UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JOÃO PEDRO VIEIRA RODRIGUES

USO DE APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE FACES FALSAS GERADAS POR INTELIGÊNCIA ARTIFICIAL

BAURU
Outubro/2023

R696u

Rodrigues, João Pedro Vieira
USO DE APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE FACES FALSAS GERADAS POR INTELIGÊNCIA ARTIFICIAL / João Pedro Vieira Rodrigues. -- Bauru, 2023
37 p. : tabs., fotos

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (Unesp), Faculdade de Ciências, Bauru
Orientadora: Kelton Augusto Pontara da Costa

1. Aprendizado de Máquina. 2. Cibersegurança. 3. Capsule Neural Network. 4. Detecção de faces. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Ciências, Bauru. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

João Pedro Vieira Rodrigues

Uso de Aprendizado de Máquina para detecção de faces falsas geradas por Inteligência Artificial

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof. Dr. Kelton Augusto Pontara da Costa
Orientador
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Ciência da Computação

Prof. Dr. Simone das Graças Domingues Prado
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Ciência da Computação

Me. Juliana da Costa Feitosa
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Ciência da Computação

Bauru, 18 de novembro de 2023.

Dedico esta monografia a todos que já passaram ou estão presentes em minha vida, meu mais sincero obrigado.

Agradecimentos

Agradeço primeiramente aos meus pais, Sandra e Sebastião, que me dão toda a força e auxílio do mundo, sem eles eu não seria quem sou hoje. Eles sacrificam suas vontades e desejos pelo meu futuro, eu os amo mais que tudo e nenhuma palavra escrita aqui pode explicar esse amor, sou eternamente grato pelas pessoas incríveis que ambos são. Junto deles agradeço ao meu irmão Rafael, que é um exemplo de pessoa e que me inspira a continuar seguindo em frente independente da situação.

Agradeço também aos meus professores, que durante a graduação sempre se empenharam em ensinar da melhor maneira possível. Em especial gostaria de agradecer o meu orientador Kelton, que me deu uma oportunidade de explorar o mundo acadêmico e me deu todo o suporte nessa jornada, tanto neste trabalho quanto na iniciação científica. Também agradeço a professora Andrea, que sem ela não seria possível estar realizando este trabalho.

Agradeço meus amigos na universidade: Arthur, Nathan, Danilo, Renato e Ronaldo, por toda a ajuda durante a graduação e de serem responsáveis por momentos inesquecíveis que vivi dentro e fora da universidade, todos eles estão no meu coração.

Agradeço minha namorada, Isadora, por acreditar em mim mesmo quando eu não acredito, por me ajudar nos momentos mais difíceis, por estar do meu lado em qualquer situação, por enxergar coisas em mim que eu não vejo, por ser essa pessoa que eu amo. Obrigado.

Por fim, agradeço a todas as pessoas que de uma maneira ou outra influenciaram na minha vida e fizeram eu chegar aonde eu cheguei, meu mais sincero obrigado.

"*Na minha cabeça, eu sou o melhor. Se não pensarmos assim não temos ambição. Eu tenho de pensar que, na minha profissão, eu sou o melhor. Posso não ser, mas na minha cabeça eu sou o melhor.*"

Cristiano Ronaldo

Resumo

O avanço no campo da Inteligência Artificial, especialmente na área de aprendizado profundo, possibilitou a criação de rostos humanos por meio de modelos de redes neurais, como a Generative Adversarial Network (GAN). Entretanto, esse avanço levantou preocupações significativas em relação à segurança, especialmente nos contextos de biometria e autenticação digital. Diante desse problema, este trabalho concentra-se na aplicação de uma técnica específica de aprendizado de máquina conhecida como Capsule Neural Network (CapsNet). Esta abordagem se mostrou promissora para o processamento de imagens e será comparada a outras técnicas, como Local Binary Pattern, Res-Net e Gram-Net. A análise detalhada desses métodos permitirá uma compreensão mais aprofundada de suas capacidades e limitações, contribuindo para o desenvolvimento de sistemas mais seguros e eficazes no contexto da inteligência artificial e suas aplicações práticas. A avaliação crítica dessas técnicas é essencial para aprimorar a segurança e a autenticidade em sistemas que dependem de reconhecimento facial e autenticação digital.
Palavras-chave: aprendizado profundo, segurança digital, rostos falsos.

Abstract

The advancement in the field of artificial intelligence, especially in the area of Deep Learning, have made it possible to create human faces using neural network models, such as the Generative Adversarial Network (GAN). However, this progress has raised significant security concerns, especially in the context of biometrics and digital authentication. Faced with this problem, this work focuses on the application of a specific machine learning technique known as Capsule Neural Network (CapsNet). This approach has shown promise for image processing and will be compared to other techniques such as Local Binary Pattern, Res-Net and Gram-Net. The detailed analysis of these methods will allow a deeper understanding of their capabilities and limitations, contributing to the development of safer and more effective systems in the context of artificial intelligence and its practical applications. A critical evaluation of these techniques is essential for improving security and authenticity in systems that rely on facial recognition and digital authentication.

Keywords: deep learning, digital cecurity, fake faces.

Lista de figuras

Lista de tabelas

Lista de abreviaturas e siglas

CapsNet	*Capsule Neural Network*
GAN	*Generative Adversarial*
CNN	*Convolutional Neural Network*
Res-Net	*Residual Neural Network*
Gram-Net	*Residual Neural Network com uma matriz de Gram*
LBP-Net	*Local Binary Pattern*

Sumário

1 Introdução

Nos últimos anos, a área de identificação de imagens criadas por Inteligência Artificial vem se destacando (ZHANG et al., 2022). Com os avanços da tecnologia, a criação dessas imagens falsas tornou-se cada vez mais comum em nossa realidade. Elas podem ser usadas para realizar ataques cibernéticos, campanhas de desinformação, criação de notícias falsas e de perfis falsos em redes sociais, entre outras aplicações maliciosas.

Para lidar com esse problema, pesquisadores e cientistas têm se empenhado a explorar técnicas inovadoras de aprendizado de máquina (DIRIK et al., 2007) com o propósito de encontrar uma solução efetiva para essa situação. Essas técnicas analisam características específicas das imagens, como distorções, padrões de cores e movimentos irregulares, a fim de identificar sinais de manipulação digital (DEHNIE; SENCAR; MEMON, 2006).

Entre as técnicas que surgem a partir de diferentes pesquisas, uma que se mostra promissora nesse contexto é a *Capsule Neural Network* (CapsNet) (DONG; LIN, 2019). Elas se mostraram excepcionais na identificação de padrões em imagens de forma eficiente, indo além das capacidades das tradicionais *Convolutional Neural Networks* (CNNs). Sua principal característica é a capacidade de identificar características sutis, como rotação de elementos dentro da imagem, algo que muitas vezes escapa à detecção das CNNs tradicionais (HOLLÓSI; POZNA, 2018). Esse avanço é fundamental para enfrentar a sofisticação crescente das manipulações de imagem.

Em conclusão, a aplicação da CapsNet representa um marco significativo na luta contra a disseminação de informações falsas e manipuladas na era digital. Sua capacidade de identificar padrões complexos e sutilezas nas imagens contribui para a preservação da integridade das informações e, por conseguinte, para a confiança do público nas imagens que circulam na internet. À medida que continuamos a enfrentar desafios crescentes no cenário da desinformação online, a pesquisa contínua e o desenvolvimento de tecnologias como a CapsNet são cruciais para proteger a autenticidade e a confiabilidade das informações em nossa sociedade digitalmente conectada.

1.1 Problema

A disseminação das imagens geradas por Inteligência Artificial tem desempenhado um papel multifacetado na sociedade contemporânea. Por um lado, essas inovações tecnológicas têm sido uma fonte de inspiração e criatividade para artistas e designers, proporcionando ferramentas inovadoras que ampliam os limites da expressão visual. No entanto, essa mesma tecnologia também tem sido explorada por indivíduos mal-intencionados, que a utilizam de

maneira prejudicial para aplicar golpes e disseminar informações falsas pela Internet, constituindo uma ameaça significativa à integridade da informação online (KHALAF; VAROL, 2019).

Esses criminosos aproveitam-se da rapidez com que a informação se propaga nas redes sociais e em portais de notícias, explorando a vulnerabilidade das plataformas digitais para disseminar desinformação de forma ampla e eficaz. A disseminação de notícias falsas e imagens manipuladas representa um desafio complexo para a sociedade, minando a confiança do público e potencialmente causando danos irreparáveis.

Diante dessa ameaça crescente, torna-se imperativo desenvolver técnicas avançadas para identificar e combater essas imagens geradas artificialmente. A criação de métodos eficazes de detecção é essencial para minimizar o impacto dessas atividades criminosas. Essas técnicas podem envolver a análise detalhada de padrões, texturas e distorções presentes nas imagens, bem como a implementação de algoritmos de aprendizado de máquina capazes de discernir entre imagens autênticas e falsificadas.

Além disso, é fundamental promover a conscientização pública sobre os riscos associados à disseminação de informações falsas e manipuladas. A educação digital desempenha um papel crucial na capacitação dos usuários para identificar e questionar a autenticidade das informações encontradas online, fortalecendo assim a resiliência da sociedade contra essas ameaças.

Em suma, a sociedade enfrenta um desafio complexo e dinâmico na era digital, onde a inovação e a criminalidade coexistem em um espaço virtual interconectado. A colaboração entre pesquisadores, empresas de tecnologia e autoridades regulatórias é essencial para desenvolver estratégias abrangentes que possam proteger a integridade da informação online e garantir um ambiente digital seguro e confiável para todos os usuários.

1.2 Justicativa

Este trabalho tem como seus pilares o uso de aprendizado de máquina e a segurança da informação, dois assuntos em alta nos dias atuais e que necessitam de pesquisas relacionadas para poder agir contra as ameaças que o avanço da informática trouxe consigo.

A técnica de aprendizado de máquina utilizada será a CapsNet pois é promissora para essa finalidade, tornando os sistemas mais robustos e menos suscetíveis a falsos positivos, trazendo assim maior segurança e diminuindo o impacto negativo que as atividades criminosas podem causar a sociedade.

1.3 Objetivos

Este trabalho tem como objetivo avaliar o desempenho da CapsNet no problema de reconhecer imagens criadas por Inteligência Artificial.

1.3.1 Objetivos Específicos

- Estudar e implementar uma CapsNet;
- Analisar um banco de imagens geradas Inteligência Artificial;
- Treinar a rede neural para conseguir reconhecer imagens falsas;
- Interpretar os resultados para entender o nível de eficácia dessa técnica; e
- Tirar conclusões sobre os resultados obtidos a fim de entender se esse modelo é efetivo ou não nesse contexto.

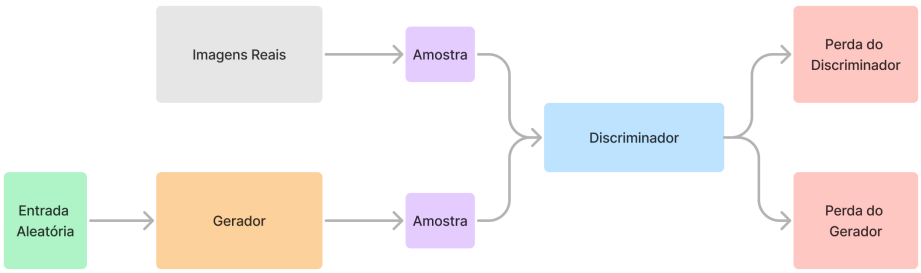
2 Fundamentação Teórica

O objetivo desta seção é apresentar e explicar todos os conceitos que envolvem a realização desse trabalho. Na Seção 2.1 é explicado a rede neural que cria os rostos falsos, na Seção 2.2 é abordado a técnica usada para realizar a identificação de imagens neste trabalho, na Seção 2.3 é destinado as funções de ativação da rede, na Seção 2.4 é mostrado as métricas de avaliação e por fim a Seção 2.5 é comentado sobre as bibliotecas usadas para implementar a rede.

2.1 Generative Adversarial Network

Generative Adversarial Network ou apenas GAN, é um modelo de aprendizado de máquina que possui duas redes neurais na sua implementação, o gerador e o discriminador, como pode ser visto na Figura 1. Essas duas redes trabalham em conjunto para gerar dados artificiais que são indistinguíveis dos dados reais. Essa rede foi concedida inicialmente por Ian Goodfellow e seus colegas de pesquisa no ano de 2014 (GOODFELLOW et al., 2014).

Figura 1 – Arquitetura de uma GAN.



Fonte: Elaborado pelo autor.

O gerador na GAN cria novos dados que parecem ser autênticos, enquanto o discriminador avalia essa autenticidade, determinando se são reais ou falsos. O processo de treinamento das GANs é composto por duas etapas principais. Na primeira etapa, o discriminador é alimentado por entradas reais e entradas geradas pelo gerador. Já na segunda etapa, o gerador desenvolve novos dados para tentar passar pelo discriminador sem ser percebido, tornando-os cada vez mais realistas.

Essa mecânica entre as duas redes tem como objetivo fazer o gerador melhorar a sua capacidade de produção de dados cada vez mais reais, enquanto o discriminador aprimora sua capacidade analítica, a fim de ser capaz de distinguir os reais dos artificiais.

O impacto dessa tecnologia é que ela tem sido cada vez mais utilizada para criar rostos falsos de alta qualidade. Antes dessa rede, a criação de rostos falsos era limitada apenas pelos métodos tradicionais como computação gráfica, que, em grande parte, produziam resultados pouco satisfatórios.

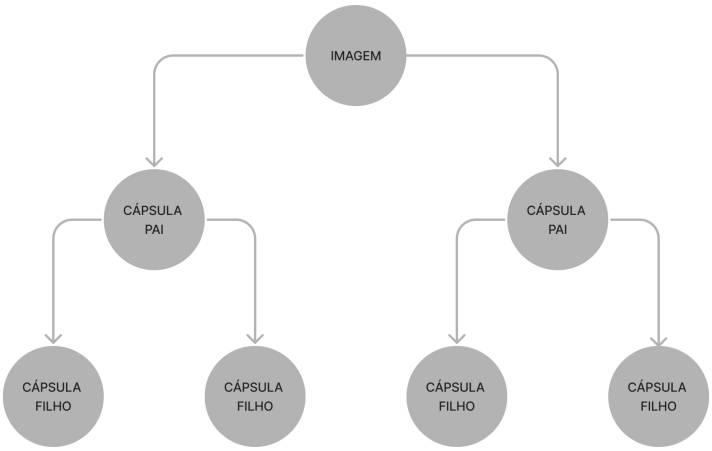
2.2 Capsule Neural Network

Proposto inicialmente em 2017 por Geoffrey E. Hilton, a *Capsule Neural Network*, ou CapsNet, é uma rede neural que usa o conceito de cápsulas para o seu funcionamento (SABOUR; FROSST; HINTON, 2017).

Essa rede utiliza as cápsulas e suas propriedades para fornecer uma solução na detecção de partes de objetos em uma imagem e representar relacionamentos espaciais entre essas partes. Isso significa que ela é capaz de reconhecer o mesmo objeto, como um rosto, em diversos ângulos diferentes e com o número típico de características (olhos, nariz, boca), mesmo que não tenha sido treinada especificamente para esta tarefa.

Esse modelo possui uma arquitetura composta de cápsulas pais e filhos, como pode ser observado na Figura 2, que por sua vez compõem uma imagem completa de um objeto.

Figura 2 – Esquema de cápsulas pai e cápsulas filho.



Fonte: Elaborado pelo autor.

Outras técnicas de processamento de imagem, como a *Convolutional Neural Network*, usam a função de *Pooling* para reduzir as dimensões da imagem como largura e altura enquanto mantém as características principais dela (AJIT; ACHARYA; SAMANTA, 2020). O problema dessa abordagem é que muita vezes o Pooling faz com que a imagem perca algumas de suas propriedades (SABOUR; FROSST; HINTON, 2017) e a cápsula entra como uma solução para este problema.

2.2.1 Cápsulas

As cápsulas pode ser definidas como um agrupamento de neurônios que são ativados individualmente e possuem diversas propriedades de um objeto, como por exemplo sua cor, posição, orientação, textura, largura. Esses neurônios juntos produzem um vetor μ que representa um objeto existente na imagem (DONG; LIN, 2019). Esse vetor μ usa sua magnitude e orientação para armazenar informações das cápsulas:

- **Magnitude** (m) = a probabilidade de que algo exista. A magnitude do vetor é um valor entre 0 e 1 que indica a probabilidade de que um objeto específico de um todo exista e que foi detectado em uma imagem.
- **Orientação** (θ) = o estado das propriedades do objeto. A orientação do vetor representa o estado das propriedades de uma parte desse todo. Esta orientação mudará se uma das propriedades mudar.

Com essas informações, a rede é capaz de reconhecer um objeto mesmo quando ele muda o seu ângulo, já que apenas a sua orientação muda e a magnitude, que representa apenas a existência dele, continua do mesmo tamanho. Isso facilita a detecção de objetos em diferentes cenários durante o treinamento.

O fato da saída de uma cápsula ser um vetor com orientação, torna possível usar um mecanismo chamado *roteamento dinâmico* para garantir que a saída de uma cápsula seja enviada para a cápsula parente mais apropriada na próxima iteração da rede.

2.2.2 Roteamento Dinâmico

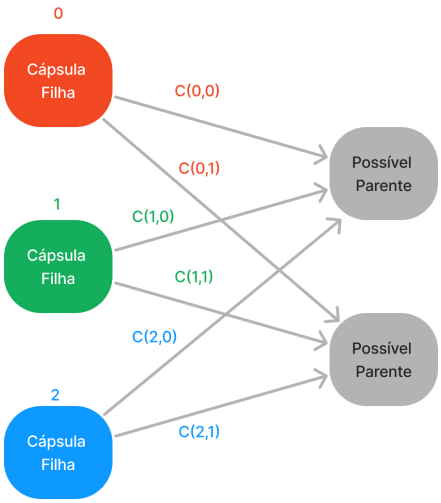
O Roteamento Dinâmico é um processo que tem como objetivo encontrar os melhores pares entre a saída de uma cápsula e as entradas da próxima camada de cápsulas.

Este método possibilita que elas se comuniquem umas com as outras e estabeleçam como os dados se movem através delas, de acordo com as mudanças em tempo real nas entradas e saídas entre as camadas da rede. Ou seja, não importa que tipo de imagem de entrada a CapsNet enxerga, o roteamento dinâmico garante que a saída de uma cápsula seja enviada para a cápsula parente apropriada (SABOUR; FROSST; HINTON, 2017).

No começo de uma rede, as cápsulas iniciais não possuem um parente para transmitir a sua saída, pois atuam como entrada para a próxima camada de cápsulas pais. Portanto, cada cápsula começa com uma lista de possíveis pais, que são todas as cápsulas da próxima camada.

Durante essa fase, é usado um valor chamado coeficiente de acoplamento $\boldsymbol{c}$, esse valor representa a probabilidade de uma informação de saída de uma cápsula filha específica seja transmitida para uma cápsula-pai. Um nó filho com dois pais possíveis começará com coeficientes de acoplamento iguais para ambos. A Figura 3 mostra o uso dos coeficientes de acoplamento

Figura 3 – Coeficiente de acoplamento entre as cápsulas.



Fonte: Fonte: Elaborado pelo autor.

Antes de passar os vetores para as próximas camadas, a CapsNet adiciona uma função de compressão na saída dos vetores. Essa função recebe o nome de *squashing*, que garante que o menor vetor seja comprimido para um número perto de zero e o maior vetor é comprimido para um número um pouco abaixo de um.

Isso garante que o módulo do vetor de saída que mostre a probabilidade do objeto representado pela cápsula está, de fato, presente na imagem atual.

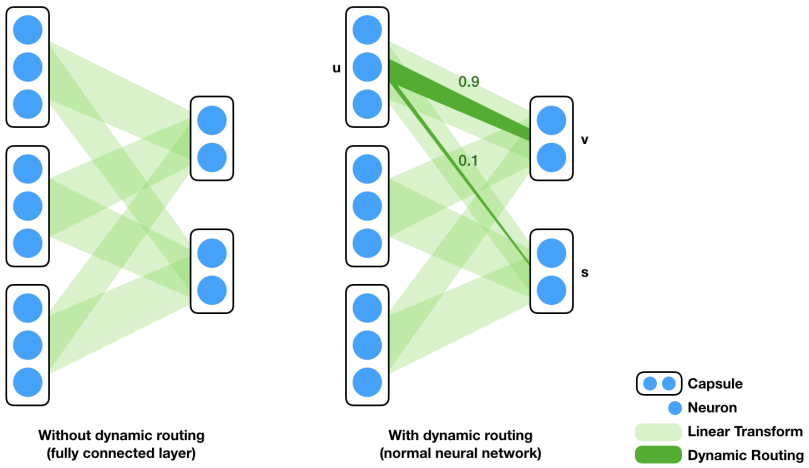
A função de *squashing* pode ser representada pela Equação 2.1 (HOLLÓSI; POZNA, 2018):

$$\mathrm{v}_j = \frac{\left\|\mathrm{s}_j\right\|^2 \mathrm{s}_j}{1 + \left\|\mathrm{s}_j\right\|^2 \left\|\mathrm{s}_j\right\|} \tag{2.1}$$

Nessa equação, *v* é o vetor de saída da cápsula *j* e *Sj* é o total de entradas. Depois de aplicada essa função nas saídas, a próxima interação pode começar.

A Figura 4 demonstra a diferença entre a rede com e sem roteamento dinâmico.

Figura 4 – Diferença da rede com e sem o roteamento dinâmico



Fonte: (UNIVERSITY OF ARIZONA, s.d.)

.

Esse processo inteiro é usado para atualizar o coeficiente de acoplamento das cápsulas. O *roteamento por acordo* entra depois dessa atualização para realizar a conexão entre filhos e pais apropriados (DONG; LIN, 2019).

2.2.3 Roteamento por Acordo

O Roteamento por Acordo funciona seguindo um conjunto de etapas para garantir que as cápsulas filhas sejam acopladas com as cápsulas pais. Em todas as camadas, exceto a primeira, ocorre os seguintes passos:

- Para cada pai possível, uma cápsula filha calcula um *vetor de previsão*, *û*, que é uma função de seu vetor de saída, *u*, vezes uma matriz de peso, *W*, como pode ser observado na Equação 2.2:

$$\mathrm{s}_j = \sum_i \mathrm{c}_{ij} \mathrm{u}_{\frac{j}{i}}, \mathrm{u}_{\frac{j}{i}} = \mathrm{W}_{ij} \mathrm{u}_i \tag{2.2}$$

Onde *Cij* é o coeficiente de acoplamento que é atualizado pelo roteamento dinâmico;

- Se o vetor de predição tem um produto escalar grande com o vetor de saída da cápsula pai, v, então esses vetores são considerados concordantes e o coeficiente de acoplamento entre aquele pai e a cápsula filha aumenta. Simultaneamente, o coeficiente de acoplamento entre aquela cápsula filho e todos os outros pais diminui;

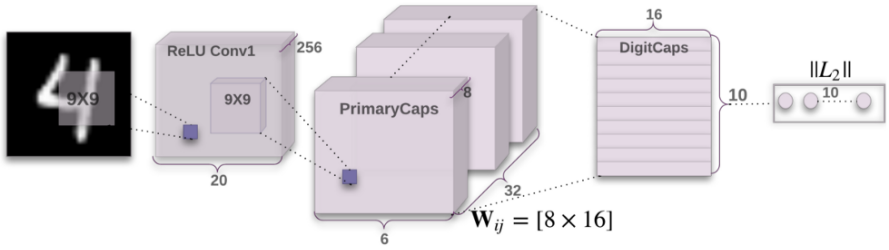
- Se a orientação dos vetores de saída das cápsulas em camadas sucessivas estiver alinhada, eles concordam que devem ser acoplados e as conexões entre eles são fortalecidas. Os coeficientes de acoplamento são calculados por uma função softmax (Seção 2.3.1) que opera sobre os acordos entre cápsulas e os transforma em probabilidades tais que a soma entre um filho e seus possíveis pais deve ser igual a um. O acordo de duas cápsulas é dado por um produto simples como é visto na Equação 2.3:

$$\mathrm{a}_{ij} = \mathrm{v}_j * \mathrm{u}_{\frac{j}{i}} \tag{2.3}$$

- Por fim, em todas as conexões entre uma cápsula filha e todas as cápsulas pai possíveis, os coeficientes de acoplamento devem somar 1.

A Figura 5 demonstra como é a representação da CapsNet implementada especificamente para o banco de dados *Modified National Institute of Standards and Technology* ou *MNIST*, que é um banco de dados de números escritos a mão.

Figura 5 – Representação da CapsNet completa para a base de dados MNIST



Fonte: (SABOUR; FROSST; HINTON, 2017)

.

A entrada da rede é uma imagem do banco de dados MNIST que representa o número quatro, logo depois temos a camada de convolução que é usada para extrair características da imagem de entrada (Seção 2.2). Imediatamente após essa parte, encontra-se as camadas das cápsulas primárias, onde tamanho do vetor ativo de cada cápsula no *DigitCaps* indica a presença de uma instância.

Pode-se observar também a matriz de peso W descrita nesta seção na parte *PrimaryCaps*. A saída é a classificação da imagem de entrada, que neste caso podem ser dez casos diferentes (dígitos de 0 a 9).

2.3 Funções de ativação

Uma função de ativação é uma função utilizada para ajudar a rede neural a entender e aprender padrões durante o seu treinamento (DING; QIAN; ZHOU, 2018). Elas são empregadas em diferentes contextos e possuem diferentes abordagens. Nesta seção 2.3, vamos abordar duas que foram usadas neste trabalho: na Seção 2.3.1 será explicado com mais detalhes a função *Softmax* utilizada no roteamento por acordo e na Seção 2.3.2 será abordado a função de ativação *Sigmoid*.

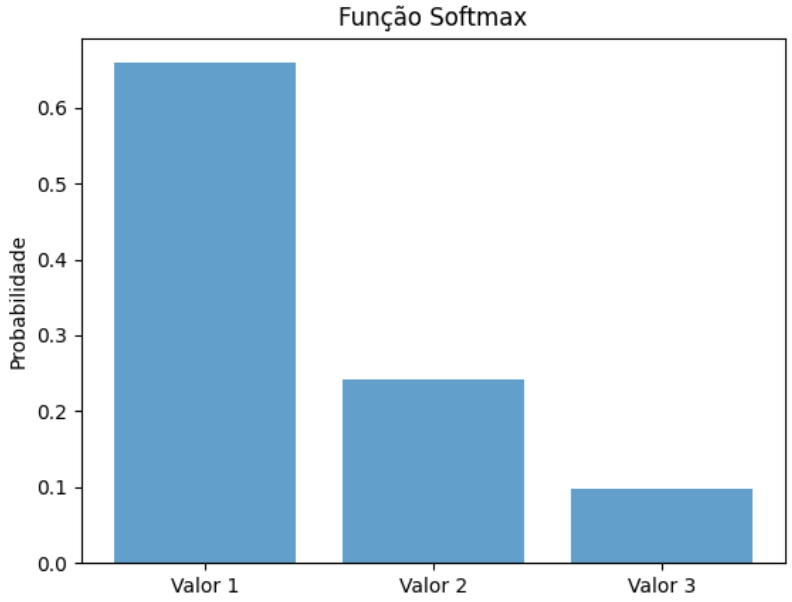
2.3.1 Softmax

A função de ativação *softmax* é uma função do tipo exponencial e é usada em problemas de classificação multi-classe, onde a saída da rede precisa ser transformada em uma probabilidade, a Equação 2.4 representa a função *softmax*.

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{2.4}$$

Nessa equação, o numerador representa a exponencial de x_i e o denominador é igual a somatória de todas as exponenciais x_j para todas as classes. Essa função normaliza as saídas para que a soma das probabilidades de todas as classes seja igual a um, tornando mais fácil a visualização da probabilidade. A Figura 6 mostra graficamente a função *softmax* para três valores distintos (2.0, 1.0 e 0.1).

Figura 6 – Gráfico da função *softmax*



Fonte: Elaborado pelo autor.

Nesse trabalho ela foi utilizada para ajudar a rede neural a visualizar se várias características de um rosto dado na entrada estão de fato na imagem. Ela retorna os valores das probabilidades dessas características.

2.3.2 Sigmoid

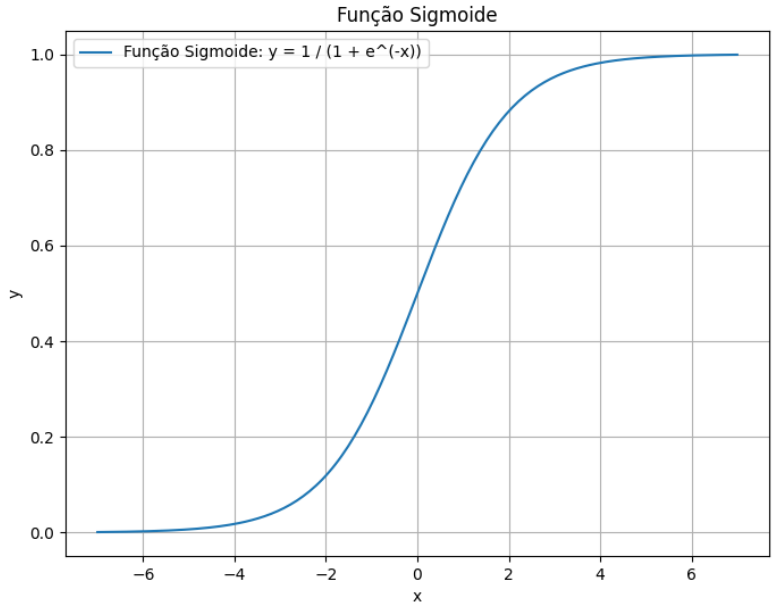
A função de ativação *sigmoid*, ou função logística, tem como objetivo transformar qualquer valor real em um intervalo de zero a um (DING; QIAN; ZHOU, 2018), sua fórmula pode ser vista na Equação 2.5.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.5}$$

Nessa equação, x é a entrada da função, ou seja, um número real qualquer. Quando o número x é grande e positivo, a função retorna um valor próximo de um e quando x é grande e negativo, a função retorna um valor próxima de zero. Isso é ideal para classificação binária, quando a saída pode assumir dois valores.

A Figura 7 mostra o gráfico da função *sigmoid* e os valores que ela pode assumir em um intervalo [-7,7].

Figura 7 – Gráfico da função *sigmoid*



Fonte: Elaborado pelo autor.

Nesse trabalho ela foi utilizada para ajudar a rede neural a predizer se um rosto dado na entrada é verdadeiro ou falso. Ela retorna um valor entre zero e um que significa a probabilidade da entrada ser falsa.

2.4 Avaliação

O campo de aprendizado de máquina vem passando por grandes avanços nos últimos anos devido ao crescimento do poder computacional e à disponibilidade de grandes conjuntos de dados. Porém, desenvolver uma rede neural que seja eficaz para seu propósito não é uma tarefa simples e para ajudar nesse desafio é comum usar métricas de avaliação, tais como acurácia e precisão (M.; M.N, 2015). A Seção 2.4.1 tem como objetivo explicar a acurácia e a Seção 2.4.2 explicará a precisão.

A matriz de confusão é usada em problemas de classificação binária e ela possui quatro entradas principais:

- Verdadeiro Positivo: caso em que o modelo acertou a previsão da classe positiva;
- Verdadeiro Negativo: caso em que o modelo acertou a previsão da classe negativa;
- Falso Positivo: caso em que o modelo errou a previsão da classe positiva; e
- Falso Negativo: caso em que o modelo errou a previsão da classe negativa.

A Tabela 1 mostra como a matriz de confusão é formada.

Tabela 1 – Matriz de Confusão para Classificadores Binários

	Classe Real	
Classe Predita	**Positiva**	**Negativa**
Positiva	Verdadeiros Positivos (VP)	Falsos Positivos (FP)
Negativa	Falsos Negativos (FN)	Verdadeiros Negativos (VN)

Fonte: Elaborado pelo autor.

A partir dessa matriz é possível visualizar as fórmulas das métricas e implementá-las no contexto ideal.

2.4.1 Acurácia

A acurácia de uma rede neural representa a proporção de previsões assertivas em relação ao total de previsões. Ela pode ser entendida através da Equação 2.6.

$$acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.6}$$

Nessa equação, o numerador é a soma dos Verdadeiros Positivos com Verdadeiros Negativos e o denominador é a soma dos Verdadeiros Positivos, Verdadeiros Negativos, Falsos Positivos e Falsos Negativos.

A acurácia foi empregada nesse trabalho pois é uma medida direta da performance da rede, sendo fácil de entender seu funcionamento, é fundamental para decisões binárias e facilitam o acompanhamento do aprendizado e aprimoramento do modelo.

2.4.2 Precisão

A precisão de uma rede neural representa a proporção de verdadeiros positivos (previsões corretas de casos que são positivos) em relação ao total de previsões positivas no geral (previsões verdadeiras positivas mais falsas positivas). Ela pode ser entendida através da Equação 2.7.

$$precisão = \frac{VP}{VP + FP} \tag{2.7}$$

Nessa equação, o numerador são os Verdadeiros Positivos e o denominador é a soma dos Verdadeiros Positivos com os Falsos Positivos.

Para este trabalho a precisão foi usada pois é uma medida informativa, que foca na qualidade das previsões da rede e também auxilia na tomada de decisão. Junto com a acurácia pode-se entender de maneira completa e simples as melhores estratégias a seguir durante o treinamento e teste da rede neural.

2.5 Python

Python é uma linguagem de programação de alto nível interpretada e versátil, que possui diversas aplicações na área da computação. Tornou-se popular devido a sua sintaxe clara e legível, que possibilita a manutenção do código de maneira simplificada.

Ela também é conhecida por conta da sua variedade de biblioteca e frameworks como Django, Flask, NumPy, Pandas, cada uma delas com o seu propósito e curvas de aprendizado (PYTHON SOFTWARE FOUNDATION, 2001). Neste trabalho foi utilizada a biblioteca de visualização de dados *Matplotlib* e também a biblioteca de aprendizado de máquina Pytorch. Aqui estão alguns pontos principais dessa linguagem:

- Sintaxe simples que permite uma fácil compreensão do código;
- Diversas bibliotecas que ajudam o Python a fazer tarefas em diferentes áreas; e
- linguagem interpretada, que significa que o código é executado linha por linha, facilitando a depuração.

2.5.1 PyTorch

PyTorch é uma biblioteca de código aberto muito utilizada na área de aprendizado de máquina e aprendizado profundo nos contextos de visão computacional e processamento

de linguagem natural. Foi desenvolvido pela Meta AI e atualmente faz parte da The Linux Foundation (THE LINUX FOUNDATION, 2000).

A estrutura de dados mais básica dessa biblioteca é o *Tensor*. Ele se assemelha aos arrays do NumPy, porém ele conta com o suporte de operações em GPUs, permitindo aceleração de hardware para cálculos intensivos. Com o Tensor podemos usar todas as ferramentas do PyTorch. As ferramentas do PyTorch incluem:

- TorchNN: o módulo torch.nn oferece uma estrutura para construir redes neurais. Ele contém classes e métodos para definir camadas, funções de ativação como acurácia e precisão, inicializações dos pesos da rede e funções de perda desses pesos;
- TorchOptim: o módulo torch.optim oferece algoritmos de otimização como, por exemplo, o Gradient Descent. Esses algoritmos são essenciais para treinar redes neurais. Eles permitem o ajuste automático dos pesos da rede para minimizar a função de perda;
- TorchAutograd: o módulo torch.autograd oferece o mecanismo de diferenciação automática do PyTorch. Ele rastreia as operações realizadas nos tensores e automaticamente calcula os gradientes durante o treinamento das redes neurais; e
- TorchVision: uma extensão do PyTorch que oferece modelos pré-treinados e métodos para transformação de imagens que auxiliam em tarefas de visão computacional, facilitando o desenvolvimento de aplicações relacionadas a essa área.

O PyTorch foi empregado neste trabalho devido à facilidade de implementar a CapsNet usando suas principais ferramentas.

2.5.2 Matplotlib

Matplotlib é uma biblioteca de visualização e representação de dados em Python que permite a criação de gráficos e *plots* para representar dados de maneira direta e completa (THE MATPLOTLIB DEVELOPMENT TEAM, 2012). Esta biblioteca é usada para análise exploratória de dados e apresentação de resultados científicos.

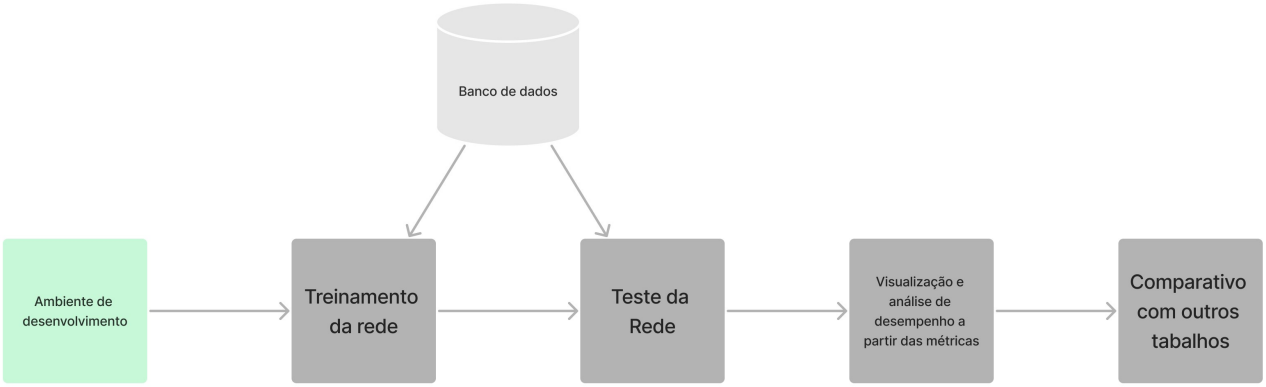
Ela foi utilizada neste trabalho com a finalidade de ajudar na criação e entendimento dos gráficos durante o treinamento, validação e teste.

3 Metodologia

Nesta seção será comentado o desenvolvimento do trabalho como um todo: a Seção 3.1 tem como objetivo apresentar e explicar a base de dados usada no trabalho, a Seção 3.2 vai apresentar o ambiente de desenvolvimento usado, a Seção 3.3 explica o treinamento, validação e teste da rede.

A Figura 8 mostra o diagrama usado para o desenvolvimento do trabalho.

Figura 8 – Diagrama do trabalho



Fonte: Elaborado pelo autor.

3.1 Base de Dados

A base de dados escolhida para esse projeto consiste em um banco de dados com 140 mil imagens falsas e reais, retirada do site Kaggle, uma plataforma mantida pela Google LLC.

A Figura 9 mostra duas imagens de rostos retiradas do banco de dados.

Figura 9 – Exemplo de imagem do banco de dados

Fonte: Elaborado pelo autor.

Essas 140 mil imagens estão divididas em 70 mil imagens de rostos falsos que foram criando por uma GAN, a *StyleGAN*, e 70 mil imagens de rostos verdadeiros que foram retirados do site Flickr, uma plataforma que hospeda imagens como fotografias e desenhos reais.

3.2 Ambiente de Desenvolvimento

O ambiente de desenvolvimento utilizado neste trabalho foi o Google Colab, um serviço de nuvem gratuito criado pela Google que possui suporte para pesquisa e implementação de modelos de aprendizado de máquina e inteligência artificial no geral.

Ele funciona utilizando os Google Collab Notebooks que são células de código que podem ser executadas de maneira individual ou em conjunto. Ao invés de ter um código monolítico, os notebooks fazem essa partição do código que facilita a implementação de modelos como o CapsNet.

Além disso, por ser um serviço da Google, ele ainda conta com integração com os diversos outros serviços oferecidos pela empresa como, por exemplo, o Google Drive que é usado para armazenar dados e informações.

Foi escolhido para ser utilizado neste trabalho pela praticidade que trás consigo através das células de código Notebooks e pela integração com outros serviços.

3.3 Treinamento, Validação e Teste

Antes de falar sobre as etapas de treinamento, validação e teste, é importante explicar o conceito de época no contexto do aprendizado de máquina. Época (ou *epoch* em inglês) é uma única passagem que a rede faz por todo o conjunto de dados durante o processo de treinamento (CARNEY; CUNNINGHAM, 1998). Resumidamente, uma época é considerada completa quando um modelo é alimentado com todos os exemplos de treinamento uma vez, permitindo que o algoritmo de aprendizado ajuste seus parâmetros para fazer previsões mais precisas.

Durante o treinamento, os dados são divididos em lotes (ou *batches* em inglês) para facilitar o processo de otimização do modelo. O número de épocas durante uma etapa de treinamento é uma escolha arbitrária e ela precisa ser definida antes de iniciá-lo. O modelo passa por todas essas épocas até aprender o padrão nos dados. A Tabela 2 mostra a divisão das imagens em cada etapa:

Tabela 2 – Tabela da divisão das imagens

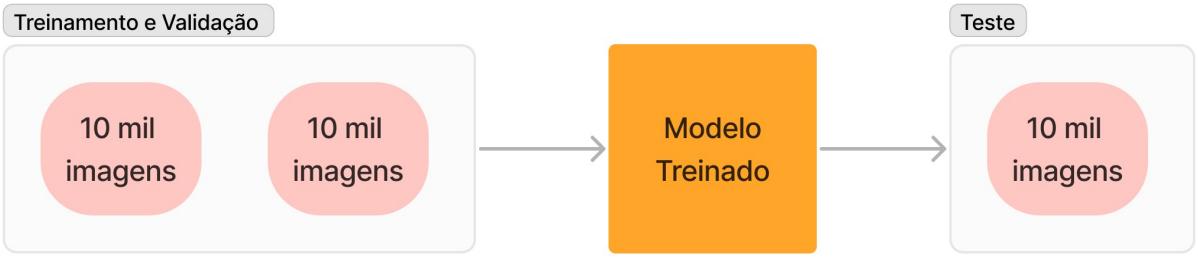
	Rostos Reais	**Rostos Falsos**
Treinamento	50 mil imagens	50 mil imagens
Validação	10 mil imagens	10 mil imagens
Teste	10 mil imagens	10 mil imagens

Fonte: Elaborado pelo autor.

Pode-se dizer que dentro de uma época ocorrem o treinamento e a validação juntos, cujo treinamento serve para ajustar os parâmetros internos de uma rede neural afim de melhorar as suas previsões. Após uma etapa de treino, o modelo é avaliado usando um conjunto de dados diferente, esse processo é chamado de validação. A validação é usada para ajustar o hiperparâmetros de uma rede neural como a taxa de aprendizado e também serve para melhorar o desempenho da rede.

O teste ocorre depois que todas as épocas de treinamento e validação são completadas. Ele usa um conjunto de dados que não foi utilizada nos processos anteriores e serve para avaliar o desempenho final do modelo, oferecendo uma estimativa imparcial dos resultados e simulando como esse modelo se comporta em situações reais. É nesse processo que se obtém as conclusões finais de acurácia e precisão. A Figura 10 demonstra todos esse processos.

Figura 10 – Processos



Elaborado pelo autor.

4 Resultados

Esta seção tem como objetivo apresentar e analisar os resultados obtidos no trabalho.

Antes de mostrar os resultados, é importante entender o que pode acontecer durante o processo de aprendizado. A rede pode ter alguns desses comportamentos:

- A rede pode entrar em *Overfitting*, que signifca que ela está viciada no conjunto de dados do treinamento e não consegue fazer previsões assertivas fora dele; e
- A rede pode entrar em *Underfitting*, que significa que ela não consegue aprender os padrões do conjunto de dados, pela simplicidade da rede ou mal ajuste dos parâmetros.

O processo de treinamento da rede neural presente nesse trabalho, ou seja, o treinamento de 140 mil imagens de rostos falsos e verdadeiros, foi realizado no total de 25 épocas. Em cada época a rede divide os dados em dois: dado de treinamento (acurácia e perda de treinamento) e um de validação (perda de validação) como está apresentado na Tabela 3. Ela também nos mostra os valores que cada dado assumiu na sua respectiva época.

Tabela 3 – Resultados obtidos por época de treinamento

Época	Perda de Treinamento	Perda de Validação	Acurácia
1	0.0348	0.0257	0.7316
2	0.0236	0.0215	0.7939
3	0.0202	0.0196	0.8201
4	0.0182	0.0189	0.8304
5	0.0166	0.0164	0.8602
6	0.0155	0.0154	0.8745
7	0.0145	0.0150	0.8798
8	0.0137	0.0174	0.8380
9	0.0130	0.0138	0.8951
10	0.0124	0.0139	0.8918
11	0.0119	0.0136	0.8972
12	0.0114	0.0127	0.9063
13	0.0110	0.0128	0.9072
14	0.0106	0.0124	0.9098
15	0.0102	0.0127	0.9091
16	0.0099	0.0129	0.9055
17	0.0097	0.0126	0.9114
18	0.0094	0.0120	0.9101
19	0.0092	0.0117	0.9170
20	0.0090	0.0117	0.9159
21	0.0087	0.0120	0.9123
22	0.0085	0.0117	0.9166
23	0.0083	0.0116	0.9144
24	0.0081	0.0114	0.9207
25	0.0080	0.0116	0.9164

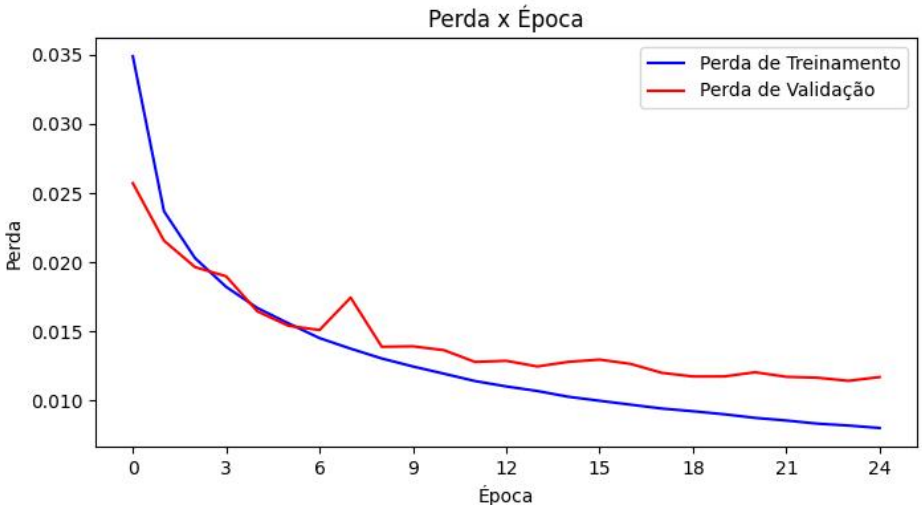
Fonte: Criado pelo autor (2023)

.

Como pode ser observado na Tabela 3, na maior parte do treinamento a acurácia aumentou e a perda de treinamento diminuiu, significando que o modelo funcionou de maneira satisfatória durante todo o treinamento, sendo possível afirmar que ela foi implementada de maneira correta.

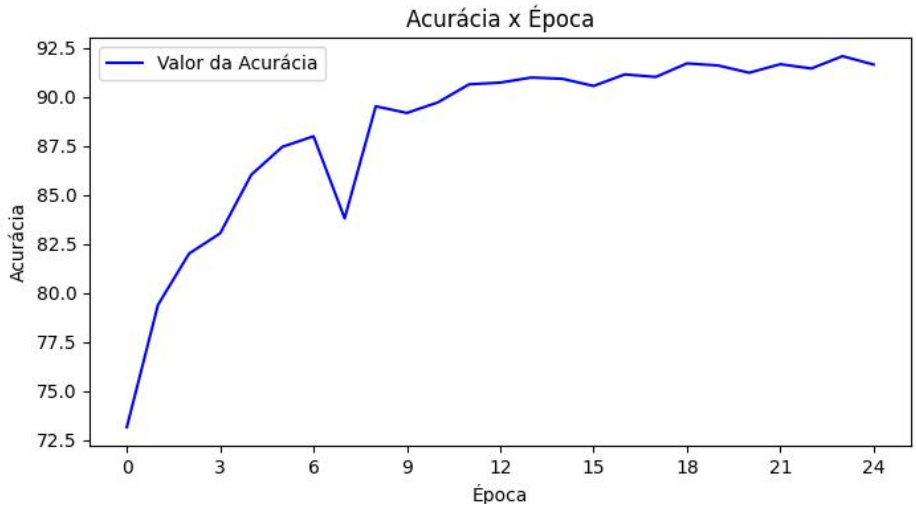
A Figura 11 demonstra o gráfico das perdas tanto de treinamento quanto de validação e a Figura 12 mostra apenas a acurácia em todas as épocas.

Figura 11 – Gráfico da perda de treinamento e validação em relação as épocas.



Fonte: Elaborado pelo autor.

Figura 12 – Gráfico da acurácia no teste em relação as épocas.



Fonte: Elaborado pelo autor.

Analisando os resultados tanto da tabela quanto dos gráficos, o modelo que obteve a maior acurácia foi o da época 24 enquanto o que teve menor perda de treinamento foi o da época 25. Para realizar o teste, foi selecionado o modelo da época 24 e sua acurácia foi de 91.87% com precisão de 93.01% após todas as 10 mil imagens do conjunto de teste.

4.1 Comparativo

Neste trabalho é utilizado um artigo chamado "Fake face detection using local binary patrern and ensemble modeling"(WANG; ZARGHAMI; CUI, 2021) que utilizou a mesma base

de dados porém modelos diferentes para classificação de rostos falsos ou verdadeiros. A Tabela 4 foi modificada afim de inserir os dados obtidos durantes os testes com a CapsNet.

No artigo foi implementado três modelos diferentes para o mesmo propósito, *Residual Neural Network* (Res-Net), Gram-Net (Res-Net com uma matriz de Gram) e *Local Binary Patern* (LBP-net).

Tabela 4 – Tabelas com os resultados obtidos (em porcentagem) com a CapsNet em comparação com o artigo de WANG, ZARGHAMI, CUI.

Modelos	**Acurácia**	**Precisão**
CapsNet	91.87	93.01
Res-Net	98.67	99.12
Gram-Net	98.71	98.98
LBP-Net	98.58	98.96

Adaptado de WANG, ZARGHAMI e CUI (2021)

5 Conclusão

O estudo realizado apresentou uma análise aprofundada sobre a eficácia da CapsNet no contexto da segurança da informação, concentrando-se especificamente na detecção de faces falsas. Os resultados obtidos no final das fases de treinamento, validação e teste, indicam um desempenho satisfatório dessa técnica na detecção de rostos gerados por Inteligencia Artificial.

Ao considerar o impacto do trabalho, destaca-se a relevância significativa do conhecimento adquirido. Tanto a natureza crítica do problema abordado quanto os desafios intrínsecos à implementação da CapsNet reforçam a importância dos resultados alcançados. Este estudo contribuiu não apenas para a resolução prática de uma questão crucial em segurança da informação, mas também para a superação bem-sucedida dos obstáculos relacionados à implementação da rede neural.

Além disso, a conclusão extraída do desempenho consistente da CapsNet nas diversas fases do processo valida a eficácia da abordagem escolhida. A complexidade do padrão facial humano, que muitas vezes confunde outros métodos de detecção, foi enfrentada de maneira satisfatória pela CapsNet, demonstrando sua capacidade única de compreender padrões em dados complexos.

A obtenção bem-sucedida dos objetivos propostos destaca não apenas a competência técnica do estudo, mas também sua relevância prática. A capacidade da CapsNet em lidar com a detecção de faces falsas contribui positivamente para a segurança da informação, um campo onde a evolução constante das tecnologias demanda soluções inovadoras e confiáveis. Dessa forma, este trabalho não apenas atingiu seus objetivos de pesquisa, mas também ofereceu *insights* valiosos para o avanço contínuo no entendimento e na aplicação de técnicas avançadas em segurança digital.

5.1 Trabalhos Futuros

Para possíveis trabalhos futuros, poderá ser analisado o aumento da precisão e acurácia diante de fotos com maior qualidade. Dessa forma, a rede será capaz de compreender, com dados mais elaborados, as nuances de cada rosto humano real e como eles se diferem em relação aos gerados por Inteligência Artificiai.

Além disso, poderá ser feita uma mudança no banco de dados para que possa ser compreendido a questão da generalização de rostos, uma vez que a rede não estará acostumada com a nova base de dados.

Referências

AJIT, A.; ACHARYA, K.; SAMANTA, A. A review of convolutional neural networks. In: *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*. [S.l.: s.n.], 2020. p. 1–5.

CARNEY, J.; CUNNINGHAM, P. The epoch interpretation of learning. In: . [s.n.], 1998. Disponível em: <https://api.semanticscholar.org/CorpusID:16392810>.

DEHNIE, S.; SENCAR, T.; MEMON, N. Digital image forensics for identifying computer generated and digital camera images. In: *2006 International Conference on Image Processing*. [S.l.: s.n.], 2006. p. 2313–2316.

DING, B.; QIAN, H.; ZHOU, J. Activation functions and their characteristics in deep neural networks. In: *2018 Chinese Control And Decision Conference (CCDC)*. [S.l.: s.n.], 2018. p. 1836–1841.

DIRIK, A. E.; BAYRAM, S.; SENCAR, H. T.; MEMON, N. New features to identify computer generated images. In: *2007 IEEE International Conference on Image Processing*. [S.l.: s.n.], 2007. v. 4, p. IV – 433–IV – 436.

DONG, Z.; LIN, S. Research on image classification based on capsnet. In: *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. [S.l.: s.n.], 2019. p. 1023–1026.

GOODFELLOW, I. J.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. *Generative Adversarial Networks*. 2014.

HOLLÓSI, J.; POZNA, C. R. Improve the accuracy of neural networks using capsule layers. In: *2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)*. [S.l.: s.n.], 2018. p. 000015–000018.

KHALAF, R. S.; VAROL, A. Digital forensics: Focusing on image forensics. In: *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*. [S.l.: s.n.], 2019. p. 1–5.

M., H.; M.N, S. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, v. 5, p. 01–11, 2015. Disponível em: <https://api.semanticscholar.org/CorpusID:61877559>.

PYTHON SOFTWARE FOUNDATION. *About Python*. 2001. Disponível em: https://www.python.org/about/. Acesso em 23 de agosto de 2023.

SABOUR, S.; FROSST, N.; HINTON, G. E. *Dynamic Routing Between Capsules*. 2017.

THE LINUX FOUNDATION. *Features*. 2000. Disponível em: https://pytorch.org/features/. Acesso em 23 de agosto de 2023.

THE MATPLOTLIB DEVELOPMENT TEAM. *History*. 2012. Disponível em: https://matplotlib.org/stable/users/project/history.html. Acesso em 23 de agosto de 2023.

UNIVERSITY OF ARIZONA. *Understanding Capsule Networks*. s.d. Disponível em: http://hdc.cs.arizona.edu/ mwli/understanding-capsule-network/writing/. Acesso em 20 de agosto de 2023.

WANG, Y.; ZARGHAMI, V.; CUI, S. Fake face detection using local binary pattern and ensemble modeling. In: *2021 IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2021. p. 3917–3921.

ZHANG, Z.; SUN, W.; MIN, X.; WANG, T.; LU, W.; ZHAI, G. Distinguishing computer-generated images from photographic images: a texture-aware deep learning-based method. In: *2022 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. [S.l.: s.n.], 2022. p. 1–5.