UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
DEPARTAMENTO DE ESTATÍSTICA E INFORMÁTICA
PROGRAMA DE PÓS–GRADUAÇÃO EM INFORMÁTICA APLICADA

EMESON JOSÉ SANTANA PEREIRA

UM ESTUDO SOBRE O IMPACTO DO AUMENTO DE DADOS NO TREINAMENTO DE REDES NEURAIS CONVOLUCIONAIS NA PRESENÇA DE RÓTULOS RUIDOSOS

RECIFE-PE
2024

EMESON JOSÉ SANTANA PEREIRA

UM ESTUDO SOBRE O IMPACTO DO AUMENTO DE DADOS NO TREINAMENTO DE REDES NEURAIS CONVOLUCIONAIS NA PRESENÇA DE RÓTULOS RUIDOSOS

Dissertação submetida à Coordenação do Programa de Pós–Graduação em Informática Aplicada do Departamento de Estatística e Informática da Universidade Federal Rural de Pernambuco, como parte dos requisitos necessários para obtenção do grau de Mestre.

ORIENTADOR: Filipe Rolim Cordeiro

RECIFE-PE

2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

P436e Pereira, Emeson José Santana
Um estudo sobre o impacto do aumento de dados no treinamento de redes neurais convolucionais na presença de rótulos ruidosos / Emeson José Santana Pereira. - 2024.
72 f. : il.

Orientador: Filipe Rolim Cordeiro.
Inclui referências.

Dissertação (Mestrado) - Universidade Federal Rural de Pernambuco, , Recife, 2024.

1. anotações ruidosas. 2. aprendizagem profunda. 3. classificação. I. Cordeiro, Filipe Rolim, orient. II. Título

CDD

EMESON JOSÉ SANTANA PEREIRA

UM ESTUDO SOBRE O IMPACTO DO AUMENTO DE DADOS NO TREINAMENTO DE REDES NEURAIS CONVOLUCIONAIS NA PRESENÇA DE RÓTULOS RUIDOSOS

Dissertação submetida à Coordenação do Programa de Pós–Graduação em Informática Aplicada do Departamento de Estatística e Informática da Universidade Federal Rural de Pernambuco, como parte dos requisitos necessários para obtenção do grau de Mestre.

Aprovada em: 29 de Maio de 2024.

BANCA EXAMINADORA

Filipe Rolim Cordeiro (Orientador)
Universidade Federal Rural de Pernambuco
Departamento de Computação - DC

Valmir Macario Filho
Universidade Federal Rural de Pernambuco
Departamento de Computação - DC

Danilo Silva
Universidade Federal de Santa Catarina
Departamento de Engenharia Elétrica e Eletrônica - EEL

À minha família, ao meu orientador, à todos aqueles que contribuem para a evolução humana através da Ciência da Computação e, em especial, à minha esposa que esteve comigo e me apoiou em todo o percurso.

Agradecimentos

Agradeço, em especial, a minha esposa, por estar sempre ao meu lado, me apoiando em todos os momentos.

À minha família que sempre me apoiou e esteve ao meu lado durante toda a jornada da minha vida que pude vivenciar até os dias de atuais.

E, por fim, ao Prof. Filipe Rolim, pela excelente orientação e por ter me apoiado em todos os momentos, até mesmo nos momentos mais difíceis e incertos deste trabalho.

"Não importa o quão rápido você anda, mas
a força de vontade para nunca parar."

(Confúcio)

Resumo

Rótulos ruidosos estão presentes nas grandes bases de dados do mundo real e representam um grande desafio na aprendizagem de máquina, pois trazem consigo grandes impactos no treinamento de modelos de aprendizagem, tais como redução de desempenho e problemas de sobreajuste. Ao longo dos anos, vários trabalhos têm abordado o tema e trazido técnicas para lidar com a presença de ruídos de rótulo em conjuntos de dados de treinamento. No entanto, poucos estudos avaliam o impacto dos aumentos de dados, como a escolha de técnicas de aumentos de dados para treinamento de redes neurais profundas com anotações ruidosas. Sendo assim, neste trabalho avaliamos o desempenho de modelos de aprendizagem profunda, no treinamento com a presença de rótulos ruidosos, utilizando diferentes aumentos de dados e suas combinações. São avaliadas estratégias clássicas, ou seja, transformações básicas de processamento de imagens e técnicas do estado da arte com diferentes níveis de ruídos sintéticos. Para este estudo, foram utilizados os conjuntos de dados MNIST, CIFAR-10, CIFAR-100 e a base de dados do mundo real Clothing1M. Os métodos foram avaliados utilizando a métrica de acurácia. Os resultados mostraram que a seleção adequada de técnicas de aumento de dados pode melhorar drasticamente o desempenho do modelo na presença de ruídos de rótulo, obtendo melhorias de até 177,62% na acurácia relativa a base de dados de teste em tarefas de classificação de imagens, quando comparadas ao desempenho de um modelo treinado sem o uso de aumento de dados. Os resultados também mostram que é possível obter um ganho absoluto de até 6% com a seleção adequada de aumentos de dados para a estratégia de treinamento do estado da arte DivideMix.

Palavras-chave: Rótulo Ruidoso. Aprendizagem profunda. Classificação

Abstract

Noisy labels are present in large real-world databases and represent a great challenge in machine learning, because they bring a great impacts on the training of learning models, such as reduced performance and overfitting problems. Over the years, several works have addressed the topic and proposed techniques to deal with the presence of label noise in training datasets. However, few studies evaluate the impact of data augmentations, such as the choice of data augmentation techniques for training deep neural networks. Therefore, in this work we evaluate the performance of deep learning models, when training with the presence of noisy labels, using different data augmentations and their combinations. Classical strategies are evaluated, that is, basic image processing transformations and state-of-the-art techniques with different levels of synthetic noise. For this work, the MNIST, CIFAR-10, CIFAR-100 datasets and the Clothing1M real-world database were used. The methods were evaluated using the accuracy metric. The results showed that the appropriate selection of data augmentation techniques can drastically improve model performance in the presence of label noise, achieving improvements up to 177.62% of best test accuracy in image classification tasks, when compared to the performance of a model trained without the use of data augmentation. They also showed that it is possible to achieve an absolute gain up to 6% with proper selection of data augmentations for the state-of-the-art DivideMix training strategy.

Keywords: Noisy Labels. Deep Learning. Classification

Lista de Figuras

Lista de tabelas

Lista de Siglas

CMYK	*Cyan-Magenta-Yellow-Black*
CNN	*Convolutional Neural Network*
ConvNet	*Convolutional Neural Network*
IA	Inteligência Artificial
ILSVRC	*ImageNet Large Scale Visual Recognition Challenge*
RGB	*Red-Green-Blue*
RNA	Rede Neural Artificial
ReLU	*Rectified Linear Unit*
ResNet	*Residual Network*
SGD	*Stochastic Gradient Descent*

Sumário

1 Introdução

1.1 Motivação

Com a chegada das redes neurais convolucionais (KHAN et al., 2020), ou CNNs, e outros modelos, a aprendizagem profundo vem ganhando ampla atenção, permitindo avanços notáveis em áreas como visão computacional (PATEL; PATEL, 2020), aplicações médicas (ANWAR et al., 2018), processamento de linguagem natural (GOLDBERG, 2022), reconhecimento de fala (ABDEL-HAMID et al., 2014) e sistemas de transporte inteligente (SIROHI et al., 2020). Apesar disso, existem muitos desafios na área de aprendizagem profunda a serem explorados, tais como métodos, parâmetros e tempo de treinamento, necessidade de grandes conjuntos de dados e base de dados com rótulos ruidosos (DONG et al., 2021; AHMED et al., 2023).

Dentre os desafios da aprendizagem profunda, os rótulos ruidosos em base de dados de imagens, cujo suas amostras têm um rótulo diferente que define sua verdadeira classe (CORDEIRO; CARNEIRO, 2020) e estão presentes nas mais diversas bases de dados, vêm ganhando atenção, uma vez que os impactos que causam no treinamento dos modelos são significativamente relevantes, como a degradação do desempenho e redução da generalização do modelo (ZHANG et al., 2021). Rotular grandes bases de dados é um processo custoso e demorado, tornando o processo difícil e criando a necessidade de explorar maneiras mais rápidas e baratas, como *crowdsourcing*[1] (YU et al., 2018) e consultas na internet (XIE et al., 2019), que podem produzir anotações incorretas. Existe também a possibilidade de termos anotações incorretas em bases de dados menores, causadas por erro humano durante o processo de anotação.

Portanto, é comum existirem anotações incorretas nas mais diversas bases de dados e, diante disso, vários métodos vêm sendo propostos para lidar com os rótulos ruidosos (CORDEIRO; CARNEIRO, 2020) e suas abordagens incluem funções de perda robustas, ajuste de rótulos, ponderação de amostras, meta aprendizagem, aprendizagem através de comitê, aprendizagem semissupervisionada, entre outras. Dentre as técnicas existentes para lidar com a presença de rótulos ruidosos, várias (LI et al., 2020; LIU et al., 2020; CORDEIRO et al., 2021) têm utilizado técnicas de aumento de dados para obter uma

[1] *Crowdsourcing* é um modelo de terceirização cujo propósito é reunir diferentes pessoas com o objetivo de realizar uma tarefa ou solucionar um problema (HOSSAIN; KAURANEN, 2015).

melhor generalização do modelo. As estratégias de aumento de dados visam aumentar o conjunto de dados (SHORTEN; KHOSHGOFTAAR, 2019), realizando transformações nos dados de entrada, como aumento de contraste de imagens, melhoria na nitidez, borrões, recortes aleatórios, entre outros, para obter uma maior variedade de dados de entrada e, assim, tendo a função de regularização durante a aprendizagem, trazendo uma melhor generalização ao modelo. Entretanto, pouco se estuda sobre a importância e o impacto das técnicas de aumento de dados na aprendizagem de modelos em bases de dados com anotações ruidosas.

Neste trabalho, é investigado o impacto das técnicas de aumento de dados utilizadas no treinamento de modelos em bases de dados contendo anotações ruidosas, comparando o impacto de transformações de imagens básicas, combinações de transformações básicas e técnicas de aumento de dados mais robustas, além de propor e avaliar modificações nos aumentos de dados utilizados em técnicas que lidam com anotações ruidosas, para obter um melhor desempenho de treinamento e generalização.

1.2 Problema de Pesquisa

Rótulos ruidosos são rótulos que não representam a classe verdadeira de sua respectiva amostra de dados (CORDEIRO; CARNEIRO, 2020). Alguns exemplos destes rótulos podem ser vistos ao analisar base de dados rotuladas por especialistas, onde as imagens são difíceis de rotular, como são os casos de imagens médicas utilizadas na detecção de câncer. Estes tipos de rótulos estão presentes nas mais diversas bases de dados utilizadas no treinamento de modelos de aprendizagem profunda e trazem um grande impacto negativo para o processo de aprendizagem, levando os modelos, muitas vezes, ao sobreajuste ou, como também é conhecido, *overfitting* (ZHANG et al., 2021). Com isso, os modelos terminam se ajustando exatamente aos dados de treinamento, acarretando em um baixo desempenho em relação ao dados desconhecidos, o que foge ao propósito dos modelos.

No mesmo contexto, têm-se as técnicas de aumento de dados, onde, através delas, se busca uma melhor generalização, pois são capazes de fornece uma regularização para o treinamento dos modelos (SHORTEN; KHOSHGOFTAAR, 2019). As estratégias de aumento de dados, motivadas principalmente pela necessidade de grandes conjuntos

de dados de treinamento na aprendizagem profunda, buscam aplicar transformações nas amostras de entrada com o intuito de diversificar ainda mais o conjunto de dados, mantendo as amostras dentro do escopo e trazendo um ganho de generalização para os modelos. Os aumentos de dados são amplamente utilizados em treinamento de modelos de aprendizagem profunda, o que incluem técnicas de treinamento de modelos em base de dados com rótulos ruidosos, porém pouco se aprofunda sobre o impacto que os aumento de dados têm no treinamento com anotações ruidosas.

Logo, é preciso investigar mais afundo o impacto das estratégias de aumento de dados em base de dados com anotações ruidosas, buscando compreender melhor o impacto da escolha das estratégias a serem utilizadas, da escolha dos possíveis parâmetros de cada estratégia e em qual contexto cada uma se aplica melhor. Também é preciso buscar soluções mais simples de lidar com anotações ruidosas, tendo em vista a complexidade e o custo computacional de técnicas do estado da arte. Com isso, as questões que o presente projeto visa responder são:

1. Qual o impacto da utilização de diferentes técnicas de aumentos de dados no treinamento de modelos em conjuntos de dados com anotações ruidosas?
2. Qual é o ganho obtido ao combinar diferentes estratégias de aumento de dados para lidar com base de dados contendo rótulos ruidosos?
3. Qual é o ganho obtido ao modificar as estratégias de aumento de dados utilizadas em técnicas que lidam com anotações ruidosas?
4. Qual a importância da escolha das estratégias de aumento de dados em contexto que envolvam bases de dados com anotações ruidosas?

1.3 Objetivos

1.3.1 Objetivo Geral

Este trabalho tem o objetivo de analisar o impacto das técnicas de aumento de dados no treinamento de modelos de redes neurais convolucionais em bases de dados que contêm rótulos ruidosos, analisando diferentes técnicas de aumento de dados aplicadas em diferentes tipos de rótulos ruidosos, juntamente com diferentes modelos, conjuntos de dados e técnicas de aprendizagem que lidam com anotações ruidosas. Para atingir este objetivo geral, alguns objetivos específicos foram propostos, como é mostrado a seguir.

1.3.2 Objetivos Específicos

Com intuito de atingir o objetivo geral deste trabalho, os seguintes objetivos específicos foram definidos:

- Análise das técnicas de aumento de dados e métodos do estado da arte, utilizados para lidar com anotações ruidosas, que serão utilizados neste trabalho;
- Análise dos resultados obtidos;
- Identificação da melhor combinação de técnicas de aumento de dados para treinamento de modelos na presença de rótulos ruidosos;
- Melhoria de técnicas de treinamento do estado da arte através da otimização de estratégias de aumento de dados.

1.4 Estrutura do Trabalho

Este trabalho é estruturado de forma a ajudar na total compreensão da análise feita. Tendo isso em vista, informações importantes, contidas neste trabalho, são devidamente esclarecidas antes de sua utilização.

Por fim, este estudo está organizado em 6 capítulos, como informado a seguir:

1. O capítulo 1, onde foi apresentado o contexto do problema, a motivação para o estudo, o problema de pesquisa tratado e os objetivos.
2. No Capítulo 2 são apresentados fundamentos básicos e indispensáveis para a total compreensão da pesquisa realizada. Nesse capítulo, são abordados os conceitos de rede neurais artificiais, aprendizagem profunda, redes neurais convolucionais, transferência de conhecimento na aprendizagem de máquina, métodos de aumento de dados e, por fim, o conceito de rótulos ruidosos.
3. As contribuições do estado da arte nos últimos anos e os trabalhos relacionados são abordados no Capítulo 3.
4. No Capítulo 4 é explorada a metodologia utilizada nesta pesquisa, abrangendo as bases de dados utilizadas, os tipos de rótulos ruidosos abordados, as técnicas de aumento de dados utilizadas e seus parâmetros, as redes neurais convolucionais utilizadas e as ferramentas utilizadas no desenvolvimento e na realização dos experimentos.
5. No Capítulo 5 são abordados os experimentos realizados e os resultados obtidos.

6. Por fim, temos no Capítulo 6 as considerações finais sobre a pesquisa realizada, conclusões obtidas a partir dos resultados dos experimentos e os trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo é trazida uma breve descrição sobre os assuntos importantes abordados neste estudo. Portanto, será apresentado um resumo do conhecimento necessário para que seja obtido um entendimento aprofundado sobre o trabalho aqui desenvolvido.

Os assuntos abordados por este capítulo são: 2.1. Fundamentos de imagens digitais, importante para entender como são constituídas as imagens e como agem as transformações nelas aplicadas; 2.2. E, por fim, conceito de aprendizado de máquina onde veremos também as técnicas de redes neurais artificiais, aprendizagem profunda, redes neurais convolucionais e suas arquiteturas mais conhecidas na literatura, além de conceitos como aumento de dados e rótulos ruidosos.

2.1 Fundamentos de Imagem Digital

2.1.1 *Pixel*

O *pixel*, que vem originalmente do termo "*picture element*", é a menor unidade de uma imagem digital que representa um único ponto de cor e brilho. Uma imagem é formada por um conjunto finito de *pixels* organizados juntos em uma grade, onde cada um possui uma localização no eixo bi-dimensional x, y (GONZALEZ et al., 2002), como pode ser visto na Figura 1.

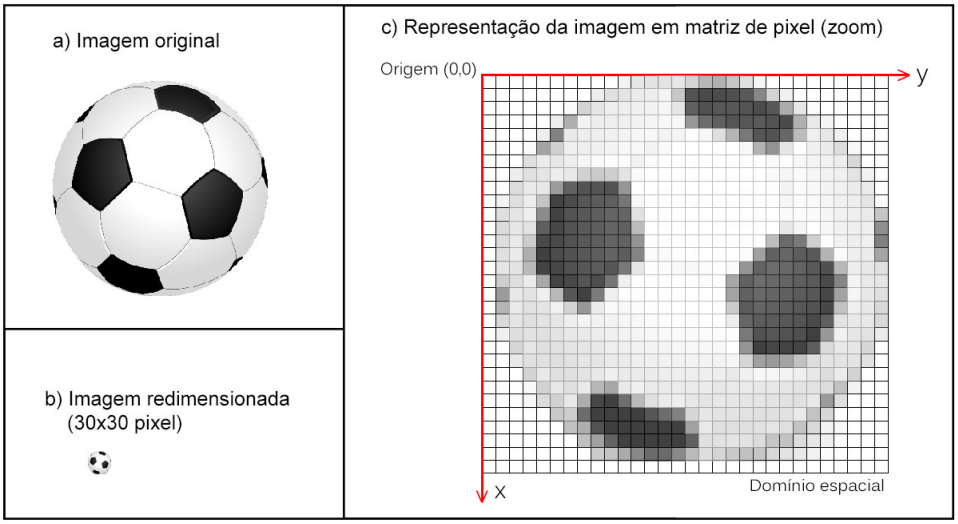
Cada *pixel* representa um único ponto na imagem e possui seu próprio valor de cor e intensidade de brilho, onde o brilho de um pixel é determinado pela intensidade dos componentes de cor, normalmente especificado usando combinações de valores de vermelho, verde e azul (RGB[1]) ou outros modelos de cores como CMYK (ciano, magenta, amarelo e preto), no caso das impressoras. Para as imagens em tons de cinza, o *pixel* apresenta apenas um canal de cor, neste caso o preto, e o valor do canal define a intensidade do tom, como mostra a Figura 2.

A resolução de uma imagem é determinada pelo número de *pixels* que ela contém. Imagens de resolução mais alta têm mais *pixels* por unidade de área, resultando em

[1] RGB significa vermelho, verde e azul - *Red-Green-Blue* - que são as cores primárias de luz usadas em monitores digitais e dispositivos de imagem, onde cada canal de cor pode ser representado por um valor entre 0 e 255.

maior detalhe e clareza. É por isso que as imagens parecem mais nítidas em telas de alta resolução em comparação com telas de baixa resolução. As imagens são projetadas em telas utilizando milhões de *pixels* organizados em uma matriz de linhas e colunas, as vezes, tão densa que chegam a parecer, a olho nu, que são um elemento único. Logo, é através de uma coordenada, que envolve uma linha e uma coluna, que o *pixel* é localizado em uma imagem (MCCONNELL, 2005).

Figura 1 – Representação da imagem digital através de uma matriz de *pixels*.

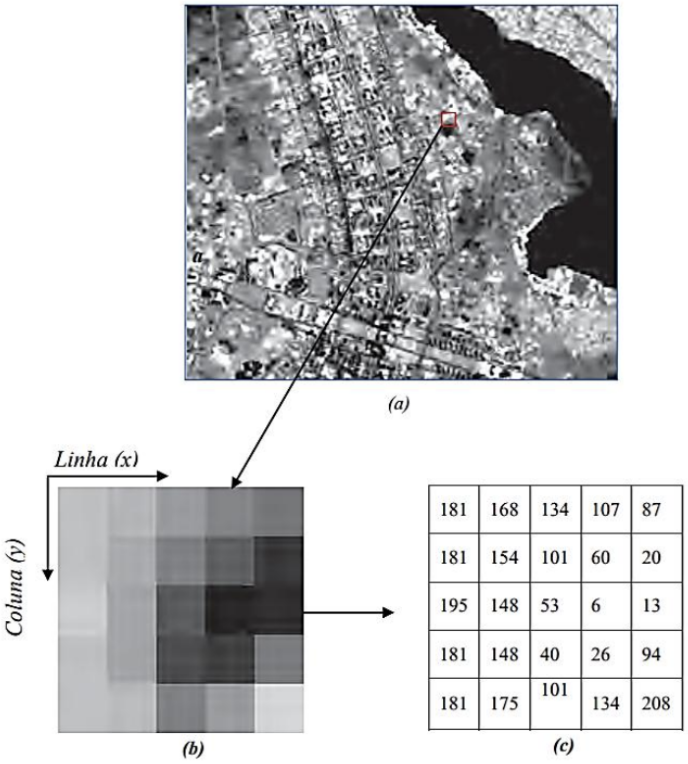


Fonte: França (2016).

2.1.2 Representação de Imagem Digital

Sendo um conceito fundamental na visão computacional e no processamento de imagens, a representação matricial de uma imagem é amplamente utilizada nos meios digitais, que são essencialmente números dispostos em linhas e colunas. Cada número na matriz representa a intensidade ou valor da cor de um *pixel* específico na imagem. O tamanho da matriz corresponde às dimensões da imagem, ou também conhecida como resolução da imagem, sendo que o número de linhas e colunas indica a altura e a largura, respectivamente (MUSCI, 2016). Na Figura 3 é mostrado um exemplo desta representação.

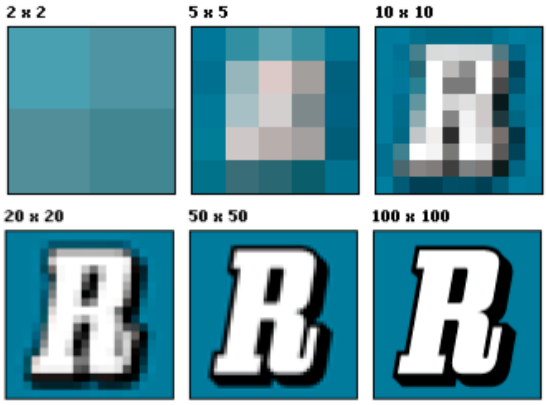
Figura 2 – Representação de *pixels* de canal único, ou tons de cinza. a) Imagem original; b) Um recorte de 5×5 *pixels* da imagem (a); c) Valores de cada *pixels*.



181	168	134	107	87
181	154	101	60	20
195	148	53	6	13
181	148	40	26	94
181	175	101	134	208

Fonte: Meneses e Almeida (2012).

Figura 3 – Representação de uma imagem em várias resoluções.

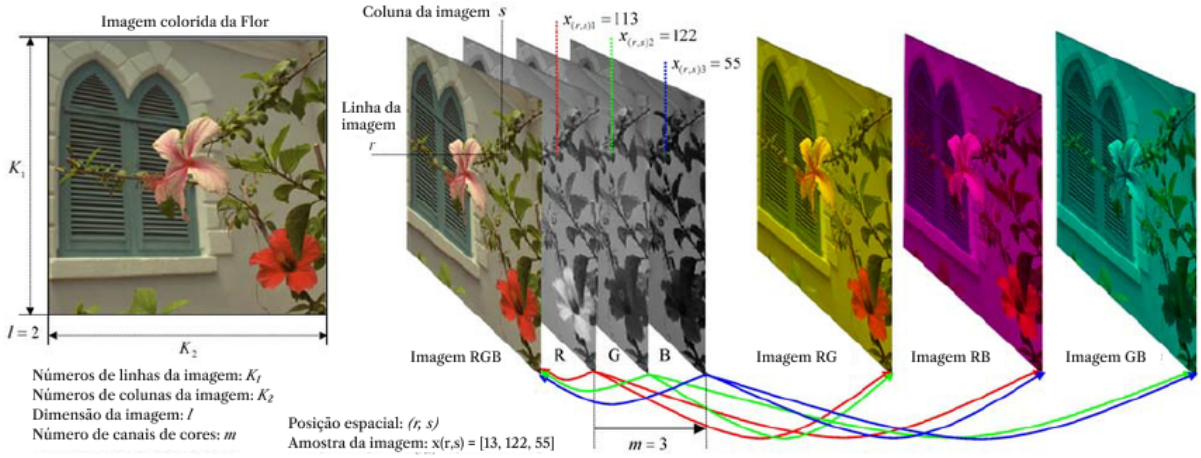


Fonte: Adaptado de Wikimedia Commons (2006).

2.1.3 Sistema de Cores RGB

Representando o sistema de cores mais utilizado no meio digital, o RGB é uma das formas mais comuns de representar cores em dispositivos e aplicativos digitais. RGB significa vermelho, verde e azul, que são as cores primárias da luz. Neste sistema, as cores são criadas combinando diferentes intensidades destas três cores primárias (IBRAHEEM et al., 2012). Cada canal de cores (vermelho, verde e azul) é normalmente representado usando 8 *bits*, o que significa que cada canal pode ter 256 níveis de intensidade variando de 0 a 255. Ao combinar diferentes intensidades de vermelho, verde e azul, uma ampla gama de cores pode ser representada. A Figura 5 mostra a composição de uma imagem RGB e como as cores são formadas.

Figura 4 – Exemplo de composição de uma imagem no padrão RGB.



Fonte: Adaptado de Lukac e Plataniotis (2018).

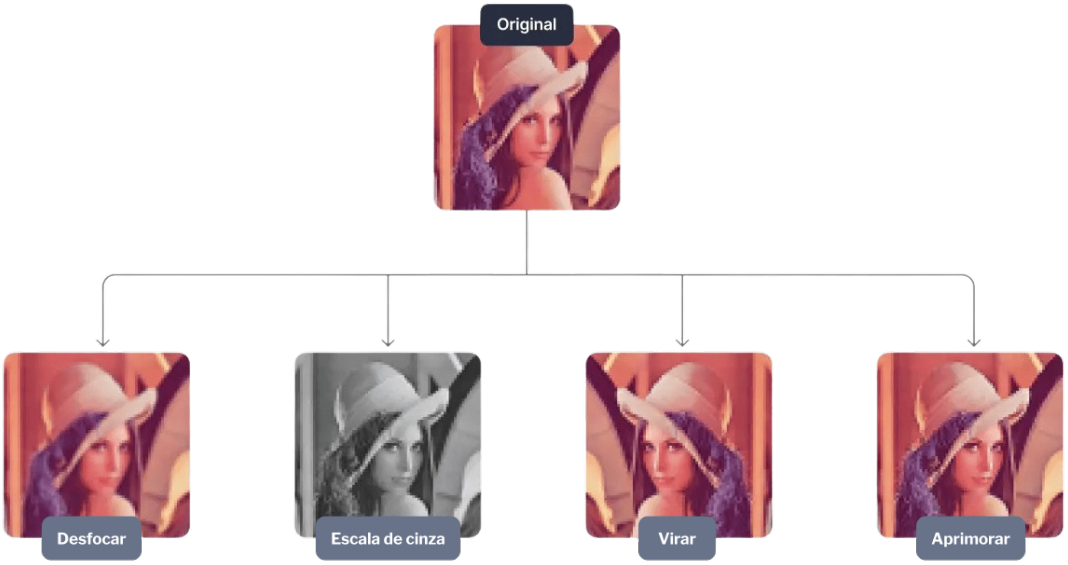
2.1.4 Processamento de Imagens

O processamento de imagens envolve a manipulação e análise de imagens digitais utilizando diversas técnicas e algoritmos. Abrange uma ampla gama de tarefas destinadas a melhorar a qualidade visual das imagens, extrair informações úteis ou tornar as imagens adequadas para aplicações específicas (KURUVILLA et al., 2016).

Dentre as técnicas de processamento de imagem, temos a melhoria de imagem, que envolve melhorar a aparência visual de uma imagem ajustando seu contraste, brilho, nitidez e equilíbrio de cores; Restauração de imagem, que visa recuperar ou melhorar a qualidade de imagens degradadas ou corrompidas; Filtragem de imagens, que é usada para

modificar as características espaciais de uma imagem aplicando operações de convolução; Segmentação de imagens, que envolve particionar uma imagem em regiões ou objetos significativos; Extração de características, que é usada para identificar e extrair informações ou características relevantes de imagens; Detecção e reconhecimento de objetos, que envolve a localização e identificação de objetos específicos dentro de uma imagem, enquanto o reconhecimento envolve a identificação do tipo ou categoria de objetos; Registro de imagem, que envolve o alinhamento de múltiplas imagens da mesma cena ou objeto para permitir comparação, análise ou fusão; E, por fim, compressão de imagem, que é usada para reduzir o espaço de armazenamento necessário para imagens e, ao mesmo tempo, minimizar a perda de qualidade visual.

Figura 5 – Exemplos de técnicas de processamento de imagem.



Fonte: Adaptada de Kundu (2022).

No geral, o processamento de imagens desempenha um papel crucial em uma ampla gama de aplicações, incluindo imagens médicas, imagens de satélite, vigilância, robótica, fotografia digital e muito mais. Além disso, nos permite extrair informações valiosas das imagens, melhorar a sua qualidade e torná-las mais adequadas para análise e interpretação.

2.2 Aprendizagem de Máquina

A aprendizagem de máquina é um subconjunto da inteligência artificial (IA) que se concentra no desenvolvimento de algoritmos e técnicas que permitem aos computadores aprender com os dados e fazer previsões ou decisões sem serem explicitamente programados. Baseia-se na ideia de que os computadores podem aprender com padrões e características derivadas de dados, em vez de depender de instruções explícitas.

Em Mitchell (1997), a aprendizagem de máquina é definida como: "Diz-se que um programa de computador aprende pela experiência E, com respeito a algum tipo de tarefa T e performance P, se sua performance P nas tarefas em T, na forma medida por P, melhoram com a experiência E"[2]. Exemplificando, pode-se dizer que T é uma tarefa como, por exemplo, a classificação de imagens de animais. P é um método quantitativo utilizado para avaliar o desempenho do algoritmo, como, por exemplo, quantos porcento das amostras o algoritmo classifica corretamente. Por fim, E seria a base de dados, contendo as imagens e suas classificações, utilizada para treinar o algoritmo.

Além disso, a aprendizagem de máquina atualmente pode ser dividida em 4 categorias (RUSSELL; NORVIG, 2016). São elas: aprendizagem supervisionada, aprendizagem não supervisionada, aprendizagem semissupervisionada e aprendizagem por reforço. Para este trabalho, daremos ênfase a aprendizagem supervisionada e não supervisionada.

Na aprendizagem supervisionada, o algoritmo é treinado em um conjunto de dados rotulados, onde cada entrada está associada a uma saída ou rótulo correspondente. O objetivo é aprender um mapeamento de entradas para saídas, para que o algoritmo possa fazer classificações ou previsões precisas sobre amostras de dados desconhecidas. Em outras palavras, a aprendizagem supervisionada funciona baseada na ideia de se ter um supervisor ou professor, que fornece as respostas corretas durante o treinamento (GOODFELLOW et al., 2016). Cada amostra apresentada ao algoritmo pode ser representada na forma de tupla (x_i, y_i), onde x_i, dado como a entrada, é o vetor de características associado a classe y_i, dada como a saída esperada. Esta técnica permite que o algoritmo possa realizar classificações ou previsões para dados de entrada não vistos anteriormente.

A ideia principal é que, através dos dados rotulados, o algoritmo possa aprender de

[2] Versão original em inglês: *"a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"* (MITCHELL, 1997).

forma generalizada as principais características do conjunto de dados apresentado, ou seja, um modelo para identificar o tipo de um carro deve aprender características importantes que os diferenciam, como os pneus, faróis, lataria, entre outras. Este tipo de aprendizagem incluem tarefas de classificação, como identificar a categoria de um item, e regressão, como prever valores futuros de uma ação na bolsa de valores.

Diferente do supervisionado, na aprendizagem não supervisiona não é informado ao algoritmo o rótulo das amostras de dados utilizadas no treinamento, portanto, o algoritmo fica responsável por reconhecer os padrões e a relação entre as amostras e agrupá-las. Neste caso, com a ausência dos rótulos ou classes das amostras, não existe mais a figura do "professor", responsável anteriormente por dar *feedbacks* sobre a acurácia dos resultados. Sendo assim, esse tipo de aprendizagem é indicado para problemas no qual existe pouca ou nenhuma informação sobre a classe das amostras de treinamento (GOODFELLOW et al., 2016). Um exemplo da utilização desta técnica são os algoritmos de *clustering* que agrupam amostras de dados semelhantes com base em alguma métrica de similaridade.

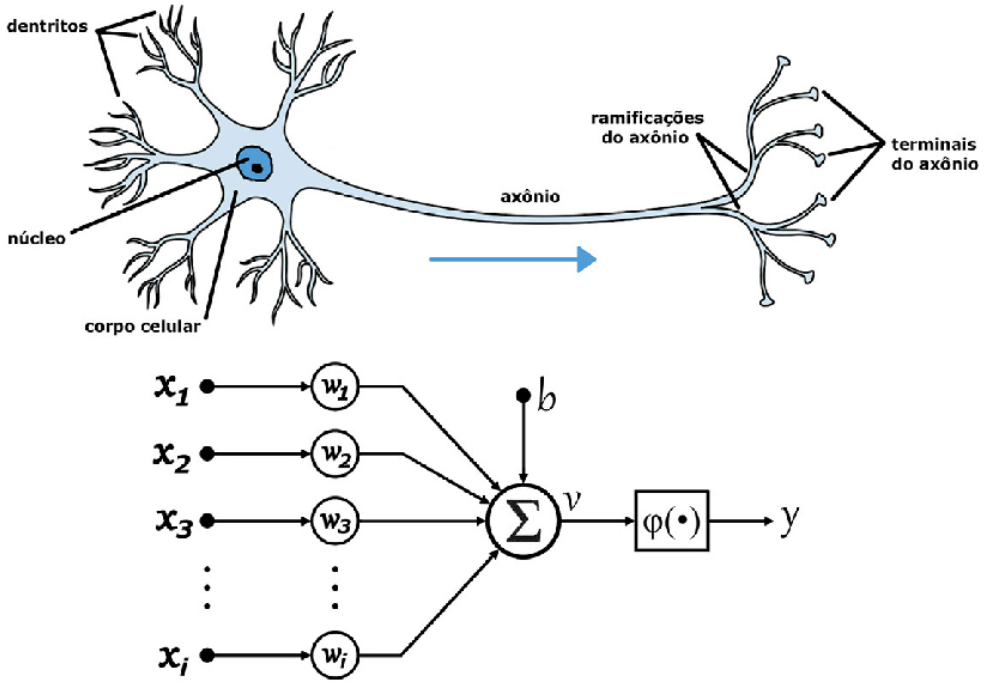
2.2.1 Redes Neurais Artificiais

Categorizada como uma aprendizagem supervisionada, as redes neurais artificiais (RNAs), comumente citadas como redes neurais, são a base da inteligência artificial moderna e do aprendizado de máquina. As RNAs são modelos computacionais inspirados na estrutura e função das redes neurais biológicas, como as encontradas no cérebro humano, consistindo em nós interconectados, ou neurônios artificiais, organizados em camadas (HAYKIN, 1994). Cada neurônio recebe sinais de entrada, os processa e produz um sinal de saída, que é então passado para outros neurônios da rede.

Formado por centenas de bilhões de neurônios interligados através de sinapses com capacidade de processar informações em paralelo, o cérebro humano é a principal inspiração para as RNAs (WANG, 2003). Baseando-se nas interações entre os neurônios, são simuladas as redes neurais artificiais capazes de aprender padrões. Um neurônio artificial simula um neurônio biológico, onde os dendritos são as unidades de entrada de dados, sendo esses dados processados pelo corpo celular ou função de ativação no caso do modelo artificial, e o axônio, sendo a unidade de saída no modelo artificial, pode ser conectado a outros neurônios formando assim a RNA (CIDRIM et al., 2019).

Na Figura 6 é possível observar um modelo de neurônio biológico e um artificial. No modelo artificial, a conexão entre os neurônios têm pesos sinápticos, ou simplesmente pesos da rede, que regulam a transmição de sinais de uma célula para a outra, representando assim o conhecimento adquirido (HAYKIN, 1994). Sendo assim, o sinal é levado pelo axônio, representado por x_i, e se comunica com os dentritos, representado pela multiplicação $x_i \times w_i$, onde w_i é o peso sináptico. A aprendizagem da rede ocorre através do ajuste dos pesos sinápticos durante o treinamento, ou seja, os pesos aumentam ou diminuem para que a rede aprenda determinados padrões. Logo, os dentritos levam o sinal para o corpo celular, onde todos os sinais são somados formando um só sinal de saída. Além disso, nas RNAs existe também o *bias* (b) que é um valor ajustável, como um peso, e que é adicionado ao somatório de sinais. Por fim, para simular a ativação de sinal de um neurônio, temos a função de ativação ($\varphi(\cdot)$) que recebe o somatório dos sinais e retorna um resultado representando a saída final do neurônio.

Figura 6 – Estrutura básica de um neurônio biológico e sua representação computacional.



Fonte: Cidrim et al. (2019).

As funções de ativação são componentes cruciais em RNAs. Eles introduzem não linearidade na rede, permitindo que a rede aprenda e represente padrões complexos em dados. Dentre as funções de ativação, temos a limiarização que retorna 1 se o sinal atingir determinado limiar ou 0, caso contrário (GERSHENSON, 2003). Além disso, temos a

função de ativação sigmoide, onde a entrada é comprimida em um intervalo entre 0 e 1, obedecendo a fórmula $\sigma(x) = \frac{1}{1+e^{-x}}$. As funções sigmoides são geralmente usadas em problemas que necessitam prever a probabilidade dos dados analisados (HAN; MORAGA, 1995). A Tanh, ou tangente hiperbólica, também uma função de ativação, semelhante à função sigmoide, mas que comprime a entrada em um intervalo entre -1 e 1 (KALMAN; KWASNY, 1992), sendo geralmente utilizada para classificações que envolvem duas classes. Ainda abordando funções de ativação, temos também a função Softmax, comumente usada na camada de saída para problemas de classificação multiclasse, onde o valor de cada saída é a probabilidade de um dado ser de uma determinada classe do problema. A função Softmax converte a saída de uma RNA em uma distribuição de probabilidade relativa a múltiplas classes, garantindo que a soma de todos os valores de saída seja 1, ou seja, $softmax(x_i) = \frac{e^{x_i}}{\sum_{N}^{j=1} e^{x_j}}$, onde N é o número de classes (BRIDLE, 1990a; BRIDLE, 1990b). Além das funções já vistas, também temos a *Rectified Linear Unit* (ReLU), função que tem sido amplamente usada em tarefas de visão computacional (DAHL et al., 2013). Esta função define todos os valores negativos como zero e deixa os valores positivos inalterados. Sua fórmula é dada por $ReLU(x) = \max(0, x)$. A função ReLU ajuda com o problema do desaparecimento do gradiente e acelera significativamente o treinamento, permitindo uma convergência mais rápida (IDE; KURITA, 2017).

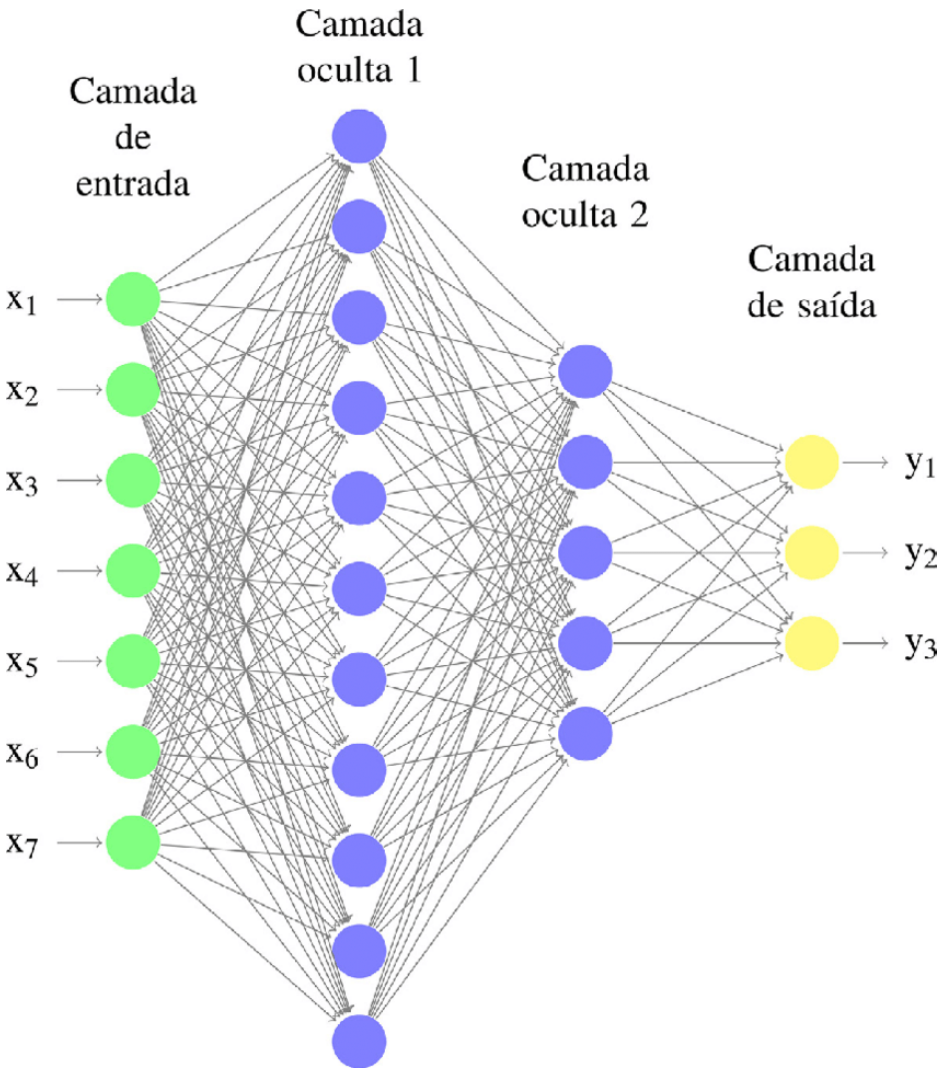
Referindo-se a aprendizagem de uma RNA, pode-se dizer que existe um processo de ajuste de pesos com o intuito de minimizar os erros na saída da rede neural, melhorando assim seu desempenho. Este processo é conhecido como *backpropagation* (WERBOS; JOHN, 1974; RUMELHART et al., 1986; LECUN et al., 1989), abreviação para *backward propagation of errors*, e trata-se de um processo de ajuste de pesos da rede neural com o objetivo de reduzir ao máximo a diferença entre a saída dada pela rede e a saída esperada (RUMELHART et al., 1986). Este processo é realizado retroativamente de modo a ajustar todos os pesos baseando-se no impacto de cada um deles na saída da RNA (ALPAYDIN, 2014).

De maneira geral, as RNAs são composta por um conjunto de neurônios interligados como num grafo acíclico[3], comumente estando totalmente conectados uns aos outros, uma camada após a outra, onde a saída de um neurônio pode ser dada como entrada de outro. Na arquitetura das RNAs existem três tipos de camadas, são elas: a camada de entrada,

[3] Grafo acíclico é um grafo dirigido sem ciclo, ou seja, dado um vértice v, não há nenhuma ligação dirigida iniciando e finalizando em v.

que basicamente são os dados de entrada; as camadas escondidas ou ocultas, podendo existir uma ou mais; e a camada de saída. Na Figura 7 pode-se observar um exemplo de arquitetura de uma RNA.

Figura 7 – Arquitetura de uma rede neural artificial.



Fonte: Cidrim et al. (2019).

Por fim, a arquitetura de uma RNA deve ser adaptada ao problema que se quer tratar e a outros fatores, como a quantidade de dados disponíveis para o treinamento da rede.

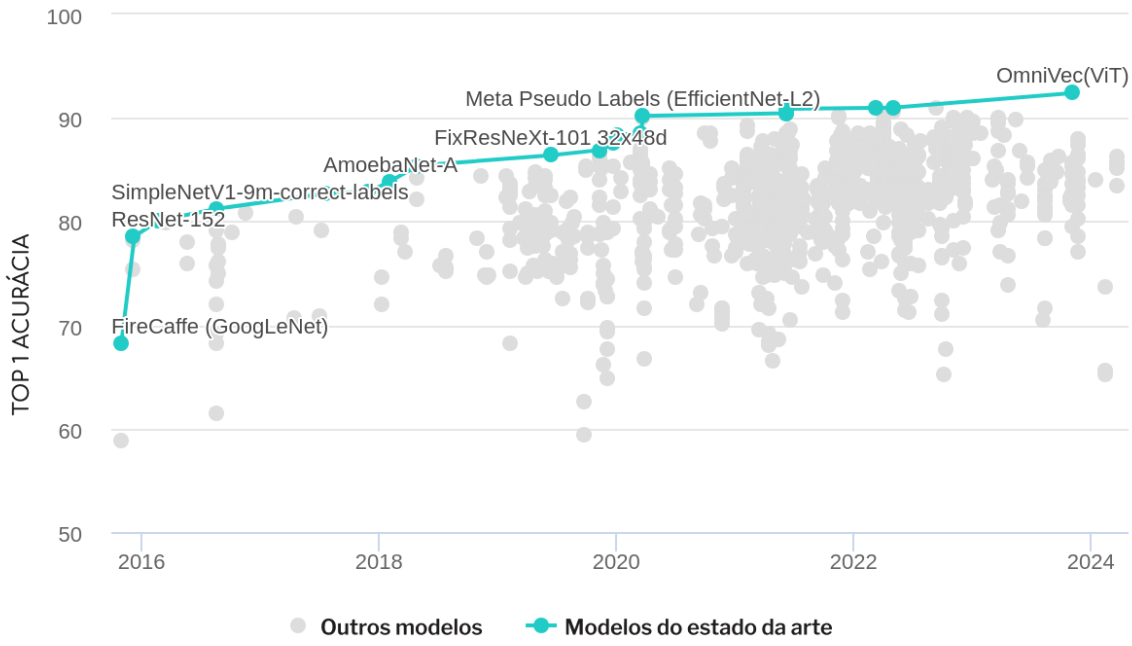
2.2.2 Aprendizagem Profunda

A aprendizagem profunda é um subcampo da inteligência artificial e da aprendizagem de máquina que tem atraído atenção significativa nos últimos anos devido

às suas capacidades na resolução de problemas complexos em vários domínios (BENGIO et al., 2013). Na Figura 8 é possível ver um dos grandes avanços da aprendizagem profunda, mais especificamente na visão computacional, quando observamos os resultados obtidos ao longo dos anos no desafio de aprendizagem utilizando a base de dados ImageNet.

Uma rede neural profunda, diferente das RNAs clássicas, pode lidar com funções de grande complexidade. Isso se dá devido ao fato de que são adicionadas a este tipo de rede mais camadas e mais neurônios dentro de cada camada, tanto quanto necessário for para lidar com o dado problema (GOODFELLOW et al., 2016). Grande parte das tarefas que estão relacionadas a um mapeamento de um vetor de entrada para um vetor de saída, podem ser realizadas através da aprendizagem profunda, considerando que o modelo seja grande o suficiente e exista um grande conjunto de dados de treinamento.

Figura 8 – O progresso das taxas de acurácia das técnicas de aprendizagem profunda em relação ao conjunto de dados ImageNet.



Fonte: Adaptado de Papers With Code (2024).

A técnica de aprendizagem profunda consiste em amplificar características relevantes dos dados de entrada e descartar as demais. Para isso, cada camada se torna responsável por extrair um conjunto distinto de características que vão sendo passadas para as camadas a frente. Logo, quanto mais profunda a camada da rede, mais complexas são as características que esta camada consegue extrair, dado que as características vão sendo agregadas ou recombinadas a cada camada que os dados vão passando. Um exemplo disto pode ser

visto na Figura 9, onde é mostrado a extração ou reconhecimento de características de uma rede neural profunda, dado um problema de reconhecimento facial.

Um ponto importante da aprendizagem profunda é que as camadas de extração de características não são projetadas manualmente, as máquinas são capazes de aprender e definir estas camadas a partir dos dados (LECUN et al., 2015). No entanto, dada a complexidade dos problemas tratados pelas técnicas de aprendizagem profunda, os modelos têm dificuldades de alcançar uma boa generalização, ou seja, um bom desempenho em conjuntos de dados de teste. Com o intuito de melhorar a performance dos modelos nos dados de teste, várias estratégias foram criadas. Esses métodos são conhecidos como regularização (GOODFELLOW et al., 2016).

Algoritmos de aprendizagem profunda são extremamente dependentes de grandes volumes de dados para obter uma boa generalização. Entretanto, em muito casos, ter disponível um grande volume de dados para treinar o modelo não é uma realidade. Pensando nisso, estratégias como o aumento de dados, ou do inglês *data augmentation*, foram criadas para melhorar o treinamento dos modelos em base de dados menores ou problemas que exijam um treinamento em um volume de dados muito grande (SHORTEN; KHOSHGOFTAAR, 2019). Mais a frente serão abordadas com mais detalhes as técnicas de aumento de dados.

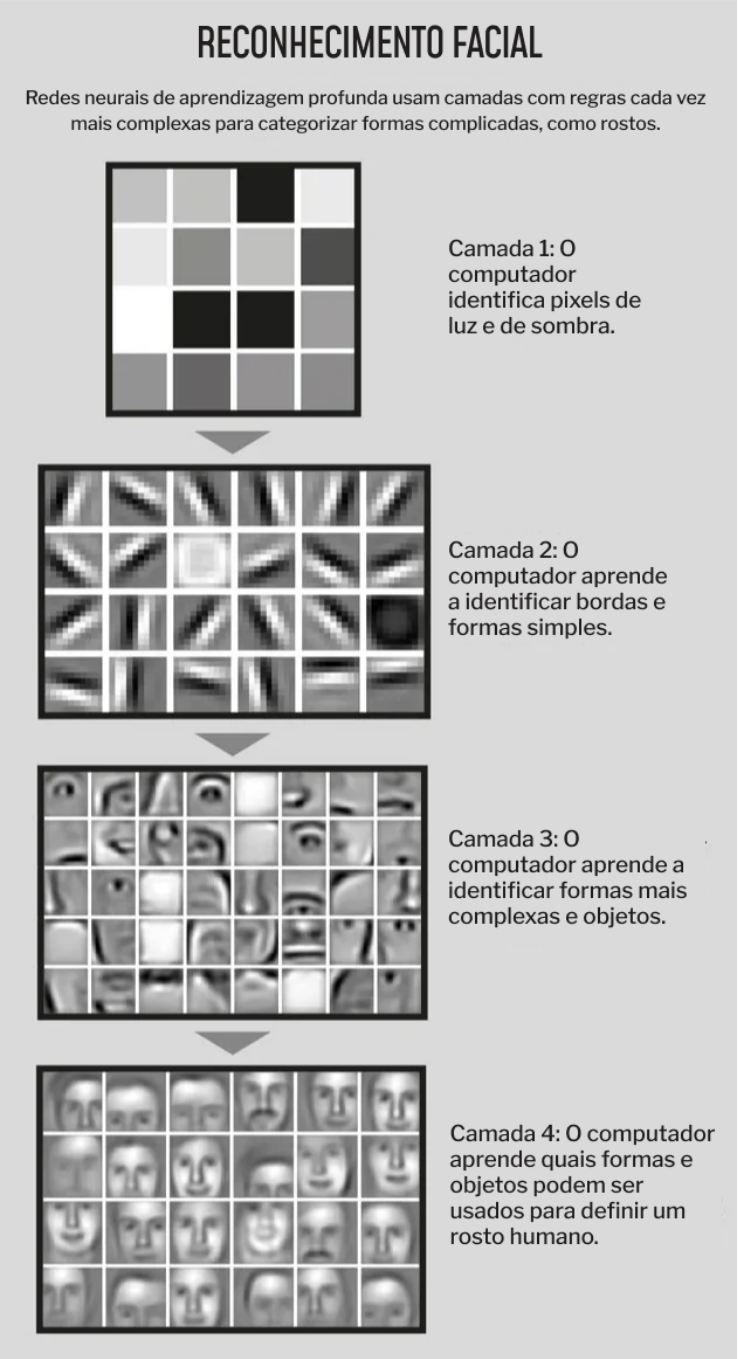
A seguir, serão abordadas as redes neurais convolucionais, tipo de rede popular na área da aprendizagem profunda por conta de seu excelente desempenho em tarefas de visão computacional.

2.2.3 Redes Neurais Convolucionais

As redes neurais convolucionais, do inglês *convolutional neural networks* (CNNs ou ConvNets), são um tipo de modelo de aprendizado profundo projetado especificamente para processar dados de grade estruturada, como imagens, e têm sido incrivelmente bem-sucedidas em várias tarefas de visão computacional (GOODFELLOW et al., 2016). Este tipo de arquitetura de rede a cada dia vem ganhando mais notoriedade, tudo isso após uma CNN vencer o desafio *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC)[4] (KRIZHEVSKY et al., 2012). Com isso, as CNNs vêm cada vez mais sendo

[4] *ImageNet Large Scale Visual Recognition Challenge* foi um concurso que avaliava os algoritmos de detecção de objetos e classificação de imagens em larga escala através de seus desempenhos no conjunto

Figura 9 – Exemplo de reconhecimento e extração de características da aprendizagem profunda.



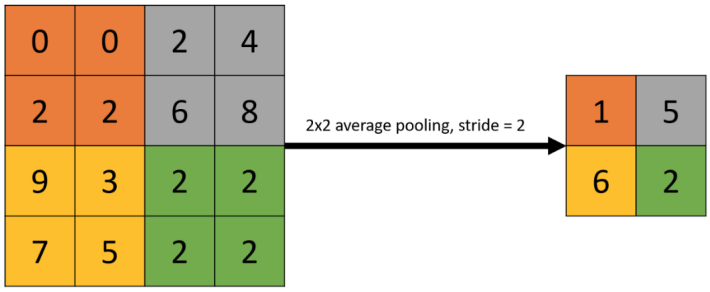
Fonte: Adaptado de Jones (2014).

aplicadas em diversas áreas do conhecimento e são, atualmente, o estado da arte em diversas aplicações, principalmente na área de visão computacional.

Em uma CNN, cada neurônio é responsável por processar os dados após passarem por várias camadas convolucionais. Essas camadas consistem em um conjunto de filtros ou *kernels* que podem ser aprendidos pela rede. Cada filtro aplica a convolução ao longo da imagem de entrada, calculando produtos escalares entre os pesos do filtro e a entrada em cada posição. Esta operação extrai características locais da entrada, preservando os relacionamentos espaciais (MAIRAL et al., 2014).

Nesse contexto, os filtros são pequenas matrizes aplicadas aos dados de entrada para extração de recursos. Esses *kernels* são deslizados sobre os dados de entrada, realizando operações de multiplicação e agregando elemento a elemento para produzir mapas de recursos, que são representações dos dados de entrada em diferentes níveis de abstração. Ainda nesse contexto, temos o *stride* que representa o tamanho do passo do movimento do filtro na imagem. Logo, com uma imagem de tamanho 4×4, um filtro de tamanho 2×2 e utilizando um *stride* de tamanho 2, significa que, após o filtro ser aplicado nos *pixels* enquadrados pela matriz 2×2, o filtro será deslocado em 2 *pixels* antes de ser executado novamente. Este processo ocorre repetidamente, até que toda a imagem seja percorrida, como mostra a Figura 10.

Figura 10 – Exemplo da aplicação de um filtro de tamanho 2×2 que calcula a média com um *stride* de tamanho 2.



Fonte: Zawadzki (2018).

O nome "rede neural convolucional" é baseado no fato de que a rede emprega uma operação matemática chamada convolução[5] com o objetivo de extrair características importantes dos dados de entrada. Redes convolucionais são, de forma simples, redes

de dados conhecido como ImageNet (RUSSAKOVSKY et al., 2015).

[5] Convolução é um operador matemático linear que resulta numa função que mede a soma do produto de outras duas funções ao longo da região subentendida pela superposição delas, conforme o deslocamento existente entre as funções (O'NEIL et al., 1963).

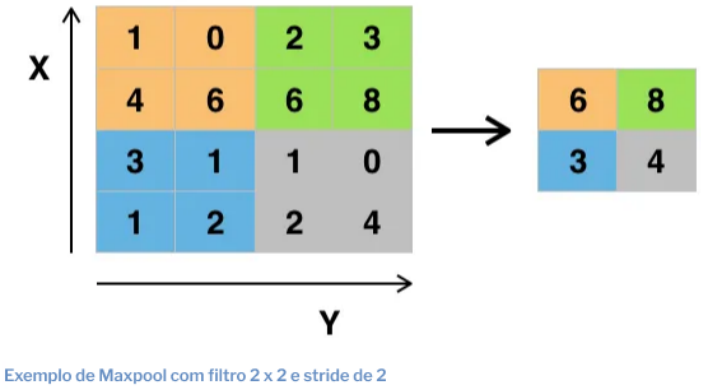
neurais que usam convolução, substituindo a multiplicação geral de matrizes, em pelo menos uma de suas camadas (O'NEIL et al., 1963).

As CNNs são, de modo geral, compostas por camadas de convolução com a função ReLU aplicada as saídas, camadas de *polling*, camadas totalmente conectadas e uma camada de saído, onde é aplicada uma função de ativação que, na maioria das vezes, é usada a função Softmax com o intuito se obter a probabilidade para cada classe existente no problema tratado (GOODFELLOW et al., 2016).

As camadas de convolução são responsáveis pela aplicação dos filtros, ou *kernels*, e a sua saída é chamada de mapa de características (*feature map*). Além disso, uma mesma camada da rede é responsável por gerar múltiplos mapas de características (HAYKIN, 1994). Já as camadas de *polling* são responsáveis por reduzir as dimensões espaciais dos mapas de características, reduzindo os cálculos e controlando o sobreajuste. Desta forma, as camadas de *pooling* reduzem as dimensões dos dados através da seleção de características relevantes (HAYKIN, 1994). Um exemplo de técnica aplicada nesta camada é a *max pooling*, que basicamente é um filtro de dimensões iguais com *stride* de tamanho baseado no problema tratado, onde é obtido como resultado o valor máximo de cada sub-região do mapa de características. É possível observar um exemplo da aplicação da técnica de *max pooling* na Figura 11. Outra técnica de *pooling* amplamente utilizada é a da média, do inglês *average pooling*. Esta técnica consiste na obtenção do valor médio de cada sub-região do mapa de características, ao invés do valor máximo, como é o caso da *max pooling*. Vale ressaltar que as CNNs consistem em várias camadas de convolução e *pooling*, mas não é necessário ser nesta ordem. Além disso, ambas as camadas fazem parte da etapa de extração de características.

De maneira geral, o processo realizado por uma CNN consiste em, dada uma matriz de dados de entrada, uma imagem por exemplo, a rede extrai características através da camada de convolução, enquanto a camada de *pooling* combina as características semanticamente similares. Desta forma, combinações de características locais produzem áreas de interesse, onde essas áreas são combinadas formando objetos (BENGIO et al., 2015). Em outras palavras, as camadas iniciais da CNN são responsáveis por aprender padrões de borda da imagem, onde quanto mais funda a camada, mais complexa são as representações que elas aprendem, enquanto a última camada é responsável pelo resultado final, onde se tratando de um problema de classificação, a rede deve retornar a probabilidade

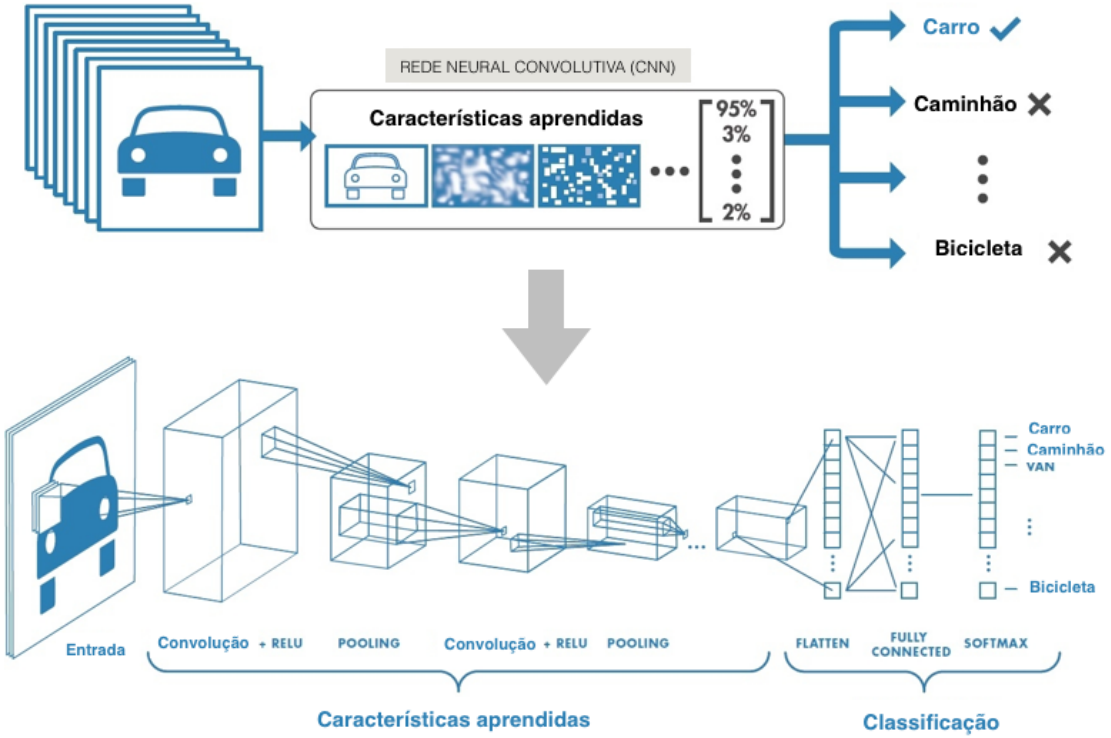
Figura 11 – Exemplo de aplicação do *max pooling*.



Fonte: Adaptado de Estevam (2019).

dos dados de entrada serem de uma determinada classe do problema tratado. A Figura 12 demonstra um passa a passo de todo o processo executado por uma CNN.

Figura 12 – Funcionamento de uma rede neural convolucional.



Fonte: Adaptado de MathWorks® (2018).

Com o passar dos anos, várias técnicas e arquiteturas de CNNs foram surgindo, cada uma com suas características singulares. Entretanto, todas compartilham a ideia de aplicar camadas de extração de características em conjunto com uma rede totalmente conectada para obter um resultado final. A diferença entre os vários modelos propostos está,

principalmente, em alguns características de suas arquiteturas, o que inclui a disposição de cada camada. A seguir falaremos um pouco sobre alguns dos principais modelos de redes neurais convolucionais.

2.2.3.1 LeNet

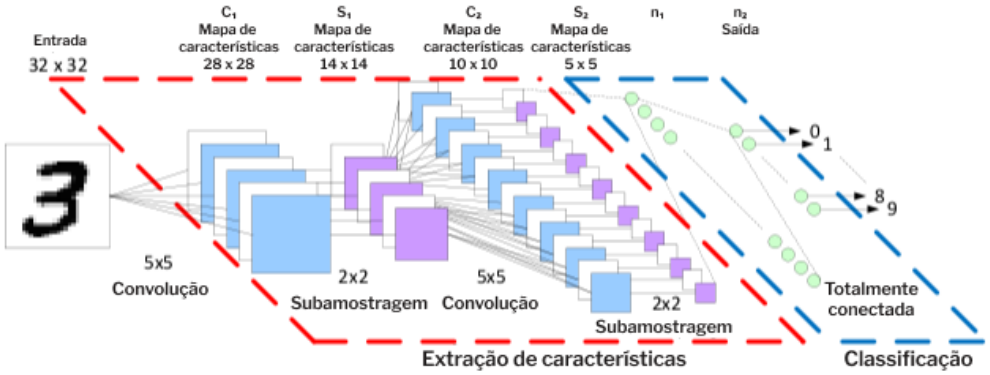
Sendo uma das CNNs pioneiras, a LeNet, proposta por LeCun et al. (1998), teve inicialmente o objetivo de alcançar a automação da detecção de dígitos manuscritos, como reconhecimento de dígitos em cheques para processamento bancário. Uma das principais inovações da LeNet foi o uso de camadas convolucionais, que permitiram à rede aprender automaticamente características a partir dos dados de entrada, reduzindo a necessidade de filtros para extração de características feitos a mão. Além disso, LeNet introduziu o conceito de compartilhamento de peso em camadas convolucionais, onde o mesmo conjunto de pesos é usado em diferentes localizações espaciais na imagem de entrada, levando à eficiência dos parâmetros.

A LeNet consiste basicamente da camada de entrada que recebe os valores brutos dos *pixels* da imagem de entrada, seguida de duas camadas convolucionais, cada uma acompanhada por uma camada de *pooling*. As camadas convolucionais aplicam filtros que podem ser aprendidos para extrair características espaciais das imagens de entrada. As camadas de *pooling* reduzem a dimensionalidade dos mapas de características, tornando a rede mais eficiente computacionalmente e reduzindo o sobreajuste. Após as camadas convolucionais e de *pooling*, a LeNet possui algumas camadas totalmente conectadas. Essas camadas pegam as características de alto nível extraídos pelas camadas convolucionais e as mapeiam para as classes de saída. As camadas totalmente conectadas empregam arquiteturas tradicionais de RNAs, conectando cada neurônio de uma camada a cada neurônio da camada seguinte. Por fim, a camada final da LeNet é normalmente seguida pela função de ativação Softmax, que produz probabilidades para cada classe do problema (LECUN et al., 1998). Na Figura 13 é possível observar a arquitetura do modelo aplicada a uma imagem de entrada de tamanho 32×32, similar a arquitetura utilizada pelos próprios autores.

A rede foi capaz de alcançar uma taxa de acerto acima dos 99% na base de dados

MNIST[6], na época, alcançando algo próximo ao resultado do estado de arte. Com o passar do tempo, estudos realizados trouxeram pequenas modificações no modelo inicialmente proposto, gerando o surgimento de variações conhecidas como LeNet-1, LeNet-4, LeNet-5 e Boosted LeNet-4 (LECUN et al., 1995).

Figura 13 – Arquitetura da rede LeNet-5.



Fonte: Adaptado de Deep Learning Tutorials (2015 apud LECUN et al., 1998).

Embora a LeNet tenha sido amplamente superada por arquiteturas mais modernas em termos de desempenho e complexidade, seus princípios e elementos de *design* continuam a influenciar o desenvolvimento das CNNs. Esta arquitetura de rede demonstrou a eficácia das abordagens de aprendizagem profunda, particularmente no campo da visão computacional, e abriu caminho para avanços subsequentes neste campo.

2.2.3.2 ResNet

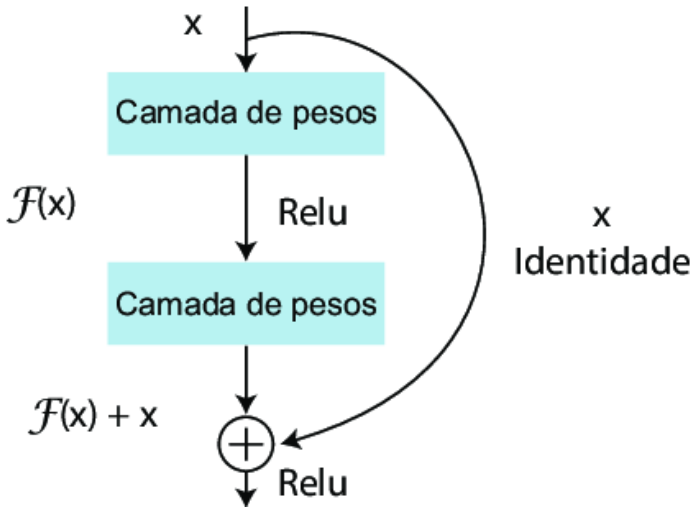
A ResNet, ou *residual network*, é uma arquitetura de rede neural convolucional que introduziu o conceito de aprendizagem residual. O modelo foi proposto por He et al. (2016a) e trouxe uma abordagem do problema do desaparecimento do gradiente em redes muito profundas, introduzindo saltos nas conexões, ou também conhecidos como atalhos. Esta arquitetura tornou possível o treinamento de centenas ou até milhares de camada, mantendo um bom desempenho. A rede ResNet foi responsável por vencer a ILSVRC 2015 e alcançar o *top-5 error* com uma taxa de erro de 3,57%, um feito que supera a taxa de erro humana, onde se espera valores entre 5 e 10%.

A proposta inicial da arquitetura, utilizada no treinamento do conjunto de dados

[6] Base de dados pública, amplamente utilizada, composta por imagens de números escritos a mão (LECUN et al., 1998).

ImageNet, é formada por blocos residuais e composta por até 152 camadas, onde, para cada bloco, a entrada é submetida a uma operação de convolução, seguida pela aplicação da função ReLU e, logo após, outra convolução, sendo o resultado adicionado aos dados de entrada, tornando-se assim a saída do bloco. Em vez de tentar aprender diretamente o mapeamento desejado, os blocos residuais aprendem os mapeamentos residuais, ou seja, a diferença entre a entrada e a saída do bloco. Esta técnica ficou conhecida como aprendizagem residual. Na Figura 14 é mostrada uma representação do bloco residual utilizados pela ResNet.

Figura 14 – Bloco residual utilizados na ResNet.



Fonte: Silva et al. (2020).

Ao introduzir a aprendizagem residual adicionando a entrada de um bloco diretamente à sua saída sem qualquer transformação, o ResNet facilita o treinamento de redes muito profundas sem encontrar degradação no desempenho. Redes mais profundas podem aprender recursos mais complexos e exibir melhor generalização, mas as arquiteturas tradicionais sofrem com o desaparecimento do gradiente, se tornando difíceis de otimizar. Com o aprendizado residual, o gradiente pode se propagar de forma mais eficaz pela rede, possibilitando o treinamento de arquiteturas mais profundas.

Capaz de alcançar o desempenho do estado da arte em vários *benchmarks* de classificação de imagens, incluindo a base de dados ImageNet, a ResNet demonstrou a eficácia do aprendizado residual no treinamento de redes muito profundas. Desde o seu surgimento, a ResNet foi amplamente utilizada e adaptada para várias tarefas de visão computacional, servindo como base para muitas arquiteturas de redes convolucionais que

foram surgindo ao longo do tempo e inspirando novas pesquisas na área da aprendizagem profunda.

2.2.3.3 Transferência de Conhecimento

Na área da aprendizagem profunda, por um tempo, existiu a crença de que os dados utilizados para treinamento dos modelos deveriam ser de classes relacionadas ao problema tratado, devendo também ter uma distribuição uniforme dos dados para cada classe. Sendo assim, se o problema tratado fosse a classificação de imagens de cachorros e gatos, a base de dados de treinamento deveria ser composta por imagens variadas de ambos os tipos de animais distribuídas uniformemente para cada classe, ou seja, a base deveria conter a mesma quantidade de imagem para cachorros e gatos.

No entanto, quando tratamos problemas do mundo real, nem sempre é possível ter dados distribuídos uniformemente, ou ainda, ter uma quantidade de dados suficientes para realizar o treinamento do modelo. Para casos como esse, sugiram as técnicas de transferência de conhecimento na aprendizagem de máquina, que consiste em utilizar um modelo pré-treinado em uma determinada tarefa, de um dado domínio A, como base de treinamento em uma tarefa similar, de um dado domínio B (PAN; YANG, 2010). A transferência de conhecimento pode melhorar significativamente o desempenho dos modelos na tarefa alvo, aproveitando o conhecimento aprendido na tarefa originaria (WEISS et al., 2016).

Na literatura é possível encontra diversas aplicações da transferência de conhecimento e em diversas áreas (ZHOU et al., 2014; HAREL; MANNOR, 2010; DUAN et al., 2012). Nos estudos voltados à área de redes neurais convolutivas focados no treinamento na presença de rótulos ruidosos, temos técnicas que se utilizam de CNNs pré-treinadas para obter um melhor desempenho em determinadas bases de dados, como a técnica vista em Li et al. (2020).

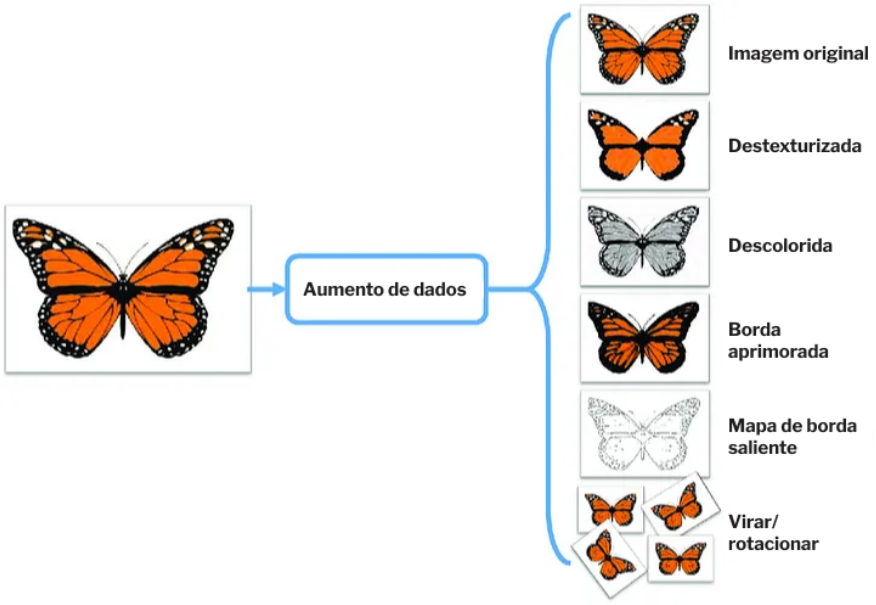
2.2.4 Aumento de Dados

Aumento de dados, do inglês *data augmentation*, é uma técnica comumente usada na aprendizagem de máquina e aprendizagem profunda para aumentar a diversidade dos dados

durante o treinamento dos modelos. A ideia básica é criar novas amostras de treinamento aplicando transformações nos dados existentes, preservando seu significado semântico. Esta técnica é muito utilizada, principalmente, no processo de treinamento de modelos onde existe uma certa deficiência ou escassez nas amostras de treinamento utilizadas. Isso ajuda a melhorar a generalização e a robustez dos modelos de aprendizagem de máquina, expondo-os a uma maior variedade nos dados de entrada (SHORTEN; KHOSHGOFTAAR, 2019).

Na Figura 15 é possível observar a aplicação de algumas técnicas de aumento de dados na imagem de uma borboleta. Os aumento de dados em imagens envolvem técnicas de transformações, tais como rotação, redimensionamento, recorte, translação, ajustes de brilho e ajustes de cores. Estas transformações simulam diferentes condições de visualização e variações na aparência dos objetos.

Figura 15 – Exemplo de aplicação de técnicas de aumento de dados em uma imagem de uma borboleta.



Fonte: Adaptador de Kumar (2019).

Além dos aumentos de dados em imagens, na aprendizagem de máquina também são utilizadas técnicas de aumento em outro tipos de dados, como textos para modelos de processamento de linguagem natural (SHORTEN et al., 2021), áudios para modelos que realizam tarefas de reconhecimento de fala (KO et al., 2015), entre outros.

Neste trabalho, foram utilizadas apenas técnicas de aumento de dados em imagens. Dentre as técnicas utilizadas, estão os seguintes transformações básicas de imagens: *random*

crop, horizontal flip, rotation, translation, shear, autocontrast, invert, equalize, posterize, contrast, brightness, sharpness, solarize. Na Tabela 1 é mostrado o tipo de cada um dos aumentos básicos e uma breve descrição de seus efeitos. Além disso, dentre as técnicas utilizadas, também estão os seguintes métodos do estado da arte: AutoAug (CUBUK et al., 2019), RandAug (CUBUK et al., 2020), Cutout (DEVRIES; TAYLOR, 2017), Mixup (ZHANG et al., 2017), CutMix (YUN et al., 2019), AugMix (HENDRYCKS et al., 2019). Na Tabela 2 é mostrada uma breve descrição dos aumentos de dados do estado da arte.

Por fim, nas Figuras 16 e 17 pode ser observada a diferença entre os aumentos de dados utilizados neste trabalho e seus efeitos em uma imagem original.

2.2.5 Rótulos Ruidosos

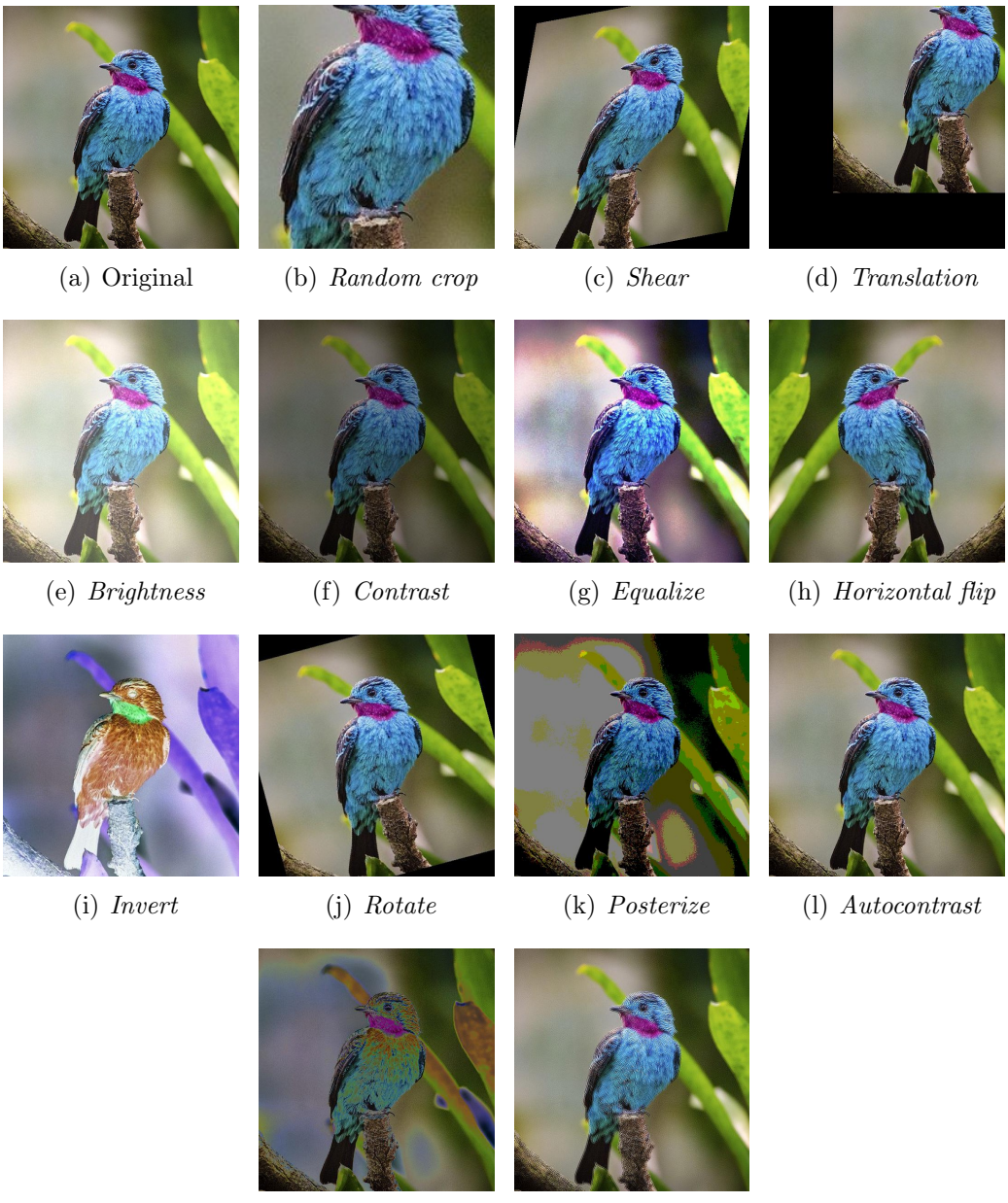
Considerando que os rótulos ruidosos podem se originar por motivos citados no Capítulo 1, a Figura 18 traz a ilustração dos tipos de processos de rotulagem de dados e alguns dos possíveis motivos da origem dos ruídos de rótulo. Na Figura 19 pode ser observado um exemplo de imagens semelhantes que pertencem a classes diferentes e que podem ocasionar ruídos de rótulo durante o processo de rotulagem. Ainda, os rótulos ruidosos podem ocorrer através de coletas de dados *online*, onde as informações estejam divergentes, como um tipo de roupa ou calçado inseridos na categoria errada de uma plataforma de vendas, onde imagens estejam sendo coletadas e rotuladas para serem utilizadas no treinamento de um modelo.

Os ruídos de rótulo podem ser divididos em dois tipos: simétricos e assimétricos (CORDEIRO; CARNEIRO, 2020). Os ruídos simétricos, também conhecidos como ruídos aleatórios ou uniformes, ocorrem quando um rótulo tem probabilidades iguais de mudar para qualquer outra classe. Já os ruídos assimétricos, se aproximam mais do mundo real, baseando-se na probabilidade de um rótulo mudar para uma classe semelhante, como visto na Figura 19. A Figura 20 traz uma representação em matriz de transição de ruídos simétricos e assimétricos para um contexto de 5 classes e probabilidade de mudança entre classes de 40%, também conhecida como taxa de ruído. Logo, para a matriz de transição de ruídos simétricos da Figura 20, uma determinada classe tem 10% de chance de mudar para cada uma das outras 4 classes. Já na matriz de transição de ruídos

Tabela 1 – Descrição dos aumentos básicos avaliados neste trabalho, separados em transformações a nível espacial e a nível de *pixel*.

Método	Nível da Transformação	Descrição
random crop	espacial	Corta uma parte aleatória da imagem de entrada.
horizontal flip	espacial	Vira a imagem de entrada horizontalmente em torno do eixo y.
rotation	espacial	Rotaciona a imagem de entrada em um angulo selecionado aleatoriamente a partir de uma distribuição uniforme.
translation	espacial	Desloca a imagem de entrada ao longo de um eixo.
shear	espacial	Distorce a imagem de entrada ao longo de um eixo.
invert	*pixel*	Inverte a imagem de entrada subtraindo os valores de *pixel* do valor máximo possível, sendo o valor máximo igual a 255 para imagens de 8-*bits*.
equalize	*pixel*	Equaliza o histograma da imagem de entrada.
posterize	*pixel*	Reduz o número de *bits* para cada canal de cor da imagem de entrada.
contrast	*pixel*	Aleatoriamente modifica o contraste da imagem de entrada.
autocontrast	*pixel*	Remapea os *pixels*, para cada canal de cor, maximizando o contraste da imagem de entrada.
brightness	*pixel*	Modifica aleatoriamente o brilho da imagem de entrada.
sharpness	*pixel*	Modifica a nitidez da imagem de entrada.
solarize	*pixel*	Inverte todos os valores de *pixel* acima de um limite.

Fonte: O Autor.



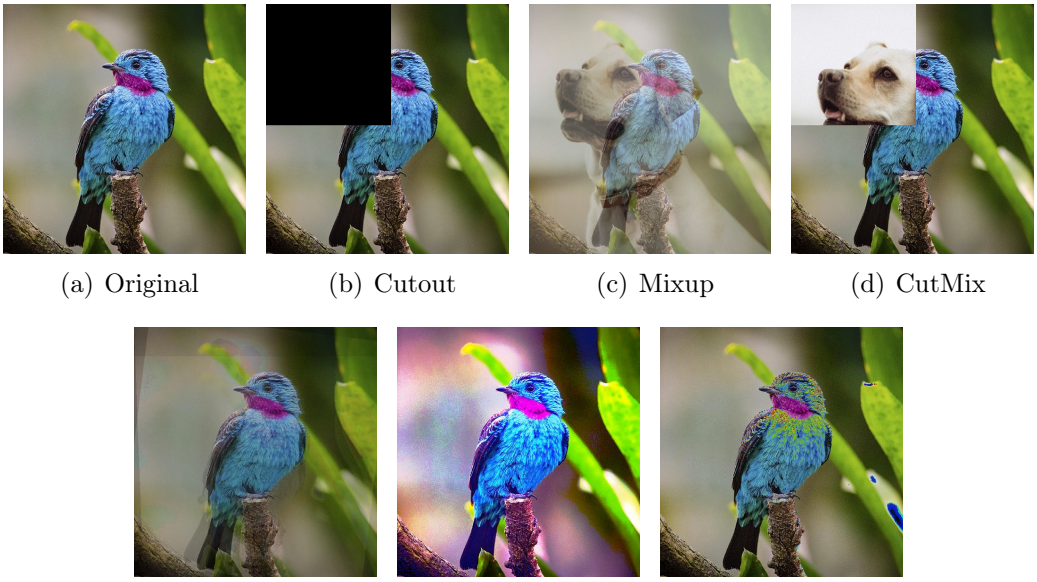
(a) Original (b) *Random crop* (c) *Shear* (d) *Translation*

(e) *Brightness* (f) *Contrast* (g) *Equalize* (h) *Horizontal flip*

(i) *Invert* (j) *Rotate* (k) *Posterize* (l) *Autocontrast*

(m) *Solarize* (n) *Sharpness*

Figura 16 – Aumento de dados básicos baseados em transformações clássicas de imagens (b)-(n).
Fonte: O Autor.

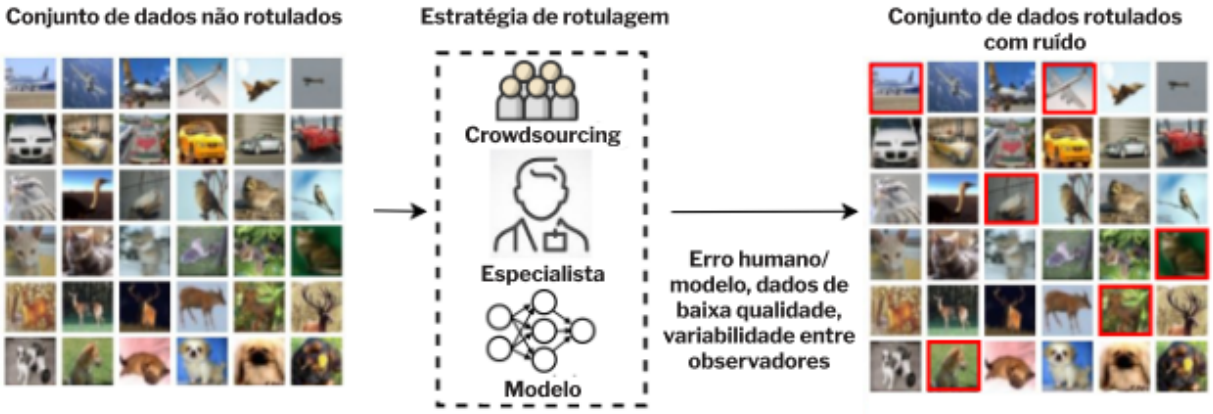


(a) Original (b) Cutout (c) Mixup (d) CutMix

(e) AugMix (f) AutoAug (g) RandAug

Figura 17 – Aumento de dados do estado da arte (b)-(g).
Fonte: O Autor.

Figura 18 – Exemplo de processos de rotulagem e origem dos ruídos.



Fonte: Adaptador de Cordeiro e Carneiro (2020).

Tabela 2 – Descrição dos aumentos do estado da arte avaliados neste trabalho.

Método	Descrição
Cutout	Remove aleatoriamente regiões quadráticas da imagem de entrada.
Mixup	Combinação linear entre imagens de entrada usando as seguintes equações: $\tilde{x} = \lambda x_i + (1-\lambda)x_j$ e $\tilde{y} = \lambda y_i + (1-\lambda)y_j$, onde x_i e x_j representam as imagens de entrada e y_i e y_j seus respectivos rótulos.
CutMix	Troca regiões removidas de uma imagem de entrada por partes de uma outra imagem.
AugMix	Para uma dada imagem de entrada, são aplicadas diferentes sequências de aumentos de dados e as imagens resultantes de cada sequência de aumentos são misturadas utilizando diferentes pesos.
AutoAug	Baseia-se em políticas de aumento de dados aplicadas a uma imagem de entrada, usando aumentos otimizados para cada conjunto de dados. Usa a aprendizagem por reforço para determinar o conjunto de métodos de aumento de dados e a ordem que devem ser aplicados.
RandAug	Baseia-se em políticas de aumento de dados aplicadas a uma imagem de entrada, usando aumentos otimizados para cada conjunto de dados. Diferente do AutoAug, usa uma busca em grade para determinar o conjunto ideal de métodos de aumento de dados a serem aplicados.

Fonte: O Autor.

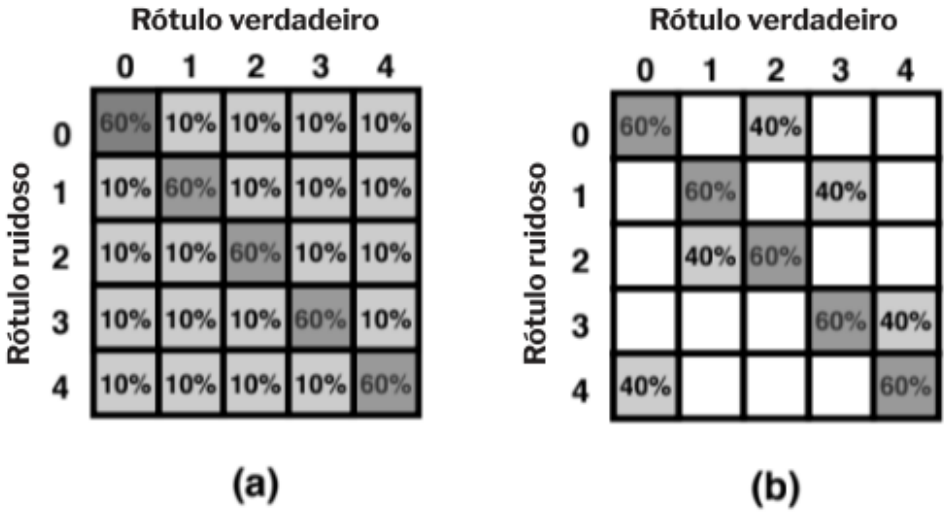
assimétricos, uma determinada classe tem 40% de chance de mudar apenas para uma outra classe semelhante. Também podem ocorrer cenários onde, para outros conjuntos de dados, uma classe tenha chance de mudar para mais de uma classe semelhante com probabilidades diferentes ou iguais.

Por fim, um importante fato sobre os rótulos ruidosos, é que eles podem degradar o desempenho dos modelos de aprendizagem de máquina ao introduzir informações incorretas durante o treinamento. Os modelos podem aprender padrões ou relacionamentos incorretos a partir dos rótulos ruidosos, levando a uma deficiência na generalização, o que ocasiona um baixo desempenho ao lidar com dados não vistos (SONG et al., 2022).

Figura 19 – Exemplo de imagens de classes semelhantes, onde (a) está relacionado a uma cobra coral verdadeira e (b) a uma falsa.

Fonte: Adaptado de Lima (2022).

Figura 20 – Matriz de transição de diferentes tipos de ruídos, onde (a) está relacionado ao ruído simétrico e (b) ao ruído assimétrico.



Fonte: Adaptador de Cordeiro e Carneiro (2020).

3 Trabalhos Relacionados

3.1 Estado da Arte

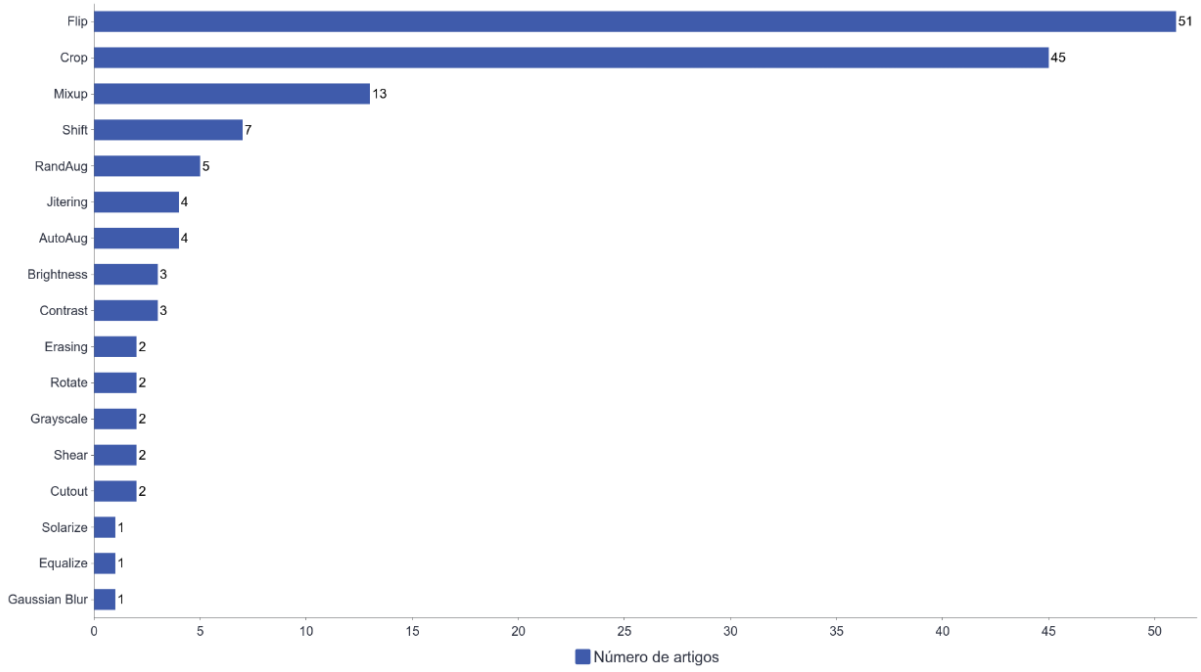
Na literatura, muitas técnicas vêm sendo propostas ao longo dos anos para lidar com a presença de rótulos ruidosos durante o treinamento das redes (ALGAN; ULUSOY, 2021). Dado os desafios de lidar com os ruídos de rótulo, presentes de forma inerente em bases de dados do mundo real, diferentes abordagens foram surgindo, que podem ser categorizadas, de acordo com o princípio aplicado pela técnica, em: técnicas de estimativa de matrizes de transição de ruídos (HENDRYCKS et al., 2018), funções de perda robustas (WANG et al., 2019), limpeza de rótulos (JAEHWAN et al., 2019), seleção de amostras (HAN et al., 2018b), ponderação de amostras (REN et al., 2018), meta-aprendizagem (HAN et al., 2018a), aprendizagem semissupervisionada (LI et al., 2020; WANG et al., 2022), entre outras (CORDEIRO; CARNEIRO, 2020).

Na Figura 21, é possível observar a quantidade de artigos relacionados ao treinamento de redes na presença de rótulos ruidosos que fazem o uso de aumento de dados. Foram avaliados 61 artigos relacionados a rótulos ruidosos e o aumentos de dados utilizados por eles. Além disso, foram listadas as principais abordagens encontradas na literatura para lidar com a presença de ruídos de rótulo, onde cada artigo pode ter utilizado mais de um método de aumento de dados. Para obter os dados, foram realizadas buscas utilizando as palavras-chave *noisy label* e *data augmentation* nas bases de dados do IEEE Xplore (IEEE, 1998) e Scopus (Elsevier, 2004) com relação aos últimos seis anos.

Dentre as abordagens mais recentes, o uso mais comum de aumento de dados é através das técnicas de processamento de imagens clássicas, como *random crop* e *horizontal flip*, mostradas na Figura 21. Algumas recentes abordagem têm utilizado métodos de aumento de dados do estado da arte (LI et al., 2020; WANG et al., 2022; ZHANG et al., 2020), como Mixup (ZHANG et al., 2017), CutMix (YUN et al., 2019) e RandAugment (RandAug) (CUBUK et al., 2020).

Chen et al. (2021) traz uma análise do treinamento de redes sem aumento de dados, com aumentos clássicos (*random crop* e *horizontal flip*) e aumento de dados forte usando o AutoAugment (AutoAug) (CUBUK et al., 2019). Os resultados mostram melhorias ao usar aumento de dados forte. No entanto, a análise feita se restringiu a uma estratégia de

Figura 21 – Número de artigos relacionados ao treinamento de modelos na presença de rótulos ruidosos que usa cada tipo de aumento de dados listado.



Fonte: O Autor.

aumento de dados do estado da arte e uma combinação de aumentos de dados clássicos.

Em Zhang et al. (2017), é proposto o Mixup, um aumento de dados que usa uma combinação linear entre amostras para melhorar a robustez do modelo. A abordagem é simplesmente uma combinação de imagens e rótulos usando a seguinte equação:

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda)x_j, \\ \tilde{y} &= \lambda y_i + (1-\lambda)y_j \end{aligned} \tag{3.1}$$

onde x_i e x_j representam as imagens de entrada e y_i e y_j seus respectivos rótulos. Esta abordagem mostrou-se eficaz na melhoria da robustez do modelo em abordagens de treinamento na presença de rótulos ruidosos como DivideMix (LI et al., 2020) e o PropMix (WANG et al., 2022).

Em DeVries e Taylor (2017), é proposto o Cutout, uma técnica de aumento de dados e regularização que, aleatoriamente, remove regiões quadráticas da imagem de entrada durante o treinamento. Já o CutMix, proposto em Yun et al. (2019), troca regiões removidas por partes de uma outra imagem, ao invés de remover os *pixels* e preenchê-los de preto, como no Cutout.

Chamado de AugMix, o método de aumento de dados proposto por Hendrycks et al. (2019) consiste em misturar os resultados de aumentos de dados aplicados em cadeia.

Para uma dada imagem de entrada, são aplicados diferentes sequências de aumentos de dados, tais como *rotation*, *shear*, *translation* e *posterize*, e a imagens resultantes de cada sequência de aumentos são misturadas utilizando diferentes pesos. No contexto de rótulos ruidos, este método não é explorado na literatura.

Por fim, temos as técnicas AutoAug (CUBUK et al., 2019) e RandAug (CUBUK et al., 2020) que se baseam em políticas de aumento de dados aplicadas durante o treinamento do modelo, usando aumentos fortes e otimizados para cada conjunto de dados. Utilizar um aumento forte está ligado ao fato das transformações de imagem serem aplicadas com maior intensidade, causando maiores mudanças nas imagens. Essas políticas são compostas por transformações de imagem básicas, como *rotation*, *shear*, *invert* e *contrast*. AutoAug usa a aprendizagem por reforço para determinar o conjunto de métodos de aumento de dados e a ordem que devem ser aplicados para trazer uma maior generalização ao modelo. Removendo a fase de busca do AutoAug e reduzindo a complexidade do treinamento, o RandAug usa uma busca em grade para determinar o conjunto ideal de métodos de aumento de dados a serem aplicados. O RandAug mostrou-se eficaz na melhoria da robustez do modelo para a técnica de treinamento PropMix, proposta por Wang et al. (2022).

3.2 Resumo do Estado da Arte e Contribuições do Trabalho Proposto

Apesar de existirem diferentes propostas para treinar modelos na presença de rótulos ruidosos, apenas alguns métodos de aumento de dados existentes foram explorados para resolver este problema. Na Figura 21 é possível observar uma lista dos métodos de aumento de dados mais usados em técnicas presentes na literatura para lidar com rótulos ruidosos durante o treinamento do modelo. No entanto, apenas Chen et al. (2021) traz uma análise de poucos métodos de aumento de dados.

Sendo assim, este trabalho propõe a análise de 13 métodos clássicos e 6 do estado da arte de aumento de dados, com o objetivo de identificar o impacto no treinamento de modelos ao usar diferentes estratégias de aumento de dados. Além disso, propõe-se avaliar os melhores métodos de aumento de dados ou a combinação deles para diferentes níveis de taxa de ruído.

4 Metodologia

Neste capítulo são apresentados os conjuntos de dados, técnicas e ferramentas utilizadas na realização deste trabalho.

4.1 Bases de Dados

Neste trabalho, as bases de dados utilizadas na realização dos experimentos foram MNIST (LECUN et al., 1998), CIFAR-10, CIFAR-100 (KRIZHEVSKY et al., 2009) e Clothing1M (XIAO et al., 2015).

O conjunto de dados MNIST consiste de imagens de dígitos escritos a mão, onde cada imagem tem dimensão de 28×28 píxeis, com 60000 amostras de treinamento e 10000 amostras de teste. Para esta base de dados, foi aplicado um ruído simétrico sintético, onde cada amostra tem seu rótulo j trocado aleatoriamente para outra classe c com probabilidade η_{jc}. Foram avaliados $\eta_{jc} \in \{20\%, 50\%, 90\%\}$ que representam os cenários de ruídos baixo, médio e alto. Além disso, também foi aplicado um ruído assimétrico sintético, onde cada amostra tem seu rótulo j trocado aleatoriamente para outra classe c com probabilidade presente na matriz de transição mostrada na Tabela 3. A matriz de transição da Tabela 3 segue um mapeamento baseado na similaridade entre alguns dígitos e foi inspirado no mapeamento de ruído assimétrico utilizado por Li et al. (2020) para a base de dado CIFAR-10.

Já os conjuntos de dados CIFAR-10 e CIFAR-100 têm 50000 amostras de treinamento e 10000 amostras de teste, cada amostra é uma imagem de 32×32 *pixels*, onde as imagens estão separadas em 10 e 100 classes para o CIFAR-10 e o CIFAR-100, respectivamente. Em relação as classes do CIFAR-10, temos *airplane*, *automobile*, *bird*, *cat*, *deer*, *dog*, *frog*, *horse*, *ship* e *truck*. Já em relação ao CIFAR-100, as 100 classes são agrupadas em 20 superclasses. Ambas as bases de dados têm a mesma quantidade de amostras por classe. Nestas bases de dados também foram aplicados ruídos sintéticos simétrico e assimétrico com probabilidade $\eta_{jc} \in \{20\%, 50\%, 80\%\}$ para o ruído simétrico e com matriz de transição representada na Tabela 4 para o ruído assimétrico aplicado à base CIFAR-10. A matriz de transição da Tabela 4 segue o mapeamento de ruído assimétrico utilizado por Li et al. (2020). Para o CIFAR-100, o ruído assimétrico foi o mesmo utilizado

Tabela 3 – Matriz de transição para ruído assimétrico sintético utilizada na base de dados MNIST.

Classe	0	1	2	3	4	5	6	7	8	9
0	100%									
1		100%								
2			60%					40%		
3				60%					40%	
4					100%					
5						60%	40%			
6							100%			
7		40%						60%		
8									100%	
9		40%								60%

Fonte: O Autor.

por Patrini et al. (2017), que agrupa as 100 classes em suas respectivas superclasses e, dentro de cada superclasse, o ruído transforma, com probabilidade $\eta_{jc} = 40\%$, cada classe na que vem a seguir.

Tabela 4 – Matriz de transição para ruído assimétrico sintético utilizada na base de dados CIFAR-10, onde *0* $\rightarrow$ *airplane*, *1* $\rightarrow$ *automobile*, *2* $\rightarrow$ *bird*, *3* $\rightarrow$ *cat*, *4* $\rightarrow$ *deer*, *5* $\rightarrow$ *dog*, *6* $\rightarrow$ *frog*, *7* $\rightarrow$ *horse*, *8* $\rightarrow$ *ship* e *9* $\rightarrow$ *truck*.

Classe	0	1	2	3	4	5	6	7	8	9
0	100%									
1		100%								
2	40%		60%							
3				60%		40%				
4					60%			40%		
5				40%		60%				
6							100%			
7								100%		
8									100%	
9		40%								60%

Fonte: O Autor.

Por fim, a base de dados Clothing1M tem 1000000 de imagens de roupas obtidas através de vários *sites* de compras na internet, consequentemente, contendo rótulos ruidosos que refletem o mundo real, o que corresponde a um ruído assimétrico semântico. O conjunto de dados também contém 14 classes e 74000 amostras sem a presença se rótulos ruidosos,

onde 50000 são para treinamento, 14000 para validação e 10000 para teste.

4.2 Implementação

Neste trabalho, foi utilizada a linguagem de programação Python (ROSSUM; JR, 1995) e a biblioteca PyTorch (PASZKE et al., 2019) como bases para a construção e execução dos algoritmos. Além disso, foram avaliadas técnicas de aumento de dados divididas em duas categorias: técnicas básicas e do estado da arte. Dentre os métodos básicos, foram avaliados: *random crop, horizontal flip, rotation, translation, shear, autocontrast, invert, equalize, posterize, contrast, brightness, sharpness, solarize*. Para o estado da arte, foram avaliados os seguintes métodos: AutoAug, RandAug, Cutout, Mixup, CutMix, AugMix. Ademais, também foram avaliadas combinações entre os métodos básicos e o estado da arte.

Na implementação dos aumentos de dados básicos, foram usados conjuntos de parâmetros baseados nas transformações de imagens utilizadas por Cubuk et al. (2020), considerando os parâmetros utilizados pelo RandAug para uma magnitude de entrada igual a 5, onde a magnitude máxima é 31, pois esta seria uma magnitude ótima encontrada, pelo criador da técnica, em experimentos feitos com a base de dados CIFAR. No entanto, o *random crop* e o *horizontal flip* não são métodos utilizados no trabalho anteriormente citado e tiveram seus parâmetros baseados em (LI et al., 2020). Na Tabela 5 são mostrados os parâmetros usados nos aumentos básicos. Para os aumentos de dados do estado da arte Cutout, Mixup, CutMix, AugMix, AutoAug e RandAug, foram utilizados os conjuntos de parâmetros padrão fornecidos por seus respectivos repositórios ou artigos. De forma independente, para a base de dados Clothing1M foram aplicados os aumentos de dados *random crop* e Mixup como feito por Li et al. (2020). Além disso, também foram aplicados à base de dados Clothing1M o aumento de dados CutMix, como um substituto do Mixup, e o AutoAug.

O modelo utilizado nos experimentos para as bases de dados MNIST, CIFAR-10 e CIFAR-100 foi o ResNet-18, seguindo He et al. (2016b). Com a base MNIST, o modelo foi treinado com gradiente descendente estocástico, ou *stochastic gradient descent* (SGD), com momento de 0,9, decaimento de peso de 0,0005, taxa de aprendizagem de 0,001 e *batches* de 64 amostras. Para CIFAR-10 e CIFAR-100, o modelo foi treinado com SGD

Tabela 5 – Parâmetros utilizados na avaliação dos aumentos de dados básicos, onde p, quando presente, é a probabilidade do método ser aplicado.

Método	Parâmetros
random crop	$size = 32, padding = 4$
horizontal flip	$p = 0{,}5$
rotation	$degrees = [-30, 30]$
translation	$fraction = 0{,}453172$
shear	$degrees = [-17{,}188733, 17{,}188733]$
invert	$p = 0{,}5$
equalize	$p = 0{,}5$
posterize	$bits = 7, p = 0{,}5$
contrast	$contrast = [0{,}1, 1{,}9]$
autocontrast	$p = 0{,}5$
brightness	$brightness = [0{,}1, 1{,}9]$
sharpness	$factor = 0{,}85, p = 0{,}5$
solarize	$threshold = 213{,}3333, p = 0{,}5$

Fonte: O Autor.

com momento de 0,9, decaimento de peso de 0,0005, taxa de aprendizagem de 0,02 e *batches* de 64 amostras. Foram usadas 100 e 200 épocas para treinar os modelos para as bases MNIST e CIFAR, respectivamente.

Além disso, foram avaliadas modificações nos aumentos de dados utilizados na estrategia de treinamento do estado da arte DivideMix (LI et al., 2020). No DivideMix, para as bases CIFAR-10 e CIFAR-100, foi usado, como modelo base, uma PreAct-ResNet-18 (PRN18) (HE et al., 2016c) de 18 camadas, como em Li et al. (2020). O modelo foi treinado com SGD com momento de 0,9, decaimento de peso de 0,0005 e *batches* de 64 amostras. Além de uma taxa de aprendizagem de 0,02 que foi reduzida para 0,002 no meio do treinamento (LI et al., 2020). O modelo foi treinado com 300 épocas.

Por fim, para a base de dados Clothing1M, foi usada uma ResNet-50 como modelo base, seguindo Li et al. (2020). Neste contexto, foi usada uma ResNet-50 com pesos pré-treinados na base de dados ImageNet (DENG et al., 2009). O modelo foi treinado

por 80 épocas com *batches* de 32 amostras, SGD com uma taxa de aprendizagem de 0,002 (dividida por 10 a cada 40 épocas), momento de 0,9 e decaimento de peso de 0,0001.

5 Resultados

Este capítulo traz os resultados, produzidos pelos experimentos realizados, referentes as avaliações dos aumentos de dados como descritas no Capítulo 4.

5.1 Resultados Experimentais

Como descrito anteriormente, foram avaliadas as base de dados MNIST, CIFAR-10, CIFAR-100 e Clothing1M, como também a estratégia de treinamento DivideMix.

5.1.1 MNIST

Para a base MNIST, foram executados experimentos utilizando o modelo ResNet-18 com ruído simétrico sintético com taxa de ruído $\eta_{jc} \in \{20\%, 50\%, 90\%\}$ e ruído assimétrico sintético com taxa de 40% para diferentes tipos de aumentos de dados e combinações entre eles. Foi usada uma ResNet-18, sem aumento de dados, como nosso modelo base. Não foram avaliados os aumentos de dados *equalize, posterize, solarize, contrast, autocontrast, brightness e solarize* para a base MNIST, por se tratar de uma base com imagens binárias. Por fim, os resultados são mostrados na Tabela 6, onde também pode ser observado o melhor resultado, na coluna identificada com M, e o último resultado, na coluna identificada com U, obtidos na avaliação das amostras de teste. Note que o *random crop* é referenciado como *rc*.

Observando a Tabela 6, nota-se que todos os aumentos melhoram a robustez para rótulos ruidosos em relação ao modelo base que não utiliza nenhum aumento de dados. Utilizado por várias estratégias (LI et al., 2020; CORDEIRO et al., 2021; LIU et al., 2020; WANG et al., 2022), o *random crop* se destaca como o principal aumento para lidar com rótulos ruidosos, o que pode ser comprovado observando-se os resultados obtidos. É possível observar que o *random crop* ajuda os métodos do estado da arte a alcançar uma alta performance, isso porque o *random crop* ajuda a lida com o sobreajuste, ou *overfitting*, em relação ao ruído de rótulo, ou seja, ajuda o modelo a aprender a classe da amostra baseado em diferentes características da imagem. Vale destacar que, embora os aumentos de dados individualmente melhorem os resultados, a combinação deles, como em *random*

Tabela 6 – Resultado da acurácia de diferentes aumentos de dados avaliados na base MNIST. As colunas identificadas com M representam o melhor valor obtido e as identificadas com U representam o último valor obtido. O melhor valor de cada coluna está destacado em negrito.

Tipo de Ruído	Simétrico						Assimétrico	
Método/Taxa de Ruído	20%		50%		90%		40%	
Acurácia	M	U	M	U	M	U	M	U
modelo base	98,31	96,70	97,16	79,21	72,96	22,04	89,46	83,04
random crop (rc)	99,45	99,08	99,09	98,41	95,19	85,44	99,23	90,85
random horizontal flip	97,62	95,51	96,06	68,76	72,80	20,68	91,39	80,45
random rotation	99,22	91,90	98,72	75,72	88,89	28,77	96,92	82,14
random translation-x	99,31	91,52	98,79	72,01	88,81	27,98	97,51	82,59
random shear-x	99,04	93,46	98,42	70,97	85,97	21,65	94,91	83,01
random invert	98,42	95,93	97,42	66,05	73,34	22,31	91,95	80,20
random sharpness	98,41	96,39	97,18	78,73	73,39	22,82	88,57	83,34
random crop + *translation-x* + *shear-x*	99,57	99,46	**99,50**	99,38	96,11	93,82	99,43	89,67
todas as transformações básicas	98,59	98,14	97,70	97,62	34,42	34,42	98,02	91,73
AutoAug (CUBUK et al., 2019)	99,42	95,79	98,91	84,40	86,33	30,85	97,76	82,13
RandAug (CUBUK et al., 2020)	99,48	95,13	99,06	80,11	88,88	27,50	98,15	82,84
Cutout (DEVRIES; TAYLOR, 2017)	99,09	80,78	98,13	65,76	81,76	16,61	95,31	79,21
Mixup (ZHANG et al., 2017)	98,82	96,45	97,50	79,11	73,34	21,91	93,19	81,17
CutMix (YUN et al., 2019)	98,36	96,05	97,20	77,11	74,90	23,16	94,52	76,59
AugMix (HENDRYCKS et al., 2019)	98,65	93,32	97,87	67,30	67,60	20,37	93,48	82,29
AutoAug + *random crop*	99,64	99,47	99,48	**99,48**	96,46	94,58	99,39	90,87
RandAug + *random crop*	**99,66**	99,52	99,38	99,25	**97,10**	**95,64**	**99,58**	89,97
Cutout + *random crop*	99,37	99,25	99,17	98,97	95,04	93,03	99,13	87,27
Mixup + *random crop*	99,47	99,31	99,17	99,05	95,40	90,68	99,27	**98,71**
CutMix + *random crop*	99,38	99,32	99,10	98,94	95,72	94,40	98,56	96,32
AugMix + *random crop*	99,55	99,28	99,20	98,74	95,80	90,92	99,29	91,07
AutoAug + rc + *translation-x* + *shear-x*	99,52	99,38	99,35	99,17	92,61	91,19	99,39	89,54
RandAug + rc + *translation-x* + *shear-x*	99,58	99,50	99,40	99,23	95,06	94,74	99,42	90,78
Cutout + rc + *translation-x* + *shear-x*	99,40	99,27	99,05	98,99	93,81	90,61	99,27	89,15
Mixup + rc + *translation-x* + *shear-x*	99,52	99,31	99,29	99,19	93,47	91,87	99,34	94,05
CutMix + rc + *translation-x* + *shear-x*	99,43	99,23	99,20	99,16	95,08	94,34	98,90	97,41
AugMix + rc + *translation-x* + *shear-x*	99,62	**99,60**	99,46	99,36	96,49	94,15	99,48	89,36

Fonte: O Autor.

crop + translation-x + shear-x, é mais eficiente. Entretanto, combinar todos os aumentos básicos, como em *todas as transformações básicas*, é menos eficiente. Os métodos do estado da arte têm melhores resultados quando combinados apenas com o *random crop*. Apesar que, combinações onde foram adicionados mais aumentos aos métodos do estado da arte, tal como *AutoAug + rc + transition + shear*, mostram resultados similares. Também observa-se que RandAug, combinado com *random crop*, mostra os melhores resultados. Em comparação com o modelo base, a escolha de um aumento de dados adequado melhorou os resultados em até 33% em altas taxas de ruído.

5.1.2 CIFAR

Além da base MNIST, também foram avaliados aumentos de dados para as bases CIFAR-10 e CIFAR-100 com ruído simétrico $\eta_{jc} \in \{20\%, 50\%, 80\%\}$ e ruído assimétrico de 40%. Para estas bases de dados, foram inclusos todos os 13 aumentos de dados básico, os métodos do estado da arte e suas combinações. Também foi avaliada a estratégia de treinamento do estado da arte DivideMix (LI et al., 2020), onde foram usados diferentes aumentos de dados. Por padrão, o DivideMix usa os aumentos *Mixup*, *random crop* e *random horizontal flip*. Neste caso, foram avaliadas a troca do Mixup pelo CutMix e a adição do AutoAug. Para as bases CIFAR-10 e CIFAR-100, os resultados são mostrados nas Tabelas 7 e 8, respectivamente.

Nos resultados das Tabelas 7 e 8, observa-se uma melhoria expressiva, em diferentes taxas de ruídos, quando adicionado aumento de dados básicos ao modelo base. No entando, usando todas as transformações básicas juntas, os resultados mostraram uma eficiência menor, como visto na base de dados MNIST. Também como visto na base MNIST, a combinação *rc + translation-x + shear-x* mostrou-se mais eficiente que o uso dos aumentos individualmente. Além disso, também é observado que o *random crop* (rc) é um aumento essencial para melhorar os resultados dos aumentos do estado da arte. A partir dos resultados observados, é possível notar uma melhoria da acurácia, em relação ao modelo base, de até 61,39% sobre o ruído assimétrico e 177,62% sobre o ruído simétrico através do uso dos aumentos de dados no processo de treinamento. Neste caso, observa-se que as combinações de *CutMix + rc + AutoAug* e *Mixup + rc + AutoAug* têm os melhores resultados quando adicionados ao modelo base. Em relação a estratégia de trainamento

Tabela 7 – Resultado da acurácia de diferentes aumentos de dados avaliados na base CIFAR-10. As colunas identificadas com M representam o melhor valor obtido e as identificadas com U representam o último valor obtido. O melhor valor de cada coluna está destacado em negrito.

Tipo de Ruído	Simétrico						Assimétrico	
Método/Taxa de Ruído	20%		50%		80%		40%	
Acurácia	M	U	M	U	M	U	M	U
modelo base	72,60	62,31	62,96	39,67	41,50	18,25	70,30	64,64
random crop (rc)	82,00	75,73	76,04	54,94	58,66	27,28	81,48	76,41
random horizontal flip	78,02	68,34	68,62	44,29	48,46	18,90	76,32	65,05
random rotation	76,32	67,82	67,76	44,60	48,93	20,24	73,99	67,83
random translation-x	78,71	70,90	69,03	45,17	50,43	20,25	77,21	69,22
random shear-x	75,50	66,87	63,42	41,89	43,56	20,79	73,03	65,94
random autocontrast	73,93	62,82	62,78	39,92	44,78	18,26	70,31	64,69
random invert	73,42	63,34	63,85	39,62	44,70	17,62	71,31	62,55
random equalize	72,94	62,66	62,89	37,30	44,07	18,78	71,37	63,36
random posterize	72,54	63,46	62,94	39,42	44,32	18,17	69,58	65,18
random contrast	71,51	62,52	63,97	36,67	46,11	18,69	72,33	63,29
random brightness	73,74	63,89	62,36	39,47	44,78	18,67	72,24	63,41
random sharpness	72,93	63,67	64,56	40,77	44,92	18,63	69,98	61,54
random solarize	73,48	63,18	62,80	38,15	44,87	18,54	71,76	63,86
rc + *translation-x* + *shear-x*	82,66	81,74	75,38	70,46	55,14	46,62	80,69	79,40
todas as transformações básicas	64,71	61,67	51,08	47,42	10,52	10,00	61,46	55,49
AutoAug	77,52	67,70	67,87	42,31	46,71	19,62	75,46	65,52
RandAug	78,89	71,86	71,86	46,03	52,22	22,19	77,18	67,00
Cutout	75,86	67,13	67,34	43,76	49,56	21,38	74,64	66,39
CutMix	79,67	74,59	70,58	51,75	50,24	22,05	76,51	68,46
AugMix	75,38	66,68	65,53	43,61	46,10	20,30	72,30	66,21
Mixup	74,40	66,11	65,10	42,66	44,26	20,47	70,33	64,89
AutoAug + *random crop*	85,05	83,56	78,84	75,88	64,10	56,76	84,16	79,57
RandAug + *random crop*	84,75	82,41	77,92	72,86	62,31	49,05	82,78	79,57
Cutout + *random crop*	84,61	83,29	78,93	75,40	60,42	45,34	83,77	79,76
Mixup + *random crop*	84,53	81,88	77,15	73,93	61,64	49,08	82,53	77,79
CutMix + *random crop*	85,19	82,94	78,29	71,03	63,37	50,05	83,33	77,53
AugMix + *random crop*	83,31	80,58	77,45	63,98	59,47	34,44	81,61	75,65
AutoAug + rc + *translation-x* + *shear-x*	81,15	79,74	73,40	72,59	49,28	42,70	79,53	75,25
RandAug + rc + *translation-x* + *shear-x*	82,24	74,66	74,78	73,12	50,42	48,29	80,34	77,22
Cutout + rc + *translation-x* + *shear-x*	80,10	77,13	72,94	69,46	50,64	42,53	80,21	76,00
Mixup + rc + *translation-x* + *shear-x*	80,18	76,15	72,12	66,08	49,45	44,05	78,60	72,95
CutMix + rc + *translation-x* + *shear-x*	77,85	76,76	71,59	61,34	45,79	36,60	75,74	74,62
AugMix + rc + *translation-x* + *shear-x*	82,13	80,24	75,58	71,89	52,48	50,22	80,13	74,84
CutMix + rc + AutoAug	85,46	82,90	80,03	75,95	64,88	58,52	83,18	79,73
Mixup + rc + AutoAug	85,70	83,74	80,35	75,34	65,08	60,77	82,91	79,79
CutMix + Mixup + rc	84,29	82,15	77,59	74,10	62,27	53,20	82,04	72,53
DivideMix (com Mixup) (LI et al., 2020)	96,27	**96,97**	94,82	94,54	93,08	92,92	93,54	91,97
DivideMix (com Mixup) + AutoAug	96,28	96,14	95,32	95,11	94,18	93,96	94,41	**94,05**
DivideMix (com CutMix)	96,42	95,96	95,25	94,82	92,00	91,46	93,22	86,97
DivideMix (com CutMix) + AutoAug	**96,91**	96,46	**95,92**	**95,52**	**94,39**	**94,02**	**94,74**	92,34

Fonte: O Autor.

Tabela 8 – Resultado da acurácia de diferentes aumentos de dados avaliados na base CIFAR-100. As colunas identificadas com M representam o melhor valor obtido e as identificadas com U representam o último valor obtido. O melhor valor de cada coluna está destacado em negrito.

Tipo de Ruído	Simétrico						Assimétrico	
Método/Taxa de Ruído	20%		50%		80%		40%	
Acurácia	M	U	M	U	M	U	M	U
modelo base	40,27	31,42	28,94	16,31	8,27	4,92	30,67	23,60
random crop (rc)	52,01	41,22	39,48	21,86	15,74	7,88	39,14	31,42
random horizontal flip	45,74	34,71	34,71	16,96	17,94	4,73	35,17	28,72
random rotation	44,79	35,21	33,75	18,01	17,12	5,86	34,54	26,59
random translation-x	45,08	35,59	34,26	19,25	18,43	5,39	33,84	30,91
random shear-x	41,96	31,88	32,95	16,06	15,58	4,97	31,71	24,93
random autocontrast	41,03	31,64	30,51	14,22	13,20	4,29	31,39	22,54
random invert	39,87	27,63	30,80	14,15	13,78	3,93	30,23	22,45
random equalize	41,74	29,77	31,16	14,53	12,91	3,85	32,25	23,47
random posterize	40,52	30,25	30,16	14,77	13,54	4,03	30,36	23,47
random contrast	39,62	29,26	29,21	13,76	14,54	3,56	30,73	23,38
random brightness	39,35	29,56	30,08	13,89	15,71	3,87	29,48	22,41
random sharpness	39,95	30,12	30,33	15,27	13,13	3,95	30,59	23,46
random solarize	41,05	29,33	29,91	14,22	14,55	3,98	30,45	22,97
rc + *translation-x* + *shear-x*	51,48	46,69	42,36	40,44	16,95	16,56	39,70	35,20
todas as transformações básicas	31,64	28,67	18,53	16,65	2,47	1,67	27,14	21,52
AutoAug	44,89	34,89	34,28	16,35	16,66	4,53	34,60	28,12
RandAug	47,49	37,46	37,67	20,22	19,54	6,42	35,87	29,29
Cutout	41,87	32,83	32,29	15,88	16,38	4,63	31,44	25,83
CutMix	50,18	42,23	38,22	23,58	18,62	6,34	39,75	31,57
AugMix	41,38	34,02	31,12	17,13	15,07	4,85	31,75	24,34
Mixup	43,54	33,39	32,85	17,06	15,18	5,13	33,90	23,82
AutoAug + *random crop*	56,13	51,73	45,26	36,97	20,39	17,09	43,40	42,26
RandAug + *random crop*	55,15	45,95	43,22	25,16	19,19	11,13	41,83	33,07
Cutout + *random crop*	51,91	41,82	40,11	23,33	16,27	9,02	39,07	30,85
Mixup + *random crop*	55,64	49,30	45,10	33,45	18,56	12,23	43,46	37,61
CutMix + *random crop*	56,61	52,16	46,04	32,85	19,29	12,63	48,88	43,88
AugMix + *random crop*	52,30	43,05	41,69	24,30	17,26	9,82	40,35	31,52
AutoAug + rc + *translation-x* + *shear-x*	51,34	51,34	43,40	42,16	16,31	16,31	40,84	36,05
RandAug + rc + *translation-x* + *shear-x*	52,38	48,53	42,95	40,99	16,59	16,55	42,73	40,43
Cutout + rc + *translation-x* + *shear-x*	53,43	49,01	42,68	41,86	23,12	21,23	40,46	33,91
Mixup + rc + *translation-x* + *shear-x*	48,72	45,61	41,20	37,67	17,87	15,93	41,40	39,91
CutMix + rc + *translation-x* + *shear-x*	47,34	44,82	37,73	34,15	13,38	13,18	39,80	34,06
AugMix + rc + *translation-x* + *shear-x*	51,81	47,26	42,94	38,36	16,22	15,71	41,51	35,34
CutMix + rc + AutoAug	58,92	65,54	49,69	45,32	22,96	20,43	48,20	44,03
Mixup + rc + AutoAug	59,32	54,51	48,12	48,12	20,24	17,26	49,42	45,36
CutMix + Mixup + rc	56,07	52,37	42,78	35,16	17,65	16,14	48,14	46,88
DivideMix (com Mixup)	77,19	76,74	74,53	74,30	54,33	54,33	60,64	53,22
DivideMix (com Mixup) + AutoAug	78,04	78,04	76,53	76,06	58,98	58,52	65,44	**60,99**
DivideMix (com CutMix)	79,76	78,84	75,64	74,11	54,42	54,15	61,74	53,79
DivideMix (com CutMix) + AutoAug	**80,70**	**80,35**	**78,68**	**77,99**	**60,25**	**60,03**	**66,67**	56,93

Fonte: O Autor.

DivideMix, onde o resultado da estratégia padrão é identificado por *DivideMix (com Mixup)* nas Tabelas 7 e 8, a troca do Mixup pelo CutMix e a adição do AutoAug traz melhorias quando comparada a todos os aumentos e combinações analisadas. Além do mais, é possível observar um ganho absoluto de até 6%, no CIFAR-100, na técnica DivideMix apenas mudando a estratégia de aumento de dados.

5.1.3 Clothing1M

Além dos experimentos em bases de dados com ruídos sintéticos, também foram realizados experimentos com a base de dados Clothing1M, trazendo assim uma análise do impacto do uso de aumento de dados em uma base do mundo real. Para esta base de dados, foram avaliados o modelo base, sem aumento de dados, e o uso da melhor estratégia observada nos experimentos realizados nas base de dados anteriores, ou seja, *CutMix + AutoAug*. Além disso, também foi analisado o DivideMix padrão com o uso de diferentes aumentos de dados, onde os resultados podem ser observados na Tabela 9. Ainda observando os resultados da Tabela 9, é possível observar um ganho de 1,55% em relação ao modelo base. Para a base Clothing1M, a combinação dos métodos CutMix e AutoAug mostrou resultados inferiores ao DivideMix padrão. No entanto, a adição de apenas o AutoAug ao DivideMix padrão trouxe melhores resultados.

Tabela 9 – Resultados para a base de dados Clothing1M.

Método	Acurácia (%)
modelo base	70,30
rc + CutMix + AutoAug	71,85
DivideMix (com Mixup)	74,32
DivideMix (com Mixup) + AutoAug	**75,12**
DivideMix (com CutMix)	72,63
DivideMix (com CutMix) + AutoAug	69,95

Fonte: O Autor.

5.2 Discussão dos Resultados

Observando os resultados na seção anterior, é possível notar melhorias significativas ao adicionar aumentos de dados e suas combinações aos respectivos modelos base. Para o MNIST, foi possível observar uma melhoria em altas taxas de ruído, relativa ao modelo base, de até 31,72%, apenas adicionado aumentos de dados básicos e suas combinações ao modelo base, onde não foram utilizados aumentos por padrão, como é o caso da combinação *random crop + translation-x + shear-x*. Além disso, é possível observar que, combinando os métodos do estado da arte com o *random crop*, foi possível obter uma melhoria em relação aos métodos do estado da arte individualmente e alcançar os melhores resultados na base de dados MNIST.

Para as bases de dados CIFAR-10 e CIFAR-100, foram observadas uma melhoria de acurácia, em altas taxas de ruído, de até 41,34% e 90,32%, respectivamente, apenas adicionando o *random crop* ao modelo base, onde não foram aplicados aumentos de dados por padrão. Ao analisar os resultados para as combinações de aumentos de dados básicos e do estado da arte, foi possível alcançar uma melhoria ainda maior em altas taxas de ruído, com resultados até 56,81% melhores para a base CIFAR-10 e 177,62% melhores para a base CIFAR-100, como é o caso, por exemplo, da combinação *CutMix + rc + AutoAug*.

Para a estratégia de treinamento DivideMix, pode ser observada, em relação a estratégia padrão identificada por *DivideMix (com Mixup)*, uma melhoria de até 1,40% e 10,89% para as bases CIFAR-10 e CIFAR-100, respectivamente, em um alto nível de ruído. Já para a base de dados Clothing1M, que traz consigo dados do mundo real que também inclui um ruído natural, é possível observar uma melhoria de 2,20%, em relação ao modelo base, quando adicionado apenas uma combinação de aumento de dados. Já utilizando o DivideMix, foi possível alcançar uma melhoria de 1,07% em relação à estratégia padrão.

Com esta análise, mostrou-se que a escolha do aumento de dados, como parte do projeto de um modelo ou estratégia, pode trazer ganhos significativos nos resultados quando comparados a um treinamento padrão na presença de rótulos ruidosos. Já em relação ao uso de estratégias de treinamento do estado da arte, também pode haver ganho na performance do modelo quando o aumento de dados apropriado for escolhido. Embora a melhor estratégia de aumento de dados pareça depender do conjunto de dados, ainda assim, podem ser identificadas as melhores e mais importantes estratégias de aumento de dados para serem utilizadas no processo de treinamento, reduzindo o espaço de busca.

Por fim, as questões trazidas por este trabalho foram respondidas da seguinte maneira:

1. **Qual o impacto da utilização de diferentes técnicas de aumentos de dados no treinamento de modelos em conjuntos de dados com anotações ruidosas?** Baseado nos resultados, é possível observar que a utilização de diferentes técnicas de aumento de dados no treinamento com a presença de rótulos ruidosos pode trazer uma melhoria de até 177,62%, quando comparadas ao modelo base, onde não foi utilizada uma estratégia de aumento de dados. Também é possível observar que quase todas as técnicas avaliadas melhoraram a robustez do modelo.
2. **Qual é o ganho obtido ao combinar diferentes estratégias de aumento de dados para lidar com base de dados contendo rótulos ruidosos?** Os resultados mostram que as combinações de estratégias de aumento de dados, como por exemplo *CutMix + rc + AutoAug* em alto nível ruído para a base de dados CIFAR-100, obtiveram os melhores resultados, quando comparadas ao modelo base, com uma melhoria relativa de até 61,39% sobre o ruído assimétrico e 177,62% sobre o ruído simétrico.
3. **Qual é o ganho obtido ao modificar as estratégias de aumento de dados utilizadas em técnicas que lidam com anotações ruidosas?** Foi possível observar um ganho absoluto de até 6% na técnica DivideMix ao modificar a estratégia de aumento de dados proposta originalmente pela técnica.
4. **Qual a importância da escolha das estratégias de aumento de dados em contexto que envolvam bases de dados com anotações ruidosas?** Os resultados mostram que a escolha da melhor estratégia de aumento de dados pode trazer ganhos significativos na aprendizagem do modelo na presença de rótulos ruidosos.

6 Conclusão

6.1 Considerações Finais e Contribuições do Trabalho

Podendo surgir de diversas fontes, como anotadores humanos, imagens de baixa qualidade ou alguns métodos de coleta de dados, lidar com rótulos ruidosos tem sido um grande desafio. Neste sentido, várias técnicas vêm sendo propostas para melhorar o treinamento dos modelos (HAN et al., 2018a; REN et al., 2018; WANG et al., 2019; JAEHWAN et al., 2019; LIU et al., 2020; LI et al., 2020; CORDEIRO et al., 2021). No entanto, muitas dessas técnicas trazem consigo uma grande complexidade e um alto custo computacional (CORDEIRO; CARNEIRO, 2020) e, apesar da grande maioria utilizarem aumentos de dados no processo de treinamento, nenhuma das estratégias trouxe uma análise aprofundada do impacto dos aumentos de dados em sua estratégia de treinamento ou, até mesmo, do impacto do uso de aumento de dados na presença de anotações ruidosas. Em razão disso, neste trabalho foi avaliado o uso de diferentes tipos de aumento de dados no treinamento de CNNs com a presença de rótulos ruidosos.

Ainda neste trabalho, foi trazida uma avaliação do uso de aumento de dados com ruídos simétricos, assimétricos e semânticos para as bases de dados MNIST, CIFAR-10, CIFAR-100 e Clothing1M. Os experimentos conduzidos mostraram que a escolha do aumento de dados impacta drasticamente o desempenho do treinamento. Na análise trazida, é possível concluir que a combinação de abordagens clássicas e aumentos de dados do estado da arte é a melhor opção, além de que a configuração ideal de aumentos de dados é uma escolha importante no projeto de uma estratégia de treinamento. No entanto, a escolha do melhor aumento de dados depende da base de dados utilizada e precisa ser avaliada separadamente.

Por fim, foi mostrado que o uso de aumento de dados é essencial para lidar com anotações ruidosas e a escolha do melhor aumento depende do conjunto de dados utilizado. Para as abordagens avaliadas, podemos concluir que a combinação de abordagens básicas e do estado da arte são as melhores opções e a configuração ideal de aumente de dados é uma escolha importante no projeto de um modelo ou estratégia de treinamento.

Os resultados iniciais deste estudo foram publicados e também se encontram disponíveis nos Anais da *35th Conference on Graphics, Patterns and Images* (SIBGRAPI

2022) com o título “A Study on the Impact of Data Augmentation for Training Convolutional Neural Networks in the Presence of Noisy Labels” (PEREIRA et al., 2022).

6.2 Trabalhos Futuros

Durante a realização deste estudo, várias possibilidades de novas contribuições foram observadas, sendo uma delas a exploração dos benefícios do uso de aumentos de dados fracos e fortes em diferentes estágios do treinamento e o desenvolvimento de algumas estratégias para definir, de forma automática, o melhor método de aumento de dados para um determinado contexto, ou seja, para um modelo e base de dados.

Considerando o custo computacional para executar os experimentos e o tempo, uma outra possibilidade seria tornar o estudo mais abrangente, trazendo outros aumentos de dados e estratégias de treinamento para terem seus resultados avaliados.

Referências

ABDEL-HAMID, O.; MOHAMED, A.-r.; JIANG, H.; DENG, L.; PENN, G.; YU, D. Convolutional neural networks for speech recognition. **IEEE/ACM Transactions on audio, speech, and language processing**, IEEE, v. 22, n. 10, p. 1533–1545, 2014.

AHMED, S. F.; ALAM, M. S. B.; HASSAN, M.; ROZBU, M. R.; ISHTIAK, T.; RAFA, N.; MOFIJUR, M.; ALI, A. S.; GANDOMI, A. H. Deep learning modelling techniques: current progress, applications, advantages, and challenges. **Artificial Intelligence Review**, Springer, p. 1–97, 2023.

ALGAN, G.; ULUSOY, I. Image classification with deep learning in the presence of noisy labels: A survey. **Knowledge-Based Systems**, Elsevier, v. 215, p. 106771, 2021.

ALPAYDIN, E. **Introduction to machine learning**. [S.l.]: MIT press, 2014.

ANWAR, S. M.; MAJID, M.; QAYYUM, A.; AWAIS, M.; ALNOWAMI, M.; KHAN, M. K. Medical image analysis using convolutional neural networks: a review. **Journal of medical systems**, Springer, v. 42, p. 1–13, 2018.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 35, n. 8, p. 1798–1828, 2013.

BENGIO, Y.; GOODFELLOW, I. J.; COURVILLE, A. Deep learning. **Nature**, v. 521, p. 436–444, 2015.

BRIDLE, J. S. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In: **Neurocomputing**. [S.l.]: Springer, 1990. p. 227–236.

_____. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In: **Advances in neural information processing systems**. [S.l.: s.n.], 1990. p. 211–217.

CHEN, P.; YE, J.; CHEN, G.; ZHAO, J.; HENG, P.-A. Robustness of accuracy metric and its inspirations in learning with noisy labels. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2021. v. 35, n. 13, p. 11451–11461.

CIDRIM, L.; LOPES, W.; MADEIRO, F. **TECNOLOGIAS e CIÊNCIAS DA LINGUAGEM: vertentes e novas aplicações**. [S.l.]: Pá de Palavra, 2019.

CORDEIRO, F. R.; BELAGIANNIS, V.; REID, I.; CARNEIRO, G. Propmix: Hard sample filtering and proportional mixup for learning with noisy labels. **arXiv preprint arXiv:2110.11809**, 2021.

CORDEIRO, F. R.; CARNEIRO, G. A survey on deep learning with noisy labels: How to train your model when you cannot trust on the annotations? In: IEEE. **2020 33rd SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)**. [S.l.], 2020. p. 9–16.

CUBUK, E. D.; ZOPH, B.; MANE, D.; VASUDEVAN, V.; LE, Q. V. Autoaugment: Learning augmentation strategies from data. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 113–123.

CUBUK, E. D.; ZOPH, B.; SHLENS, J.; LE, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops**. [S.l.: s.n.], 2020. p. 702–703.

DAHL, G. E.; SAINATH, T. N.; HINTON, G. E. Improving deep neural networks for lvcsr using rectified linear units and dropout. In: IEEE. **Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on**. [S.l.], 2013. p. 8609–8613.

Deep Learning Tutorials. **Tutorials**. 2015. Último acesso 11 de Maio de 2024. Disponível em: <"http://swinghu.github.io/deep_learning/2015/10/09/dl-tutorials.html">.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255.

DEVRIES, T.; TAYLOR, G. W. Improved regularization of convolutional neural networks with cutout. **arXiv preprint arXiv:1708.04552**, 2017.

DONG, S.; WANG, P.; ABBAS, K. A survey on deep learning and its applications. **Computer Science Review**, Elsevier, v. 40, p. 100379, 2021.

DUAN, L.; XU, D.; TSANG, I. Learning with augmented features for heterogeneous domain adaptation. **arXiv preprint arXiv:1206.4660**, 2012.

Elsevier. **Scopus**. 2004. Último acesso 18 de Maio de 2024. Disponível em: <"https://www.scopus.com">.

ESTEVAM, R. **Redes Neurais | Redes Neurais Convolucionais**. 2019. Último acesso 11 de Maio de 2024. Disponível em: <"https://medium.com/turing-talks/turing-talks-23-modelos-de-predi%C3%A7%C3%A3o-redes-neurais-convolucionais-d364654a34de">.

FRANçA, R. N. **File:Representação de uma imagem digital - Matriz de pixel**. 2016. Último acesso 11 de Maio de 2024. Disponível em: <"https://commons.wikimedia.org/wiki/File:Representa%C3%A7%C3%A3o_de_uma_imagem_digital_-_Matriz_de_pixel.jpg">.

GERSHENSON, C. Artificial neural networks for beginners. **arXiv preprint cs/0308031**, 2003.

GOLDBERG, Y. **Neural network methods for natural language processing**. [S.l.]: Springer Nature, 2022.

GONZALEZ, R. C.; WOODS, R. E. et al. **Digital image processing**. [S.l.]: Prentice hall Upper Saddle River, NJ:, 2002.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. **Deep learning**. [S.l.]: MIT press, 2016. v. 1. <http://www.deeplearningbook.org>.

HAN, B.; NIU, G.; YAO, J.; YU, X.; XU, M.; TSANG, I.; SUGIYAMA, M. Pumpout: A meta approach for robustly training deep neural networks with noisy labels. 2018.

HAN, B.; YAO, Q.; YU, X.; NIU, G.; XU, M.; HU, W.; TSANG, I.; SUGIYAMA, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. **Advances in neural information processing systems**, v. 31, 2018.

HAN, J.; MORAGA, C. The influence of the sigmoid function parameters on the speed of backpropagation learning. In: SPRINGER. **International Workshop on Artificial Neural Networks**. [S.l.], 1995. p. 195–201.

HAREL, M.; MANNOR, S. Learning from multiple outlooks. **arXiv preprint arXiv:1005.0027**, 2010.

HAYKIN, S. **Neural networks: a comprehensive foundation**. [S.l.]: Prentice Hall PTR, 1994.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.

_____. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.

_____. Identity mappings in deep residual networks. In: SPRINGER. **European conference on computer vision**. [S.l.], 2016. p. 630–645.

HENDRYCKS, D.; MAZEIKA, M.; WILSON, D.; GIMPEL, K. Using trusted data to train deep networks on labels corrupted by severe noise. **Advances in neural information processing systems**, v. 31, 2018.

HENDRYCKS, D.; MU, N.; CUBUK, E. D.; ZOPH, B.; GILMER, J.; LAKSHMINARAYANAN, B. Augmix: A simple data processing method to improve robustness and uncertainty. **arXiv preprint arXiv:1912.02781**, 2019.

HOSSAIN, M.; KAURANEN, I. Crowdsourcing: a comprehensive literature review. **Strategic Outsourcing: An International Journal**, Emerald Group Publishing Limited, v. 8, n. 1, p. 2–22, 2015.

IBRAHEEM, N. A.; HASAN, M. M.; KHAN, R. Z.; MISHRA, P. K. Understanding color models: a review. **ARPN Journal of science and technology**, Citeseer, v. 2, n. 3, p. 265–275, 2012.

IDE, H.; KURITA, T. Improvement of learning for cnn with relu activation by sparse regularization. In: IEEE. **2017 international joint conference on neural networks (IJCNN)**. [S.l.], 2017. p. 2684–2691.

IEEE. **IEEE Xplore**. 1998. Último acesso 18 de Maio de 2024. Disponível em: <"https://ieeexplore.ieee.org">.

JAEHWAN, L.; DONGGEUN, Y.; HYO-EUN, K. Photometric transformer networks and label adjustment for breast density prediction. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops**. [S.l.: s.n.], 2019. p. 0–0.

JONES, N. Using massive amounts of data to recognize photos and speech, deep-learning computers are taking a big step toward true artificial intelligence. **Nature**, v. 505, n. 7482, p. 146–148, 2014.

KALMAN, B. L.; KWASNY, S. C. Why tanh: choosing a sigmoidal function. In: IEEE. **[Proceedings 1992] IJCNN International Joint Conference on Neural Networks**. [S.l.], 1992. v. 4, p. 578–581.

KHAN, A.; SOHAIL, A.; ZAHOORA, U.; QURESHI, A. S. A survey of the recent architectures of deep convolutional neural networks. **Artificial intelligence review**, Springer, v. 53, p. 5455–5516, 2020.

KO, T.; PEDDINTI, V.; POVEY, D.; KHUDANPUR, S. Audio augmentation for speech recognition. In: **Interspeech**. [S.l.: s.n.], 2015. v. 2015, p. 3586.

KRIZHEVSKY, A.; HINTON, G. et al. Learning multiple layers of features from tiny images. Toronto, ON, Canada, 2009.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 1097–1105.

KUMAR, S. **ML Practicum: Image Classification**. 2019. Último acesso 11 de Maio de 2024. Disponível em: <"https://medium.com/secure-and-private-ai-writing-challenge/data-augmentation-increases-accuracy-of-your-model-but-how-aa1913468722">.

KUNDU, R. **Image Processing: Techniques, Types, Applications [2023]**. 2022. Último acesso 11 de Maio de 2024. Disponível em: <"https://assets-global.website-files.com/5d7b77b063a9066d83e1209c/62fe3b32a805ee3e647f00bc_IN%20TEXT%20ASSET.webp">.

KURUVILLA, J.; SUKUMARAN, D.; SANKAR, A.; JOY, S. P. A review on image processing and image segmentation. In: IEEE. **2016 international conference on data mining and advanced computing (SAPIENCE)**. [S.l.], 2016. p. 198–203.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015.

LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; JACKEL, L. D. Backpropagation applied to handwritten zip code recognition. **Neural computation**, MIT Press, v. 1, n. 4, p. 541–551, 1989.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Ieee, v. 86, n. 11, p. 2278–2324, 1998.

LECUN, Y.; JACKEL, L.; BOTTOU, L.; BRUNOT, A.; CORTES, C.; DENKER, J.; DRUCKER, H.; GUYON, I.; MULLER, U.; SACKINGER, E. et al. Comparison of learning algorithms for handwritten digit recognition. In: PERTH, AUSTRALIA. **International conference on artificial neural networks**. [S.l.], 1995. v. 60, p. 53–60.

LI, J.; SOCHER, R.; HOI, S. C. Dividemix: Learning with noisy labels as semi-supervised learning. **arXiv preprint arXiv:2002.07394**, 2020.

LIMA, I. **Entenda a diferença entre as cobras coral-verdadeira e falsa-coral**. 2022. Último acesso 09 de Julho de 2024. Disponível em: <"https://portalamazonia.com/amazonia/voce-conhece-a-diferenca-entre-as-cobras-coral-verdadeira-e-falsa-coral/">.

LIU, S.; NILES-WEED, J.; RAZAVIAN, N.; FERNANDEZ-GRANDA, C. Early-learning regularization prevents memorization of noisy labels. **Advances in neural information processing systems**, v. 33, p. 20331–20342, 2020.

LUKAC, R.; PLATANIOTIS, K. N. Single-sensor camera image processing. In: **Color Image Processing**. [S.l.]: CRC Press, 2018. p. 409–438.

MAIRAL, J.; KONIUSZ, P.; HARCHAOUI, Z.; SCHMID, C. Convolutional kernel networks. **Advances in neural information processing systems**, v. 27, 2014.

MathWorks®. **Convolutional Neural Network**. 2018. Último acesso 17 de Fevereiro de 2018. Disponível em: <"https://www.mathworks.com/discovery/convolutional-neural-network.html">.

MCCONNELL, J. **Computer Graphics: Theory Into Practice**. 1ª edição. ed. [S.l.]: Jones & Bartlett Learning, 2005. ISBN 978-0763722500.

MENESES, P. R.; ALMEIDA, T. d. Introdução ao processamento de imagens de sensoriamento remoto. **Universidade de Brasília, Brasília**, 2012.

MITCHELL, T. M. Machine learning. 1997. **Burr Ridge, IL: McGraw Hill**, v. 45, n. 37, p. 870–877, 1997.

MUSCI, M. **Representação Digital de Imagens**. 2016. Último acesso 22 de Setembro de 2016. Disponível em: <"http://www.musci.com.br/multimidia/ImagensDesenhos3D.pdf">.

O'NEIL, R. et al. Convolution operators and $l(p, q)$ spaces. **Duke Mathematical Journal**, Duke University Press, v. 30, n. 1, p. 129–142, 1963.

PAN, S. J.; YANG, Q. A survey on transfer learning. **IEEE Transactions on knowledge and data engineering**, IEEE, v. 22, n. 10, p. 1345–1359, 2010.

Papers With Code. **Image Classification on ImageNet**. 2024. Último acesso 17 de Maio de 2024. Disponível em: <"https://paperswithcode.com/sota/image-classification-on-imagenet">.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; DESMAISON, A.; KOPF, A.; YANG, E.; DEVITO, Z.; RAISON, M.; TEJANI, A.; CHILAMKURTHY, S.; STEINER, B.; FANG, L.; BAI, J.; CHINTALA, S. Pytorch: An imperative style, high-performance deep learning library. In: **Advances in Neural Information Processing Systems 32**. Curran Associates, Inc., 2019. p. 8024–8035. Disponível em: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.

PATEL, R.; PATEL, S. A comprehensive study of applying convolutional neural network for computer vision. **International Journal of Advanced Science and Technology**, v. 6, n. 6, p. 2161–2174, 2020.

PATRINI, G.; ROZZA, A.; MENON, A. K.; NOCK, R.; QU, L. Making deep neural networks robust to label noise: A loss correction approach. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 1944–1952.

PEREIRA, E.; CARNEIRO, G.; CORDEIRO, F. R. A study on the impact of data augmentation for training convolutional neural networks in the presence of noisy labels. In: IEEE. **2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.], 2022. v. 1, p. 25–30.

REN, M.; ZENG, W.; YANG, B.; URTASUN, R. Learning to reweight examples for robust deep learning. In: PMLR. **International conference on machine learning**. [S.l.], 2018. p. 4334–4343.

ROSSUM, G. V.; JR, F. L. D. **Python reference manual**. [S.l.]: Centrum voor Wiskunde en Informatica Amsterdam, 1995.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group, v. 323, n. 6088, p. 533, 1986.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. [S.l.]: Pearson, 2016.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. **Journal of big data**, SpringerOpen, v. 6, n. 1, p. 1–48, 2019.

SHORTEN, C.; KHOSHGOFTAAR, T. M.; FURHT, B. Text data augmentation for deep learning. **Journal of big Data**, Springer, v. 8, n. 1, p. 101, 2021.

SILVA, L.; ARAUJO, L.; SOUZA, V.; SANTOS, A.; NETO, R. Redes neurais convolucionais aplicadas na detecção de pneumonia através de imagens de raio-x. In: . [S.l.: s.n.], 2020. p. 1–8.

SIROHI, D.; KUMAR, N.; RANA, P. S. Convolutional neural networks for 5g-enabled intelligent transportation system: A systematic review. **Computer Communications**, Elsevier, v. 153, p. 459–498, 2020.

SONG, H.; KIM, M.; PARK, D.; SHIN, Y.; LEE, J.-G. Learning from noisy labels with deep neural networks: A survey. **IEEE transactions on neural networks and learning systems**, IEEE, 2022.

WANG, H.; XIAO, R.; DONG, Y.; FENG, L.; ZHAO, J. Promix: Combating label noise via maximizing clean sample utility. **arXiv preprint arXiv:2207.10276**, 2022.

WANG, S.-C. Artificial neural network. **Interdisciplinary computing in java programming**, Springer, p. 81–100, 2003.

WANG, X.; HUA, Y.; KODIROV, E.; CLIFTON, D. A.; ROBERTSON, N. M. Imae for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude's variance matters. **arXiv preprint arXiv:1903.12141**, 2019.

WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. **Journal of Big Data**, SpringerOpen, v. 3, n. 1, p. 9, 2016.

WERBOS, P.; JOHN, P. J. P. Beyond regression: new tools for prediction and analysis in the behavioral sciences. 01 1974.

Wikimedia Commons. **File:Resolution illustration**. 2006. Último acesso 11 de Maio de 2024. Disponível em: <"https://commons.wikimedia.org/wiki/File:Resolution_illustration.png">.

XIAO, T.; XIA, T.; YANG, Y.; HUANG, C.; WANG, X. Learning from massive noisy labeled data for image classification. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 2691–2699.

XIE, X.; MAO, J.; LIU, Y.; RIJKE, M. de; AI, Q.; HUANG, Y.; ZHANG, M.; MA, S. Improving web image search with contextual information. In: **Proceedings of the 28th ACM International Conference on Information and Knowledge Management**. [S.l.: s.n.], 2019. p. 1683–1692.

YU, X.; LIU, T.; GONG, M.; TAO, D. Learning with biased complementary labels. In: **Proceedings of the European conference on computer vision (ECCV)**. [S.l.: s.n.], 2018. p. 68–83.

YUN, S.; HAN, D.; OH, S. J.; CHUN, S.; CHOE, J.; YOO, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [S.l.: s.n.], 2019. p. 6023–6032.

ZAWADZKI, J. **The Deep Learning(.ai) Dictionary**. 2018. Último acesso 11 de Maio de 2024. Disponível em: <"https://towardsdatascience.com/the-deep-learning-ai-dictionary-ade421df39e4">.

ZHANG, C.; BENGIO, S.; HARDT, M.; RECHT, B.; VINYALS, O. Understanding deep learning (still) requires rethinking generalization. **Communications of the ACM**, ACM New York, NY, USA, v. 64, n. 3, p. 107–115, 2021.

ZHANG, H.; CISSE, M.; DAUPHIN, Y. N.; LOPEZ-PAZ, D. mixup: Beyond empirical risk minimization. **arXiv preprint arXiv:1710.09412**, 2017.

ZHANG, Z.; ZHANG, H.; ARIK, S. O.; LEE, H.; PFISTER, T. Distilling effective supervision from severe label noise. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020. p. 9294–9303.

ZHOU, J. T.; PAN, S. J.; TSANG, I. W.; YAN, Y. Hybrid heterogeneous transfer learning through deep learning. In: **Twenty-eighth AAAI conference on artificial intelligence**. [S.l.: s.n.], 2014.