

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

MATEUS FIORESE MACHADO NETTO

Treinamento de Redes Neurais Convolucionais utilizando
Dados Gerados Artificialmente para Reconhecimento de
Imagens

São Carlos
2024

MATEUS FIORESE MACHADO NETTO

Treinamento de Redes Neurais Convolucionais utilizando
Dados Gerados Artificialmente para Reconhecimento de
Imagens

Monografia apresentada ao Curso
de Engenharia Elétrica com
Ênfase em Eletrônica, da Escola
de Engenharia de São Carlos
da Universidade de São Paulo,
como parte dos requisitos para
obtenção do Título de Engenheiro
Eletricista

Orientador: Prof. Dr. Valdir Grassi
Junior

VERSÃO CORRIGIDA

São Carlos

2024

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

N469t Netto, Mateus Fiorese Machado
Treinamento de Redes Neurais Convolucionais
utilizando Dados Gerados Artificialmente para
Reconhecimento de Imagens / Mateus Fiorese Machado
Netto; orientador Valdir Grassi. São Carlos, 2024.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2024.

1. Imagens sintéticas. 2. Redes generativas
adversariais. 3. Redes neurais convolucionais.. 4.
Reconhecimento de imagens. I. Título.

FOLHA DE APROVAÇÃO

Nome: Mateus Fiorese Machado Netto

Título: “Treinamento de Redes Neurais Convolucionais utilizando Dados Gerados Artificialmente para Reconhecimento de Imagens”

Trabalho de Conclusão de Curso defendido e aprovado
em 09 / 12 / 2024,

com NOTA 7,8 (sete , oito), pela Comissão
Julgadora:

Prof. Associado Valdir Grassi Junior - Orientador SEL/EESC/USP

Mestre Nicolas dos Santos Rosa - Doutorando EESC/USP

Mestre Eduardo Godinho Ribeiro - Doutorando EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Professor Associado José Carlos de Melo Vieira Júnior

RESUMO

MACHADO NETTO, M. F. **Treinamento de Redes Neurais Convolucionais utilizando Dados Gerados Artificialmente para Reconhecimento de Imagens**. 2024. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2024.

Este trabalho aborda a aplicação de imagens sintéticas geradas via redes generativas adversariais (GANs) no treinamento de redes neurais convolucionais (CNNs) para reconhecimento de imagens. A pesquisa é motivada pela dificuldade muitas vezes encontrada em obter um banco de dados relevante para o treinamento de redes neurais artificiais. O estudo tem como objetivo principal verificar a validade de utilizar *datasets* constituídos de maneira total ou parcial por imagens sintéticas em comparação com bancos de dados de imagens reais. Na pesquisa foram utilizadas GANs para a geração de imagens similares às do *dataset* MNIST de modo a constituir múltiplos bancos de imagens que foram utilizados para o treinamento de CNNs para reconhecimento de imagens. Assim foi analisado o desempenho destas redes e os resultados comparados com uma base preestabelecida. Os resultados indicam que não há diferenças significativas entre a utilização de imagens sintéticas ou reais no treinamento de CNNs para reconhecimento de imagens, quando aplicado a dados de baixa complexidade, porém nota-se que a geração destas imagens sintéticas requer certos cuidados.

Palavras-chave: Imagens sintéticas. Redes generativas adversariais. Redes neurais convolucionais. Reconhecimento de imagens.

ABSTRACT

MACAHDO NETTO, M. F. **Training Convolutional Neural Networks Using Artificially Generated Data for Image Recognition**. 2024. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2024.

This work addresses the application of synthetic images generated through Generative Adversarial Networks (GANs) in training Convolutional Neural Networks (CNNs) for image recognition. The research is motivated by the frequent difficulty of obtaining a relevant database for training artificial neural networks. The primary objective of the study is to verify the validity of using datasets composed entirely or partially of synthetic images in comparison to databases consisting solely of real images. In this research, GANs were used to generate images similar to those in the MNIST dataset to create multiple image datasets, which were employed in training CNNs for image recognition. The performance of these networks was then analyzed, and the results were compared with a pre-established baseline. The results indicate no significant differences between the use of synthetic or real images in training CNNs for image recognition when applied to low complexity data. However, it is noted that generating synthetic images requires certain precautions.

Keywords: Synthetic images. Generative Adversarial Networks. Convolutional Neural Networks. Image recognition.

LISTA DE ILUSTRAÇÕES

Figura 1 – Representação de um neurônio artificial	21
Figura 2 – Representação de uma rede multicamadas	22
Figura 3 – Representação de uma rede neural convolucional	23
Figura 4 – Fluxograma do processo de geração de imagens	32
Figura 5 – Fluxograma do processo de montagem dos <i>datasets</i>	32
Figura 6 – Exemplos de imagens sintéticas geradas	35
Figura 7 – Gráfico de acurácia por CNN	37
Figura 8 – Gráfico de precisão por CNN	37
Figura 9 – Gráfico de <i>recall</i> por CNN	38
Figura 10 – Gráfico de <i>F1-score</i> por CNN	38

LISTA DE TABELAS

Tabela 1 – Bancos de dados gerados	33
Tabela 2 – Arquitetura das CNNs treinadas	33
Tabela 3 – Resultados e desempenho	36

LISTA DE ABREVIATURAS E SIGLAS

CNN	<i>Convolutional Neural Network</i>
CAD	<i>Computer Aided Diagnosis</i>
RNA	Rede Neural Artificial
ReLU	<i>Rectified Linear Unit</i>
FC	<i>Fully Connected</i>
SGD	<i>Stochastic Gradient Descent</i>
GAN	<i>Generative Artificial Network</i>
VAE	<i>Variational Autoencoders</i>
TP	<i>True Positive</i>
TN	<i>True Negative</i>
FP	<i>False Positive</i>
FN	<i>False Negative</i>

LISTA DE SÍMBOLOS

s	Sinal de saída
f	Função de ativação
a	Sinais de entrada
w	Pesos
b	Viés
l	Índice de camada da rede
Θ	Pesos
η	taxa de aprendizado
$J(\Theta)$	Função de perda
$\hat{y}_i$	Predição estimada
y_i	Predição desejada

SUMÁRIO

1	INTRODUÇÃO	19
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	Histórico e Evolução das Redes Neurais	21
2.2	Princípios de Funcionamento das Redes Neurais	21
2.3	Redes Neurais Convolucionais (CNNs)	23
2.3.1	Componentes Principais das CNNs	23
2.3.2	Vantagens das CNNs no Reconhecimento de Imagens	24
2.3.3	Aplicações Práticas das CNNs	24
2.4	Treinamento de Redes Neurais	24
2.4.1	Processo de Treinamento	25
2.4.2	Funções de Perda e Algoritmos de Otimização	25
2.4.3	Técnicas para Melhorar o Treinamento	26
2.5	Dados para Treinamento de Redes Neurais	27
2.5.1	Características dos Conjuntos de Dados para o Treinamento de Redes Neurais	27
2.5.2	Problemas Associados à Obtenção de Dados Reais	27
2.5.3	Exemplos de Conjuntos de Dados Reais	28
2.6	Geração de Dados Sintéticos	28
2.6.1	Conceito de Dados Sintéticos	28
2.6.2	Métodos para Geração de Dados Sintéticos	28
2.6.3	Vantagens e Limitações dos Dados Sintéticos	29
2.7	Generalização e Desempenho	29
2.7.1	Definição de Generalização	29
2.7.2	Métricas para Avaliação de Desempenho	30
3	METODOLOGIA	31
3.1	Geração dos <i>Datasets</i> sintéticos	31
3.2	Treinamento e Validação	33
4	RESULTADOS	35
4.1	Treinamento e Validação	36
5	CONCLUSÃO	41
	REFERÊNCIAS	43

1 INTRODUÇÃO

As redes neurais convolucionais, *Convolutional Neural Networks* (CNN), destacam-se nos últimos anos como umas das melhores ferramentas para o reconhecimento de imagens, com ótimos desempenhos em aplicações como diagnósticos médicos auxiliados por computador, *Computer Aided Diagnosis* (CAD), veículos autônomos, dentre outras. Krizhevsky A. (2012) afirmam, em tradução livre, "uma rede neural convolucional profunda é capaz de atingir resultados sem precedentes em *datasets* desafiadores somente com aprendizagem supervisionada". Entretanto, esse treinamento requer um grande volume de dados de modo a evitar problemas como *overfitting*, cuja obtenção pode ser um desafio significativo devido a disponibilidade limitada e custos associados para adquirir dados reais. Segundo Fei-Fei L. (2009), em tradução livre, "a construção de grandes bases de dados anotados manualmente é um processo intensivo e caro".

A falta desses dados em quantidade suficiente pode trazer problemas na capacidade de generalização da CNN, reduzindo sua precisão e robustez, como verificado por He K. (2016), quanto mais profunda a CNN, maior deve ser o banco de dados de modo a evitar *overfitting* e garantir precisão e robustez. Nesse contexto, surge a ideia de contornar este problema com o uso de dados sintéticos, gerados artificialmente, possibilitando assim a criação de grandes conjuntos de dados personalizados de acordo com a necessidade de cada projeto.

O objetivo deste trabalho é avaliar a viabilidade do treinamento de CNNs para reconhecimento de imagens exclusivamente com dados sintéticos, assim como os seus efeitos. Neste trabalho, foi realizada a geração de conjuntos de dados sintéticos para utilização no treinamento de redes neurais convolucionais, assim como a comparação entre redes treinadas por dados reais e por dados artificiais, de modo a analisar a eficácia desta técnica.

Este trabalho é motivado pela dificuldade em obter dados para treinamento de redes neurais em quantidade suficiente para certas aplicações, como exemplo para aplicações em veículos autônomos em condições ambientais adversas, como neblina, neve ou chuva. LeCun Y. (2015) explicam que "a falta de dados rotulados suficientes é um dos principais obstáculos no treinamento de redes neurais profundas".

Primeiramente, neste estudo, foi realizada a geração de conjuntos de imagens sintéticas utilizando redes generativas, com tema escolhido semelhante a banco de dados de imagens reais existente, o MNIST, possibilitando o uso de uma base de comparação para os resultados obtidos. A partir destes dois conjuntos de dados, um sintético e um real, foram treinadas diversas CNNs para classificação de imagens, utilizando múltiplos modelos de modo a otimizar o desempenho.

Em seguida as redes obtidas tiveram seu desempenho, capacidade de generalização e robustez avaliados utilizando métricas de acurácia, precisão e *F1-score*, assim como comparadas entre si. Com isso foram exploradas possíveis limitações e desafios no uso de

dados sintéticos para o treinamento de CNNs.

2 FUNDAMENTAÇÃO TEÓRICA

As redes neurais artificiais têm sido amplamente estudadas e aplicadas em diversas áreas do conhecimento. A seguir, apresentam-se o histórico e a evolução dessas redes, bem como seus principais conceitos.

2.1 Histórico e Evolução das Redes Neurais

Na década de 1940, com a publicação do artigo de Warren McCulloch e Walter Pitts propondo um modelo matemático para um neurônio artificial, criado com base em estudos sobre o cérebro humano e a forma como este processa informações, tem-se o início do desenvolvimento das redes neurais artificiais (RNA) (MCCULLOCH W.S., 1943).

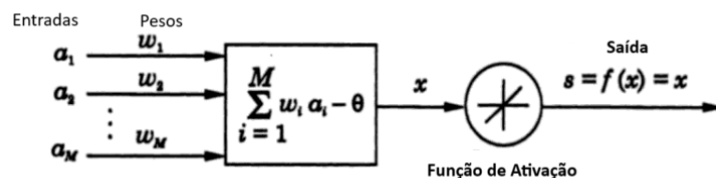
Na década de 1950, foi desenvolvida uma das primeiras arquiteturas de RNA, o *perceptron*, desenvolvido por Frank Rosenblatt (ROSENBLATT, 1953). Porém, como este era somente capaz de resolver problemas lineares, ou que possam ser separados em problemas lineares, como destacado por Marvin Minsky e Seymour Papert em 1969, não houve grande interesse e financiamento para pesquisas nessa área (MINSKY M., 1969).

Na década de 1980, houve uma retomada no interesse pela área com o algoritmo de retropropagação de erro (*backpropagation*) por Geoffrey Hinton, David Rumelhart e Ronald Williams, permitindo o treinamento de redes neurais profundas, redes neurais com múltiplas camadas (RUMELHART DAVID E.; WILLIAMS, 1986).

2.2 Princípios de Funcionamento das Redes Neurais

As RNAs podem ser divididas em três componentes para a explicação de seu funcionamento: os neurônios, as camadas e a função de ativação, conforme ilustrado na Figura 1.

Figura 1 – Representação de um neurônio artificial



Fonte: Adaptado de Yegnanarayana (2009)

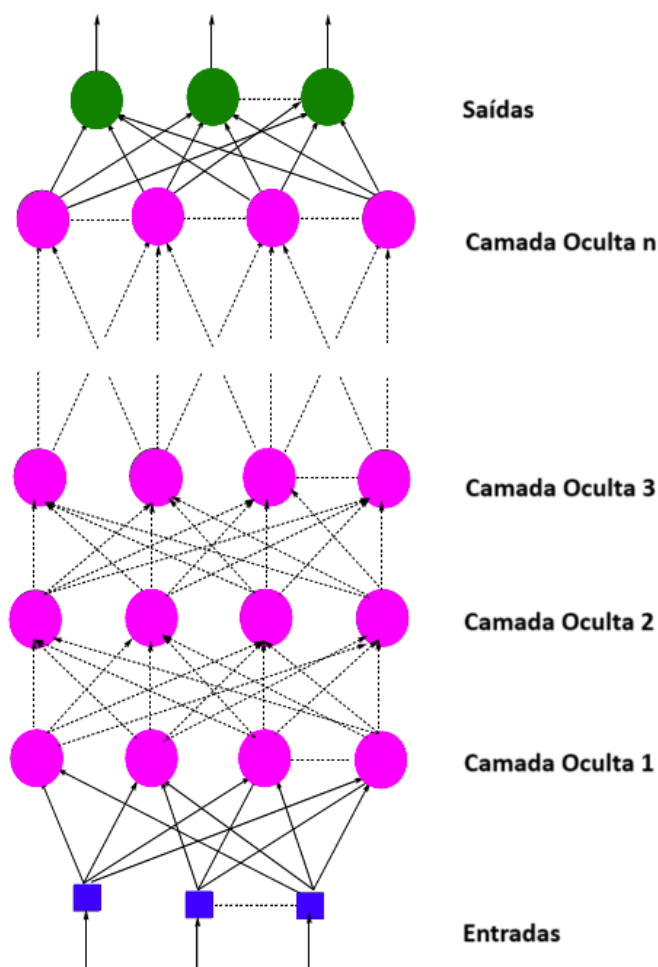
Os Neurônios recebem entradas que são ponderadas por um conjunto de pesos e em seguida passam por uma função de ativação para determinar a saída. Matematicamente essa operação pode ser representada por:

$$s = f(\sum w_i a_i - \Theta) \quad (1)$$

onde f é a função de ativação, a as entradas, w os pesos e Θ o viés (HAYKIN, 2009).

Esses neurônios são divididos em camadas, conforme mostra a Figura 2, sendo a primeira a camada de entrada, que recebe os dados brutos. Cada neurônio nessa camada, portanto, utiliza os pesos e vieses para extrair e representar características relevantes do conjunto de dados a ser analisado. A última camada é denominada camada de saída, fornecendo o resultado final da rede. As camadas intermediárias, ou camadas ocultas, aumentam a complexidade da rede. A profundidade de uma rede é determinada pelo número de camadas que ela possui (HAYKIN, 2009).

Figura 2 – Representação de uma rede multicamadas



Fonte: Adaptado de Ramchoun H. (2016)

A função de ativação, aplicada no final da análise do neurônio, após a ponderação por pesos, introduz elementos não lineares ao processo, permitindo a análise de problemas com

maior complexidade. Alguns exemplos de funções de ativação são a sigmóide, a *softmax* (JOULIN A., 2014), a tangente hiperbólica e a ReLU, *Rectified Linear Unit* (GLOROT X., 2011). A escolha da função de ativação é dada a partir do problema a ser resolvido.

2.3 Redes Neurais Convolucionais (CNNs)

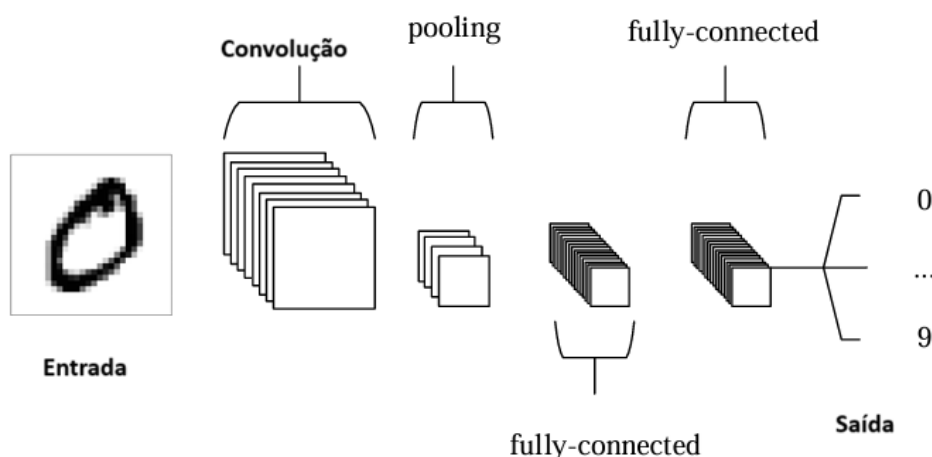
As redes neurais convolucionais, Convolutional Neural Network (CNN), são estruturas de redes neurais artificiais baseadas no córtex visual animal (FUKUSHIMA, 1980). Foram desenvolvidas especificamente para lidar com dados bidimensionais, como imagens, que são organizados no formato de matrizes e são bastante eficazes no reconhecimento e classificação de padrões visuais.

A nomenclatura desse tipo de rede vem das camadas convolucionais, que utilizam operações de convolução nas entradas de modo a extrair características relevantes como bordas, texturas e formas sem antes estas precisarem ser extraídas manualmente dos dados de entrada (KRIZHEVSKY A., 2012).

2.3.1 Componentes Principais das CNNs

Há diferentes tipos de camadas em uma CNN, conforme ilustra a Figura 3, dentre elas tem-se as camadas convolucionais, as camadas de *pooling* e as camadas *fullyconnected* (FC).

Figura 3 – Representação de uma rede neural convolucional



Fonte: Adaptado de O'Shea (2015)

As camadas convolucionais aplicam filtros às entradas utilizando operações de convolução. Cada filtro detecta um padrão específico, como detecção de bordas horizontais

ou bordas verticais, sendo treinado durante o processo de aprendizagem para otimizar a detecção das características mais relevantes (GOODFELLOW I., 2016).

As camadas de *pooling* agregam informações locais, reduzindo a dimensão do mapa de características e assim minimizando o efeito de pequenas variações e deslocamentos na entrada. Duas camadas de pooling comumente utilizadas são, *max pooling*, que usa o valor máximo da região, e *average pooling*, que utiliza o valor médio (GOODFELLOW I., 2016).

As camadas *fullyconnected* são assim chamadas pois tem como característica a conexão de cada neurônio com todos os presentes na camada anterior, de forma semelhante as camadas presentes de redes neurais artificiais tradicionais. São geralmente utilizadas no final de redes convolucionais para tarefas de classificação, de modo a agregar as informações obtidas nas camadas de *pooling* em uma forma mais adequada para a classificação final dos resultados (GOODFELLOW I., 2016).

2.3.2 Vantagens das CNNs no Reconhecimento de Imagens

Conforme visto, diferentemente de redes como as MLPs, as CNNs são especialmente projetadas para aprender as características relevantes dos dados de entrada de forma hierárquica, sem que essas características precisem ser extraídas manualmente (KRIZHEVSKY A., 2012). Além disso, outras vantagens apresentadas pelas CNNs no reconhecimento de imagens são a invariância espacial, derivado das operações de convolução realizadas nas camadas de convolução, proporcionando robustez a deslocamentos e variações locais. A redução de dimensão dos mapas de características proporcionado pelas camadas de pooling melhora a eficiência computacional da rede enquanto preserva as informações mais importantes (GOODFELLOW I., 2016).

2.3.3 Aplicações Práticas das CNNs

Algumas das principais aplicações práticas das CNNs são o reconhecimento de imagens, sendo amplamente utilizadas em sistemas de classificação e identificação de objetos em fotografias e vídeos, segmentação de imagens, reconhecimento facial e análise médica.

2.4 Treinamento de Redes Neurais

O treinamento de uma rede neural é o processo onde ocorre o aprendizado por parte da rede a realizar uma determinada tarefa, como classificação de objetos ou detecção de padrões. Este processo envolve diversos componentes e técnicas desenvolvidas ao longo dos anos, apresentadas a seguir.

2.4.1 Processo de Treinamento

O processo de treinamento de uma rede neural consiste essencialmente em abastecer essa rede com dados de treinamento, em caso de treinamento supervisionado o resultado final desejado é conhecido, comparar o resultado obtido pelo conjunto de pesos atual com o desejado e ajustar esses pesos de modo a minimizar a diferença entre esses resultados. Esse processo pode ser dividido em duas etapas principais, *forward propagation* e *backpropagation*.

A etapa de *forward propagation*, propagação direta, consiste em alimentar os dados na rede, passando camada por camada até gerar um resultado final, dessa forma obtendo um resultado para o atual conjunto de pesos e entradas. Matematicamente, para cada camada, isso pode ser descrito por:

$$z^l = W^l a^{l-1} + b^l \quad (2)$$

$$a^l = f(z^l) \quad (3)$$

onde W são os pesos, b é o viés, l é o índice da camada da rede e f é a função de ativação (GOODFELLOW I., 2016).

A etapa de *backpropagation*, propagação reversa, consiste em, após o processo de *forward propagation*, calcular o erro de predição em relação ao resultado esperado utilizando uma função de perda. Em seguida, ajustar os pesos de modo a minimizar essa diferença de acordo com o algoritmo de otimização, propagando o erro de volta por meio da rede. Esse processo pode ser matematicamente descrito por:

$$\Theta \leftarrow \Theta - \eta \frac{\delta J(\Theta)}{\delta \Theta} \quad (4)$$

onde Θ representa os pesos, η representa a taxa de aprendizado e $J(\Theta)$ é a função de perda (GOODFELLOW I., 2016).

2.4.2 Funções de Perda e Algoritmos de Otimização

As funções de perda ou funções de custo medem a diferença entre a predição estimada e a predição desejada. Alguns exemplos mais comuns são, o erro quadrático médio:

$$MSE = \frac{1}{n} \sum (y_i - \hat{y}_i)^2 \quad (5)$$

e a entropia cruzada (*CrossEntropy*)

$$CrossEntropy = -\frac{1}{n} \sum [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (6)$$

onde $\hat{y}_i$ é a predição estimada e y_i é a predição desejada.

Os algoritmos de otimização definem o ajuste dos pesos de modo a minimizar o erro. Alguns dos mais comuns são o Gradiente Descendente Estocástico (SGD, do inglês *Stochastic Gradient Descent*), onde a atualização dos pesos se dá utilizando somente o gradiente de uma única iteração de treinamento, e o RMSProp, que mantém para iterações futuras um decaimento exponencial das médias dos quadrados dos gradientes passados (BOTTOU, 2010) (LECUN Y., 1998) (TIELEMAN T., 2012) (HINTON, 2012).

2.4.3 Técnicas para Melhorar o Treinamento

Diversas técnicas podem ser adicionadas ao treinamento para melhorar sua eficiência e evitar problemas como *overfitting*, como por exemplo a regularização, o *dropout* e o *batch normalization*.

A regularização consiste em adicionar um termo à função de perda de modo a penalizar grandes pesos, como por exemplo, a regularização *Lasso* (TIBSHIRANI, 1996) (HASTIE T., 2009):

$$Loss_L = Loss + \lambda \sum |w_i| \quad (7)$$

e a regularização *Ridge* (HOERL A. E., 1970) (HASTIE T., 2009):

$$Loss_L = Loss + \lambda \sum w_i^2 \quad (8)$$

O *dropout* consiste, em cada iteração do treinamento desativar aleatoriamente uma fração dos neurônios em cada camada, dando a rede mais robustez, uma vez que ela passa a não depender de neurônios específicos (SHRIVASTAVA A., 2017).

O *batch normalization* consiste em normalizar as entradas de cada camada, assim estabilizando a distribuição dos dados e acelerando a convergência (IOFFE S., 2015).

2.5 Dados para Treinamento de Redes Neurais

O banco de dados é um componente essencial para o treinamento de redes neurais, pois é a partir dele que a rede irá aprender a reconhecer padrões e realizar previsões precisas. A qualidade e quantidade dos dados é, portanto, fator crucial no sucesso ou falha de uma aplicação. A seguir são exploradas as características desses dados e os problemas associados a sua obtenção.

2.5.1 Características dos Conjuntos de Dados para o Treinamento de Redes Neurais

Há diversas características que podem ser mencionadas em relação aos conjuntos de dados utilizados no treinamento de redes neurais. Algumas das mais importantes são a variedade e diversidade dos dados, que auxiliam na obtenção da capacidade de generalização da rede, evitando problemas de overfitting, situação onde a rede memoriza os dados de treinamento, tornando-se ineficaz na análise de dados novos.

Outra característica é a representatividade, uma vez que para que a rede possa aprender todas as características necessárias durante o processo de treinamento o banco de dados deve ser suficientemente representativo do problema a ser analisado. Também deve-se mencionar a quantidade, já que o banco de dados deve ser suficientemente grande para realizar o treinamento da rede, isso se acentua quando treinando redes profundas, que requerem, às vezes, milhões de exemplos.

Há também a qualidade do conjunto, tanto das imagens como de suas anotações, pois erros de anotação trazem efeitos negativos para a rede. E para conjuntos com múltiplas classes, um equilíbrio entre elas, afim de evitar um viés da rede para classes mais representadas.

2.5.2 Problemas Associados à Obtenção de Dados Reais

A obtenção de dados reais para o treinamento de redes neurais traz consigo diversos desafios e problemas a serem explorados, como o custo e a escassez.

Entre custos com equipamentos de captura, infraestrutura e gastos com profissionais tanto para a coleta quanto para a anotação dos dados, a geração de grandes conjuntos de dados pode ser um processo bastante caro e demorado.

Além disso, em algumas áreas, a obtenção de dados suficientes pode ser bastante difícil, seja pela raridade dos dados em questão, como por exemplo uma situação muito específica de trânsito no caso de direção autônoma, ou uma condição médica rara. Seja por questões de privacidade e ética como, novamente, na obtenção de dados hospitalares, seja para questões de segurança como reconhecimento facial.

2.5.3 Exemplos de Conjuntos de Dados Reais

Alguns exemplos de conjuntos de dados disponíveis para uso são o ImageNet, o CIFAR-10 e o MNIST.

O ImageNet é um dos maiores e mais utilizados bancos de dados, com mais de 14 milhões de imagens anotadas em mais de 20 mil categorias. Utilizado principalmente para classificação de objetos (FEI-FEI L., 2009).

O CIFAR-10 contém 60 mil imagens divididas em 10 classes, 6 mil imagens por classe. É usado com frequência para *benchmarking* de algoritmos de aprendizado profundo (KRIZHEVSKY A., 2009).

O MNIST é um conjunto de 70 mil exemplos de dados manuscritos, com 10 categorias, sendo estas os dígitos de 0 a 9 (DENG, 2012).

2.6 Geração de Dados Sintéticos

A geração de dados sintéticos possibilita uma alternativa para o treinamento de redes neurais, principalmente em situações onde a obtenção de dados reais é difícil, seja por custo, disponibilidade ou questões de privacidade.

2.6.1 Conceito de Dados Sintéticos

Dados sintéticos são dados gerados artificialmente por meio de algoritmos, modelos computacionais ou simulações, imitando as características observadas em dados reais, como padrões e distribuições, de modo a complementar e ampliar um banco de dados existente, amenizar ou resolver problemas de escassez de dados em uma situação particular ou criar cenários para testes e validação.

2.6.2 Métodos para Geração de Dados Sintéticos

Embora existam diversos métodos para a geração de dados sintéticos, a seguir são abordados os dois métodos principais: os modelos generativos e os algoritmos procedurais.

Os modelos generativos estão entre o tipo de modelos de aprendizagem que podem gerar novos dados a partir de uma distribuição conhecida. Alguns dos modelos generativos mais comuns incluem as Redes Generativas Adversariais (GANs) (GOODFELLOW I. J., 2014), que consistem de um gerador e de um discriminador, o gerador cria dados sintéticos enquanto o discriminador tenta diferenciá-lo de dados reais. Ambos os componentes são treinados juntos, dessa forma o gerador melhora enquanto tenta enganar o discriminador. Outros modelos generativos são os *Variational Autoencoders* (VAEs), modelos baseados em *autoencoders* que aprendem como representar os dados em um espaço contínuo latente e, a partir desse espaço gerar novos dados escolhendo amostras e as decodificando de volta para o espaço original (GOODFELLOW I., 2016).

Os algoritmos procedurais produzem dados sintéticos utilizando de regras predefinidas e algoritmos procedurais. Esses métodos são amplamente aplicados em situações onde pode-se explicitamente definir padrões de dados, como simulações computacionais, usadas em campos como física, biologia e finanças, em que dados podem ser obtidos por meio de simulações de fenômenos naturais.

2.6.3 Vantagens e Limitações dos Dados Sintéticos

As principais vantagens do uso de dados sintéticos se encontram em sua acessibilidade, pois dados sintéticos podem ser gerados em grande escala, ultrapassando limitações associadas com a disponibilidade de dados reais. Outra vantagem está relacionada com o custo, pois a geração de dados sintéticos, embora ainda apresente custos, estes podem ser menores que os envolvidos na coleta de dados reais. Também deve ser mencionado o controle, uma vez que o controle total sobre as variáveis e condições experimentais oferecido pelos dados sintéticos facilita teste e validação em cenários específicos. Além disso a privacidade também é uma vantagem, já que dados sintéticos podem ser utilizados para evitar problemas de privacidade em áreas sensíveis, como saúde e finanças, onde o uso de dados reais pode ser restrito (BROCK ANDREW; SIMONYAN, 2018).

Por outro lado, as maiores limitações apresentadas por dados sintéticos são a fidelidade, uma vez que os dados sintéticos podem não representar perfeitamente a complexidade e variabilidade de dados reais, levando a modelos que não generalizam bem, problema conhecido como *Reality Gap* (TOBIN J., 2017). E, portanto, é essencial a validação desses dados sintéticos, de modo a garantir sua representatividade. Outra limitação é a dependência do modelo, uma vez que a qualidade dos dados artificiais depende fortemente do modelo e algoritmos usados em sua geração. Também deve ser mencionada a complexidade computacional, já que métodos avançados podem ser bastante computacionalmente intensivos, necessitando de recursos para a criação (BROCK ANDREW; SIMONYAN, 2018).

2.7 Generalização e Desempenho

A generalização é um conceito essencial para as redes neurais convolucionais, uma vez que essa capacidade de estender para dados novos as características obtidas no treinamento é o que vai de fato determinar a eficácia da rede em aplicações reais (GOODFELLOW I., 2016).

2.7.1 Definição de Generalização

A generalização, no ambiente de aprendizado de máquinas, refere-se a capacidade de uma rede neural treinada de reconhecer e classificar corretamente imagens, objetos, que não fizeram parte de seu conjunto de treinamento. Ou seja, uma rede com boa capacidade

de generalização consegue aplicar os padrões aprendidos no processo de treinamento mesmo quando lidando com dados novos (BISHOP C. M., 2006).

2.7.2 Métricas para Avaliação de Desempenho

Existem múltiplas ferramentas utilizadas para medir o desempenho de uma rede neural, em sua maioria estas se baseiam na utilização de quatro dados obtidos de testes e validação, TP – *true positive*, se refere ao número de dados que a rede corretamente afirmou como positivos, TN – *true negative*, se refere ao número de dados que a rede corretamente identificou como negativos, FP – *false positive*, se refere ao número de dados que a rede incorretamente identificou como positivo e FN – *false negative*, dados que a rede incorretamente identificou como negativos.

A métrica de acurácia, *accuracy*, é determinado pela relação de predições corretas por predições totais:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

A métrica da precisão, *precision*, é determinado pela relação entre o número de positivos verdadeiros pelo número total de positivos:

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

A métrica do *recall* é a relação entre o número de positivos verdadeiros pelo número de positivos reais:

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

A métrica *F1-score* engloba tanto a precisão quanto o recall:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (12)$$

3 METODOLOGIA

Conforme mencionado anteriormente, tem-se como objetivo a comparação entre resultados obtidos no treinamento de CNNs a partir de *datasets* formados de imagens reais e de imagens sintéticas. Para tal, inicialmente foram gerados bancos de dados por meio de redes generativas, feitos de modo a serem similares ao *dataset* MNIST, para assim permitir a comparação dos resultados de teste.

Em seguida foram treinadas CNNs em três condições, utilizando o *dataset* MNIST, utilizando o *dataset* sintético e utilizando um *dataset* misto. As redes obtidas então foram testadas utilizando os dados de teste do banco de dados MNIST, e os resultados comparados.

3.1 Geração dos *Datasets* sintéticos

Para geração dos bancos de dados de imagens sintéticas foram utilizadas ferramentas disponíveis na internet, sites que utilizam redes generativas para gerar imagens sintéticas utilizando *prompts*, comandos de texto que a rede utiliza para guiar a geração da imagem. Para auxiliar na geração foram utilizados exemplos de imagens pertencentes ao banco de dados MNIST. As seguintes plataformas foram utilizadas para geração de imagens: Canva, Pixlr, DeepAI, Picsart e Crayon (CANVA, Acesso em: 30 set. 2024) (PIXLR, Acesso em: 30 set. 2024) (DEEPAI, Acesso em: 30 set. 2024) (PICSART, Acesso em: 30 set. 2024) (CRAYON, Acesso em: 30 set. 2024).

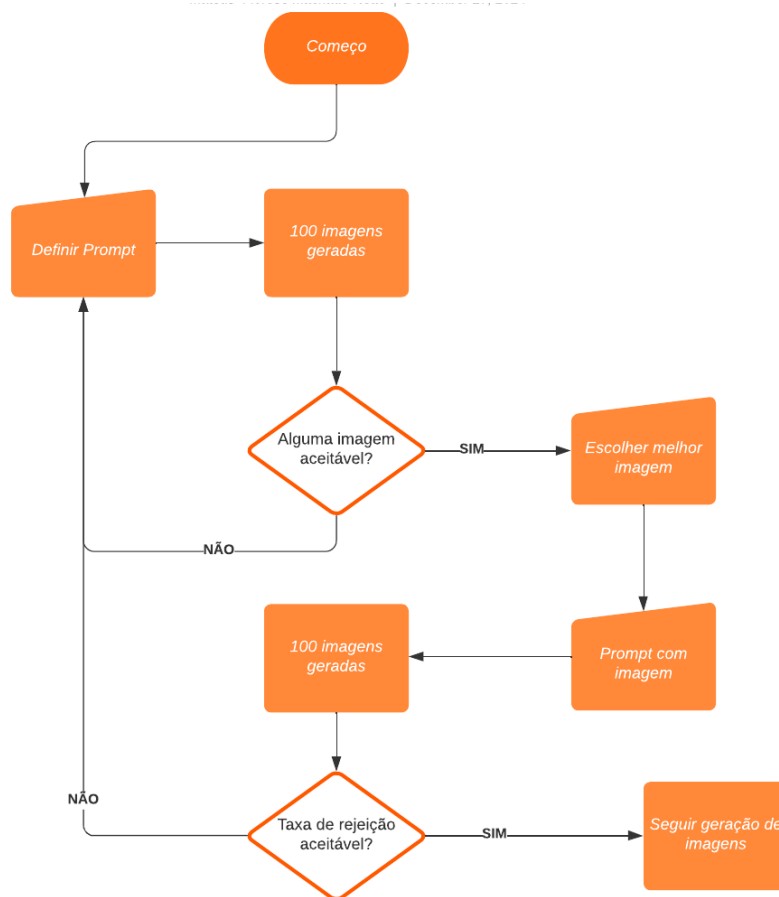
Exemplo de *prompt* utilizado: “Produza uma imagem de 28 pixels por 28 pixels de um único algarismo, sendo ele o número 1, em preto sobre um fundo branco, exclua quaisquer outros símbolos da imagem”. *Prompt* de autoria própria. Para os demais algarismos presentes no banco de dados MNIST foram utilizados *prompts* análogos.

O processo é ilustrado na Figura 4.

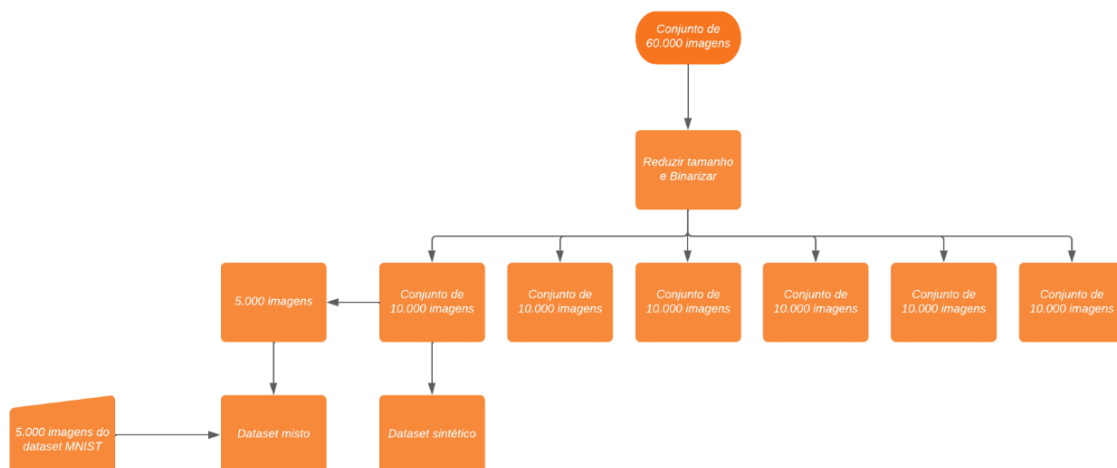
As imagens sintéticas obtidas foram, por conta de limitações das ferramentas utilizadas, de dimensão superiores as do banco de dados MNIST, portanto estas foram reduzidas para as dimensões apropriadas. Nem todas as imagens obtidas puderam ser utilizadas nos *datasets*, parte teve de ser descartada por fugir do contexto desejado, seja por presença de componentes indesejados ou por excesso de estilização do algarismo.

Uma vez obtidas as imagens, estas foram divididas em um total de 6 bancos de dados, cada um constituído de um total de 10.000 imagens. Em seguida foram formados mais 6 bancos de dados, cada um constituído de 5.000 imagens retiradas de um dos 6 bancos de dados originais e mais 5.000 imagens retiradas do banco de dados MNIST. Do *dataset* MNIST foram retiradas 10.000 imagens para constituir um banco de dados de comparação. Este processo é ilustrado na Figura 5.

Figura 4 – Fluxograma do processo de geração de imagens



Fonte: Elaboração própria.

Figura 5 – Fluxograma do processo de montagem dos *datasets*

Fonte: Elaboração própria.

A escolha das imagens para cada banco de dados foi feita de maneira aleatória, porém exclusiva, isto é, a mesma imagem não aparece em mais de um *dataset*. A Tabela 1 ilustra os bancos de dados gerados

Tabela 1 – Bancos de dados gerados

Dataset	tipo de imagens
MNIST	real
sintético1	sintético
sintético2	sintético
sintético3	sintético
sintético4	sintético
sintético5	sintético
sintético6	sintético
misto1	real e sintético
misto2	real e sintético
misto3	real e sintético
misto4	real e sintético
misto5	real e sintético
misto6	real e sintético

Fonte: Elaboração própria.

3.2 Treinamento e Validação

Após gerados os *datasets*, foi realizado o treinamento das CNNs, para cada um dos 13 bancos de dados foi treinada uma rede neural, sendo eles 6 constituídos somente por imagens sintéticas, 1 constituído por imagens do banco de dados MNIST e mais 6 de imagens mistas. O treinamento foi realizado utilizando técnica de *backpropagation* com *cross entropy*. A Tabela 2 apresenta a arquitetura das redes treinadas.

Tabela 2 – Arquitetura das CNNs treinadas

Tipo de Camada	Parâmetros	Função de ativação
Convolutacional 2D	32 filtros	ReLU
Pooling	Max pooling	-
Convolutacional 2D	64 filtros	ReLU
Pooling	Max pooling	-
Flatten	-	-
Fullyconnected	128 neurônios	ReLU
Fullyconnected	10 neurônios	Softmax

Fonte: Elaboração própria.

Com isso foi realizada a validação das CNNs utilizando o *dataset* de validação do banco de dados MNIST. O mesmo *dataset* de validação foi utilizado para todas as redes neurais treinadas, de modo a permitir melhor comparação dos resultados obtidos.

Assim o desempenho de cada CNN foi medido por meio das seguintes métricas: *Accuracy*, *Precision*, *recall* e *F1-score*.

4 RESULTADOS

Foram geradas aproximadamente 100.000 imagens sintéticas para a formação dos bancos de dados, das quais cerca de 20.000 vieram de cada uma das seguintes plataformas: Canva, Pixlr, DeepAI, Picsart e Crayon. Estes valores apresentam tolerância de $\pm 2\%$.

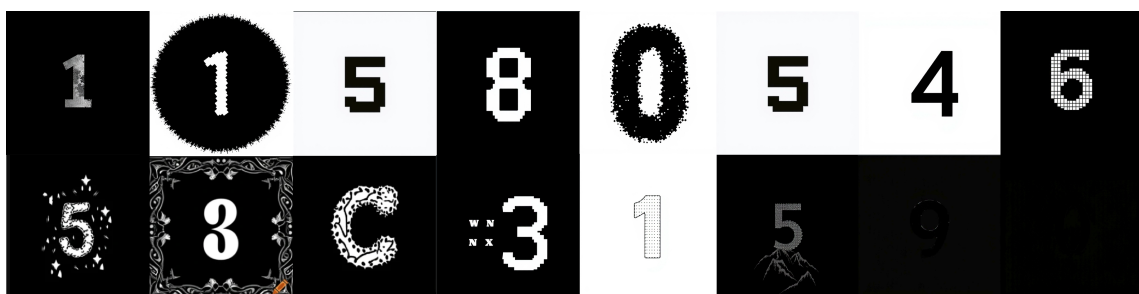
Este total foi dividido entre os algarismos 0 a 9, conforme o *dataset* MNIST, com cerca de 10.000 imagens geradas para cada algarismo. Neste caso, o erro estimado para os valores é de $\pm 4\%$, uma vez que durante o período de confecção e teste do *prompt*, a geração ficou focada em alguns algarismos em déficit dos demais, levando a um número maior de imagens referentes aos mesmos.

Durante a geração das imagens foi de grande auxílio em obter resultados coerentes com o desejado a utilização de imagens extraídas do *dataset* MNIST para ajudar a guiar a criação das imagens.

Deste total, 60.000 foram de fato utilizadas na confecção dos datasets, o restante foi descartado por não atender às necessidades do banco de dados. As principais causas que levaram ao descarte de imagens geradas foram a presença de elementos supérfluos na imagem, o excesso de estilização do algarismo na imagem e a geração de algarismo não correspondente a nenhum dos desejados. O descarte das imagens foi feito de maneira manual ao longo do processo de geração.

Houve uma taxa de utilização das imagens geradas de cerca de 60%, resultado que se deve fortemente a utilização de imagens extraídas do *dataset* MNIST para guiar a geração, prévia ao uso desta abordagem a taxa de rejeição de imagens foi de aproximadamente 60%. A Figura 6 apresenta os resultados mencionados, com a primeira linha indicando imagens aceitas e a segunda imagens descartadas.

Figura 6 – Exemplos de imagens sintéticas geradas



Fonte: Elaboração própria.

As imagens sintéticas obtidas foram de tamanho superior ao desejado de 28x28 pixels utilizado no banco de dados MNIST devido a limitações presentes nas plataformas utilizadas, estas geram imagens de tamanho fixo, a menção do tamanho no *prompt* serve para reduzir a perda de informação na redução das dimensões das imagens. Portanto, as imagens resultantes foram reduzidas às dimensões desejadas.

4.1 Treinamento e Validação

Durante o processo de treinamento foi utilizado um *batch size* de 64, *learning rate* de 0,001 e 10 épocas. Cada CNN foi treinada 5 vezes, em seguida foram utilizados os valores médios obtidos para análise.

A Tabela 3 apresenta o resultado dos testes de desempenho de cada uma das 13 RNAs treinadas, com as linhas representando cada RNA e as colunas indicando a métrica utilizada. As nomenclaturas ‘sintética1’ à ‘sintética6’ indicam as redes neurais treinadas utilizando somente imagens sintéticas, as nomenclaturas ‘mista1’ à ‘mista6’ indicam as redes treinadas utilizando imagens sintéticas e imagens reais e a nomenclatura ‘MNIST’ indica a CNN treinada somente com as imagens do banco de dados MNIST.

Tabela 3 – Resultados e desempenho

Condição	Accuracy	Precision	Recall	F1-Score
MNIST	0.9877 $\pm$ 0.000057	0.9881 $\pm$ 0.000057	0.9877 $\pm$ 0.000057	0.9877 $\pm$ 0.000057
sintética1	0.9822 $\pm$ 0.000056	0.9822 $\pm$ 0.000056	0.9822 $\pm$ 0.000056	0.9822 $\pm$ 0.000056
sintética2	0.9392 $\pm$ 0.000057	0.9392 $\pm$ 0.000057	0.9392 $\pm$ 0.000057	0.9392 $\pm$ 0.000057
sintética3	0.9688 $\pm$ 0.000055	0.9688 $\pm$ 0.000055	0.9688 $\pm$ 0.000055	0.9688 $\pm$ 0.000055
sintética4	0.9838 $\pm$ 0.000058	0.9838 $\pm$ 0.000058	0.9838 $\pm$ 0.000058	0.9838 $\pm$ 0.000058
sintética5	0.9767 $\pm$ 0.000060	0.9768 $\pm$ 0.000060	0.9767 $\pm$ 0.000060	0.9767 $\pm$ 0.000060
sintética6	0.9684 $\pm$ 0.000056	0.9685 $\pm$ 0.000056	0.9684 $\pm$ 0.000056	0.9684 $\pm$ 0.000056
mista1	0.9895 $\pm$ 0.000058	0.9895 $\pm$ 0.000058	0.9895 $\pm$ 0.000058	0.9895 $\pm$ 0.000058
mista2	0.9857 $\pm$ 0.000060	0.9859 $\pm$ 0.000060	0.9857 $\pm$ 0.000060	0.9857 $\pm$ 0.000060
mista3	0.9774 $\pm$ 0.000055	0.9775 $\pm$ 0.000055	0.9774 $\pm$ 0.000055	0.9775 $\pm$ 0.000055
mista4	0.9717 $\pm$ 0.000057	0.9721 $\pm$ 0.000057	0.9717 $\pm$ 0.000057	0.9717 $\pm$ 0.000057
mista5	0.9814 $\pm$ 0.000056	0.9815 $\pm$ 0.000056	0.9814 $\pm$ 0.000056	0.9814 $\pm$ 0.000056
mista6	0.9697 $\pm$ 0.000055	0.9698 $\pm$ 0.000055	0.9697 $\pm$ 0.000055	0.9697 $\pm$ 0.000055

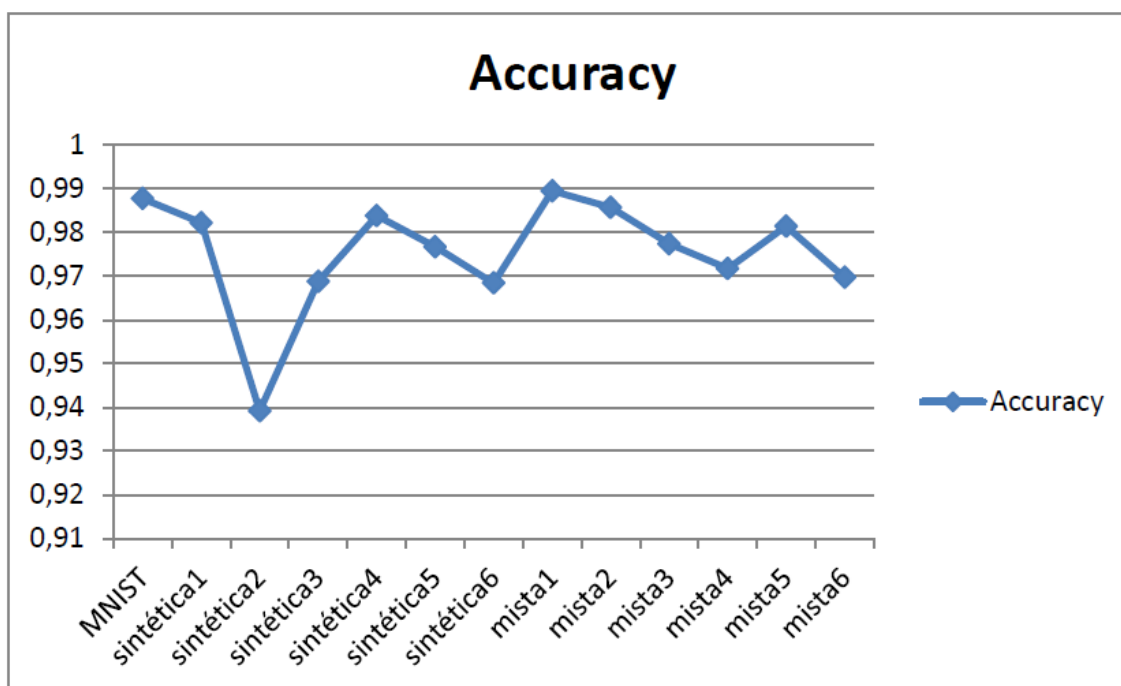
Fonte: Elaboração própria.

A partir da Tabela 3 foram confeccionados 4 gráficos, uma para cada métrica de desempenho utilizada.

A partir dos gráficos apresentados nas Figuras 7 a 10 foi realizado um teste de hipótese de modo a verificar se existem diferenças significativas entre as CNNs treinadas via imagens sintéticas e imagens reais. Para tal foram utilizados os valores de acurácia das redes. Foi calculado um valor médio de amostra de 0.98678, um valor de média de população de 0.9877, um desvio padrão da amostra de 0.0164 e foi utilizado um nível de significância $\alpha = 0.05$. Com isso foi calculado um valor t de -0.2520 , um valor crítico t de 2.0452 e um valor p de 0.8029.

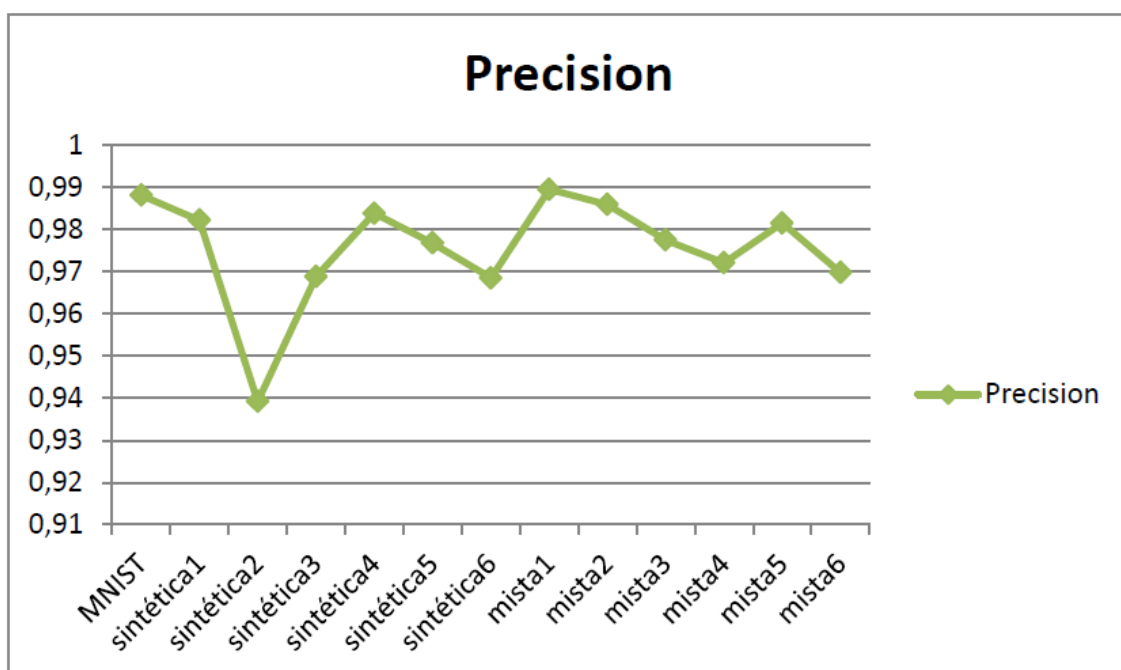
Este teste foi repetido então para as CNNs treinadas com dados mistos, obtendo valor médio de amostra de 0.9867, desvio padrão da amostra de 0.0027, valor t de -2.0286 , valor crítico t de 2.0452 e valor p de 0.0518.

Figura 7 – Gráfico de acurácia por CNN

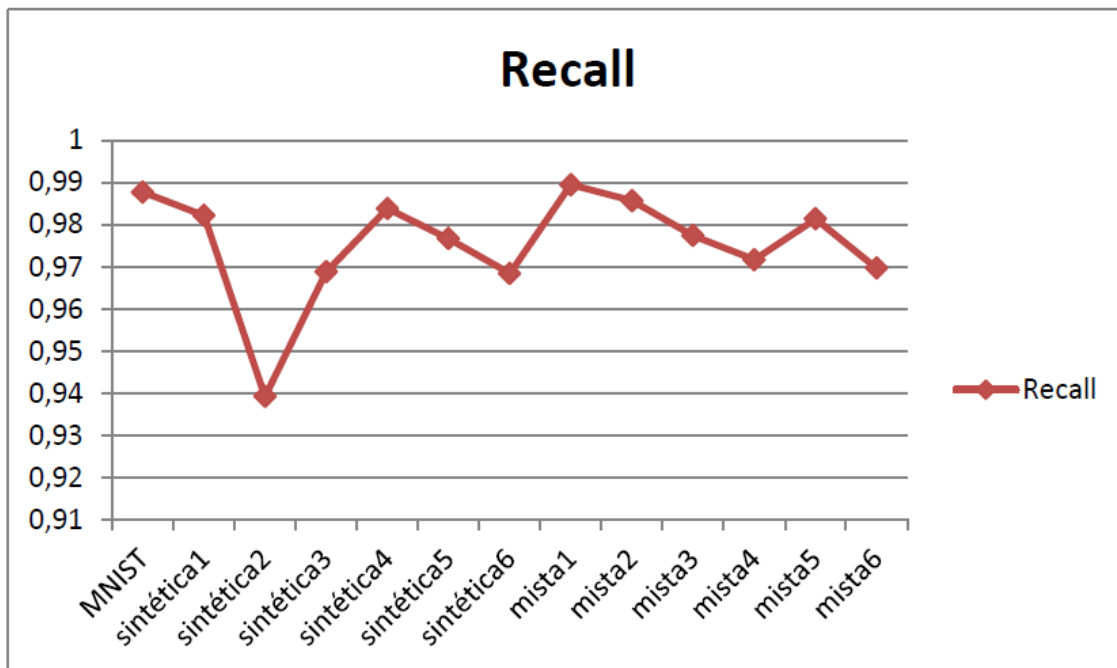


Fonte: Elaboração própria.

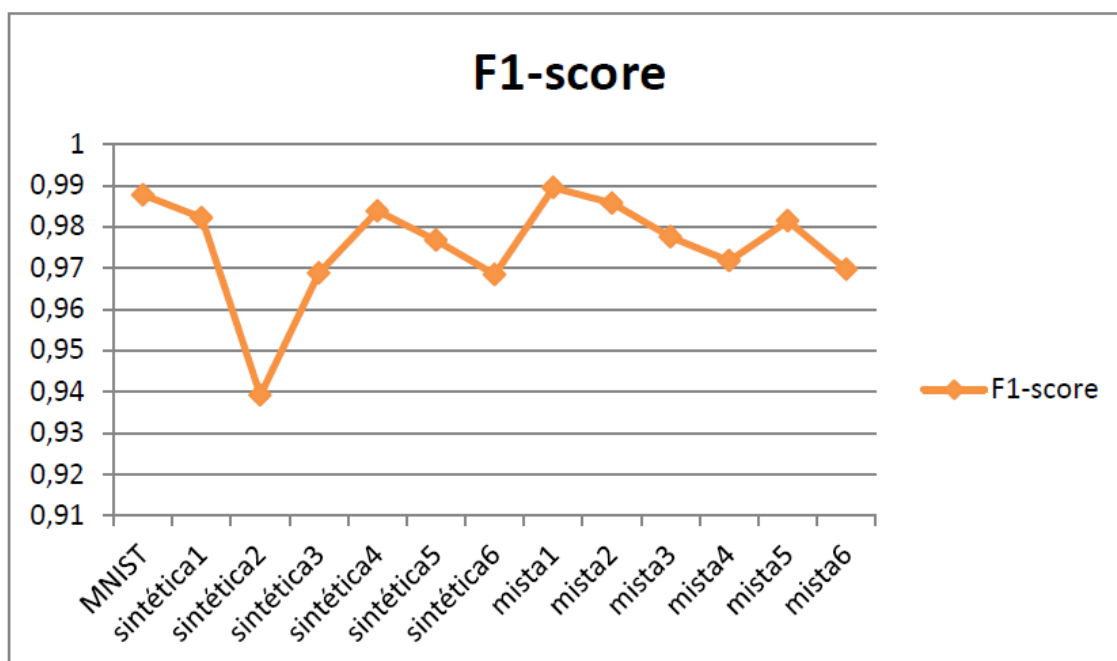
Figura 8 – Gráfico de precisão por CNN



Fonte: Elaboração própria.

Figura 9 – Gráfico de *recall* por CNN

Fonte: Elaboração própria.

Figura 10 – Gráfico de *F1-score* por CNN

Fonte: Elaboração própria.

Assim, a partir destes testes de hipótese pode-se observar que não houveram diferenças significativas entre as CNNs treinadas via imagens sintéticas e imagens reais, com a maior divergência notada não excedendo 5% e as demais não excedendo 2%. Esta diferença é ainda menos perceptível com as redes neurais mistas, onde a maior divergência vista é de 1,8%.

Nota-se que a utilização do *dataset* MNIST como base de comparação para esta pesquisa pode ter influenciado os resultados, uma vez que as imagens deste banco de dados são de maior simplicidade.

5 CONCLUSÃO

Pelos resultados obtidos pode-se concluir a viabilidade do uso de imagens sintéticas para o treinamento de redes neurais artificiais para a classificação de imagens de baixa complexidade, conforme proposto, já que as diferenças observadas nos testes de desempenho entre a CNN de comparação e as redes treinadas com imagens sintéticas e mistas foram pequenas.

Porém, nota-se que o processo de obtenção destas imagens se mostrou mais complexo que o estimado, com alta taxa de rejeição das imagens geradas, cerca de 40% das quais foram descartadas. A utilização de imagens para utilizar como guia e exemplo durante a geração provou-se essencial para reduzir essa taxa de reprovação. Porém, é importante notar que é possível a utilização de imagens sintéticas obtidas como exemplo e guia.

Conforme observado, é possível que a simplicidade do banco de dados MNIST utilizado como base de comparação tenha influenciado os resultados obtidos, tanto na geração das imagens, em particular a taxa de reprovação, como os resultados dos testes de desempenho. Fica, portanto, aberto para pesquisas futuras a verificação destes resultados com um banco de dados de imagens de maior complexidade.

REFERÊNCIAS

- BISHOP C. M., . N. N. M. *Pattern Recognition and Machine Learning*. [S.l.]: New York: springer., 2006.
- BOTTOU, L. Large-scale machine learning with stochastic gradient descent. In: *In Proceedings of COMPSTAT 2010, 10th Symposium of the European Society of Computational Methods in Statistics*. [S.l.: s.n.], 2010.
- BROCK ANDREW, J. D.; SIMONYAN, K. Large scale gan training for high fidelity natural image synthesis. *arXiv:1809.11096*, 2018.
- CANVA. Ai image generator. <https://www.canva.com/ai-image-generator/>, Acesso em: 30 set. 2024.
- CRAYON. Ai image generator. <https://www.craiyon.com/>, Acesso em: 30 set. 2024.
- DEEPAI. Ai image generator. <https://deepai.org/machine-learning-model/text2img>, Acesso em: 30 set. 2024.
- DENG, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6), 2012.
- FEI-FEI L., D. J. . L. K. Imagenet: Constructing a large-scale image database. *Journal of vision*, 9(8), 2009.
- FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4), p. 193–202, 1980.
- GLOROT X., B. A. . B. Y. Deep sparse rectifier neural networks. In: *In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*. [S.l.: s.n.], 2011. p. 315–323.
- GOODFELLOW I., B. Y. . C. A. *Deep Learning*. [S.l.]: MIT Press, 2016.
- GOODFELLOW I. J., P.-A. J. M. M. X. B. W.-F. D. O. S. . . B. Y. Generative adversarial nets. In: *In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 27. [S.l.: s.n.], 2014. p. 2672–2680.
- HASTIE T., T. R.-. F. J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. [S.l.]: Springer, 2009.
- HAYKIN, S. *Neural Networks and Learning Machines*. [S.l.]: Pearson Education India, 2009.
- HE K., Z. X.-R. S. . S. J. Deep residual learning for image recognition. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. *Proceedings...* [S.l.]: IEEE, 2016. p. 770–778.
- HINTON, G. E. Neural networks for machine learning. *Coursera*, 2012.
- HOERL A. E., . K. R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), p. 55–67, 1970.

IOFFE S., . S. C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, p. 448–456, 2015.

JOULIN A., M. T. . M. P. Inferring networks of explanations in large knowledge bases. In: *In Proceedings of the 27th International Conference on Machine Learning*. [S.l.: s.n.], 2014. p. 357–364.

KRIZHEVSKY A., . H. G. Learning multiple layers of features from tiny images. 2009.

KRIZHEVSKY A., S. I. . H. G. E. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems 25*, 2012.

LECUN Y., B. L. O. G. B. . M. K. *Efficient Backpropagation*. In *Neural Networks: Tricks of the Trade*. [S.l.]: Springer, 1998.

LECUN Y., B. Y. . H. G. Deep learning. *nature*, 521(7553), p. 436–444, 2015.

MCCULLOCH W.S., P. W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5, p. 115–133, 1943.

MINSKY M., . P. S. Marvin minsky and seymour papert, perceptrons. *Cambridge, MA: MIT Press, Introduction*, p. 1–20, 1969.

O'SHEA, K. An introduction to convolutional neural networks. *arXiv:1511.08458*, 2015.

PICSART. Ai image generator. <https://picsart.com/ai-image-generator/>, Acesso em: 30 set. 2024.

PIXLR. Ai image generator. <https://pixlr.com/image-generator/>, Acesso em: 30 set. 2024.

RAMCHOUN H., G. Y. E. M. J. I. M. A. *Multilayer perceptron: Architecture optimization and training*. [S.l.: s.n.], 2016.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), p. 386–408, 1953.

RUMELHART DAVID E., G. E. H.; WILLIAMS, R. J. Learning internal representations by error propagation, parallel distributed processing: Explorations in the microstructures of cognition. *Cambridge, MA: MIT Press*, p. 318–362, 1986.

SHRIVASTAVA A., P. T. T. O. S. J. W. W. . W. R. Learning from simulated and unsupervised images through adversarial training. In: *IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 2107–2116.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), p. 267–288, 1996.

TIELEMAN T., . H. G. E. Lecture 6.5—rmsprop: Divide the gradient by a running average of its recent magnitude. *Neural Networks for Machine Learning.*, 2012.

TOBIN J., F. R. R. A. S. J. Z. W. . A. P. Domain randomization for transferring deep neural networks from simulation to the real world. In: *In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2017. p. 23–30.

YEGNANARAYANA, B. *Artificial neural networks*. [S.l.]: PHI Learning Pvt. Ltd., 2009.