maioria das instâncias reais dessa categoria. O F1-score para AiArtData, que combina precisão e recall, foi de 88,8%, refletindo um bom equilíbrio entre as duas métricas. A acurácia geral foi de 87,5%, destacando a eficácia global do modelo para ambas as classes.Esses resultados validam a robustez do modelo em diferenciar entre imagens geradas por IA e por humanos, evidenciando sua capacidade precisa e equilibrada de identificar a maioria das amostras corretamente.

Figura 4 - Matriz de confusão ResNet152



Fonte: Autor

Na comparação da Figura 4 da tabela de confusão com as anteriores (Figuras 2 e 3), algumas diferenças podem ser observadas nos resultados do modelo de classificação entre as classes AiArtData e RealArt:

Precisão para AiArtData: Na terceira tabela, a precisão para AiArtData é ligeiramente menor, aproximadamente 83,2%, enquanto nas tabelas anteriores foi em torno de 84,9% a 84,4%. Isso indica que o modelo teve uma taxa ligeiramente maior de falsos positivos para AiArtData nesta execução.

Recall para AiArtData: O recall para AiArtData na terceira tabela é similar às tabelas anteriores, mantendo-se elevado em torno de 92,7% a 93,2%. Isso significa que o modelo continua sendo eficaz em identificar a maioria das instâncias verdadeiras de AiArtData.

Precisão para RealArt: A precisão para RealArt na terceira tabela é consistente com as anteriores, variando entre 91,2% e 91,8%, mostrando que o modelo mantém a capacidade de classificar corretamente a maioria das instâncias de RealArt.

Recall para RealArt: Assim como a precisão, o recall para RealArt na terceira tabela é similar às tabelas anteriores, variando entre 81,1% e 81,4%, indicando que o modelo continua identificando a maioria das instâncias verdadeiras de RealArt.

Acurácia e F1-score: A acurácia geral na terceira tabela varia entre 87,4% e 87,9%, alinhando-se com as tabelas anteriores que variaram de 87,5% a 87,6%. Os valores de F1-score também são consistentes, com variações mínimas entre 86,3% e 88,8%.

Em suma, embora haja uma pequena variação na precisão para AiArtData na terceira figura da tabela, o modelo mantém um desempenho geral robusto e consistente na diferenciação entre imagens geradas por inteligência artificial e imagens reais. As métricas de recall, precisão, acurácia e F1-score mostram que o modelo é capaz de realizar previsões precisas para ambas as classes com uma performance estável ao longo das diferentes execuções.**(MITCHELL ano 1997, POWERS ano 2011, MURPHY ano 2012)**

6 CONSIDERAÇÕES FINAIS

Com base nos resultados obtidos após testes realizados com um ciclo de treinamento de 10 épocas, seguido pelo descongelamento e um ciclo subsequente de 30 épocas com 10 repetições para cada rede neural utilizada, podemos concluir que o modelo demonstra uma eficácia consistente na diferenciação entre imagens

geradas por inteligência artificial (AiArtData) e imagens criadas por humanos (RealArt). As métricas de precisão, recall, F1-score e acurácia revelam um desempenho robusto e estável ao longo das diferentes execuções.

Embora tenha sido observada uma pequena variação na precisão para AiArtData na Figura 4 da matriz de confusão, com uma leve queda em relação às tabelas anteriores, o recall para AiArtData manteve-se elevado e consistente. Isso indica que o modelo continua identificando a maioria das instâncias verdadeiras dessa classe, apesar das variações nas predições. Para RealArt, tanto a precisão quanto o recall na terceira figura da tabela são comparáveis às tabelas anteriores, demonstrando a eficácia contínua do modelo em classificar corretamente a maioria das instâncias de imagens criadas por humanos.

A acurácia geral e os valores de F1-score também permanecem estáveis, destacando a capacidade do modelo de fazer previsões precisas para ambas as classes. Portanto, os resultados consolidam a robustez do modelo na tarefa de classificação de imagens entre as duas categorias, evidenciando sua confiabilidade mesmo diante de variações nos dados de entrada. Essa consistência reforça a aplicabilidade do modelo em cenários práticos que exigem precisão na diferenciação entre imagens geradas por IA e imagens reais, consolidando sua utilidade em diversas aplicações.

Com todos resultados obtidos através dos testes realizados nos parâmetros estabelecidos durante o ciclo de treinamento, identificou-se que os dados obtidos na experimentação podem ser aprimorados. Aumentando o número de épocas e treinando as redes neurais com uma quantidade maior de iterações podendo levar a um desempenho superior ao observado nos testes atuais.

REFERÊNCIAS

Anthony Scopatz um dos membros e Leah Silen diretor executivo da **Empresa de Código aberto NUMFOCUS**. NumFOCUS. Disponível em: https://numfocus.org/. Acesso em: 27 maio 2024.

Bird J, Jordan Departamento de Ciência da Computação, Nottingham Trent University (Reino Unido). **Artigo:** CIFAKE Classificação de imagens e explicável Identificação de imagens sintéticas geradas por IA. 19 de janeiro de 2024 Disponivel: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10409290

C.Gonzalez, Rafael, E. Woods Richard, **Processamento Digital de Imagem**. 3ª Edição. Tradução Cristina Yamagami e Leonardo Piamonte, Cidade: São Paulo, Editora: Pearson Education do Brasil , ano 2010.

Cornell University Boston(EUA) Computer Science **Artigo**: Aprendizado residual profundo para reconhecimento de imagens, Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 10 Dez 2015 Disponível: https://arxiv.org/pdf/1512.03385.pdf.

ESCOLA DE COMUNICAÇÃO E ARTE USP (Brasil). **Artigo** : Uso de Imagens pode ser Perigoso, Mas Incentiva Criatividade Lopes Rosiane do LAC Laboratório Agência de Comunicação. São Paulo, 28 de Set. 2023.Disponível em: https://www.eca.usp.br/noticias/pos/uso-de-ia-na-geracao-de-imagens-pode-ser-perigoso-mas-incentiva-criticidade.

Fastai. "**ImageDataLoaders.from_folder.**" fastai documentation. Disponível em: https://docs.fast.ai/vision.data.html#ImageDataLoaders.from_folder.

Fastai: **A Layered API for Deep Learning". fastai** documentation. Acesso em: data de acesso, Disponível em: https://docs.fast.ai/
Gugger Sylvain. **A Política de 1 ciclo (the 1Cycle Policy)** postado em 07 abr.de 2018 Disponível em: https://sgugger.github.io/the-1cycle-policy.html.

Géron Aurélien, **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow Conceitos, Ferramentas e Técnicas para a Construção de Sistemas Inteligentes.** versão em inglês 1ª Edição, Editora: Alta Books, ano 2019.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, **Aaron. Deep Learning**. Cambridge, MA: MIT Press, 2016.

Howard, Jeremy., & Gugger, Sylvain. **Deep Learning for Coders with Fastai and PyTorch: AI Applications Without a PhD**. O'Reilly Media. 29 jun.de 2020.

HOWARD, Jeremy; GUGGER, Sylvain. **Fastai: A Layered API for Deep Learning**. Information, v. 11, n. 2, p. 108, 2020. Disponível em: https://www.mdpi.com/2078-2489/11/2/108. Acesso em: 27 maio 2024.

Hakim Nukman,**Keggle AI vs Real using Fast.AI** codigo e documentation, dateset Acesso em 29 mar. 2024. Disponível em: https://www.kaggle.com/code/alkidiarete/ai-vs-real-using-fast-ai

HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. **Deep Residual Learning for Image Recognition**. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 2016. p. 770-778.

José Demisio Simões da Silva ,Título**.Uso De Redes Neurais Em Visão Computacional e Processamento de Imagens,** cidade São José dos Campos publicação, INPE-11279-PRP/242 no ano de 2004

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. I**mageNet classification with deep convolutional neural networks**. *Communications of the ACM*, v. 60, n. 6, p. 84-90, 2017.

MITCHELL, T. M. ***Machine Learning*. 1st ed. New York: McGraw-Hill,** 1997. 414 p

MURPHY, K. P. ***Machine Learning: A Probabilistic Perspective*. Cambridge**: The MIT Press, 2012. 1098 p.

Netto Amilcar, Maciel Francisco, **Python Para Data Science e Machine Learning Descomplicado.** versão em inglês 1ª Edição. Cidade Rio de Janeiro, Editora Atlas Books, ano de 2021

Python Software Foundation. "**os.listdir()". Python 3.10.2** documentation. Acesso em: data de acesso, Disponível em:https://docs.python.org/3/library/os.html#os.listdir

Python Software Foundation. **"pathlib — Object-oriented filesystem paths". Python 3.10.2** documentation. Acesso em: data de acesso, Disponível em: https://docs.python.org/3/library/pathlib.html

PASZKE, Adam; GROSS, Sam; MASSA, Francisco; et al. **PyTorch: An Imperative Style, High-Performance Deep Learning Library**. In: Advances in Neural Information Processing Systems. v. 32, p. 8024-8035, 2019. Disponível em: <https://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library>. Acesso em: 27 maio 2024.

POWERS, D. M. W. ***Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation*. Journal of Machine Learning Technologies**, v. 2, n. 1, p. 37-63, 2011.

Stevens, E., Antiga, L., & Viehmann, T. **Deep Learning with PyTorch.** 1ª Edição em Inglês Editora Manning Publications 4 ago.de 2020.

SOCIEDADE BRASILEIRA DA COMPUTAÇÃO (Brasil). ERCEMAPI: Coleção Livros de Minicursos. Piauí, 19 set. 2020. SBCOPENLIB,**Capítulo 3 Utilização de Técnicas de Data Augmentation em Imagens: Teoria e Prática** (pág. 48 e pág. 72). Disponível em: https://sol.sbc.org.br/livros/index.php/sbc/catalog/book/48 Acesso em: 10 set. 2020.

SMITH, Leslie N. **Cyclical Learning Rates for Training Neural Networks**. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2017. p. 464-472. Disponível em: <https://arxiv.org/abs/1506.01186>. Acesso em: 27 maio 2024.

TORCH, Py. **PyTorch Documentation**. Disponível em: https://pytorch.org/docs/stable/index.html. Acesso em: 14 jun. 2024.

Universidade Federal do Ceará UFC(Brasil) Tutorial:um guia rápido para você entender agora os fundamentos do PyTorch, Sandra Lemos Agente de comunicação. Ceará, 19 de Jan. 2022. Disponível em : https://www.insightlab.ufc.br/tutorial-pytorch-um-guia-rapido-para-voce-entender-agora-os-fundamentos-do-pytorch/.

Universidade Federal Fluminense (UFF) Ciência de dados, **Visualização de dados com Python,** postado no ano de 2020. Disponível em: https://cienciadedadosuff.github.io/cursos/notebooks/caderno-3.html

VanderPlas, J, **Python Data Science Handbook Essential Tools for Working with Data.** versão em Inglês Estados Unidos, Editora: O'Reilly Media, no ano de 2016.

Wes McKinney, **Python para Data Analysis.** versão em Inglês 1ª Edição Estados Unidos Editora: O'Reilly Media, ano de 2018.