



Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal Catarinense
Campus Videira

Julia Klopffleisch Schaedler

**O Universo em Bits: Mineração e Análise de Dados
Aplicado à Astronomia para Classificação de Galáxias
Usando Redes Neurais Convolucionais**

Videira – SC
2024

Julia Klopffleisch Schaedler

**O Universo em Bits: Mineração e Análise de Dados
Aplicado à Astronomia para Classificação de Galáxias
Usando Redes Neurais Convolucionais**

Trabalho de Conclusão de Curso apresentado ao
Curso de graduação em Ciência da Computação
do Instituto Federal de Educação, Ciência e
Tecnologia Catarinense — Campus Videira para
obtenção do título de bacharel em Ciência da
Computação
Orientador: Diego Ricardo Krohl, Me

Videira–SC
2024

Julia Klopffleisch Schaedler

O Universo em Bits: Mineração e Análise de Dados Aplicado à Astronomia para Classificação de Galáxias Usando Redes Neurais Convolucionais

Este Trabalho de Curso foi julgado adequado para a obtenção do título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo curso de graduação em Ciência da Computação do Instituto Federal Catarinense – Campus Videira.

Videira (SC), 02 de dezembro, 2024

Orientador Me. Diego Ricardo Krohl
Instituto Federal de Educação, Ciência e Tecnologia Catarinense — Campus Videira

BANCA EXAMINADORA

Dra. Angelita Rettore de Araujo Zanella
Instituto Federal de Educação Ciência e Tecnologia Catarinense – Campus Videira

Dra. Cíntia Fernandes da Silva
Instituto Federal de Educação Ciência e Tecnologia Catarinense – Campus Videira

RESUMO

Este trabalho se propõe a investigar a aplicação de técnicas de mineração de dados astronômicos para classificar galáxias automaticamente, utilizando Redes Neurais Convolucionais (*CNNs*) com diferentes otimizações. O objetivo central é aprimorar a precisão e o desempenho na classificação de galáxias em grandes conjuntos de dados, extraídos de repositórios como o *DES Project* e o *Galaxy Zoo*. Para isso, foram implementados três modelos distintos de *CNN* (A, B e C), cada um com otimizações específicas em sua arquitetura e hiperparâmetros, incluindo variações no número de camadas convolucionais, filtros, funções de ativação e algoritmos de otimização. Os modelos foram treinados e avaliados em relação à sua capacidade de classificar galáxias entre espirais e elípticas, utilizando métricas como acurácia, precisão e *recall*. Esse estudo demonstrou a eficácia das *CNNs* otimizadas na classificação de galáxias, e a importância de buscar os melhores parâmetros. Os modelos A, B e C obtiveram uma taxa de, respectivamente, 73%, 98% e 95% de acurácia ao classificar as galáxias entre os dois grupos. A mineração de dados trouxe informações referentes aos brilhos das galáxias, bem como anomalias relacionadas. Os resultados contribuíram para o desenvolvimento de ferramentas mais robustas e eficientes para análise de dados astronômicos, com potencial para impulsionar descobertas em áreas como a formação de estruturas cósmicas e a detecção de galáxias incomuns. A visualização dos resultados auxiliou na interpretação dos padrões e correlações identificados pelas *CNNs*, fornecendo informações sobre as características morfológicas das galáxias e sua relação com a evolução do universo.

Palavras-chave: mineração de dados, astronomia, galáxias, redes neurais convolucionais, otimizações.

ABSTRACT

This article proposes to investigate the application of astronomical data mining techniques to automatically classify galaxies, using Convolutional Neural Networks (CNNs) with different optimizations. The main objective is to improve the accuracy and performance in classifying galaxies in large datasets, extracted from repositories such as the DES Project and Galaxy Zoo. For this, three distinct CNN models (A, B and C) were implemented, each with specific optimizations in their architecture and hyperparameters, including variations in the number of convolutional layers, filters, activation functions, and optimization algorithms. The models were trained and evaluated in relation to their ability to classify galaxies between spirals and ellipticals, using metrics such as accuracy, precision, and recall. This study demonstrated the effectiveness of optimized CNNs in galaxy classification, and the importance of searching for the best parameters. Models A, B, and C achieved accuracy rates of 73%, 98%, and 95%, respectively, when classifying galaxies between the two groups. Data mining provided information regarding the galaxies' luminosities, as well as related anomalies. The results contributed to the development of more robust and efficient tools for astronomical data analysis, with the potential to drive discoveries in areas such as the formation of cosmic structures and the detection of unusual galaxies. The visualization of the results helped in the interpretation of the patterns and correlations identified by CNNs, providing information on the morphological characteristics of galaxies and their relationship with the evolution of the universe.

Key-words: data mining, astronomy, galaxies, convolutional neural networks, optimizations.

LISTA DE ILUSTRAÇÕES

Figura 1 — Classificação de galáxias por Hubble.....	15
Figura 2 — Subclasses da classificação de galáxias elípticas.....	15
Figura 3 — Subclasses das galáxias espirais.....	16
Figura 4 — Exemplo de galáxia lenticular, a Galáxia do Fuso.....	17
Figura 5 — Subclasses da classificação de galáxias espirais barradas.....	17
Figura 6 — Exemplo de galáxias irregulares: Grande e Pequena Nuvens de Magalhães.....	18
Figura 7 — Etapas para recebimento de dados astronômicos.....	19
Figura 8 — Imagens de um cluster de galáxias.....	20
Figura 9 — Curva de luz do sistema binário GX301-2.....	20
Figura 10 — Espectro do sistema variável cataclísmico BY Cam.....	21
Figura 11 — Demonstração de como dados podem se tornar informações importantes.....	22
Figura 12 — Diagrama ilustrando as principais etapas da mineração de dados.....	23
Figura 13 — Comparação nas etapas de processamento entre Machine Learning e Deep Learning.....	25
Figura 14 — Diagrama de comparação entre um neurônio real e um neurônio artificial.....	26
Figura 15 — Exemplo de estrutura de uma rede neural.....	26
Figura 16 — Função de ativação Sigmoid.....	27
Figura 17 — Função de ativação Tanh.....	28
Figura 18 — Função de ativação ReLU.....	28
Figura 19 — Função de ativação Softmax.....	29
Figura 20 — Técnica de Dropout sendo aplicado em uma rede neural com <i>overfitting</i>	32
Figura 21 — Diagrama de uma rede neural convolucional.....	33
Figura 22 — Programa em Python de uma CNN.....	34
Figura 23 — Resultados do programa anterior.....	35
Figura 24 — Normalização dos dados.....	44
Figura 25 — Pré-processamento das imagens.....	45
Figura 26 — Estrutura do modelo A.....	46
Figura 27 — Estrutura do modelo B.....	48
Figura 28 — Estrutura do modelo C.....	49

Figura 29 — Métricas de avaliação do treinamento do modelo A.....	52
Figura 30 — Matriz de Confusão do modelo A.....	53
Figura 31 — Gráficos de perda e acurácia durante o treinamento do modelo A.....	54
Figura 32 — Métricas de avaliação do modelo B.....	55
Figura 33 — Matriz de Confusão do modelo B.....	56
Figura 34 — Gráficos de perda e acurácia durante o treinamento do modelo B.....	57
Figura 35 — Métricas de avaliação do modelo C.....	58
Figura 36 — Matriz de Confusão do modelo C.....	59
Figura 37 — Gráficos de perda e acurácia durante o treinamento do modelo C.....	60
Figura 38 — Gráfico de perda e acurácia durante o treinamento e validação do modelo C, usando Optuna.....	61
Figura 39 — Gráfico combinado das matrizes de confusão e da curva ROC compara as previsões da nossa CNN com os rótulos do Galaxy Zoo 1 (GZ1), definidos pelos votos corrigidos para viés com um limite de 0,8.....	63
Figura 40 — Galáxias previstas e seus rótulos reais.....	64
Figura 41 — Gráfico da previsão do brilho.....	65
Figura 42 — Gráficos das bandas G, H e I.....	66
Figura 43 — Detecção de anomalias no brilho das galáxias.....	67

LISTA DE ACRÔNIMOS

NASA	The National Aeronautics and Space Administration
STSCI	Space Telescope Science Institute
DES	Dark Energy Survey
JWST	James Webb Space Telescope
ESA	European Space Agency
ALMA	Atacama Large Millimeter Array
VLA	Very Large Array
HST	Hubble Space Telescope
MAST	Mikulski Archive for Space Telescopes
CSA	Canadian Space Agencies
KDD	Knowledge Discovery in Databases
CNN	Convolutional Neural Network

SUMÁRIO

1	INTRODUÇÃO.....	11
	1.1 OBJETIVOS.....	12
	1.1.1 Gerais.....	12
	1.1.2 Específicos.....	12
	1.2 JUSTIFICATIVA.....	12
2	REFERENCIAL TEÓRICO.....	14
	2.1 SOBRE A ASTRONOMIA.....	14
	2.1.1 Dados Astronômicos: Telescópios e missões.....	18
	2.2 MINERAÇÃO DE DADOS.....	22
	2.2.1 Técnicas de Mineração de dados.....	24
	2.3 REDES NEURAIS ARTIFICIAIS.....	25
	2.3.1 Redes Neurais Convolucionais.....	32
	2.4 TRABALHOS RELACIONADOS.....	35
	2.4.1 Otimizando a classificação morfológica automática de galáxias com aprendizado de máquina e aprendizado profundo usando imagens do Dark Energy Survey.....	36
	2.4.2 Uma abordagem de <i>Machine Learning</i> para propriedades de galáxias: distribuições de probabilidade conjuntas de redshift e massa estelar com Random Forest.....	37
	2.4.3 DeepMerge - II. Construindo algoritmos robustos de Deep Learning para identificação de fusão de galáxias em diferentes domínios.....	38
	2.4.4 Um Estudo Robusto de Galáxias de Alto Redshift: <i>Machine Learning</i> Não Supervisionado para Caracterizar Morfologia com o JWST até z ~ 8.....	39
3	METODOLOGIA.....	41
	3.1 LINGUAGEM DE PROGRAMAÇÃO E BIBLIOTECAS.....	41
	3.2 ALGORITMOS.....	42
	3.3 BASES DE DADOS.....	42
	3.4 PLATAFORMAS E OUTRAS FERRAMENTAS.....	43
	3.5 ETAPAS DE DESENVOLVIMENTO.....	43
	3.5.1 Pré-processamento.....	43
	3.5.2 Treinamento dos modelos e avaliação.....	45
	3.5.2.1 Modelo A.....	46
	3.5.2.2 Modelo B.....	47
	3.5.2.3 Modelo C.....	48
	3.5.3 Mineração de dados.....	50
	3.5.4 Visualização e discussão de resultados.....	51

3.6 ORÇAMENTO.....	51
4 RESULTADOS OBTIDOS.....	52
4.1 MODELO A.....	52
4.2 MODELO B.....	55
4.3 MODELO C.....	58
4.4 COMPARAÇÃO DOS RESULTADOS COM DE OUTROS AUTORES.....	62
4.5 RESULTADOS DA MINERAÇÃO DE DADOS.....	63
5 CONCLUSÃO.....	69
REFERÊNCIAS.....	71

INTRODUÇÃO

O advento tecnológico atual permite que as pessoas obtenham e divulguem informações de maneira rápida e precisa, assim gerando uma diversidade de dados. Estes podem ser coletados por dispositivos, como sensores instalados em máquinas, radares e, principalmente, satélites. Conforme Jäger (2020), esses dados constituem o “ouro do século”, demandando processamento e análise para a extração de conhecimento útil. A mineração de dados, área central da Ciência de Dados, surge como ferramenta essencial para essa tarefa, permitindo a identificação de padrões e anomalias em conjuntos de dados complexos (Shashko, 2022).

Na Astronomia, a mineração de dados desempenha um papel fundamental na análise de grandes volumes de informações provenientes de telescópios, sondas e observatórios. A identificação de padrões e anomalias nesses dados contribui para novas descobertas sobre corpos celestes, como galáxias, abordando questões como medições de luminosidade, coordenadas celestes, distribuição de luz e cor, curvatura de luz, agrupamentos galácticos e detecção de objetos (Ivezic et al., 2020). A partir do pré-processamento dessas informações, é possível aplicar técnicas de análise, como: descritiva, agrupamento, predição, associação e detecção de anomalias, utilizando algoritmos que englobam práticas de Inteligência Artificial, como Redes Neurais Artificiais (Castro e Ferrari, 2016).

No entanto, a aplicação da mineração de dados em conjuntos de dados astronômicos massivos, como os das missões DES, Hubble e Gaia, apresenta desafios na escolha das técnicas e algoritmos mais eficazes. O desempenho desses algoritmos, especialmente em tarefas complexas como a classificação morfológica de galáxias, pode ser afetado por fatores como o volume de dados, a qualidade das imagens e a escolha de parâmetros.

Diante dessa problemática, este estudo visou responder à seguinte questão: como otimizar algoritmos para a classificação de galáxias? Para tanto, foram utilizados dados fotométricos do projeto DES (*Dark Energy Survey*), combinados com informações morfológicas dos catálogos do *Galaxy Zoo*. O projeto DES, um levantamento fotométrico que mapeou uma grande área do céu, oferece dados de alta qualidade para a análise de galáxias. Os catálogos do *Galaxy Zoo*, por sua vez, fornecem classificações morfológicas de galáxias realizadas por cientistas cidadãos, complementando os dados do DES.

A aplicação de modelos de Redes Neurais Convolucionais (CNNs), com o auxílio de ferramentas de Python, permitiu a identificação e classificação de galáxias, buscando diferentes otimizações para a detecção de padrões e anomalias. O trabalho foi desenvolvido em etapas, incluindo a seleção e pré-processamento dos dados, o treinamento e ajuste de diferentes algoritmos de CNN, a avaliação do desempenho geral e a análise dos resultados.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Aprimorar algoritmos de redes neurais convolucionais e aplicá-los nas técnicas de mineração de dados astronômicos para classificação de galáxias e melhoria no desempenho geral.

1.1.2 Objetivos Específicos

- Coletar dados disponíveis de diversas missões nos repositórios da *DES Project* e *Galaxy Zoo*;
- Aprimorar algoritmos de redes neurais convolucionais para detecção de padrões, correlações e fenômenos astronômicos, bem como classificar as galáxias entre espirais e elípticas;
- Comparar resultados de testes entre os algoritmos desenvolvidos;
- Possibilitar a visualização de dados para representar os resultados de forma comprehensível.

1.2 JUSTIFICATIVA

O mundo atualmente lida com uma quantidade exorbitante de dados, o *Big Data*, e seu gerenciamento e análise pode ser considerada um desafio, afinal, o *Big Data* consiste em *petabytes* e *exabytes* de informações coletadas por diferentes fontes, e isso, na Astronomia, não é diferente. Por conta da evolução das tecnologias de telescópios e outras ferramentas, esta área está enfrentando desafios para poder processar estes dados e transformá-los em

informações importantes, considerando principalmente a estrutura e complexidade destes (Sen et al., 2022).

Com isso, a ciência de dados, principalmente a mineração de dados, se torna fundamental neste cenário. É estimado que as sondas, bem como os telescópios terrestres e espaciais, geram cerca de 15 TB por noite, acarretando uma demanda para encontrar modelos, aplicações e/ou algoritmos que processam estes conjuntos de dados mais complexos de forma rápida e eficiente (Sen et al., 2022). A mineração de dados dispõe de métodos para analisar estes conjuntos e trazer novas soluções para este problema, trazendo também informações até então desconhecidas que favorecem o avanço científico na área (Chaudhry et al., 2023).

Quando alinhado com modelos de inteligência artificial, como redes neurais artificiais, os resultados podem ser ainda melhores. A tarefa de encontrar padrões ocultos nestes *datasets* é muito mais simples para modelos bem treinados do que feitos manualmente por pessoas (Sen et al., 2022). Entretanto, dependendo da forma em que é programado o algoritmo, este pode não trazer os resultados desejados: com grandes conjuntos de dados, pode tornar o processamento dos modelos muito lento, além de haver possibilidade de erros de classificação, por exemplo. Um estudo publicado por Cheng et al. (2020) revela que aproximadamente 2,5% das galáxias são classificadas incorretamente pelo *Galaxy Zoo 1*, mostrando a importância de otimizações para estes algoritmos conforme a situação.

Por fim, a mineração de dados astronômicos, em conjunto de ferramentas e técnicas otimizadas, pode trazer resultados promissores tanto para área da computação quanto para a astronomia, afinal, tem-se a necessidade de evoluir o desenvolvimento de ferramentas para análise da crescente quantidade de dados, possibilitando novas descobertas nos ramos mencionados.

2 REFERENCIAL TEÓRICO

Nesta seção foram abordadas as fundamentações teóricas, tanto da parte astronômica quanto da parte computacional, para melhor entendimento do trabalho. Foram descritos ainda trabalhos relacionados ao tema que servem de base para o referencial utilizado.

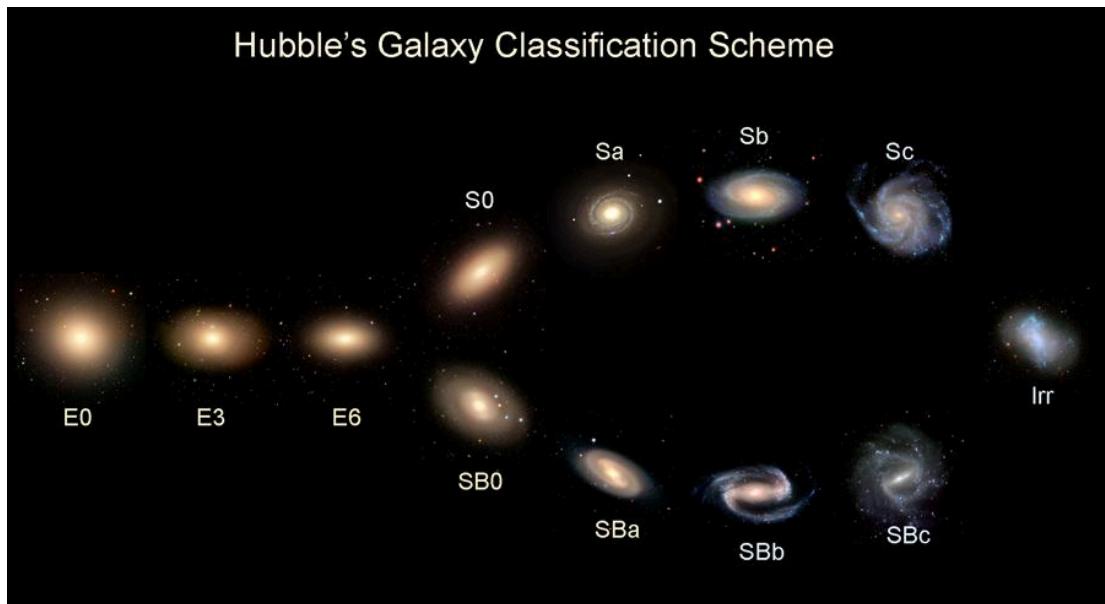
2.1 SOBRE A ASTRONOMIA

A Astronomia é uma das ciências naturais mais antigas, sendo dedicada à observação e interpretação dos fenômenos celestes, como: galáxias, quasares, buracos negros, entre outros. Segundo Borges e Rodrigues (2022), esta área é dividida em duas: teórica e observacional, em que os dados recolhidos pelos observadores por diversos meios, como os telescópios, são usados pelos teóricos para criar e testar modelos e teorias, para explicar e prever fenômenos, anomalias e classificar objetos espaciais.

O objeto celeste em foco deste trabalho são as galáxias. Elas são estruturas massivas compostas por gases, poeira cósmica e matéria escura, além de outros objetos como estrelas e planetas, todos agrupados gravitacionalmente. Essas características morfológicas, segundo Teixeira (2022), contribuem para o entendimento de como se formam e como evoluem, bem como auxiliam na compreensão da evolução cosmológica, na distribuição de massa no universo e da relação de buracos negros.

As galáxias podem ser divididas em dois grandes grupos, segundo a classificação criada pelo astrônomo Edwin Hubble, em 1936: elípticas e螺旋的. Existem algumas galáxias, que por motivos variados, não possuem forma definida, sendo assim classificadas como irregulares. A figura 1 ilustra esta classificação, assim como as subclassificações de cada grupo, incluindo as galáxias irregulares.

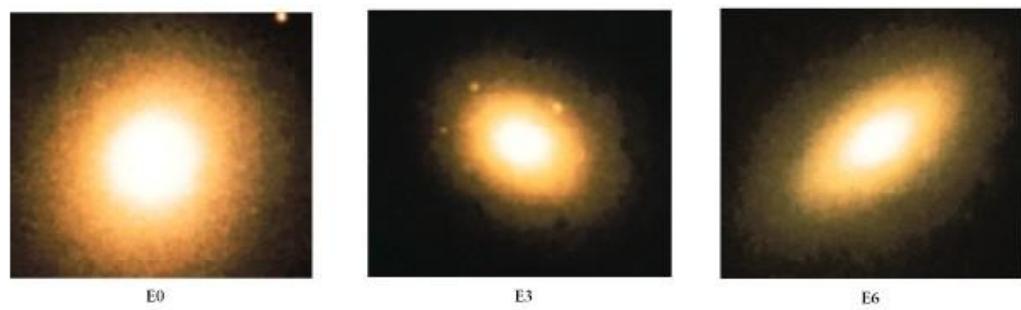
Figura 1 — Classificação de galáxias por Hubble.



Fonte: Boston University Arts & Sciences, 2020

As galáxias elípticas, segundo Saraiva e Oliveira Filho (2015), possuem baixas quantidades de gás, poeira e estrelas jovens. Elas também variam muito de tamanho, podendo ser super-gigantes ou anãs. Na classificação de Hubble, são divididas em classes de E0 a E7 segundo o grau de achatamento. A figura 2 traz exemplos de galáxias das categorias E0, E3 e E6, que ilustram os diferentes níveis de achatamento.

Figura 2 — Subclasses da classificação de galáxias elípticas.

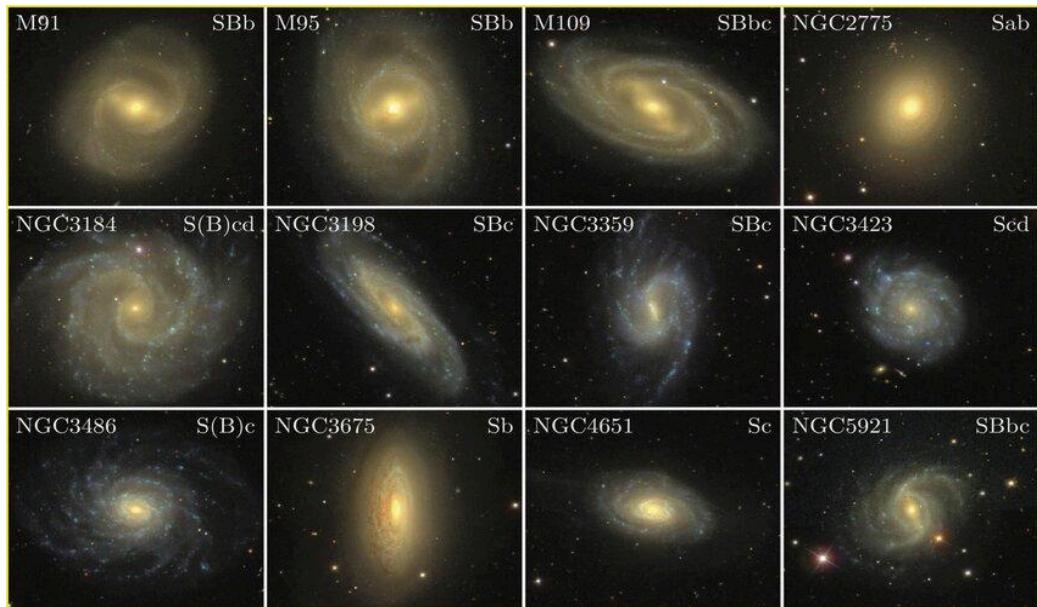


Fonte: Saraiva e Oliveira Filho, 2015.

Galáxias espirais, ainda segundo os conceitos definidos por Saraiva e Oliveira Filho (2015), podem ser classificadas em dois grupos: espirais normais e espirais barradas. As

espirais normais possuem núcleo, disco, halo e braços espirais, podendo variar muito de tamanho. Ainda, podem ser classificadas em subgrupos, conforme a classificação Hubble, sendo: Sa (núcleo maior, braços pequenos e enrolados), Sb (núcleo e braços intermediários) e Sc (núcleo menor, braços grandes e abertos). Para identificação de galáxias barradas e seus subgrupos, é utilizado a nomenclatura SBa, SBb e SBc. A figura 3 traz exemplos de galáxias espirais.

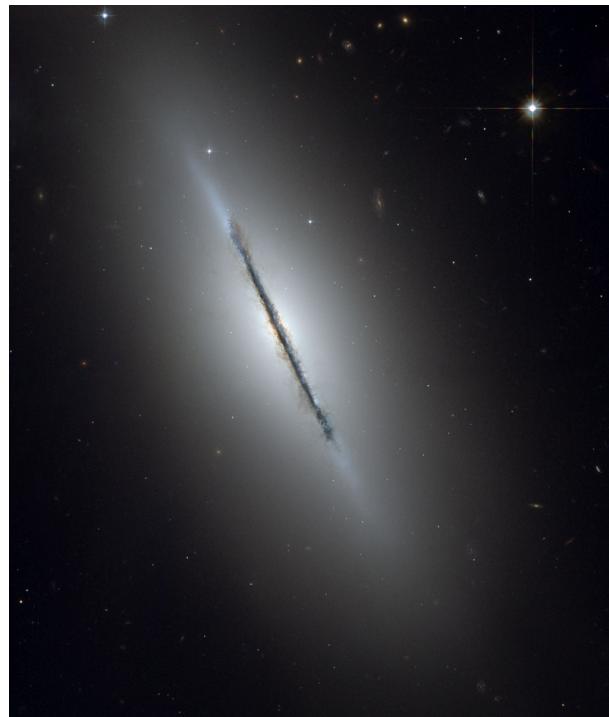
Figura 3 — Subclasses das galáxias espirais.



Fonte: Lima, 2024.

Algumas galáxias entram nessa classificação, mas não possuem braços espirais, sendo assim chamadas de lenticulares ou S0 e seu conjunto formam galáxias discoidais. Um exemplo de galáxia lenticular seria a Galáxia do Fuso (NGC 5866), localizada na Constelação do Dragão, representada na figura 4.

Figura 4 — Exemplo de galáxia lenticular, a Galáxia do Fuso.



Fonte: Wikipédia, s.d.

Além das espirais normais, galáxias espirais podem ser do tipo barrada, ou seja, apresentam uma forma de barra que atravessa seu núcleo, constituindo a sigla SB pela classificação Hubble. Podem ser divididas em subgrupos de acordo com suas formas. Em ambos os grupos, há a presença de nebulosas, poeira e estrelas jovens. Exemplos mais conhecidos são a própria Via Láctea e a galáxia vizinha Andrômeda. Na figura 5 há exemplos das subclasses das galáxias espirais barradas.

Figura 5 — Subclasses da classificação de galáxias espirais barradas.



Fonte: Saraiva e Oliveira Filho, 2015.

Já no caso de galáxias irregulares, para Saraiva e Oliveira Filho (2015), constituem galáxias com estruturas caóticas. É possível que essas galáxias estejam com forte formação estelar, explicando seus formatos mais peculiares. Alguns exemplos conhecidos são a Grande e a Pequena Nuvens de Magalhães, demonstrado na figura 6.

Figura 6 — Exemplo de galáxias irregulares: Grande e Pequena Nuvens de Magalhães.



Fonte: Saraiva e Oliveira Filho, 2015.

Muitos telescópios e missões vêm observando essas estruturas por muito tempo, como na missão Gaia, da ESA, e o Hubble, da NASA. A missão JWST também vem fazendo seu trabalho na observação de galáxias primordiais, que possuem baixa intensidade de luz se comparadas às galáxias mais próximas.

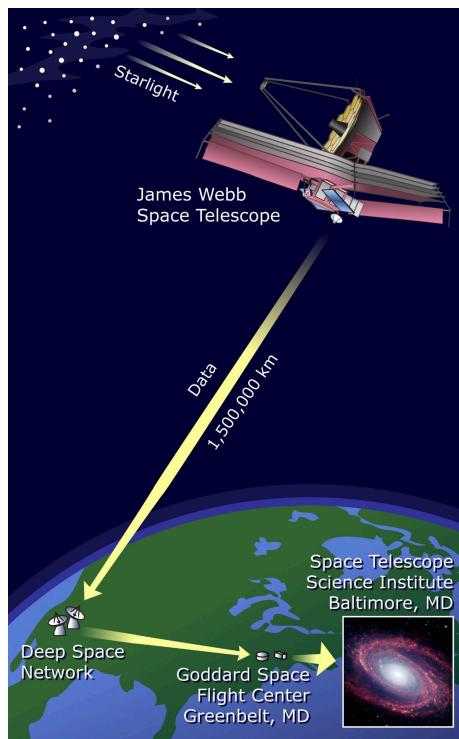
2.1.1 Dados Astronômicos: Telescópios e missões

A captura de dados astronômicos ocorre em diversas etapas, e requer uso de instrumentos específicos. No caso de telescópios espaciais, como no Hubble e JWST, é possível a coleta de dados sem a interferência da atmosfera terrestre. Já os telescópios terrestres, como o Sloan Digital Sky Survey (SDSS), são usados para fazer levantamentos do céu e coleta de dados em grandes volumes (Sands, 2017).

Segundo a NASA (2018), há três etapas principais para conseguir esses dados dos telescópios: acúmulo de dados no instrumento, envio destes dados para sistemas receptores, e a chegada destes dados aos centros científicos, que usam as linhas de telecomunicações comuns. A figura 7 ilustra este cenário. Ainda segundo a NASA (2018), os telescópios

possuem armazenamento de dois tipos de dados que são importantes: *housekeeping data* e *science data*. O primeiro diz respeito a saúde e segurança do próprio telescópio, bem como temperaturas e informações de onde ele está apontando. O segundo tipo é relacionado às informações dos corpos celestes, como imagens e metadados.

Figura 7 — Etapas para recebimento de dados astronômicos.

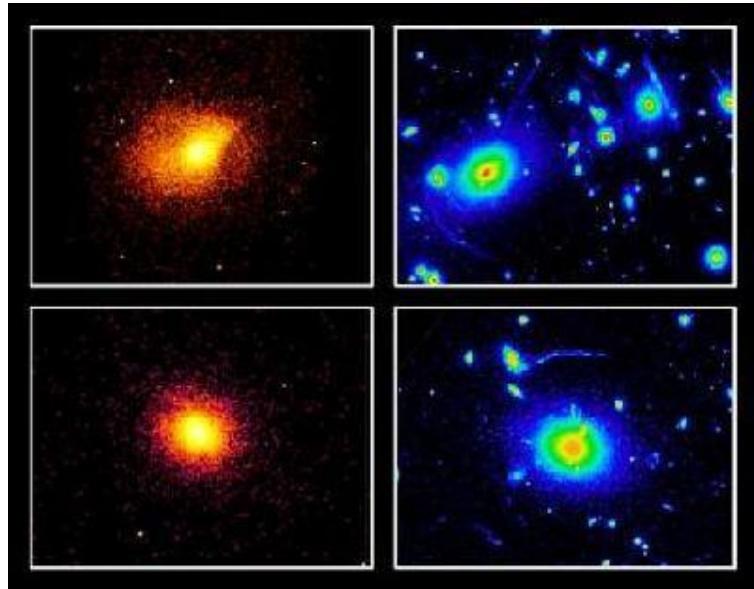


Fonte: Nasa, 2018.

Os dados científicos, neste caso os dados astronômicos, podem ser basicamente de quatro tipos, conforme o *Foundation of Astronomical Studies and Exploration* (2024):

- Imagens, que, na verdade, são um *array* bidimensional de *pixels*. A figura 8 traz um exemplo de como ficam essas imagens em raio-x, na esquerda, e luz óptica, na direita.

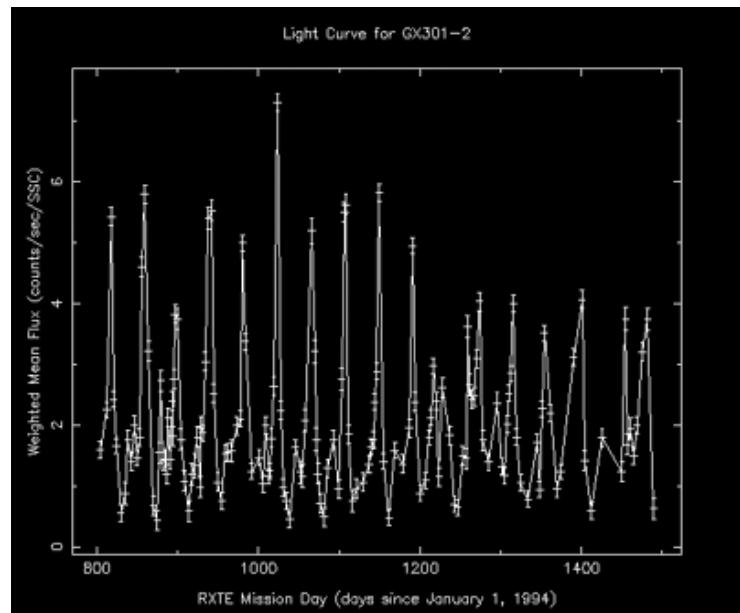
Figura 8 — Imagens de um cluster de galáxias.



Fonte: NASA, 2018.

- Curvas de luz, que são gráficos de luminosidade de um objeto durante um período de tempo. Alguns objetos têm curvas de luz distintas, como sistemas binários, que ilustra duas quedas, representado na figura 9 com o sistema GX301-2;

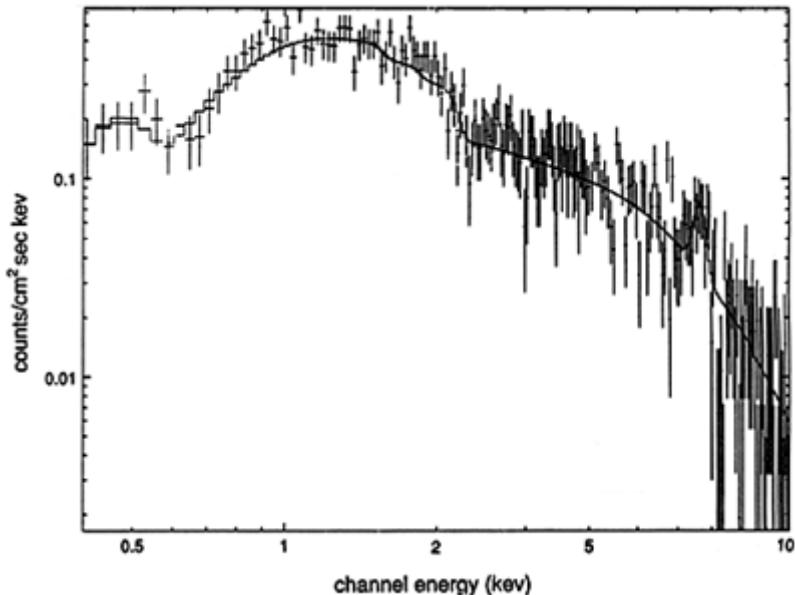
Figura 9 — Curva de luz do sistema binário GX301-2.



Fonte: NASA, 2018.

- Espectro, gráfico que mostra como a intensidade de energia emitida por um objeto celeste é distribuída pelos comprimentos de onda. É possível determinar quais elementos estão no objeto ou entre o objeto e a Terra. A figura 10 representa um gráfico do sistema variável cataclísmico¹ BY Cam (sistema estelar binário);

Figura 10 — Espectro do sistema variável cataclísmico BY Cam.



Fonte: NASA, 2018

- Cubos de dados, que são como imagens bidimensionais que contém todo o espectro recebido de uma região do céu, dando um efeito tridimensional;
- Catálogos, que possuem as propriedades de diferentes tipos de objetos astronômicos.

Esses dados vêm normalmente com meta-data, sendo informações mais aprofundadas sobre os dados coletados, por exemplo: como e quando as imagens são capturadas, localização do objeto, etc. Eles são enviados pelos satélites, comprimidamente, e eventualmente transformados em arquivos FITS (*Flexible Image Transport Standard*) para processamento e análise (NASA, 2018). Muitos destes dados podem ser acessados publicamente em diferentes repositórios, como MAST, Cosmic Hub e DES Project.

¹ O termo refere-se às erupções imprevisíveis no brilho que esses tipos de sistema podem apresentar. No caso de BY Cam, as erupções ocorrem quando a matéria da entre companheira é puxada pela intensa gravidade da estrela anã branca, formando um disco de acreção ao redor dela.

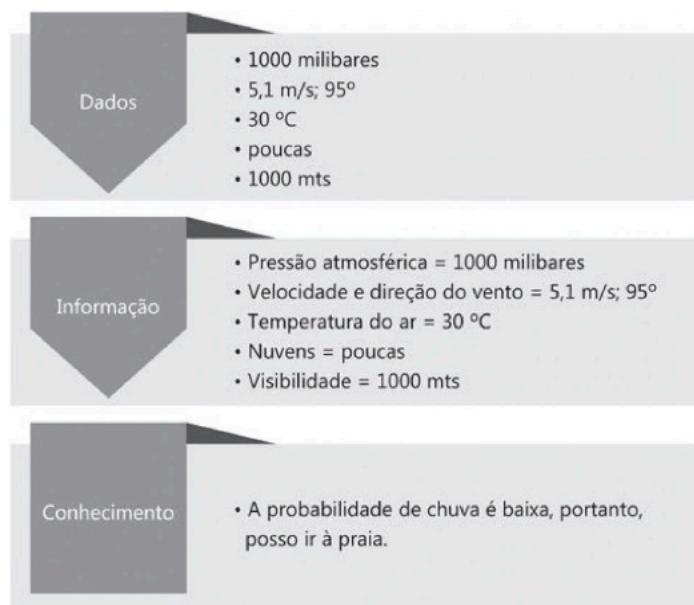
2.2 MINERAÇÃO DE DADOS

Na era da informação, os dados, conhecidos como novo ouro ou novo petróleo do século, são fundamentais para obtenção de informações importantes para novos padrões e tendências, tanto no mundo financeiro quanto no mundo científico (Tan *et al.*, 2019). Esses dados podem ser classificados em dois grupos: estruturados e não estruturados.

Dados estruturados são dados bem padronizados, normalmente apresentando uma tabela que define claramente os seus atributos, como números, textos curtos, datas, etc. São normalmente armazenados em bancos de dados relacionais, espaciais e/ou cubos OLAP, podendo chegar a se tornar uma *Data Warehouse*. Já os dados não estruturados são aqueles que variam de formato e tamanho, podendo ser áudios, vídeos, imagens, etc. Um exemplo de armazenamento destes dados não estruturados está no sistema de arquivos do computador, e uma abundância destes pode ser denominada de *Data Lakes*.

Considerando o cenário, a mineração de dados torna-se uma ferramenta essencial para analisar grandes conjuntos de dados e descobrir padrões, correlações, anomalias e novos fenômenos em base de dados que, caso contrário, manter-se-iam ocultos (Tan *et al.*, 2019). Entretanto, esses dados só têm sentido quando agrupados com outros dados, criando informações e posteriormente conhecimento, como indica a figura 11.

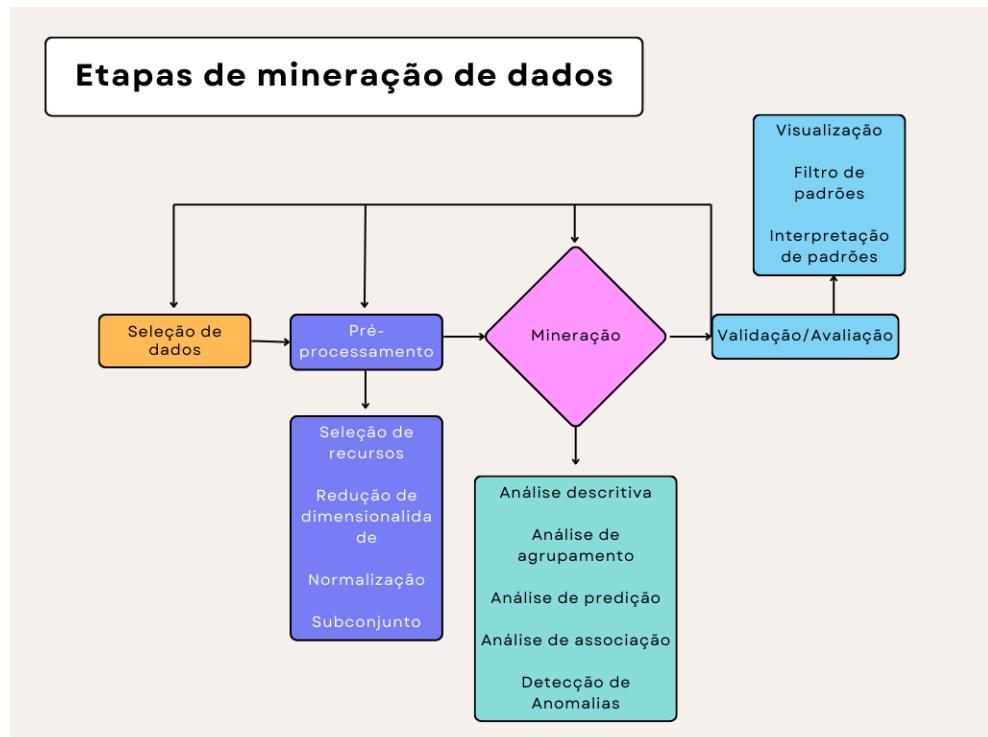
Figura 11 — Demonstração de como dados podem se tornar informações importantes.



Fonte: Castro e Ferrari, 2016.

A mineração de dados é parte de um processo chamado descoberta de conhecimentos em bases de dados (KDD), e pode ser dividida em quatro etapas, como ilustra a figura 12: separação de dados, pré-processamento de dados, mineração de dados e avaliação/validação.

Figura 12 — Diagrama ilustrando as principais etapas da mineração de dados.



Fonte: A autora.

A etapa de separação de dados consiste basicamente em selecionar a base de dados a ser utilizada no processo. O pré-processamento dos dados, para Castro e Ferrari (2016), permite preparar os dados a fim de ter uma análise mais eficaz, e inclui as seguintes sub-etapas: limpeza, para remover ruído e observações duplicadas e/ou incoerentes; integração, que combina dados de várias fontes; redução, a qual é a escolha dados relevantes; e por fim a transformação, que consolida os dados em formatos apropriados. Para Tan *et al.* (2019), a etapa de pré-processamento é a fase que é mais trabalhosa e consome mais tempo dentro de todo o processo de mineração de dados, sendo uma das mais importantes.

A etapa de mineração é o processo em que serão aplicados algoritmos que irão obter conhecimentos a partir dos dados pré-processados. Há diversas técnicas para isso, e Castro e Ferrari (2016) trazem as principais: análise preditiva, usada para classificação e estimativa; análise descritiva, em que aborda questões de tendência central e variância, distribuição e

visualização; agrupamento, com a segmentação de base de dados; associação, para determinação de atributos; e também a detecção de anomalias.

Por fim, a etapa de avaliação ou validação corresponde identificar quais conhecimentos são úteis para o objetivo final. Essa etapa, também chamada de pós-processamento, consiste na filtragem e interpretação de padrões e tem em vista oferecer uma visualização, em que os analistas usam para analisar os resultados a fim de aplicar métodos de testagem de hipóteses (Tan *et al.*, 2019).

Outro aspecto importante da mineração de dados é que é uma área altamente influenciada por outras áreas, como estatística, inteligência artificial, aprendizagem de máquina, processamento de imagens, análise espacial de dados, entre muitas outras (Castro e Ferrari, 2016). Para este trabalho, serão abordados, principalmente, questões de inteligência artificial e aprendizagem de máquina para realizar a mineração e análise dos dados astronômicos.

2.2.1 Técnicas de Mineração de dados

Como visto anteriormente, o processo de mineração de dados consiste em etapas e diversas técnicas diferentes. Essas técnicas são divididas em diferentes grupos com objetivos distintos, o mesmo ocorre com os algoritmos que podem ser utilizados. Muitas destas técnicas são de *Machine Learning* e *Deep Learning*, logo, é importante definir estes conceitos e apresentar exemplos de usabilidade na mineração de dados.

O aprendizado de máquina, ou *Machine Learning*, é uma técnica em inteligência artificial que foca em desenvolver algoritmos para melhor representar um conjunto de dados (Choi *et al.*, 2020). Essa dinâmica permite a vantagem de ter algoritmos que fazem decisões com base nos dados inseridos, assim podendo reconhecer padrões e prever resultados, diferente de outros que apenas seguem instruções estáticas de programação (Bouchefry e Souza, 2020).

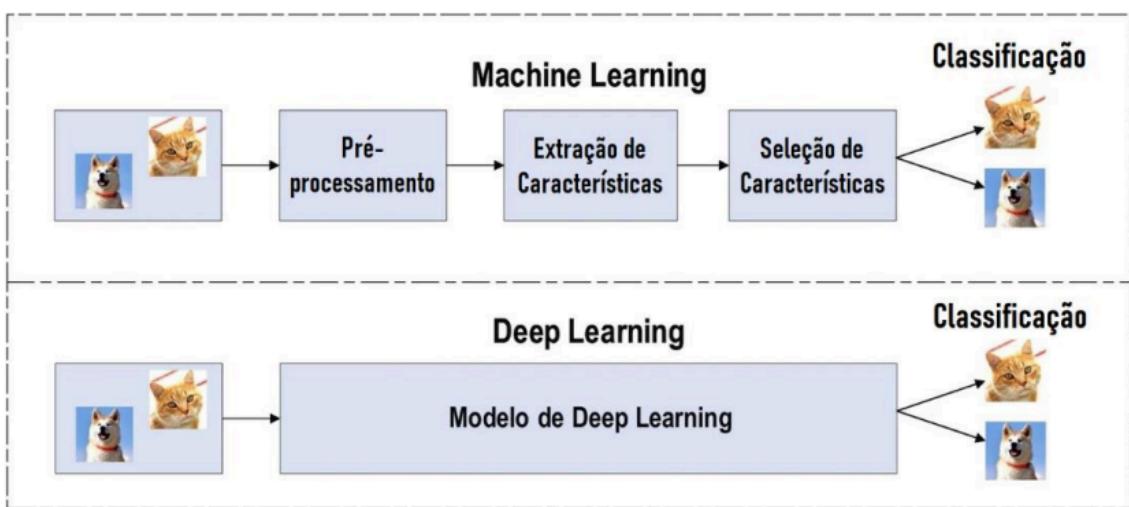
O *Deep Learning*, ou aprendizado de máquina profundo, é um subconjunto do *Machine Learning*, que por suas vez está na área de inteligência artificial. É formada basicamente por redes neurais artificiais com três ou mais camadas. O principal uso está em

questões de dados não estruturados, como texto, imagens e vídeos, capturando relações e padrões complexos em conjuntos de dados (Rai, 2019).

2.3 REDES NEURAIS ARTIFICIAIS

Inspirado nos padrões que o cérebro humano interpreta e processa as informações, as redes neurais artificiais (RNA) usam dados de diversas fontes e analisam em tempo real (Hamaguti e Breve, 2022). Esse modelo aprimora e diminui o número de etapas para aprender e classificar estes dados, como mostra a figura 13 a seguir. O aprendizado profundo traz diversas vantagens para o desenvolvimento de novas tecnologias usadas em outras áreas, como em serviços financeiros, assistência médica, aplicação da lei, entre outros exemplos.

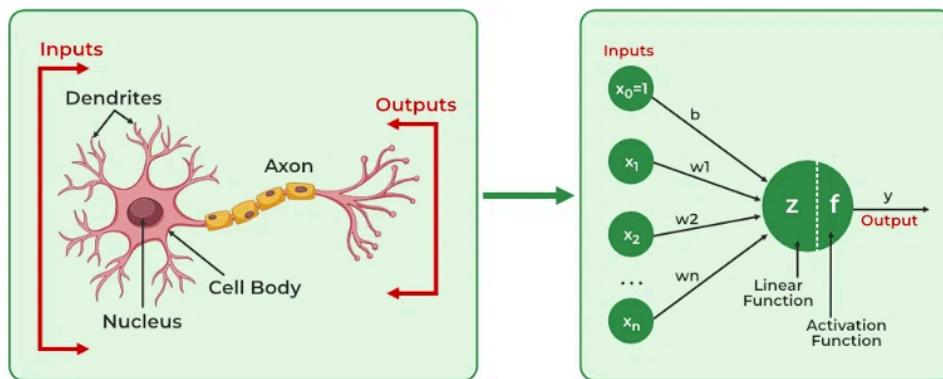
Figura 13 — Comparação nas etapas de processamento entre *Machine Learning* e *Deep Learning*.



Fonte: Hamaguti e Breve, (2022).

Estas redes neurais possuem neurônios artificiais, também chamados de unidades, compostos por entradas (inputs), pesos, bias, funções de ativação e saída. Segundo Cunha (2024), as entradas representam os dados que serão processados pelo neurônio. A cada uma destas entradas está associado um peso, os quais são parâmetros ajustáveis a fim de indicar a importância relativa de cada entrada para a saída do neurônio. Ele também tem um parâmetro adicional que permite ajustar a saída independente das suas entradas, sendo o parâmetro bias. A figura 14 ilustra o neurônio artificial, e o compara com o neurônio biológico.

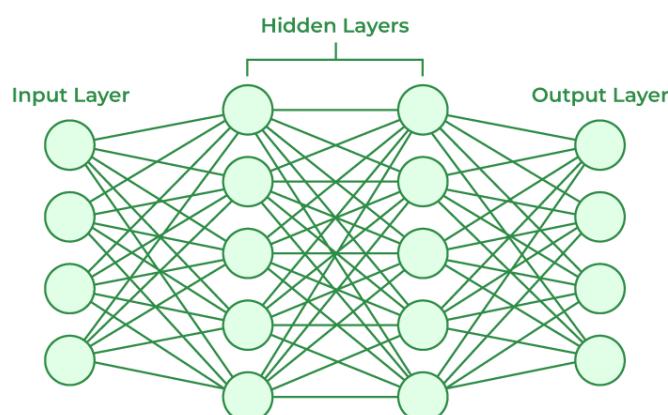
Figura 14 — Diagrama de comparação entre um neurônio real e um neurônio artificial.



Fonte: Geek for Geeks, (2023).

Esses neurônios estão espalhados numa série de camadas que em conjunto formam a estrutura de uma rede neural artificial, como mostra na figura 15, podendo conter dezenas até milhões de neurônios por camada. Conforme o artigo '*Artificial Neural Networks and its Applications*' (2023), cada camada está interligada uma na outra, com pesos que determinam a influência de cada um destes neurônios. A primeira camada é a camada de entrada, responsável por receber os dados para análise e repassá-los para as camadas ocultas, que variam de número segundo o modelo/algoritmo. Após a etapa de processamento nestas camadas, os resultados são passados para a camada de saída.

Figura 15 — Exemplo de estrutura de uma rede neural.

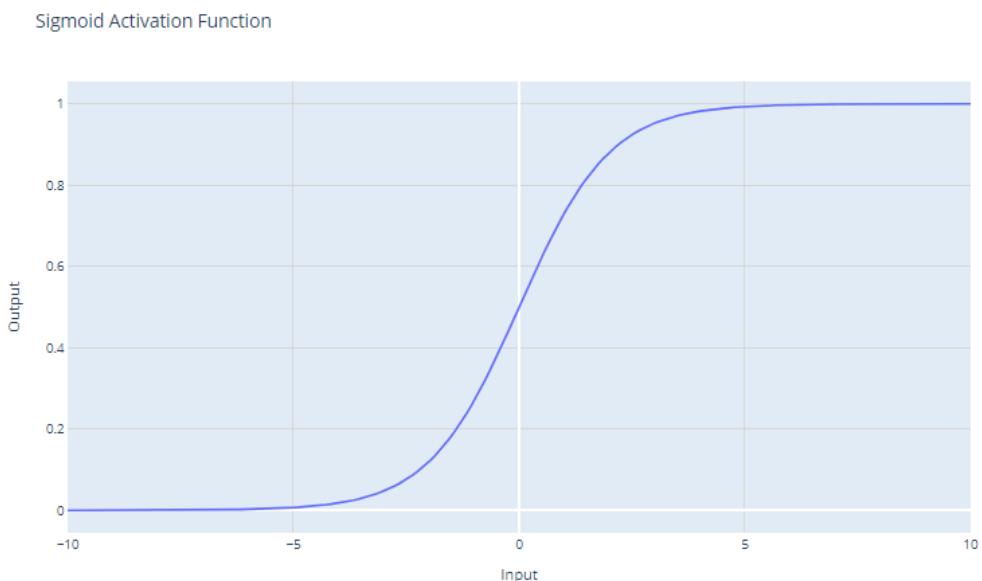


Fonte: Geek for Geeks, (2023).

As funções de ativação recebem a soma ponderada das entradas e do bias, que a transforma de maneira específica, introduzindo assim a não-linearidade ao modelo. Assim, as redes neurais aprendem padrões complexos nos dados (All, 2024). Existem alguns tipos de funções de ativação, e seus usos dependem do tipo de treinamento a ser feito, bem como dos dados a serem analisados. Alguns exemplos são: sigmóide, Tanh, reLU e softmax.

De acordo com All (2024), a função de sigmóide é uma função suave e continuamente diferenciável. Ele recebe um valor real de entrada e a transforma em um valor no intervalo de $[0,1]$. A figura 16 ilustra a curva em forma de “S” da função, normalmente usada para problemas de classificação binária na camada de saída.

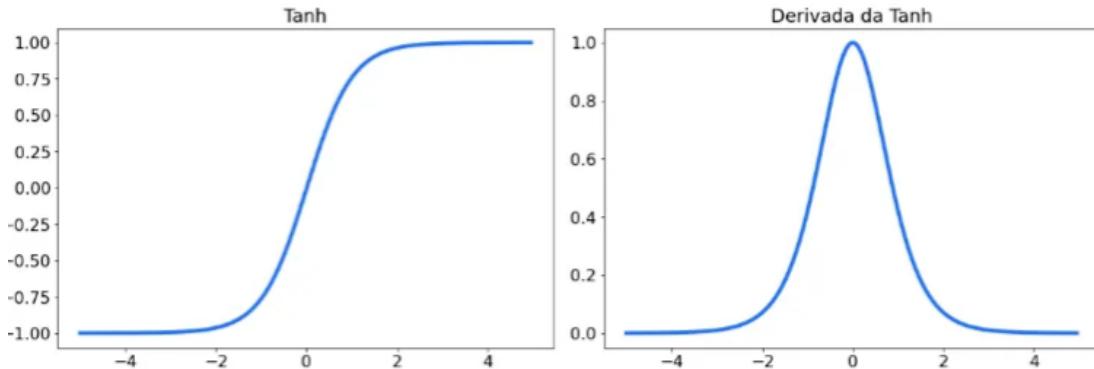
Figura 16 — Função de ativação Sigmóide.



Fonte: All, 2024.

Já a função tangente hiperbólica (Tanh) possui o intervalo de distribuição entre $[-1,1]$, e consequentemente é preferível à função anterior, ao trazer um ganho de facilidade em interpretar os dados de saída (Silva, 2023). A figura 17 ilustra a curva da função Tanh. Um problema deste tipo de função, bem como da função anterior, é o extremo das funções, que pode ocasionar o gradiente de desaparecimento, pois, durante a retropropagação, os gradientes podem se tornar muito pequenos, levando para convergências ruins (All, 2024). Esse tipo é normalmente usado nas camadas ocultas das redes neurais.

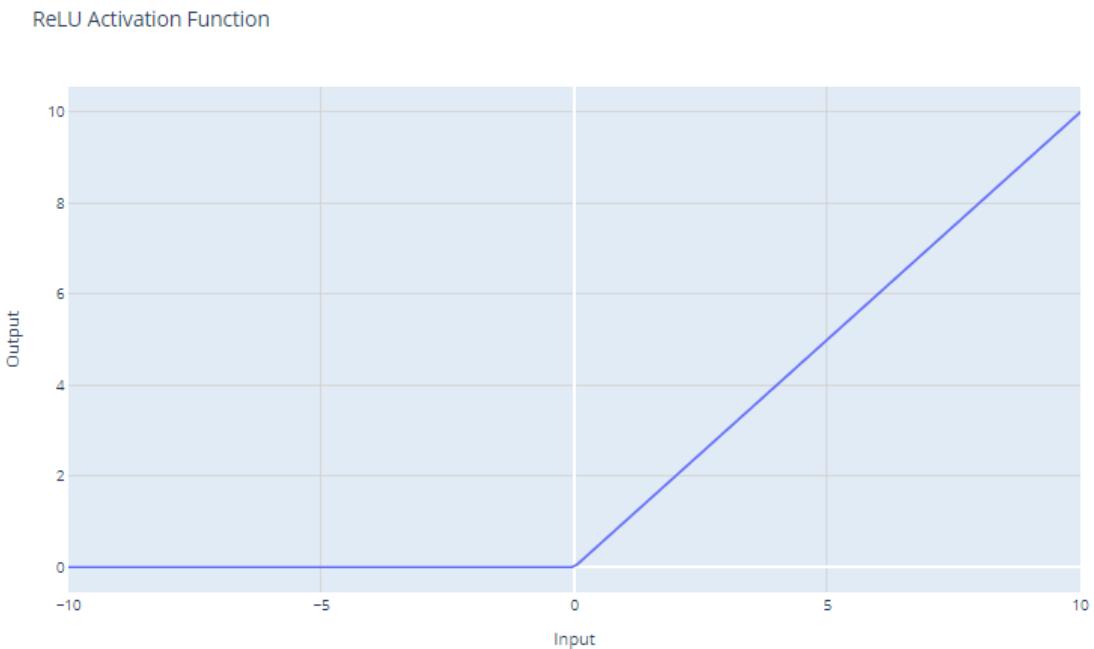
Figura 17 — Função de ativação Tanh.



Fonte: Silva, 2023.

A função de ativação ReLU, ou unidade linear retificada, trabalha com resultados no intervalo $[0, \infty[$. Isso significa que ela retorna 0 para todos os valores negativos, com a tendência de apagar alguns neurônios durante o treinamento, aumentando assim a velocidade. Para os valores positivos, ela retorna o próprio valor. É uma das funções mais utilizadas durante os treinamentos, mas não costuma ser utilizada na camada de saída (Ceccon, 2020). A figura 18 ilustra graficamente a função ReLU.

Figura 18 — Função de ativação ReLU.

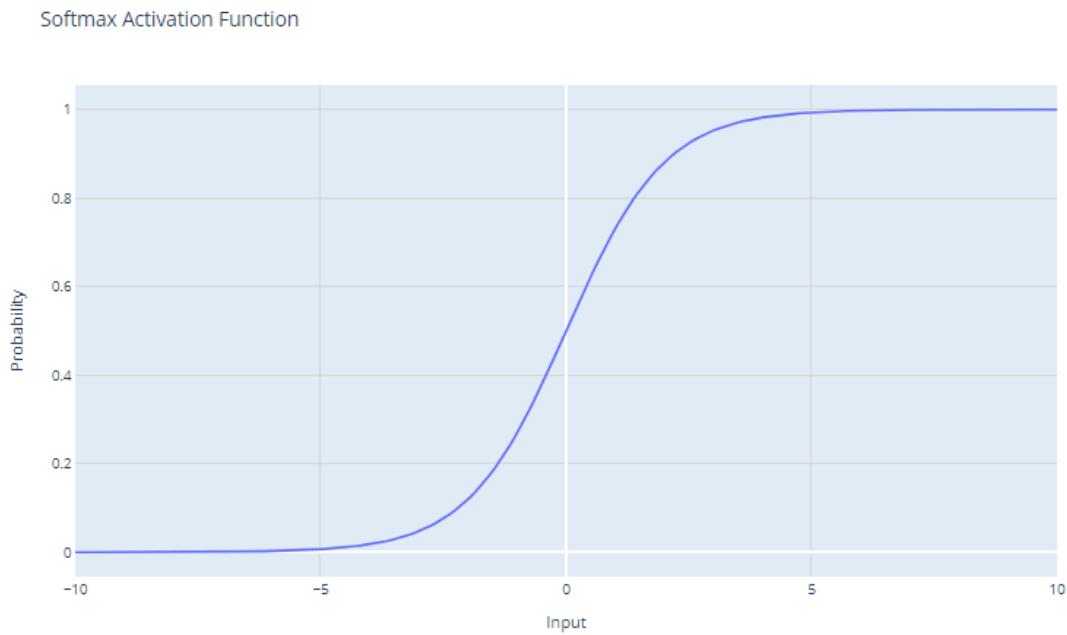


Fonte: All, 2024

Por fim, outro tipo de função de ativação é a softmax, ilustrado na figura 19, também conhecida com função exponencial normalizada (All, 2024). Segundo Ceccon (2020), é

considerada uma generalização da função sigmoide para casos não-binários, sendo comum sua aplicação na camada de saída de problemas de classificação multiclasse, onde produz valores no intervalo $[0, 1]$ em que a soma é igual a 1. A figura 19 demonstra graficamente a função softmax.

Figura 19 — Função de ativação *Softmax*.



Fonte: All, 2024

Referente aos tipos de treinamento, as redes neurais podem aprender de maneira supervisionada, semi-supervisionada, não supervisionada e por reforço. No aprendizado supervisionado as soluções desejadas, os rótulos, já estão inclusos nos dados de treinamento (Géron, 2019). São usados padrões para mapear características de um determinado alvo numa base de dados de modo a fazer previsões em novas bases de dados (Choi et al., 2020). Os dois métodos mais comuns no aprendizado supervisionado são a regressão, que se concentra na previsão de valores numéricos contínuos, e a classificação, que foca na atribuição de categorias aos dados de entrada.

Ainda no aprendizado supervisionado, no cenário das redes neurais, há também duas subcategorias: *transfer learning* e *training from scratch*. No *transfer learning*, ou transferência de aprendizado, é aproveitado o conhecimento adquirido por uma rede em uma tarefa para resolver outra similar. As redes já treinadas são alimentadas com novos dados a fim de realizar tarefas mais específicas (Kleina, 2023). No *training from scratch*, ou

treinamento do zero, como o próprio nome diz, é um método que consiste em construir uma grande base de dados rotulados para a rede aprender diversos atributos, sendo usada em novas aplicações (Kleina, 2023).

O aprendizado semi-supervisionado junta técnicas de ambas as aprendizagens, supervisionada e não supervisionada, sendo ideal para *datasets* que tem tanto dados rotulados quanto sem rótulos (Choi et al., 2020). Esse método pode ser utilizado para classificação de imagens, análise de sentimento e detecção de spam.

No aprendizado não supervisionado, é analisado dados não rotulados e prevê-se os resultados sem a interferência humana, visando juntar dados não organizados por semelhanças, diferenças e outros padrões (Kanade, 2022). As práticas dentro deste método são: clustering, que organiza dados em grupos com base em suas semelhanças; detecção de anomalias; e descoberta de regras de associação, que visa encontrar padrões frequentes nos conjuntos de dados.

Por fim, o último método é a aprendizagem por reforço, em que determinados resultados são desejados e a aprendizagem é feita por tentativa e erro, semelhante ao aprendizado humano (Choi et al., 2020). É uma abordagem interessante, por abrir diversas possibilidades para resolver problemas mais complexos, onde as ações corretas ou incorretas não são triviais. Alguns exemplos de aplicação estão no mundo dos jogos, na robótica e também em controle de sistemas.

Além da função de ativação e do tipo de treinamento, outro conceito importante para as redes neurais artificiais são as otimizações. Umas das técnicas mais essenciais para as redes neurais é o *Backpropagation*, ou Retropropagação. Ele é responsável por ajustar os erros da rede neural com base na taxa de erro obtido na época ou iteração anterior (Bergmann e Stryker, 2024). Consequentemente, minimiza o erro entre a saída obtida pela rede e a saída desejada (Rizzon, 2024). Ele funciona em duas etapas, sendo a *forward pass* e *backward pass*. Na *forward pass*, os dados de entrada são passados pela rede, camada por camada, até que se tenha uma saída e o cálculo entre a saída desejada e saída obtida. No *backward pass*, esse erro é utilizado para ajustar os pesos das conexões a fim de minimizá-lo (Rizzon, 2024). Esses ajustes de pesos podem ser feitos por diferentes otimizadores, como Gradiente Descendente, RMS Prop e Adam.

O gradiente descendente é um dos métodos de otimização mais populares, sendo utilizado para encontrar o mínimo de uma função, e, neste caso, é minimizar a função de erro. Ele utiliza a derivada da função de erro em relação aos pesos das conexões para determinar a direção e o tamanho do ajuste a ser feito em cada peso (Rizzon, 2024). O RMS prop (*root mean square propagation*) é um otimizador que utiliza o conceito de taxa de aprendizado adaptativa, e veio com a proposta de resolver problemas de oscilações extremas nos gradientes, como os problemas de gradiente evanescente e explosivo. A proposta é calcular uma média móvel quadrática dos gradientes, normalizando-os, permitindo assim um ajuste dinâmico do passo (Sanghvirajit, 2021).

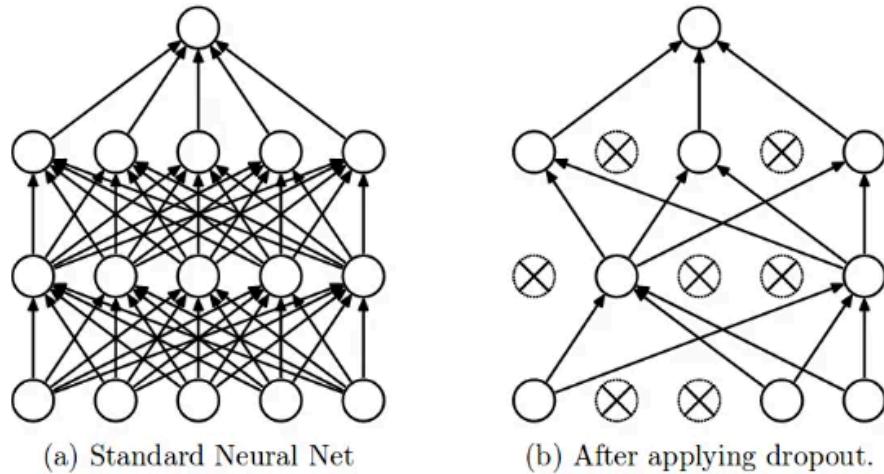
Por fim, o otimizador adam é um algoritmo baseado em gradiente de primeira ordem para ajustar os parâmetros das redes neurais (Sanghvirajit, 2021). É projetado para lidar com funções de objetivo estocásticas, e é comum em problemas de aprendizado de redes neurais onde os dados são processados em lotes aleatórios (*mini-batches*). Esses otimizadores ainda podem recorrer a outras técnicas, como learning rate decay, que diminui a taxa de aprendizado ao longo do treinamento da rede para controlar o quanto grande é o passo que o otimizador dá na direção do mínimo da função de erro.

Um dos principais problemas que as redes neurais podem passar durante seu treinamento é o *overfitting*. Esse problema normalmente se manifesta quando o algoritmo se adapta excessivamente aos dados de treinamento, falhando em fazer previsões ou ter conclusões mais precisas generalizando, ou seja, utilizando outros dados (IBM, s.d.). Os principais indicadores do *overfitting* são baixas taxas de erro e uma alta variância. O contrário também é possível de acontecer, sendo assim chamado de *underfitting*. Os impactos deste resultam em desempenho ruim e previsões imprecisas e ineficazes.

Uma das principais medidas para resolver esse problema é a técnica de *dropout*. Essa técnica de regularização funciona de modo que, durante o treinamento, neurônios selecionados aleatoriamente, tanto na camada de entrada quanto nas camadas ocultas, são desativados (Yadav, 2022). Essa prática resulta em uma sub-rede derivada da rede original, como ilustra a figura 20, e a probabilidade de um neurônio ser desativado é definida pelo parâmetro p (probabilidade de *dropout*). A técnica força a rede a aprender representações mais

robustas e menos dependentes de neurônios individuais, já que a cada iteração do treinamento a rede se depara com uma arquitetura ligeiramente diferente.

Figura 20 — Técnica de *Dropout* sendo aplicado em uma rede neural com *overfitting*.



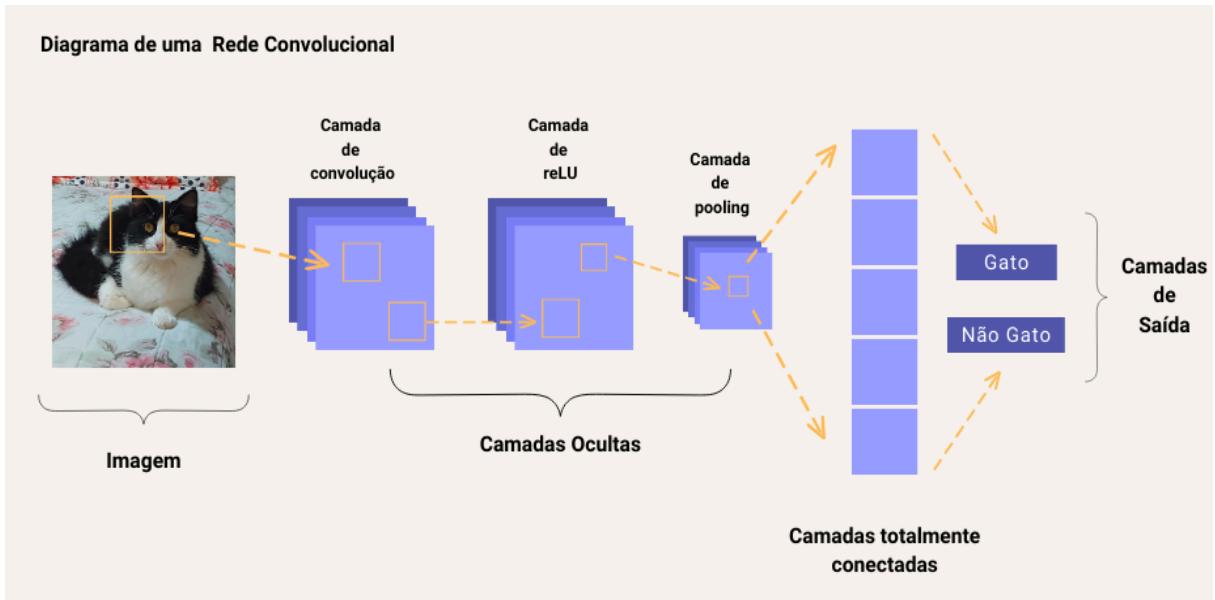
Fonte: Yadav, 2022.

É importante também revisar quais tipos de redes neurais é possível trabalhar. Existem alguns tipos mais comuns, como: redes neurais *feedforward*, recorrentes e convolucionais. A rede neural *feedforward* é um dos modelos mais simples, onde os dados que passam pela rede são unidirecionais e não se estendem ao usar métodos de classificação comuns. Já as redes neurais recorrentes são pensadas para resolver problemas de dados sequenciais, pois os *inputs* de uma etapa são ligados com os *outputs* da etapa anterior, os tornando dependentes um dos outros (Gurjar e Patel, 2022).

2.3.1 Redes Neurais Convolucionais

As redes neurais convolucionais (CNN) são muito utilizadas no reconhecimento de imagens, segmentação e detecção de objetos. Esse tipo de rede pode conter diversas camadas, em que cada uma é treinada automaticamente para poder detectar diferentes características de uma imagem (Millstein, 2018). A figura 21 traz um exemplo de como funciona o modelo de rede neural convolucional, a qual é o modelo escolhido para estudo deste trabalho.

Figura 21 — Diagrama de uma rede neural convolucional.



Fonte: A autora.

Essas camadas têm funções diferentes. Nas camadas de entrada, terá a presença de matrizes tridimensionais com altura, largura e profundidade, determinada pela quantidade de canais de cores (Alves, 2018). Na camada de convolução, são extraídas informações da imagem e mantidas regiões relevantes. Quanto mais profundas são as camadas das convoluções, mais detalhados são os traços identificados com o *activation map* (Alves, 2018). O filtro é outro componente importante, formado por pesos inicializados aleatoriamente, sendo atualizados a cada nova entrada durante o processo de *backpropagation*. Essa região é chamada de receptive field.

Logo, é passado para a camada de função de ativação, neste caso o reLU. A reLU é a função mais utilizada em CNNS por ser mais eficiente computacionalmente (Alves, 2018). A camada de *pooling* simplifica essas informações que chegam da camada anterior, passando as informações para camada totalmente conectada, que realiza a classificação da imagem conforme o treinamento. Sua entrada é a saída da camada anterior, e a saída são N neurônios, cuja quantidade é definida conforme o número de classes para a classificação.

Abaixo, na figura 22, há uma demonstração de um código feito em Python para uma CNN visando implementar um *pipeline* básico de processamento de uma imagem. Primeiro são importadas as bibliotecas essenciais e depois configurados os parâmetros.

Figura 22 — Programa em Python de uma CNN.

```

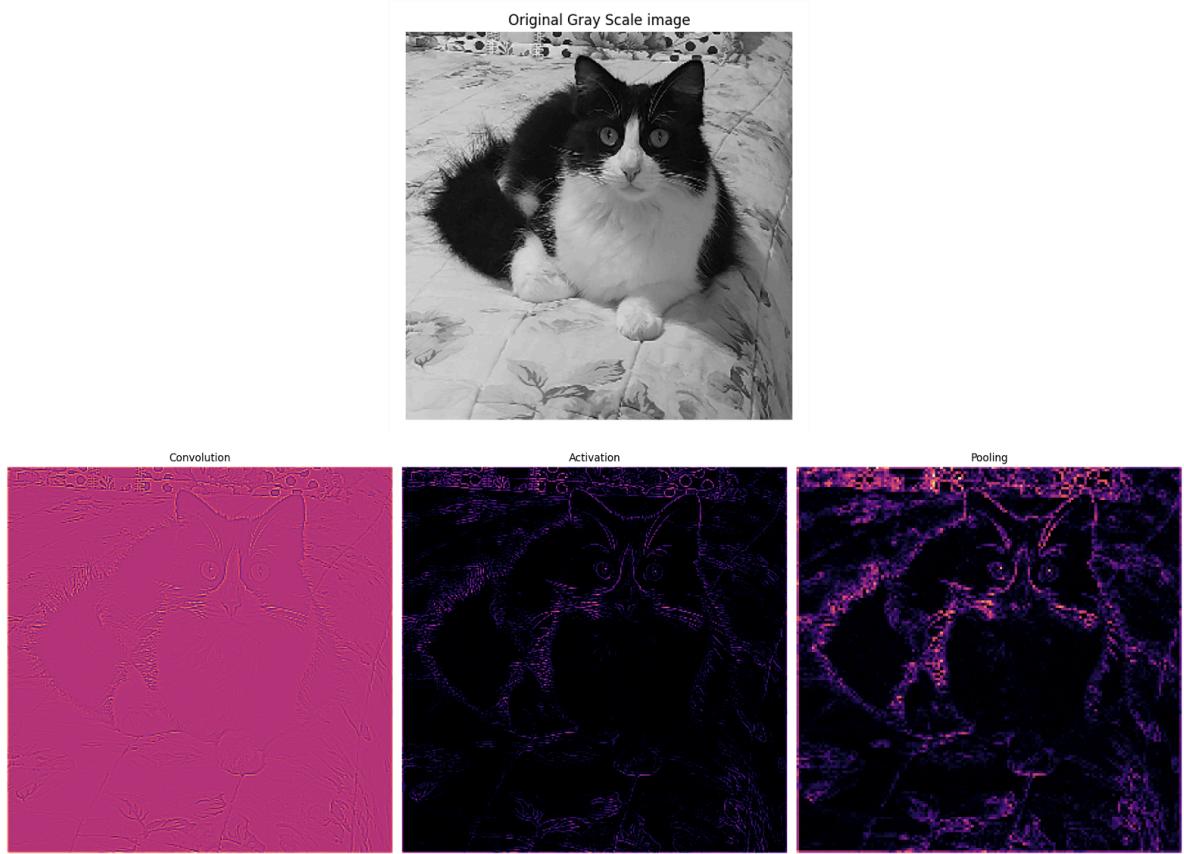
1  # import the necessary Libraries
2  import numpy as np
3  import tensorflow as tf
4  import matplotlib.pyplot as plt
5  from itertools import product
6
7  # set the param
8  plt.rcParams['figure.autolayout=True']
9  plt.rcParams['image.cmap=magma']
10
11 # define the kernel
12 kernel = tf.constant([[-1, -1, -1],
13                      [-1, 8, -1],
14                      [-1, -1, -1],
15                      ])
16
17 # Load the image
18 image = tf.io.read_file('gato2.jpeg')
19 image = tf.io.decode_jpeg(image, channels=1)
20 image = tf.image.resize(image, size=[300, 300])
21
22 # plot the image
23 img = tf.squeeze(image).numpy()
24 plt.figure(figsize=(5, 5))
25 plt.imshow(img, cmap='gray')
26 plt.axis('off')
27 plt.title('Original Gray Scale image')
28 plt.show();
29
30
31 # Reformat
32 image = tf.image.convert_image_dtype(image, dtype=tf.float32)
33 image = tf.expand_dims(image, axis=0)
34 kernel = tf.reshape(kernel, [*kernel.shape, 1, 1])
35 kernel = tf.cast(kernel, dtype=tf.float32)
36
37 # convolution layer
38 conv_fn = tf.nn.conv2d
39
40 image_filter = conv_fn(
41     input=image,
42     filters=kernel,
43     strides=1, # or (1, 1)
44     padding='SAME',
45 )
46
47 plt.figure(figsize=(15, 5))
48
49 # Plot the convolved image
50 plt.subplot(1, 3, 1)
51
52 plt.imshow(
53     tf.squeeze(image_filter)
54 )
55 plt.axis('off')
56 plt.title('Convolution')
57
58 # activation Layer
59 relu_fn = tf.nn.relu
60 # Image detection
61 image_detect = relu_fn(image_filter)
62
63 plt.subplot(1, 3, 2)
64 plt.imshow(
65     #Reformat for plotting
66     tf.squeeze(image_detect)
67 )
68
69 plt.axis('off')
70 plt.title('Activation')
71
72 # Pooling Layer
73 pool = tf.nn.pool
74 image_condense = pool(input=image_detect,
75                         window_shape=(2, 2),
76                         pooling_type='MAX',
77                         strides=(2, 2),
78                         padding='SAME',
79                         )
80
81 plt.subplot(1, 3, 3)
82 plt.imshow(tf.squeeze(image_condense))
83 plt.axis('off')
84 plt.title('Pooling')
85 plt.show()

```

Fonte: Adaptado de Geeks for Geeks, (2024).

O *kernel* é definido e recebe uma imagem para a análise, sendo reformatada e aplicadas as camadas de convolução, com filtro de detecção de bordas; de ativação, para destacar características importantes; e de *pooling*, para extrair as características mais marcantes e também reduzir dimensionalidade. O *output* do programa será a imagem original numa escala cinza, seguida pelas imagens nas respectivas camadas de convolução, ativação e *pooling*, exibido na figura 23.

Figura 23 — Resultados do programa anterior.



Fonte: A autora.

O exemplo acima demonstra o funcionamento básico de uma rede neural convolucional. Contudo, as CNN podem ser adaptadas para diferentes tipos de cenários, e consequentemente, recorrer a técnicas em métodos de aprendizado e otimização diferentes.

2.4 TRABALHOS RELACIONADOS

Nesta seção serão abordados quatro estudos que envolvem a mineração de dados astronômicos a fim de serem utilizados como referência na execução deste trabalho.

2.4.1 Otimizando a classificação morfológica automática de galáxias com aprendizado de máquina e aprendizado profundo usando imagens do Dark Energy Survey.

O estudo publicado no MNRAS 493, 4209–4228 (2020), por Cheng et al, avalia métodos de aprendizado de máquina (AM) e aprendizado profundo para classificar morfologicamente galáxias usando imagens do *Dark Energy Survey* (DES) e classificações visuais do projeto *Galaxy Zoo 1* (GZ1). O objetivo é otimizar a classificação automática de galáxias em elípticas e espirais. Por via de uma amostra de aproximadamente 2800 galáxias, o estudo conclui que a Rede Neural Convolucional (CNN) é o método mais eficiente, alcançando uma precisão de cerca de 99%. A investigação adicional revela que aproximadamente 2,5% das galáxias são classificadas incorretamente pelo GZ1, e após a correção desses rótulos, o desempenho da CNN melhora ainda mais, alcançando uma precisão média de mais de 99% (99,4% como melhor resultado).

O estudo também explora a influência da rotação de imagens na classificação, a importância do equilíbrio entre o número de amostras de cada tipo na base de treinamento e o efeito de diferentes tipos de entrada de dados (imagem bruta, Histograma de Gradientes Orientados - HOG é uma combinação das duas). A CNN mostra melhor desempenho quando alimentada com uma combinação de imagens brutas e HOG. Além disso, o *Random Forest* (RF) é destacado como um método eficaz e eficiente, comparando-se favoravelmente com outras abordagens de AM.

Os resultados indicam que a CNN é o melhor algoritmo para classificar morfologicamente galáxias a partir de dados de imagem, permitindo a construção de um catálogo de morfologia de galáxias detalhado. O estudo sugere que a classificação automática pode redescobrir que as galáxias lenticulares (S0) são distintas de galáxias elípticas e espirais, e que a correção dos rótulos incorretos em conjuntos de dados existentes pode melhorar significativamente a precisão das classificações de galáxias.

2.4.2 Uma abordagem de *Machine Learning* para propriedades de galáxias: distribuições de probabilidade conjuntas de redshift e massa estelar com *Random Forest*.

Este estudo, de Sunil et al (2021), descreve um método para estimar distribuições de probabilidade conjuntas de *redshift* e massa estelar de galáxias usando o algoritmo de aprendizado de máquina *Random Forest* (RF). O estudo foi realizado com dados do *Dark Energy Survey* e do catálogo COSMOS2015, visando demonstrar que o RF pode produzir estimativas precisas mesmo com dados fotométricos limitados a poucas bandas.

Os autores construíram dois modelos de aprendizado de máquina: um com dados fotométricos profundos nas bandas griz e outro que reflete a dispersão fotométrica presente na pesquisa principal do DES. Eles validaram as distribuições de probabilidade conjuntas para um conjunto de teste de 10.699 galáxias usando a transformada integral de probabilidade cópula e a função de distribuição de Kendall, e suas contrapartes univariadas para validar as marginais. O método baseado em RF superou os métodos de ajuste de modelo em todas as métricas de desempenho pré-definidas e conseguiu computar distribuições de probabilidade conjuntas para um milhão de galáxias em menos de 6 minutos com hardware de computador convencional.

Além disso, os autores desenvolveram o GALPRO, um pacote Python eficiente e intuitivo para geração rápida de distribuições de probabilidade multivariadas em tempo real. O pacote está disponível para uso por pesquisadores em estudos de cosmologia e evolução de galáxias. O documento também compara o desempenho do RF com o código de ajuste de modelo BAGPIPES, demonstrando que o método baseado em RF é superior em termos de acurácia e velocidade.

As principais conclusões do estudo são que o método RF consegue produzir estimativas precisas e rápidas de *redshift* e massa estelar, fundamentais para estudos de evolução de galáxias e estrutura do universo em grande escala. O pacote GALPRO oferece uma solução eficiente para a geração de distribuições de probabilidade conjuntas, potencialmente permitindo uma redução significativa no tempo de execução em comparação com métodos baseados em ajuste de modelo, o que é particularmente relevante para as próximas gerações de grandes levantamentos fotométricos.

2.4.3 DeepMerge - II. Construindo algoritmos robustos de *Deep Learning* para identificação de fusão de galáxias em diferentes domínios.

O artigo publicado no MNRAS 506, 677–691 (2021) aborda o desafio de treinar modelos de redes neurais em dados de simulação astronômica e aplicá-los a observações telescópicas reais. Os autores, liderados por A. Ciprijanović, investigam técnicas de adaptação de domínio, como a Discrepância Máxima Média (MMD) e Redes Neurais Adversárias de Domínio (DANNs), para melhorar a classificação de galáxias em colisão e não colisão em diferentes domínios de dados. Eles também exploram a utilização de perdas de Fisher e minimização de entropia para melhorar a discriminabilidade entre classes nos domínios.

O estudo demonstra que a aplicação de técnicas de adaptação de domínio pode aumentar a precisão da classificação no domínio alvo em até 20%. A pesquisa foi realizada em dois conjuntos de dados: um entre dois conjuntos de dados simulados de galáxias colidentes distantes e outro entre dados simulados de galáxias colidentes próximas e dados observacionais do Sloan Digital Sky Survey. Os resultados mostram que o uso de técnicas de adaptação de domínio permite que os astrônomos apliquem com sucesso modelos de redes neurais treinadas em dados de simulação para detectar e estudar objetos astrofísicos em levantamentos astronômicos atuais e futuros de grande escala.

Os autores também discutem a interpretabilidade dos modelos usando t-SNE e Grad-CAMs, e consideram questões como a transferência negativa e o manejo de pequenos conjuntos de dados na ciência. O apêndice do artigo contém detalhes adicionais sobre hiperparâmetros de treinamento e desempenho da rede neural.

As contribuições dos autores incluem o desenvolvimento de técnicas de aprendizado profundo para a identificação de galáxias em colisão em diferentes domínios de dados, bem como a melhoria da classificação de imagens astronômicas através da aplicação de métodos de adaptação de domínio. Os dados simulados e observacionais utilizados no estudo estão disponíveis no Zenodo, e o código está disponível no GitHub. O artigo enfatiza o potencial das técnicas de adaptação de domínio para a astronomia e outras ciências naturais, permitindo a combinação e o aproveitamento de todos os dados observacionais e simulados disponíveis.

2.4.4 Um Estudo Robusto de Galáxias de Alto *Redshift*: *Machine Learning* Não Supervisionado para Caracterizar Morfologia com o JWST até $z \sim 8$

Neste estudo publicado pelos autores Tohill et al (2024), é investigado a morfologia de galáxias de alto redshift utilizando dados do JWST além de técnicas de aprendizado de máquina não supervisionado. O objetivo é desenvolver um sistema robusto para caracterizar a morfologia de galáxias em épocas cósmicas mais antigas, evitando viéses introduzidos por classificações baseadas em nomenclatura de galáxias de baixo redshift. Os autores, liderados por L. Ferreira, utilizaram *variational autoencoders* para extrair características de 6869 galáxias com *redshift* maior que 2, incluindo 255 galáxias com *redshift* maior que 5, detectadas nos campos do *Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey* (CANDELS) e do *Cosmic Evolution Early Release Science Survey* (CEERS).

Através da eliminação de efeitos de orientação e fontes de fundo antes da codificação das características das galáxias, os pesquisadores construíram um espaço de características fisicamente significativas. Eles identificaram 11 classes morfológicas distintas, separadas por parâmetros estruturais como concentração, assimetria, suavidade (CAS) e o índice de Sérsic. Observou-se um declínio na presença de galáxias do tipo esferoidal com o aumento do *redshift*, sugerindo a dominância de galáxias do tipo disco no universo primordial. Os autores concluem que os sistemas de classificação visuais convencionais são inadequados para classificar a morfologia de galáxias de alto *redshift* e defendem a necessidade de um esquema de classificação mais detalhado e refinado.

É demonstrado que os recursos extraídos por máquinas se alinham melhor com os parâmetros medidos do que os métodos tradicionais, oferecendo maior relevância física. Os autores propõem um novo sistema de classificação baseado nos recursos extraídos pelas máquinas e ilustram como esses clusters se alinham com os parâmetros medidos. O estudo também explora a evolução das galáxias massivas com o *redshift* e a morfologia de galáxias com alta taxa específica de formação estelar, encontrando uma diversidade de morfologias já presentes em épocas cósmicas iniciais.

Em resumo, o estudo conclui haver eficácia de um sistema de classificação visual para galáxias em alto *redshift*. Ele confirma também a predominância de galáxias do tipo disco

nesses *redshifts* e destaca a relevância dos métodos de aprendizado de máquina não supervisionado para identificar características morfológicas chave e seu impacto nas propriedades físicas das galáxias.

3 METODOLOGIA

O objetivo deste trabalho é aprimorar algoritmos para que eles tenham um melhor desempenho e, consequentemente, melhores resultados, buscando por informações ou detalhes antes não encontrados nos conjuntos de dados selecionados. Como a pesquisa visa analisar dados astronômicos e descobrir padrões desconhecidos, o trabalho se encaixa no tipo de pesquisa exploratória com uma abordagem predominantemente quantitativa.

Nesta seção foram abordadas as ferramentas e técnicas utilizadas para a mineração de dados astronômicos para detecção de galáxias, com base nas pesquisas já feitas sobre o assunto.

3.1 LINGUAGEM DE PROGRAMAÇÃO E BIBLIOTECAS

A linguagem de programação escolhida para o desenvolvimento dos algoritmos a serem utilizados na mineração e análise de dados é a linguagem Python, que possui uma tipagem dinâmica e forte. Além disso, é multiparadigma e possui um modelo de desenvolvimento comunitário, e justamente por este motivo a linguagem desenvolveu-se inovadoramente e passou a ser uma das mais importantes em ciência de dados (McKinney, 2018). Com a linguagem, foram utilizadas técnicas de programação paralela para otimizar o processamento dos algoritmos, sendo usado bibliotecas como *multiprocessing* e *threadPoolExecutor*, e também ferramentas como CUDA e cuDNN para realizar o processamento com uso de GPU. Além disso, outras bibliotecas utilizadas para realizar esse processo de análise de dados, foram, principalmente:

- NumPy: base de computação científica em Python, fornecendo ferramentas para trabalhar com *arrays* multidimensionais, além de funções matemáticas e operações de álgebra linear, números aleatórios, etc.
- Astropy: oferece ferramentas para lidar com manipulação de coordenadas celestes, leitura e escrita de dados astronômicos, acesso a catálogos astronômicos online, entre outros.
- Matplotlib: biblioteca de visualização de dados por gráficos 2D, como histogramas, gráficos de dispersão, mapas de calor, entre outros. Ele permite também personalizar estes gráficos, para a visualização ficar conforme o esperado.

- TensorFlow: é uma biblioteca que facilita o processo de *Machine Learning* e *Deep Learning*, permitindo construir e treinar modelos para diversas tarefas, como reconhecimento de imagem, processamento de linguagem natural, etc.

3.2 ALGORITMOS

Tratando-se do processo de mineração de dados, é importante definir quais ferramentas e algoritmos a serem utilizados para analisar as bases de dados encontradas, tanto da etapa de pré-processamento dos dados como a mineração propriamente dita. O escolhido para ser abordado nesta pesquisa é o algoritmo de redes neurais convolucionais (CNN), explicado com exemplo na seção 2.3.1. Na etapa de pré-processamento foi utilizado um algoritmo para a normalização e limpeza nos metadados, assim como a criação de rótulos para classificar as imagens. No pré-processamento das imagens, foram usadas técnicas de redimensionamento, normalização, rotação, aplicação de diferentes escalas e balanceamento de dados. Já na etapa de mineração, as imagens foram analisadas e comparadas com base no treinamento da rede neural, e os resultados foram exibidos e discutidos por via de gráficos.

O algoritmo foi treinado com dados coletados e classificados de galáxias, e comparado com os resultados dos outros algoritmos de teste, bem como com os resultados de outros estudos, principalmente com o estudo referenciado na seção 2.4.1.

3.3 BASES DE DADOS

As bases de dados escolhidas são aquelas disponibilizadas publicamente nas plataformas da *DES Project* e *Galaxy Zoo*, sendo recolhidas de diferentes missões. Os conjuntos de dados são divididos em *datasets* de metadados, com informações e rótulos das galáxias, e também em *datasets* com imagens propriamente ditas. Os *datasets* de metadados são provenientes dos catálogos *DES Y3 GOLD* e também do *DES Y1* com o *Galaxy Zoo 1*. O primeiro contém cerca de 5 GB com mais de 21 mil entradas com informações de milhares de galáxias, tanto espirais quanto elípticas. O segundo, são o conjunto de cerca de 2800 galáxias, com os seus metadados. Os dados disponíveis são de anos anteriores, analisados em artigos com os de Cheng et al. (2021) e Lintott et al. (2011), dadas as políticas internas das

instituições e pesquisadores que detém direitos a estes dados. Também serão coletados dados de objetos celestes rotulados como desconhecidos a fim de analisar as possibilidades de haver alguma galáxia neste conjunto, como mencionado anteriormente. Os dados serão baixados dos sites oficiais destas plataformas e armazenados em nuvem.

3.4 PLATAFORMAS E OUTRAS FERRAMENTAS

As plataformas utilizadas foram o próprio computador físico, com 32 *gigabytes*, processador Intel I7 e placa de vídeo Nvidia *Geforce GTX 1660 6GB*. Foi utilizado a IDE (do inglês, Ambiente Desenvolvimento Integrado) *Visual Studio Code* (VS Code) para o desenvolvimento, com auxílio do Github para controle de versões e repositório. Parte das ferramentas utilizadas foram provenientes de bibliotecas do Python. Além disso, foram usados os gráficos gerados pelas bibliotecas Python, para dispor os resultados.

3.5 ETAPAS DE DESENVOLVIMENTO

Para compreender melhor a metodologia deste trabalho, foi destacado abaixo as principais etapas de desenvolvimento deste trabalho.

3.5.1 Pré-processamento

Como mencionado anteriormente, as bases de dados, ou *datasets*, utilizadas são provenientes dos projetos *DES* e *Galaxy Zoo*. Para etapa de pré-processamento, foram feitos dois processos: o pré-processamento de metadados (dados tabulares) e o pré-processamento de imagens. No primeiro processo, foi realizada a extração de rótulos, para utilizar no treinamento dos modelos, e também a normalização dos dados. Os dados brutos, armazenados em formato FITS (*Flexible Image Transport System*), são carregados e convertidos em um *DataFrame Pandas*. Esse processo, permite a manipulação e análise dos dados eficientemente. Com isso, os rótulos que identificam as imagens no *dataset* são extraídos para facilitar o treinamento dos modelos na próxima etapa.

O restante dos dados passa pelo processo de normalização. A normalização de dados é utilizada para transformar os dados em uma escala comum, sem distorcer as diferenças nas faixas de valores ou perder informações. As colunas contendo magnitudes e erros nas bandas G, R, I, Z e Y são normalizadas utilizando a técnica de padronização, que subtrai a média e divide pelo desvio padrão de cada coluna. A figura 24 demonstra como foi implementado a normalização dos dados durante o pré-processamento destes dados tabulares.

Figura 24 — Normalização dos dados.

```
# Função para normalizar colunas de magnitude e erros
def normalizar_colunas(df):
    logging.info("Normalizando colunas de magnitude e erros")
    colunas_mag = ['MAG_AUTO_G', 'MAG_AUTO_R', 'MAG_AUTO_I', 'MAG_AUTO_Z', 'MAG_AUTO_Y']
    colunas_magerr = ['MAGERR_AUTO_G', 'MAGERR_AUTO_R', 'MAGERR_AUTO_I', 'MAGERR_AUTO_Z', 'MAGERR_AUTO_Y']

    df[colunas_mag] = df[colunas_mag].apply(lambda x: (x - x.mean()) / x.std())
    df[colunas_magerr] = df[colunas_magerr].apply(lambda x: (x - x.mean()) / x.std())

    return df
```

Fonte: A autora.

Essa etapa garante que todas as *features* tenham a mesma escala, evitando que *features* com magnitudes maiores dominem o processo de treinamento da CNN. Todo o pré-processamento foi realizado com programação paralela, a fim de agilizar o processo.

No pré-processamento das imagens, foi feita, inicialmente, a conversão para o formato RGB de 3 canais e redimensionadas para um tamanho padrão, garantindo uniformidade no input da rede neural. Outras medidas realizadas foram: conversão para escala de cinza e aplicação de filtro de mediana para redução de ruído, detecção de bordas usando o algoritmo *Canny*, equalização de histograma adaptativo, CLAHE, para realçar o contraste, e também a normalização para garantir que o valor dos *pixels* estejam em uma faixa específica. A figura 25 demonstra essa implementação, bem como a implementação do data augmentation.

Figura 25 — Pré-processamento das imagens.

```

# Ajustes no pré-processamento
image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
image_gray = cv2.medianBlur(image_gray, 3)
image_gray = cv2.Canny(image_gray, 50, 150)
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
image_clahe = clahe.apply(image_gray)
image_norm = cv2.normalize(image_clahe, None, 0, 255, cv2.NORM_MINMAX, cv2.CV_8U)
image_norm = (image_norm - np.mean(image_norm)) / np.std(image_norm)
image_norm = cv2.normalize(image_norm, None, 0, 255, cv2.NORM_MINMAX, cv2.CV_8U)
image_processed = cv2.cvtColor(image_norm, cv2.COLOR_GRAY2BGR)

# Data augmentation
data_gen = ImageDataGenerator(
    rotation_range=60,
    width_shift_range=0.5,
    height_shift_range=0.5,
    shear_range=0.4,
    zoom_range=0.5,
    horizontal_flip=True,
    vertical_flip=True,
    brightness_range=[0.5, 1.5],
    channel_shift_range=0.4,
    fill_mode='nearest'
)

```

Fonte: A autora.

O aumento de dados (*data augmentation*) aumenta a diversidade do conjunto de dados, com transformações aleatórias nas imagens, como rotações, translações, *flips* e distorções. Essa técnica é essencial, principalmente, para *datasets* relativamente menores, auxiliando na generalização do modelo e prevenindo o *overfitting*.

3.5.2 Treinamento dos modelos e avaliação

Para o treinamento, foram realizadas algumas etapas. Após carregar os arquivos necessários e configurar o ambiente de execução, incluindo a utilização de GPU para acelerar o processamento, foi realizada a extração do rótulo das galáxias. As amostras com rótulo “incerto” são removidas e as classes restantes são balanceadas para evitar vieses no treinamento. O conjunto de dados é dividido em conjuntos de treinamento e teste, e um modelo CNN é criado com camadas convolucionais, de *pooling*, de normalização e de *dropout* para prevenir o *overfitting*.

Foram feitos 3 algoritmos com três redes neurais convolucionais diferentes, a fim de buscar pelas melhores otimizações e também garantir o melhor desempenho, demonstrando

também problemas comuns de se enfrentar durante essa etapa e como resolvê-los. Ao final, foi realizada uma avaliação, com algumas métricas como matriz de confusão, perdas² e acuráncias nos treinos e validação, além de um relatório contendo informações de precisão, *recall* e *F1-score*. Nas subseções seguintes, foram apresentadas e analisadas as estruturas da rede neural de cada um dos modelos.

3.5.2.1 Modelo A

O modelo A é o modelo mais simples dos três, pensado em ser utilizado como comparação para demonstrar a funcionalidade das otimizações feitas nos outros dois modelos. A estrutura da CNN é um *pipeline* que extrai *features* das imagens de entrada por camadas convolucionais, reduz a dimensionalidade com camadas de *pooling*, aplica normalização e *dropout* para melhorar o treinamento e prevenir *overfitting*, e classifica finalmente a imagem utilizando camadas densas. A combinação dessas camadas permite que a rede aprenda padrões complexos nas imagens e realize a classificação de galáxias. A figura 26 demonstra a estrutura da rede.

Figura 26 — Estrutura do modelo A.

```
def criar_modelo(input_shape):
    model = tf.keras.models.Sequential([
        tf.keras.layers.Conv2D(32, (3, 3), activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001), input_shape=input_shape),
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.MaxPooling2D((2, 2)),
        tf.keras.layers.Dropout(0.3),

        tf.keras.layers.Conv2D(64, (3, 3), activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)),
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.MaxPooling2D((2, 2)),
        tf.keras.layers.Dropout(0.3),

        tf.keras.layers.Conv2D(128, (3, 3), activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)),
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.MaxPooling2D((2, 2)),
        tf.keras.layers.Dropout(0.4),

        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(128, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)),
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.Dropout(0.4),

        tf.keras.layers.Dense(2, activation='softmax', dtype='float32') # Modificado para 2 saídas: Espirais e Elípticas
    ])
    return model
```

Fonte: A autora.

² A perda durante o treinamento de uma CNN (Rede Neural Convolucional) é uma medida de quão errada a rede está em suas previsões. É a diferença entre a previsão da rede e o valor real que ela deveria prever.

A primeira camada convolucional possui 32 filtros, cada um com tamanho 3×3 , utilizando a função de ativação ReLU e também a regularização L2 aos pesos dos filtros, ajudando a prevenir *overfitting*. A segunda e terceira camada seguiram no mesmo padrão, apenas alterando a quantidade de filtros para 64 e 128 respectivamente. Cada camada convolucional há uma camada de max pooling com tamanho 2×2 . Ela é responsável por reduzir a dimensionalidade dos mapas de *features*, diminuindo assim o número de parâmetros e a complexidade computacional, além de aumentar a invariância a pequenas translações na imagem.

Outras camadas presentes são as de normalização, aplicadas após cada camada convolucional e antes da função de ativação. São responsáveis por normalizar as ativações das camadas anteriores, acelerando o treinamento e melhorando a estabilidade do modelo. Há também as camadas de *dropout*, outra medida para prevenir *overfitting*, que desligam aleatoriamente neurônios com uma determinada probabilidade, forçando o modelo a aprender *features* mais robustas. A camada de *flatten* transforma a saída multidimensional da última camada convolucional para um vetor unidimensional para conectá-lo à camada densa. As camadas densas são as últimas camadas da rede, sendo uma com 128 neurônios e função de ativação ReLU, e outra com saída com 2 neurônios, representando ambas as classes. A função de ativação *softmax* produz uma distribuição de probabilidade sobre as classes. O treinamento utilizou também o otimizador Adam.

3.5.2.2 Modelo B

O modelo B, apesar de trazer uma estrutura de rede mais simplificada, traz diversas otimizações que o modelo A não teve, justamente para contornar problemas como *overfitting* e *underfitting*. O modelo utiliza uma arquitetura de Rede Neural Convolutinal (CNN) pré-treinada, ResNet50, como base, aproveitando o conhecimento aprendido em um conjunto de dados massivo (ImageNet) para extraír características relevantes das imagens de galáxias. O código aborda o problema de classes desbalanceadas, onde o número de amostras de cada classe (espiral e elíptica) é desigual, utilizando a técnica de *oversampling*. O *RandomOverSampler* gera novas amostras da classe minoritária, equilibrando a distribuição das classes e evitando vieses no treinamento.

A arquitetura do modelo consiste na ResNet50 como extrator de características, seguida por camadas de *Flatten*, Densas, *BatchNormalization* e *Dropout*. As camadas Densas são responsáveis pela classificação, enquanto as camadas de *BatchNormalization* normalizam as ativações, acelerando o treinamento e melhorando a estabilidade. As camadas de *Dropout*, como mencionado anteriormente, desligam aleatoriamente neurônios durante o treinamento, prevenindo o *overfitting* e forçando o modelo a aprender características mais robustas. A figura 27 traz a implementação dessa rede.

Figura 27 — Estrutura do modelo B.

```
# Early Stopping e reduzir a taxa de aprendizado para evitar overfitting
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=5, min_lr=1e-6)

def criar_modelo(input_shape):
    base_model = tf.keras.applications.ResNet50(
        include_top=False, weights='imagenet', input_shape=input_shape, pooling='avg')

    model = tf.keras.models.Sequential([
        base_model,
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(128, activation='relu', kernel_regularizer=tf.keras.regularizers.l1_l2(0.001)), # L1 e L2
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.Dropout(0.5),

        tf.keras.layers.Dense(64, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)),
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.Dropout(0.5),

        tf.keras.layers.Dense(2, activation='softmax', dtype='float32')
    ])

    return model
```

Fonte: A autora.

O treinamento do modelo é realizado com o otimizador Adam, que ajusta a taxa de aprendizado durante o processo, e a função de perda “*sparse categorical crossentropy*”, adequada para problemas de classificação multiclasse. Para evitar *overfitting* e otimizar o desempenho, o código implementa *callbacks* como *EarlyStopping*, que interrompe o treinamento quando a perda de validação para de diminuir, e *ReduceLROnPlateau*, que reduz a taxa de aprendizado quando o modelo atinge um platô.

3.5.2.3 Modelo C

O modelo C seguiu um caminho diferente dos outros dois modelos. Esse modelo não usou redes prontas como a ResNet50, e sim de outras otimizações, como técnicas de busca Bayesiana. Em paralelo ao processamento das imagens, metadados relevantes são extraídos, incluindo informações sobre a posição (ra, dec), *redshift* (z), magnitude (mag_auto_g,

mag_auto_r, mag_auto_i), massa e probabilidades morfológicas (P_EL, P_CW, P_ACW, P_EDGE). Esses metadados são combinados com dados auxiliares de um conjunto de dados pré-processados, utilizando correspondência de posição para integrar informações adicionais sobre as galáxias.

O conjunto de dados resultante, composto por imagens e metadados, é então balanceado para garantir a representação equitativa das classes (espiral e elíptica). Para isso, o código emprega uma combinação de técnicas de *oversampling* (SMOTE) e *undersampling*, gerando amostras sintéticas da classe minoritária e reduzindo a classe majoritária, respectivamente. A figura 28 traz a implementação da rede.

Figura 28 — Estrutura do modelo C.

```

def criar_modelo(input_shape, num_metadados_imagem, num_metadados_antigos, learning_rate=0.001, dropout_rate=0.5, l2_reg=0.01):
    input_imagem = tf.keras.Input(shape=input_shape)

    def bloco_residual(x, filtros, kernel_size=(3, 3)):
        y = tf.keras.layers.Conv2D(filtros, kernel_size, activation='elu', padding='same', kernel_regularizer=tf.keras.regularizers.l2(l2_reg))(x)
        y = tf.keras.layers.BatchNormalization()(y)
        y = tf.keras.layers.Conv2D(filtros, kernel_size, activation=None, padding='same', kernel_regularizer=tf.keras.regularizers.l2(l2_reg))(y)
        y = tf.keras.layers.BatchNormalization()(y)
        y = tf.keras.layers.Add()([x, y])
        y = tf.keras.layers.Activation("relu")(y)
        return y

    x = tf.keras.layers.Conv2D(32, (3, 3), activation='elu', kernel_regularizer=tf.keras.regularizers.l2(l2_reg))(input_imagem)
    x = tf.keras.layers.BatchNormalization()(x)
    x = bloco_residual(x, 32)
    x = tf.keras.layers.MaxPooling2D((2, 2))(x)
    x = tf.keras.layers.Dropout(dropout_rate)(x)

    x = tf.keras.layers.Conv2D(64, (3, 3), activation='elu', kernel_regularizer=tf.keras.regularizers.l2(l2_reg))(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = bloco_residual(x, 64)
    x = tf.keras.layers.MaxPooling2D((2, 2))(x)
    x = tf.keras.layers.Dropout(dropout_rate)(x)

    x = tf.keras.layers.Conv2D(128, (3, 3), activation='relu', kernel_regularizer=tf.keras.regularizers.l2(l2_reg))(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = bloco_residual(x, 128)
    x = tf.keras.layers.MaxPooling2D((2, 2))(x)
    x = tf.keras.layers.Dropout(dropout_rate)(x)

    x = tf.keras.layers.Conv2D(256, (3, 3), activation='relu', kernel_regularizer=tf.keras.regularizers.l2(l2_reg))(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = bloco_residual(x, 256)
    x = tf.keras.layers.MaxPooling2D((2, 2))(x)
    x = tf.keras.layers.Dropout(dropout_rate)(x)
    x = tf.keras.layers.Flatten()(x)

    input_metadados_imagem = tf.keras.Input(shape=(num_metadados_imagem,))
    input_metadados_antigos = tf.keras.Input(shape=(num_metadados_antigos,))

    x = tf.keras.layers.concatenate([x, input_metadados_imagem, input_metadados_antigos])

    x = tf.keras.layers.Dense(1024, activation=LeakyReLU(), kernel_regularizer=tf.keras.regularizers.l2(0.03))(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.Dropout(0.7)(x)

    output = tf.keras.layers.Dense(2, activation='softmax', dtype='float32')(x)

    # --- Opções de otimizadores ---
    initial_learning_rate = learning_rate # Usar Learning_rate do argumento
    lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(
        initial_learning_rate,
        decay_steps=10000,
        decay_rate=0.96,
        staircase=True)

    optimizer = tf.keras.optimizers.SGD(learning_rate=lr_schedule, momentum=0.9, nesterov=True) # SGD com momentum e Nesterov momentum
    # optimizer = AdamW(learning_rate=lr_schedule, weight_decay=1e-4) # AdamW com weight decay

    model = tf.keras.Model(inputs=[input_imagem, input_metadados_imagem, input_metadados_antigos], outputs=output)
    model.compile(optimizer=optimizer, loss='sparse_categorical_crossentropy', metrics=['accuracy'])

    return model

```

Fonte: A autora.

A arquitetura da CNN foi construída com blocos residuais, que permitem o fluxo eficiente de gradientes durante o treinamento, camadas de convolução para extrair características, camadas de pooling para reduzir a dimensionalidade, camadas de normalização em lote (*BatchNormalization*) para acelerar o treinamento e camadas de *dropout* para prevenir *overfitting*. Adicionalmente, a rede incorpora os metadados como entrada, concatenando-os às características extraídas das imagens.

A otimização dos hiperparâmetros do modelo, como taxa de aprendizado, tamanho do *batch*, número de épocas, regularização L2 e taxa de *dropout*, é realizada por meio de uma busca Bayesiana. Essa técnica explora o espaço de hiperparâmetros eficientemente, encontrando a combinação ideal que minimiza a perda na validação. O treinamento do modelo utiliza o otimizador SGD com *momentum* e *Nesterov momentum*, que acelera a convergência e melhora a estabilidade. *Callbacks* como *EarlyStopping* e *ReduceLROnPlateau* são empregados para evitar o *overfitting* e ajustar a taxa de aprendizado dinamicamente durante o processo.

3.5.3 Mineração de Dados

Com os modelos treinados na fase anterior, foi realizada a classificação e análise das galáxias. Nessa etapa, foi criado o pipeline que inclui: carregamento dos modelos, previsão de classes, previsão de brilho, detecção de anomalias e análise de clusters, bem como visualização das imagens das galáxias. Foram carregadas as imagens anteriormente processadas, redimensionando-as para o tamanho adequado para a previsão da classe com os modelos, exibindo algumas imagens de exemplo com os rótulos verdadeiros e os previstos.

Outra técnica utilizada nas imagens é o método de suavização exponencial para prever o brilho das galáxias ao longo do tempo, permitindo identificar tendências e padrões nos dados de brilho. Para detectar as anomalias nos dados, é utilizado o algoritmo *Isolation Forest* para identificar galáxias com brilho anômalo, detectando *outliers* em conjuntos de dados mais complexos. Para criar clusters com galáxias semelhantes em relação às suas características, como tamanho e brilho, foi usado os algoritmos *KMeans* e *DBSCAN* para realizar o agrupamento.

3.5.4 Visualização e discussão de resultados

Todos os gráficos, imagens e outras informações importantes realizadas nas etapas anteriores são descritas e analisadas na seção 4. O objetivo é obter resultados que sejam fáceis tanto de visualizar quanto de interpretar.

3.6 ORÇAMENTO

Não há orçamento definido para este projeto, sendo projetado para utilizar recursos próprios, ferramentas, plataformas e/ou softwares gratuitos.

4 RESULTADOS OBTIDOS

Nessa seção serão apresentados e discutidos os resultados dos três modelos. Os resultados são demonstrados em detalhes para cada modelo, incluindo métricas de desempenho, visualizações e análises comparativas. Também foram mostrados os resultados da etapa de mineração de dados em relação aos conjuntos de dados utilizados.

4.1 MODELO A

Dentre os modelos desenvolvidos, o modelo A é o mais genérico, justamente para demonstrar, através dos outros modelos, a importância de buscar otimizações. O desempenho do modelo foi avaliado com base em métricas relevantes para problemas de classificação, como acurácia, precisão, *recall* e *F1-score*. Além disso, foram fornecidas visualizações, como gráfico de perda e acurácia durante o treinamento e validação, além da matriz confusão, que permitam uma análise mais detalhada do comportamento do modelo. A figura 29 a seguir traz o relatório com as métricas.

Figura 29 — Métricas de avaliação do treinamento do modelo A.

2024-10-24 10:43:57,514 - INFO - Treinamento finalizado				
27/27 [=====] - 1s 30ms/step				
	precision	recall	f1-score	support
Espiral	0.66	0.99	0.79	435
Elíptica	0.98	0.45	0.62	407
accuracy			0.73	842
macro avg	0.82	0.72	0.70	842
weighted avg	0.82	0.73	0.71	842

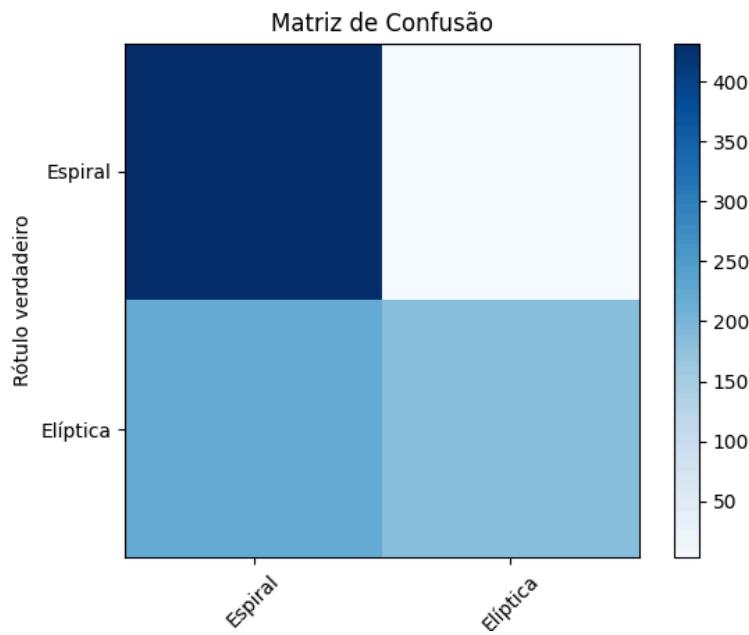
Fonte: A autora.

Começando com a métrica de precisão, que foca em não ter falsos positivos, as galáxias classificadas pelo modelo A como espirais, 66% realmente eram espirais. Já nas elípticas, o modelo classificou 98% que realmente eram elípticas. Durante a revocação (*recall*), que foca em não ter falsos negativos, de todas as galáxias espirais reais, o modelo conseguiu identificar corretamente 99%, enquanto para as elípticas ele identificou corretamente apenas 45%. O modelo se mostrou excelente em encontrar galáxias espirais, mas perdeu muitas galáxias elípticas.

Na *F1-Score*, que traz a média harmônica entre precisão e *recall*, fornecendo uma medida geral do desempenho do modelo, retornou as médias para a classificação de espirais e elípticas: 79% e 62% respectivamente. A diferença significativa entre os *F1-scores* das duas classes reforça a ideia de que o modelo tem um viés para classificar as galáxias como espirais. Isso pode ser causado por um desbalanceamento no conjunto de dados ou por características mais proeminentes nas espirais, que facilitam sua identificação. Isso é visível pela métrica de Suporte, que mostrou haver mais galáxias espirais na amostra (435) que elípticas (407).

A acurácia geral que o modelo retornou a classificação correta de 73%. Como os valores de *macro avg* e *weighted avg* são bem próximos, indica que o desbalanceamento entre as classes não é muito grande. As diferenças, embora pequenas, mostram que o modelo tende a ter um desempenho um pouco melhor nas galáxias espirais, sendo mais numerosas no conjunto de dados. Outra análise importante é a matriz de confusão, a figura 30 traz mais claramente o desempenho do modelo A.

Figura 30 — Matriz de Confusão do modelo A.

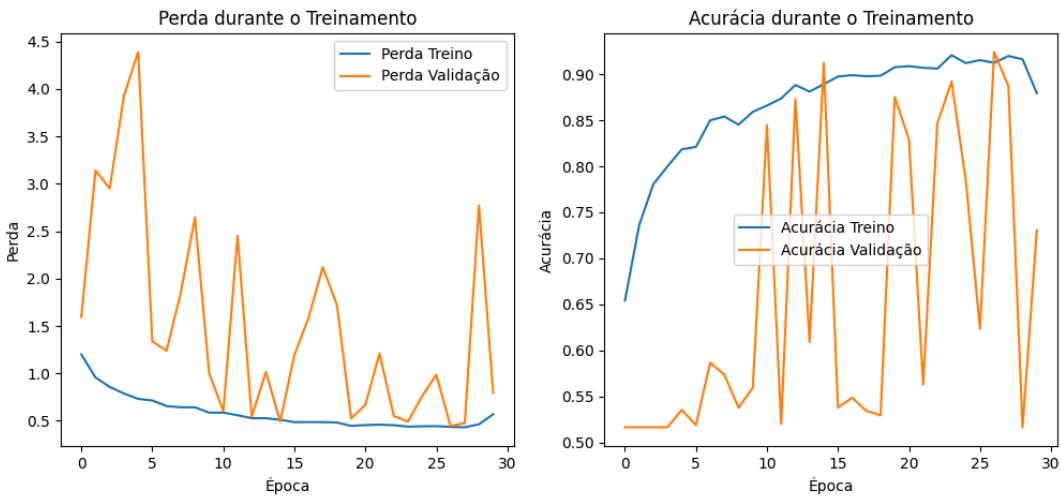


Fonte: A autora.

O quadrado superior esquerdo, sendo os verdadeiros positivos, representa as galáxias que são realmente espirais, sendo classificadas corretamente como espirais pelo modelo. Isso indica, como mencionado anteriormente, que o modelo é bom em identificar galáxias espirais. O quadrado superior direito, os falsos positivos, indica que não houve casos em que o modelo

classificou erroneamente uma galáxia elíptica como espiral. O quadrado inferior esquerdo, sendo os falsos negativos, representa as galáxias que são realmente elípticas, mas foram classificadas incorretamente como espirais. Esse valor é alto, indicando que o modelo frequentemente classifica erroneamente galáxias elípticas como espirais. O quadrado inferior direito, os verdadeiros negativos, representa as galáxias que são realmente elípticas, sendo classificadas corretamente como elípticas. A matriz confirma as observações feitas com as métricas de precisão e *recall*, afirmando que o modelo tem mais facilidade para espirais que elípticas. A figura 31 abaixo traz os gráficos de acurácia e perda durante o treinamento e validação do modelo.

Figura 31 — Gráficos de perda e acurácia durante o treinamento do modelo A.



Fonte: A autora.

O gráfico de perda mostra que no início do treinamento, ambas as perdas são altas, mas diminuem rapidamente. A perda de treino diminui de forma mais suave e consistente, enquanto a perda de validação apresenta algumas oscilações e picos. As oscilações na perda de validação podem indicar que o modelo está começando o *overfitting*, ou seja, está se ajustando muito aos dados de treinamento e não generalizando bem para novos dados, que é melhor evidenciado no gráfico de acurácia. A acurácia de treino aumenta consistentemente ao longo do treinamento, chegando a valores próximos de 100%. A acurácia de validação também aumenta, porém, com mais oscilações e em um ritmo mais lento, confirmando o *overfitting*.

Apesar do modelo ter aprendido sobre as galáxias, os gráficos mostram que ele estava se ajustando muito aos dados de treinamento e não generalizando bem para os novos dados.

Isso exemplifica a importância de adotar diferentes medidas para otimizar os modelos, visto nos modelos B e C.

4.2 MODELO B

O modelo B traz diversas melhorias em relação ao modelo A previamente discutido, como soluções melhores de aumento de dados, regularizações maiores com *dropout* e L1/L2, e com uso de rede pronta como base, a *ResNet50*, além de medidas para reduzir taxa de aprendizado dinamicamente e funções de parada. Consequentemente, o modelo B teve resultados melhores que o modelo anterior, ilustrados pelos relatórios finais na figura 32. O valor 0 corresponde a galáxias espirais quanto o valor 1 para galáxias elípticas.

Figura 32 — Métricas de avaliação do modelo B.

	<code>precision</code>	<code>recall</code>	<code>f1-score</code>	<code>support</code>
0	1.00	0.96	0.98	435
1	0.96	1.00	0.98	407
<code>accuracy</code>			0.98	842
<code>macro avg</code>	0.98	0.98	0.98	842
<code>weighted avg</code>	0.98	0.98	0.98	842

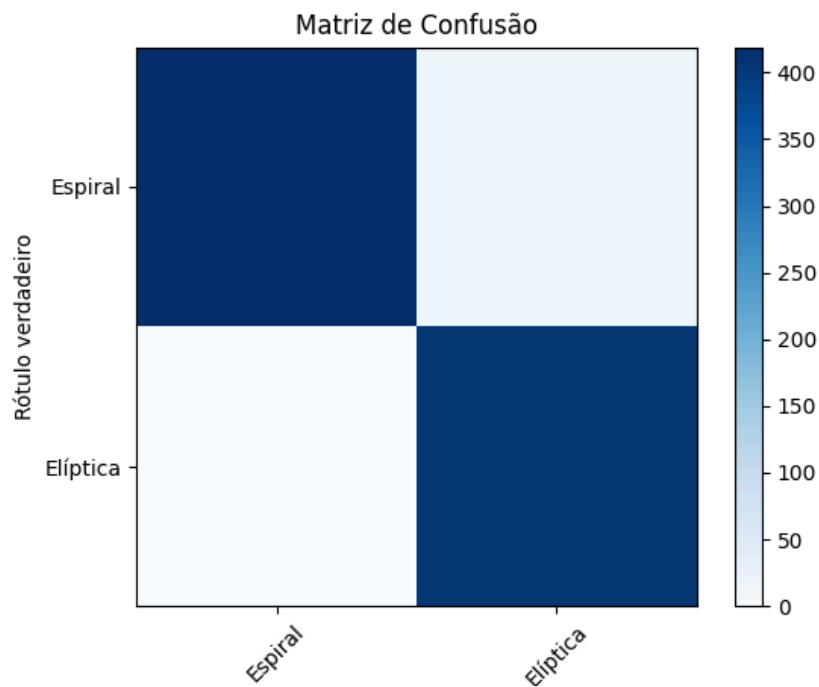
Fonte: A autora

A métrica de precisão traz que, de todas as galáxias classificadas como espirais, 100% realmente eram da classe espiral, enquanto para a classe elíptica, 96% realmente eram elípticas. Isso significa que o modelo foi extremamente preciso na classificação, com uma taxa de erro muito baixa (4%). Na métrica de *recall*, de todas as galáxias que realmente eram da classe espiral, o modelo identificou corretamente 96%. Para as galáxias elípticas, o modelo identificou que 100% eram realmente da classe elíptica. Com isso, o modelo também demonstra um excelente *recall*.

Em relação ao *F1-Score*, as classes espiral e elíptica ambas receberam um média de 98% entre precisão e recall. O Suporte, assim como nos resultados anteriores no modelo A, retorna o número de galáxias usadas para cada classe na amostra, sendo 435 para espirais e 407 para elípticas. No geral, o modelo classificou corretamente 98% de todas as galáxias utilizadas, e as medidas de *macro avg* e *weighted avg*, responsáveis por calcular a média das

métricas para cada classe e o número de amostra em cada classes, respectivamente, confirmam o bom desempenho do modelo. A figura 33 traz a matriz de confusão do modelo B.

Figura 33 — Matriz de Confusão do modelo B.



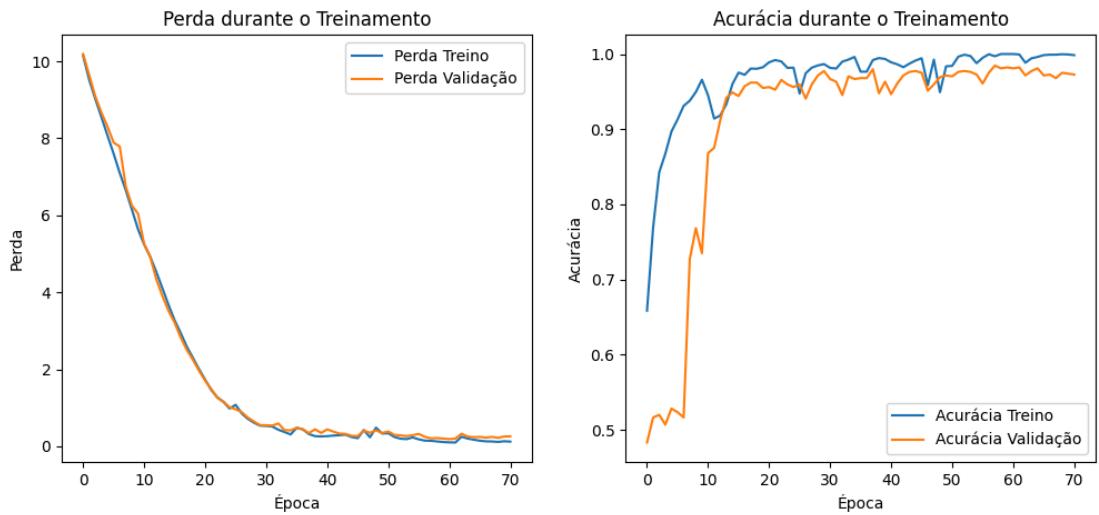
Fonte: A autora.

O quadro superior esquerdo (verdadeiros positivos) representa as galáxias que são realmente espirais, tendo sido classificadas corretamente como espirais. No quadrado inferior direito, representa as galáxias que são realmente elípticas, sendo classificadas corretamente como elípticas. Para ambos os casos, o modelo desempenhou muito bem na identificação de cada tipo de galáxias. Em relação aos falsos positivos (quadro superior direito), o quadrado está levemente azulado, indicando que mesmo com um bom desempenho geral, o modelo pode ter classificado galáxias elípticas erroneamente como espirais. Com falsos negativos (quadro inferior esquerdo), indica que não houve casos em que o modelo classificou erroneamente as galáxias espirais como elípticas.

Mesmo com essa pequena quantidade de falsos positivos, o modelo ainda apresenta um desempenho excelente. Comparando com a matriz de confusão anterior, do modelo A, podemos observar uma grande melhora. No modelo anterior, havia um número significativo de falsos negativos, indicando que o modelo classificou erroneamente galáxias elípticas como

espirais. Neste modelo, essa falha foi completamente corrigida. Abaixo, na figura 34, estão os gráficos de perda e acurácia durante o treinamento e validação do modelo.

Figura 34 — Gráficos de perda e acurácia durante o treinamento do modelo B.



Fonte: A autora.

No gráfico de perda, em ambas as partes (treino e validação), diminuem rapidamente nas primeiras épocas. Após cerca de 20 épocas, a perda de treino continua diminuindo, mas a perda de validação se estabiliza e até apresenta pequenas oscilações. Essa diferença entre as curvas, principalmente após 20 épocas, indicou ainda um leve *overfitting*, ou seja, o modelo está se ajustando muito aos dados de treinamento e não generalizando tão bem para novos dados. Isso acontece, pois, mesmo com boas medidas para evitar o *overfitting*, o conjunto de dados não é muito grande, mesmo com aumento de dados, ocasionando ainda esse tipo de problema.

Com o gráfico de acurácia, há algumas oscilações. A acurácia de treino aumenta rapidamente e se estabiliza em um valor alto, próximo de 100%. A acurácia de validação também aumenta, mas com um ritmo mais lento e algumas oscilações, sem atingir o mesmo nível da acurácia de treino. Isso reforçou a suspeita de *overfitting*, que, como mencionado anteriormente, pode ser referente ao conjunto de dados utilizado na análise. Entretanto, é possível afirmar que o modelo B é que apresentou bons resultados, e isso se dá com a facilidade de uma rede pronta implementada como base, assim como as outras otimizações.

4.3 MODELO C

O modelo C é a rede convolucional mais diferente dos outros dois exemplos anteriores. O modelo não recorreu a uma rede pronta, como o modelo B utilizou, tendo então uma arquitetura mais complexa que os outros dois modelos. Ele inclui mais camadas e também blocos residuais, bem como medidas mais reforçadas para evitar o *overfitting*, como regularizadores, funções de parada e uma função para otimizar hiperparâmetros (busca Bayesiana, mencionada na seção 3.5.2.4). Outra novidade foi um treinamento híbrido entre imagens de galáxias e também metadados relacionados às imagens, a fim de melhorar o treinamento. A imagem 35 traz o relatório final do treinamento com as métricas para análise.

Figura 35 — Métricas de avaliação do modelo C.

	<code>precision</code>	<code>recall</code>	<code>f1-score</code>	<code>support</code>
<code>Espiral</code>	<code>0.96</code>	<code>0.95</code>	<code>0.95</code>	<code>352</code>
<code>Elíptica</code>	<code>0.94</code>	<code>0.95</code>	<code>0.95</code>	<code>321</code>
<code>accuracy</code>			<code>0.95</code>	<code>673</code>
<code>macro avg</code>	<code>0.95</code>	<code>0.95</code>	<code>0.95</code>	<code>673</code>
<code>weighted avg</code>	<code>0.95</code>	<code>0.95</code>	<code>0.95</code>	<code>673</code>

Fonte: A autora.

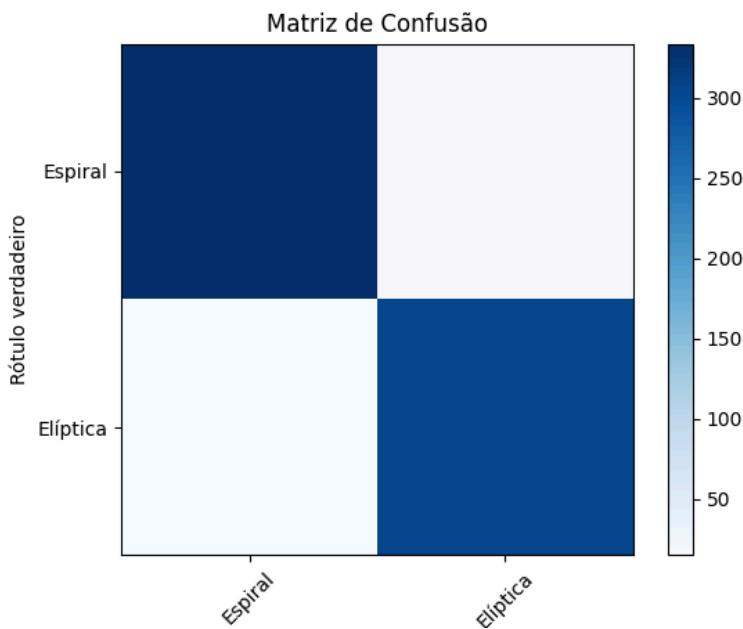
A precisão para galáxias espirais é de 0.96, o que significa que 96% das galáxias classificadas como espirais realmente são espirais. O recall para espirais é de 0.95, indicando que o modelo consegue identificar corretamente 95% de todas as galáxias espirais presentes no conjunto de dados. O *F1-score*, que traz a média da precisão e recall, é de 95% para espirais, confirmando o bom desempenho nessa classe. Já para as galáxias elípticas, 94% das galáxias classificadas como elípticas realmente são elípticas. O recall para elípticas também é de 0.95, indicando que o modelo identifica corretamente 95% de todas as galáxias elípticas presentes no conjunto de dados. O F1-score para elípticas é de 95%, mostrando um desempenho similar ao das espirais.

A acurácia do modelo também ficou com 95%, indicando um desempenho equilibrado e ótimo entre as classes, confirmado pelos valores de macro-avg e *weighted avg*. O suporte indicou uma pequena diferença em relação ao suporte dos outros dois modelos, apontando que a amostra utilizou 352 galáxias espirais e 321 galáxias elípticas. Isso se dá ao fato que o

modelo também utilizou outros dados relacionados às galáxias, como os metadados retirados dos *datasets* escolhidos.

Os resultados indicam que o modelo aprendeu efetivamente a distinguir as características que diferenciam esses dois tipos de galáxias, com uma taxa de erro muito baixa. Apesar de não ter atingido a mesma precisão que o modelo B, o modelo C se manteve com um alto desempenho. A figura 36 traz a matriz de confusão do modelo.

Figura 36 — Matriz de Confusão do modelo C.



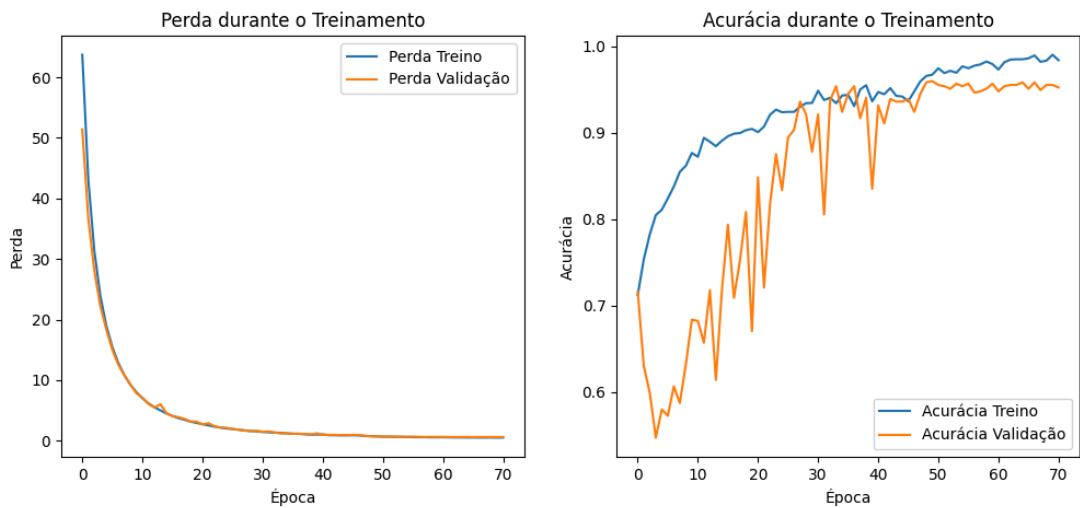
Fonte: A autora.

A matriz de confusão confirma o bom desempenho previsto pelas métricas e também pela diagonal principal da figura 37. O quadrado superior esquerdo, que representa os verdadeiros positivos para galáxias espirais, mostra que o modelo classificou corretamente a grande maioria das galáxias que realmente são espirais. Da mesma forma, o quadrado inferior direito, que representa os verdadeiros positivos para galáxias elípticas, também indica que o modelo acertou na classificação da maioria das galáxias elípticas.

Comparando com as matrizes de confusão anteriores, é possível notar uma evolução no desempenho dos modelos. O Modelo A apresentava um número considerável de falsos negativos, indicando que ele classificou erroneamente muitas galáxias elípticas como espirais. Essa falha foi corrigida no Modelo B e se mantém ausente nesta matriz de confusão. Essa comparação reforça a importância do processo de otimização do modelo e do conjunto de

dados. A figura 37 a seguir, mostra o desempenho geral do modelo durante o treinamento e validação, com os gráficos de perda e acurácia.

Figura 37 — Gráficos de perda e acurácia durante o treinamento do modelo C.



Fonte: A autora.

No gráfico de perda, é possível notar que tanto durante o treinamento quanto durante a validação, ambos começaram com valores altos e caíram drasticamente nas primeiras *épocas*, se aproximando de zero rapidamente. Isso indica que o modelo aprendeu com facilidade e rapidez a classificar as galáxias do conjunto de treinamento. Contudo, há ainda pequenas oscilações na perda de validação, o que indica que, mesmo com um treinamento híbrido, o modelo ainda estava com um pouco de *overfitting*, sendo mais evidenciado no segundo gráfico, de acurácia.

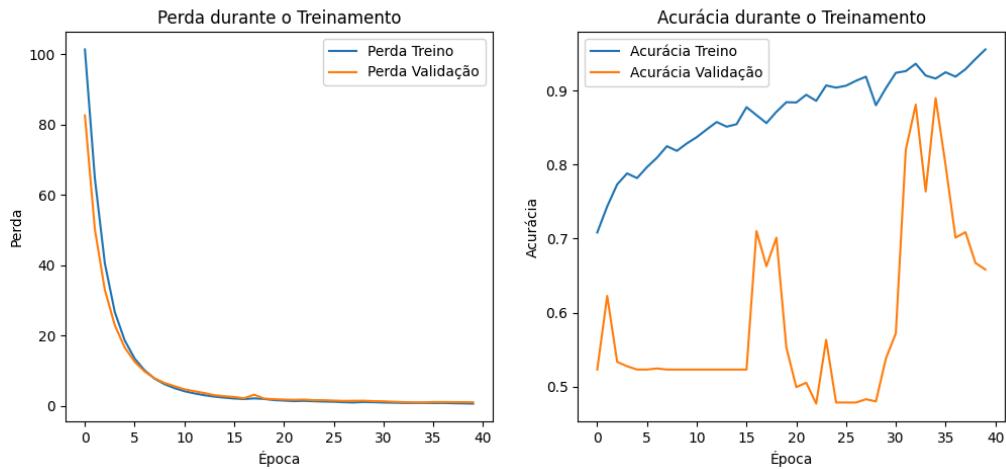
O gráfico de acurácia mostra que o treinamento rapidamente atingiu valores próximos de 100%, mostrando que o modelo se tornou ótimo em classificar as galáxias que ele já conhecia. Contudo, na validação, ela também aumentou, porém, com mais oscilações e em ritmo mais lento. Isso reforça a suspeita de *overfitting*, já que o modelo tem um desempenho muito melhor nos dados que ele “decorou” durante o treino do que nos dados novos. Como no modelo anterior (B), apesar das medidas tomadas para contornar esse problema, ele não sumiu 100%, apontando necessário um conjunto de dados maior para obter um desempenho ainda melhor.

Comparando com os gráficos anteriores, podemos notar algumas diferenças interessantes. Em relação ao gráfico do Modelo C, a perda neste gráfico diminui de forma

mais abrupta nas primeiras épocas, tanto para o treino quanto para a validação, indicando um aprendizado mais rápido. No entanto, a diferença entre a perda do treino e da validação ainda persiste, sugerindo *overfitting*, assim como no Modelo B. Olhando para a acurácia, este modelo atinge valores mais altos no treino do que o Modelo A e B, mostrando um aprendizado mais eficaz com os dados de treino. Mas, assim como no Modelo B, a acurácia da validação fica abaixo da acurácia do treino e apresenta oscilações, reforçando a possibilidade de *overfitting*.

Outros testes foram feitos com outras otimizações, como o Optuna, entretanto nenhuma retornou um melhor desempenho comparada com a busca Bayesiana. Optuna é uma biblioteca para otimização de hiperparâmetros. Ele define esse espaço de forma flexível, permitindo que seja especificado diferentes tipos de hiperparâmetros (inteiros, números de ponto flutuante, categóricos) e seus intervalos. Apesar de ser mais flexível e eficiente que a busca Bayesiana em muitos casos, neste teste, como a figura 38 abaixo ilustra, não teve bons resultados para a classificação de galáxias.

Figura 38 — Gráfico de perda e acurácia durante o treinamento e validação do modelo C, usando Optuna.



Fonte: A autora.

É possível observar que, assim como no modelo otimizado pela busca Bayesiana, houve uma queda abrupta logo nas épocas iniciais, tanto durante o treinamento quanto durante a validação, sugerindo uma rápida convergência do modelo. Entretanto, a linha de validação apresenta pequenas oscilações, que, assim como no treinamento anterior e no modelo B, podem indicar um *overfitting*. A análise da acurácia confirma essa indicação. A acurácia no treino aumenta consistentemente ao longo das épocas, atingindo valores próximos a 95%. Em

contraste, a acurácia na validação demonstra grande instabilidade, com picos e vales pronunciados.

As oscilações abruptas e a falta de uma tendência clara de aumento sugerem que o modelo não estava conseguindo generalizar o aprendizado para os dados de validação. Esse fenômeno foi observado também no treinamento com a otimização Bayesiana, porém de forma menos abrupta. O uso do Optuna para otimização de hiperparâmetros pode ter contribuído para esse problema, ao selecionar valores que maximizam o desempenho no treino, mas não necessariamente a capacidade de generalização. Com isso, foi possível concluir que a busca Bayesiana, apesar de menos flexível que o Optuna, é mais indicada para esse cenário.

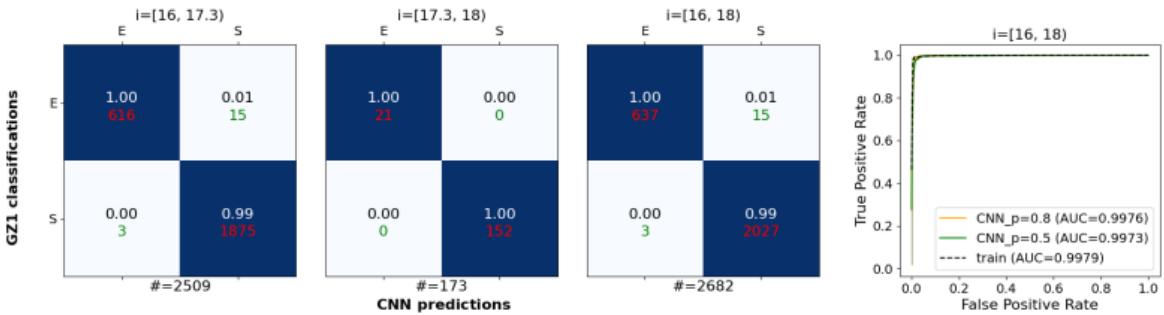
4.4 COMPARAÇÃO DOS RESULTADOS COM DE OUTROS AUTORES

Os resultados dos modelos A, B e C são comparáveis a estudos de outros autores, utilizados como base do desenvolvimento deste trabalho. Em uma análise comparativa entre diferentes modelos de classificação de galáxias, foi investigado o desempenho dos três modelos distintos desenvolvidos (A, B e C) em relação ao modelo desenvolvido por Cheng et al. (2021).

Ambos os estudos utilizaram o mesmo conjunto de dados, porém, com abordagens metodológicas distintas. Cheng et al. (2021) empregaram uma técnica de “votação” baseada em características morfológicas das galáxias, como assimetria e concentração de luz, combinada com um método de “*debiasing*” para corrigir vieses na classificação.

Em contraste, os modelos A, B e C do presente estudo foram construídos utilizando Redes Neurais Convolucionais (CNNs), com diferentes arquiteturas e hiperparâmetros. A análise comparativa das matrizes de confusão revelou similaridades e diferenças entre os modelos. A figura 39 traz as matrizes de confusão realizadas por Cheng et al. (2021).

Figura 39 — Gráfico combinado das matrizes de confusão e da curva ROC compara as previsões da nossa CNN com os rótulos do Galaxy Zoo 1 (GZ1), definidos pelos votos corrigidos para viés com um limite de 0,8.



Fonte: Cheng et al., 2021.

Tanto os modelos B (figura 34) e C (figura 37) quanto o modelo produzido por Cheng et al. (2021) demonstraram alta precisão na classificação de galáxias espirais e elípticas, com a maioria das galáxias classificadas corretamente. Observou-se, em ambos os estudos, uma tendência a classificar erroneamente galáxias espirais como elípticas, sugerindo que a distinção entre esses dois tipos morfológicos pode ser desafiadora em certos casos. O modelo dos autores exibiu um desempenho consistente em diferentes faixas de magnitude, indicando uma boa capacidade de generalização.

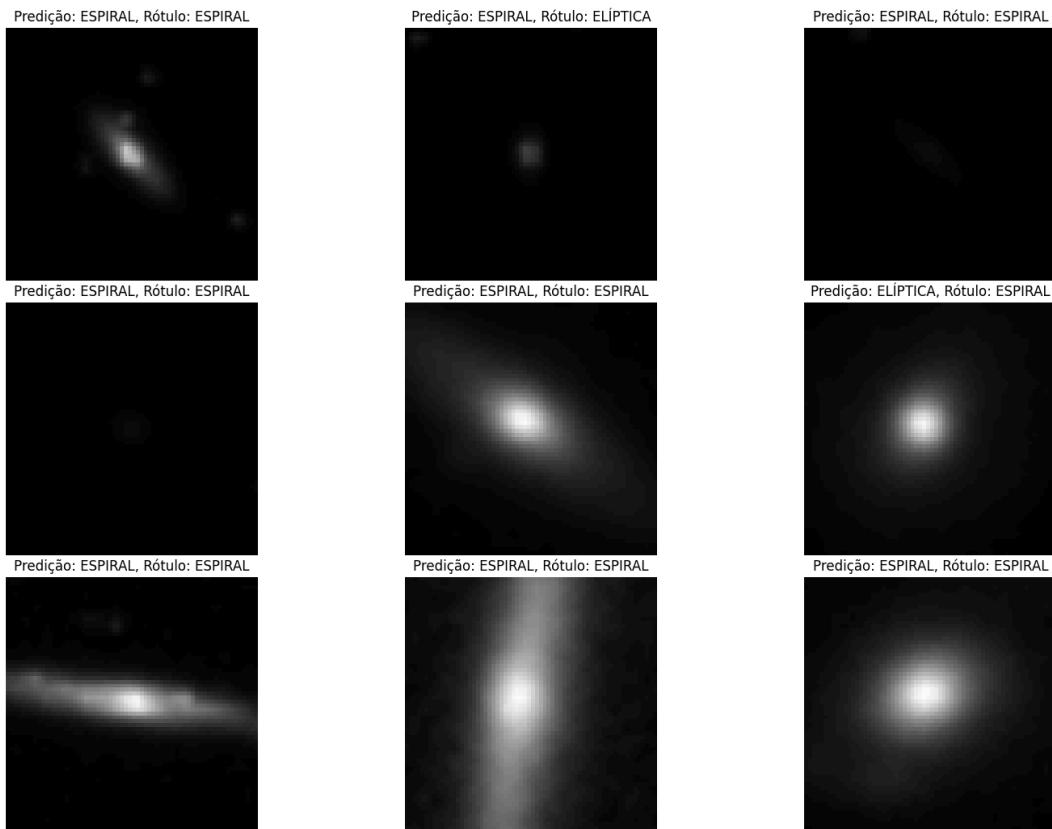
Comparando o desempenho geral dos modelos, o modelo A (figura 30) apresentou o menor desempenho, especialmente na classificação de galáxias elípticas. Os Modelos B e C, por sua vez, atingiram um desempenho comparável ao modelo dos autores, com alta precisão e poucos erros de classificação. As diferenças na distribuição dos erros podem ser atribuídas às distintas abordagens metodológicas. A análise comparativa permite contextualizar os resultados do presente estudo e validar a eficácia das CNNs na classificação de galáxias. As similaridades no desempenho entre os modelos indicam que ambos os estudos conseguiram capturar características morfológicas relevantes para a distinção entre espirais e elípticas.

4.5 RESULTADOS DA MINERAÇÃO DE DADOS

Como mencionado na seção 3.5.3 e 3.5.4, foi criado um pipeline que inclui a previsão de classes, previsão de brilho das galáxias ao longo do tempo, detecção de anomalias. Com isso, foram gerados gráficos e outras visualizações para demonstrar esse processo. O modelo usado de exemplo para criar essas observações foi o modelo B, visto que foi o modelo com melhor estabilidade no treinamento, e também com maior acurácia. A figura 40 traz a

previsão das classes, com rótulos reais e as imagens das respectivas galáxias. As imagens foram escolhidas aleatórias da amostra.

Figura 40 — Galáxias previstas e seus rótulos reais.



Fonte: A autora.

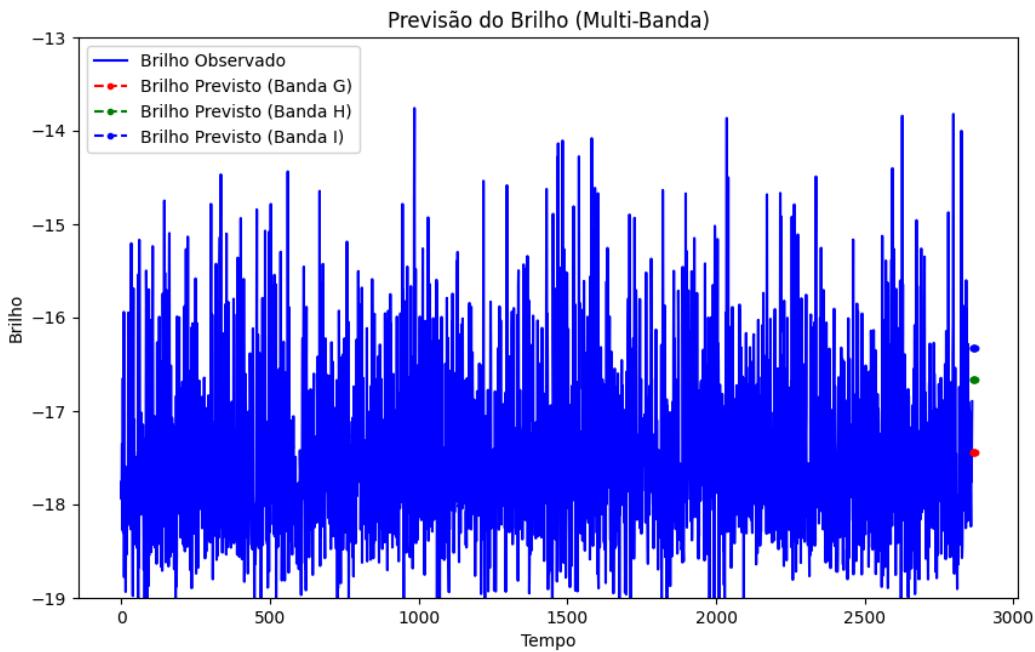
É possível notar que a maioria das galáxias foi classificada corretamente pelo modelo, com exceção da segunda imagem da primeira linha, que foi classificada com espiral, sendo uma elíptica. Esse tipo de erro é classificado como um falso positivo. Isso é consistente com a análise feita anteriormente das métricas do Modelo B, onde o recall para elípticas era excelente (1.00), mas a precisão era um pouco menor (0.96). O recall perfeito significa que o modelo não deixou de identificar nenhuma galáxia elíptica (não houve falsos negativos), mas a precisão menor indica que, entre as galáxias que ele classificou como elípticas, algumas, na verdade, não eram.

Analizando visualmente as imagens, é possível compreender por que o modelo cometeu esses erros. As galáxias elípticas classificadas erroneamente como espirais podem ter alguma característica que as torna visualmente semelhantes às espirais, como uma forma

alongada ou a presença de alguma estrutura que lembra os braços de uma espiral. Essa análise visual ajuda a entender as limitações do modelo e a pensar em estratégias para melhorá-lo.

A figura 41 a seguir mostra um gráfico da previsão de brilho das galáxias. O gráfico mostra a previsão do brilho de galáxias em diferentes bandas espectrais (G, H e I). No eixo horizontal, representado pelo tempo, observam-se cerca de 3000 amostras, correspondendo a diferentes momentos de observação ou à sequência de dados analisados. No eixo vertical, o brilho das galáxias é representado em uma escala logarítmica inversa, característica da astronomia, onde valores menores indicam maior intensidade luminosa.

Figura 41 — Gráfico da previsão do brilho.

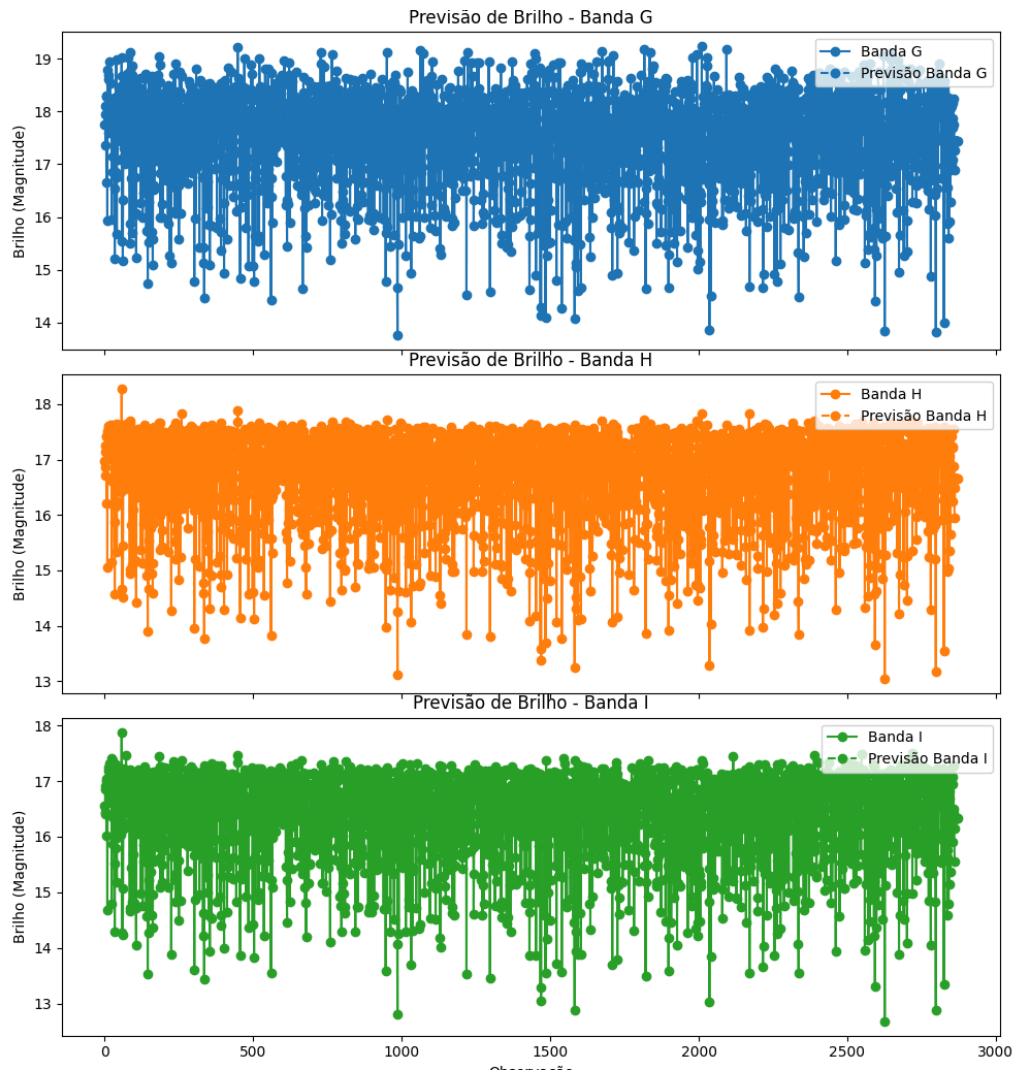


Fonte: A autora.

A linha azul sólida no gráfico reflete o brilho observado, ou seja, os valores reais registrados nas observações astronômicas. Ela apresenta uma alta variabilidade, sendo esperada devido à diversidade de características luminosas das galáxias. Sobreposta a essa linha, aparecem as linhas tracejadas que representam as previsões do brilho feitas pelo modelo B para cada banda espectral. A linha vermelha, que corresponde à banda G, mostra um padrão de previsão que frequentemente subestima o brilho observado, o que indica que o modelo apresenta maior dificuldade em prever com precisão essa banda específica. Já as linhas verde e azul, que representam as bandas H e I, respectivamente, seguem mais de perto o

comportamento do brilho observado, sugerindo que as previsões para essas bandas são mais precisas, ainda que algumas discrepâncias permaneçam. A figura 42 traz as bandas separadas, para melhor visualização.

Figura 42 — Gráficos das bandas G, H e I.



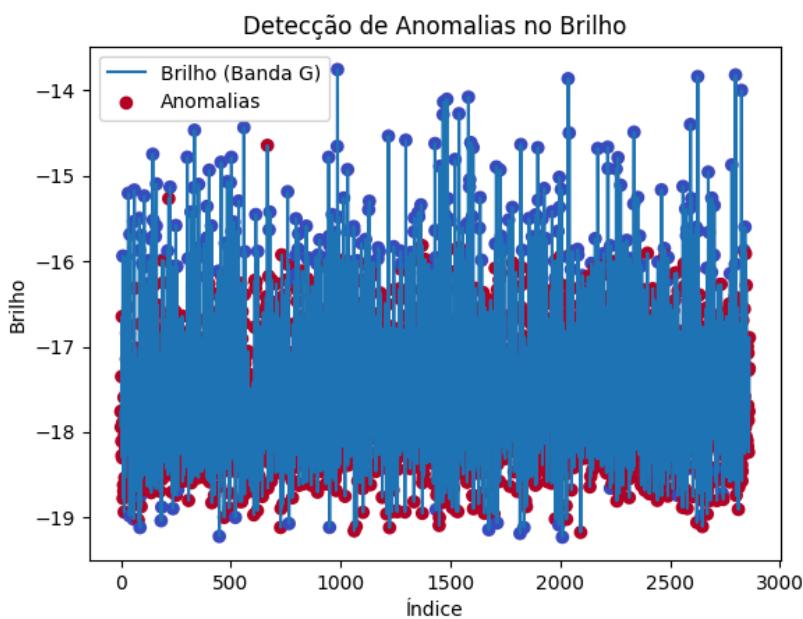
Fonte: A autora.

É possível observar, de forma geral, uma relativa estabilidade no brilho ao longo do tempo para todas as bandas, com pequenas flutuações em torno de um valor médio, sugerindo que as galáxias em questão podem estar em uma fase evolutiva quiescente, sem grandes eventos de formação estelar ou atividade nuclear. No entanto, as magnitudes médias variam entre as bandas, indicando que as galáxias emitem diferentes quantidades de luz em cada faixa do espectro eletromagnético. A banda G, por exemplo, apresenta a maior magnitude (menor brilho), seguida pelas bandas H e I, respectivamente. Essa diferença pode estar relacionada à

distribuição espectral de energia das estrelas que compõem cada galáxia, bem como à presença de poeira interestelar que pode absorver a luz em determinadas faixas do espectro.

No contexto astronômico, a análise de brilho em múltiplas bandas espetrais é crucial para a classificação de galáxias, já que galáxias espirais e elípticas exibem diferentes padrões de luminosidade que as CNNs podem explorar para distinguir entre os tipos. As galáxias elípticas, por exemplo, tendem a ser mais homogêneas em brilho, enquanto as espirais possuem uma distribuição mais variada devido às suas estruturas complexas. A discrepância notada na banda G pode indicar que essa banda está mais associada a galáxias elípticas. A figura 43 a seguir mostra a detecção de anomalias ainda relacionadas ao brilho.

Figura 43 — Detecção de anomalias no brilho das galáxias.



Fonte: A autora.

O gráfico analisa o brilho das galáxias na banda G, que se refere a um filtro específico usado para observar a luz em uma determinada faixa de comprimentos de onda. É possível notar que a maioria das galáxias tem brilho em torno de -16 magnitudes na Banda G, formando um agrupamento denso de pontos azuis. No entanto, algumas galáxias se destacam com um brilho significativamente diferente, tanto para mais quanto para menos, sendo marcadas como anomalias pelos pontos vermelhos. Essas anomalias podem ser causadas por diversos fatores, como a presença de estrelas muito brilhantes na galáxia, a atividade do núcleo galáctico, ou até mesmo erros de medição.

É importante notar que a detecção de anomalias depende dos critérios utilizados. O modelo B, treinado para classificar galáxias, pode ter aprendido a identificar como anomalias as galáxias desviadas do padrão “normal” de brilho para espirais e elípticas. Isso pode ser útil para identificar galáxias com características incomuns, que podem ser objeto de estudos mais aprofundados, como as galáxias irregulares.

5 CONCLUSÃO

Esse trabalho demonstrou o potencial das redes neurais convolucionais na classificação morfológica de galáxias, buscando distinguir entre galáxias espirais e elípticas, e na mineração de dados astronômicos como um todo. Com o auxílio do referencial teórico para compreender o funcionamento das redes neurais artificiais, da mineração de dados e a classificação de galáxias, bem como trabalhos relacionados de outros autores, foi possível obter resultados satisfatórios durante o desenvolvimento do trabalho.

Os resultados obtidos demonstram que ambos os modelos desenvolvidos B e C conseguiram atingir resultados excelentes de acurácia, 98% e 95% respectivamente, em relação ao modelo A, com apenas 73% de acurácia. Isso demonstra a importância de buscar por diferentes otimizações para aprimoramento dos modelos a serem utilizados, bem como validou-se a eficácia do uso de redes neurais convolucionais para classificação de galáxias.

Durante o desenvolvimento dos modelos, foram enfrentados alguns desafios que trouxeram algumas limitações. O primeiro deles foi em relação ao conjunto de dados, visto que é relativamente pequeno e acaba ocasionando problemas de treinamento como o *overfitting*. Isso acontece devido à dificuldade de encontrar *datasets* de galáxias rotulados, ou até mesmo no geral. Os dados coletados pelas missões espaciais dos observatórios e agências espaciais são normalmente divulgados apenas depois de um determinado tempo, variando de 2 a 5 anos, após a coleta, e ainda muitos dados acabam não sendo divulgados ou têm restrições, se tornando exclusivos de outros pesquisadores.

Além disso, outras dificuldades são referentes ao próprio processamento destes dados e também no treinamento dos modelos, visto que é necessário bons recursos computacionais para obter resultados mais rápidos. Essa problemática foi contornada razoavelmente bem com o uso de programação paralela e também com o uso de GPU para processamento dos dados e treinamento dos modelos. As otimizações feitas pelos modelos também ajudaram a diminuir o tempo de execução. Porém, como no modelo C, ainda teve uma demora considerável para obter resultados, dado a sua estrutura híbrida que utiliza imagens e seus metadados durante o treinamento, bem como o uso da busca de melhores parâmetros feito pela busca Bayesiana.

Apesar das dificuldades encontradas, ainda foi possível realizar o estudo de mineração de dados astronômicos, e encontrar bons aprimoramentos para o uso de redes neurais

convolucionais para classificação das galáxias. Os resultados se mostraram satisfatórios, contudo, há espaço para melhorias. Para trabalhos futuros, sugere-se a aplicação dos modelos em conjuntos de dados maiores; investigação de outras otimizações para lidar totalmente com overfitting em conjuntos menores de dados como o do estudo apresentado; bem como exploração de novas tecnologias para contornar problemas de processamento computacional, onde os recursos computacionais são mais limitados, trazendo soluções mais eficientes e escaláveis.

Este estudo contribui para o avanço da aplicação de técnicas de IA na Astronomia, demonstrando o potencial das redes neurais convolucionais na classificação automatizada de galáxias. A automação dessa tarefa pode auxiliar pesquisadores na análise de pequenos e grandes volumes de dados, acelerando o processo de descoberta e permitindo uma exploração mais eficiente do cosmos.

REFERÊNCIAS

- ALL, M. 2024. Introdução às funções de ativação em redes neurais. Disponível em: <<https://www.datacamp.com/pt/tutorial/introduction-to-activation-functions-in-neural-networks>>. Acesso em: 20 out. 2024.
- ALVES, G. 2018. Entendendo Redes Convolucionais (CNNs). Disponível em: <<https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184>>.
- AND, S. Astronomical Data Analysis - Foundation of Astronomical Studies and Exploration - Medium. Disponível em: <<https://medium.com/@fase.srilanka/astronomical-data-analysis-6ef086abe6e6>>. Acesso em: 19 out. 2024.
- BERGMANN, D. STRYKER C. 2024. What is Backpropagation? | IBM. Disponível em: <<https://www.ibm.com/think/topics/backpropagation>>.
- BORGES, C. L. S.; RODRIGUES, C. G. Astronomia: breve história, principais conceitos e campos de atuação / Astronomy: brief history, main concepts and fields of activity. Brazilian Applied Science Review, v. 6, n. 2, p. 545–577, 10 abr. 2022.
- BOSTON UNIVERSITY ARTS & SCIENCES. <https://www.bu.edu/astronomy.com>. Galaxy Classification. [S.I.]. Boston University, 2020. Disponível em: <https://www.bu.edu/astronomy/files/2023/01/Galaxy-Classification-2020.pdf>. Acesso em: 30 out. 2024.
- Brazil Astronomy. Classificação morfológica de galáxias. Disponível em: <https://brazilastronomy.wordpress.com/classificacao-morfologica-de-galaxias/>. Acesso em: 15 maio. 2024.
- CECCON, D. 2020. Funções de ativação: definição, características, e quando usar cada uma. Disponível em:

<<https://iaexpert.academy/2020/05/25/funcoes-de-ativacao-definicao-caracteristicas-e-quando-usar-cada-uma/>>. Acesso em: 20 out. 2024.

ĆIPRIJANOVIĆ, A. et al. DeepMerge II: Building Robust Deep Learning Algorithms for Merging Galaxy Identification Across Domains. *Monthly Notices of the Royal Astronomical Society*, v. 506, n. 1, p. 677–691, 9 jul. 2021.

CHAUDHRY, M. et al. A Systematic Literature Review on Identifying Patterns Using Unsupervised Clustering Algorithms: A Data Mining Perspective. *Symmetry*, v. 15, n. 9, p. 1679, 1 set. 2023.

CHENG, T.-Y. et al. Optimizing automatic morphological classification of galaxies with machine learning and deep learning using Dark Energy Survey imaging. *Monthly Notices of the Royal Astronomical Society*, v. 493, n. 3, p. 4209–4228, 19 fev. 2020.

CHOI, R. Y. et al. Introduction to Machine Learning, Neural Networks, and Deep Learning. *Translational Vision Science & Technology*, v. 9, n. 2, p. 14–14, 28 jan. 2020.

COUTO, G. Lenticulares: um grupo de galáxias híbridas. Disponível em: <https://astropontos.org/2018/09/19/lenticulares-um-grupo-de-galaxias-hibridas/>. Acesso em: 15 maio. 2024.

CUNHA, A. A. L., 2024. Neurônios e redes neurais: Os blocos de construção. Disponível em: <<https://www.linkedin.com/pulse/neuronios-e-redes-neurais-os-blocos-de-construcao-intelltech-it-ptikf/>>. Acesso em: 20 out. 2024.

Dados estruturados x dados não estruturados — Diferença entre dados coletáveis — AWS. Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-structured-data-and-unstructured-data/>. Acesso em: 22 maio. 2024.

Disponível em: <http://burro.case.edu/Academics/Astr222/Galaxies/Intro/galaxies.html>. Acesso em: 23 maio. 2024.

EL BOUCHEFRY, K.; DE SOUZA, R. S. Chapter 12 - Learning in Big Data: Introduction to Machine Learning. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/B9780128191545000230>.

FERRARI, D. G.; DE CASTRO, L. N. Introdução à mineração de dados: conceitos básicos, algoritmos e aplicações. 1. ed. [s.l.] Saraiva Uni, 2016. p. 29.

HAMAGUTI1, É.; FABRICIO, A.; BREVE2. Introdução sobre Machine Learning e Deep Learning. [s.l.: s.n.]. Disponível em: <http://www.jornacitec.fatecbt.edu.br/index.php/XIJTC/XIJTC/paper/viewFile/2852/3204>. Acesso em: 23 maio. 2024.

HARKIRAN78. Artificial Neural Networks and its Applications. Disponível em: <https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications>.

Imagine the Universe! Disponível em: <<https://imagine.gsfc.nasa.gov/observatories/data/>>.

IVEZIĆ, Ž. et al. Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data, Updated Edition. [s.l.] Princeton University Press, 2020.

KANADE, V. What Is Machine Learning? Definition, Types, Applications, and Trends for 2022. Disponível em: <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/>.

KLEINA, Olivia. 2023. A diferença entre machine learning e deep learning. Disponível em: <<https://posdigital.pucpr.br/blog/machine-learning-deep-learning>>. Acesso: 21 de out. 2024.

McKINNEY, W. Python for data analysis: data wrangling with pandas, NumPy, and IPython. [s.l.] O'reilly Uuuu-Uuuu, 2018.

MILLSTEIN, F. Convolutional Neural Networks In Python: Beginner's Guide To Convolutional Neural Networks In Python. [s.l.] Frank Millstein, 2020.

Mineração de Dados na Web: Data Mining. Disponível em: <https://brightdata.com.br/blog/dados-do-site/mineracao-de-dados-na-web>. Acesso em: 21 abr. 2024.

O que é *overfitting*? | IBM. Disponível em: <<https://www.ibm.com/br-pt/topics/overfitting>>.

RIZZON, J. O que é: Backpropagation. Disponível em: <<https://napoleon.com.br/glossario/o-que-e-backpropagation/>>. Acesso em: 21 out. 2024.

SANDS, A. E. A. D. (2017). Managing astronomy research data: Data practices in the Sloan digital sky survey and large synoptic survey telescope projects. Disponível em: <<https://www.proquest.com/openview/0959291177dad60c9d38b8c31765fcf7/1?pq-origsite=gscholar&cbl=18750>>. Acesso em: 19 out. 2024.

SANGHVIRAJIT, A. 2021. Complete Guide to Adam and RMSprop Optimizer. Disponível em: <<https://medium.com/analytics-vidhya/a-complete-guide-to-adam-and-rmsprop-optimizer-75f4502d83be>>.

SEN, S. et al. Astronomical big data processing using machine learning: A comprehensive review. *Experimental Astronomy*, v. 53, n. 1, p. 1–43, 14 jan. 2022.

SHASHKO, B. Data mining and knowledge discovery. In: *Encyclopedia of Data Science and Analytics*. Springer.

SILVA, J. A., 2023. Como funcionam as funções de ativação em Redes Neurais. Disponível em: <<https://medium.com/@jeapsilva/funções-de-ativação-em-redes-neurais-90d095a8c2ef>>. Acesso em: 20 out. 2024.

SMITH, M. J.; GEACH, J. E. Astronomia ex machina: a history, primer and outlook on neural networks in astronomy. v. 10, n. 5, 1 maio 2023.

SUNIL MUCESH et al. A machine learning approach to galaxy properties: joint redshift–stellar mass probability distributions with Random Forest. Monthly Notices of the Royal Astronomical Society, v. 502, n. 2, p. 2770–2786, 21 jan. 2021.

TAN, P.-N. et al. Introduction to Data Mining. 2. ed. [s.l.] Pearson Education Limited, 2019.

TEIXEIRA, R. Missão Espacial Gaia. Cadernos de Astronomia, v. 3, n. 2, p. 101–111, 26 ago. 2022.

TING-YUN, C. et al. Galaxy morphological classification catalogue of the Dark Energy Survey Year 3 data with Convolutional Neural Networks. 2021. Disponível em: <<http://arxiv.org/abs/2107.10210>>.

TOHILL, C. et al. A Robust Study of High-redshift Galaxies: Unsupervised Machine Learning for Characterizing Morphology with JWST up to $z \sim 8$. The Astrophysical Journal, v. 962, p. 164, 1 fev. 2024.

WHO RUNS THE WORLD: DATA Editors Sevinç GÜLSEÇEN, Sushil SHARMA, Emre AKADAL. [s.l]: s.n.]. Disponível em: <https://etalpykla.vilniustech.lt/bitstream/handle/123456789/151156/Full%20book.pdf?sequence=1&isAllowed=y#page=39>. Acesso em: 23 abr. 2024.

YADAV, H. 2022. Dropout in Neural Networks. Disponível em: <<https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9>>.