

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

ENYA LUÍSA GOMES DOS SANTOS

**APLICAÇÃO DE *DEEP LEARNING* PARA CLASSIFICAÇÃO
MORFOLÓGICA DE GALÁXIAS**

Ouro Preto, MG
2024

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

ENYA LUÍSA GOMES DOS SANTOS

**APLICAÇÃO DE *DEEP LEARNING* PARA CLASSIFICAÇÃO MORFOLÓGICA DE
GALÁXIAS**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Pedro Henrique Lopes Silva

Ouro Preto, MG
2024

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

S237a Santos, Enya Luisa Gomes dos.
Aplicação de Deep Learning para classificação morfológica de galáxias. [manuscrito] / Enya Luisa Gomes dos Santos. - 2024.
62 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Pedro Henrique Lopes Silva.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Biológicas. Graduação em Ciência da Computação .

1. Ciência da computação. 2. Redes neurais (Computação). 3. Inteligência artificial. 4. Aprendizado do computador. I. Silva, Pedro Henrique Lopes. II. Universidade Federal de Ouro Preto. III. Título.

CDU 519.254

Bibliotecário(a) Responsável: Paulo Vitor Oliveira - CRB6/2551



FOLHA DE APROVAÇÃO

Enya Luísa Gomes dos Santos

Aplicação de Deep Learning para Classificação Morfológica de Galáxias

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 8 de Fevereiro de 2024.

Membros da banca:

Pedro Henrique Lopes Silva (Orientador) - Doutor - Universidade Federal de Ouro Preto
Débora Nasser Diniz (Examinadora) - Doutora - TIM Brasil
Geovani Lopes Martins (Examinador) - Mestre - Universidade Federal de Ouro Preto

Pedro Henrique Lopes Silva, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 8/02/2024.



Documento assinado eletronicamente por **Pedro Henrique Lopes Silva, PROFESSOR DE MAGISTERIO SUPERIOR**, em 09/02/2024, às 10:00, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0665688** e o código CRC **61C5589E**.

Dedico este trabalho com todo o carinho e amor às minhas avós, Maria das Graças Gomes e Olindina Petrina da Paixão Santos. Vocês são a personificação do amor, sabedoria e força, e são as principais inspirações em minha vida.

Agradecimentos

Primeiramente, quero expressar minha profunda gratidão a minha família, que sempre estiveram ao meu lado, me apoiando em todas as etapas da minha jornada acadêmica. Sem o amor, encorajamento e dedicação deles, não teria sido possível alcançar esse momento tão especial em minha vida.

Aos meus queridos amigos da universidade, que compartilharam comigo momentos de alegria, estudo e superação ao longo desses anos. Juntos, enfrentamos desafios, crescemos como profissionais e construímos laços que levo para a vida toda.

Agradeço também ao meu orientador, Pedro Henrique Lopes Silva, pela orientação, paciência e compromisso durante todo o desenvolvimento deste trabalho. Suas contribuições e conhecimentos foram fundamentais para o aprimoramento deste projeto.

Por fim, agradeço à Universidade Federal de Ouro Preto por proporcionar o ambiente propício ao aprendizado, ao desenvolvimento de habilidades e ao crescimento pessoal. Sou grata por ter tido a oportunidade de estudar nesta instituição.

Que todos aqueles que de alguma maneira colaboraram para tornar este trabalho realidade se sintam reconhecidos e valorizados. Cada um de vocês fez parte dessa conquista e tenho certeza de que levarei cada lembrança e aprendizado comigo ao longo da vida.

“Na vida, não existe nada a se temer, apenas a ser compreendido”

Marie Currie (Frank, 1977, p. iii)

Resumo

A classificação morfológica de galáxias desempenha um papel crucial na compreensão da origem e evolução do universo. Tradicionalmente conduzida por especialistas, a automação desse processo por meio de técnicas de aprendizado profundo surge como uma alternativa promissora. Neste estudo, foram comparados o desempenho de uma nova arquitetura proposta por [Gharat e Dandawate \(2022\)](#) e seis arquiteturas diferentes de redes neurais convolucionais: EfficientNetV2M, DenseNet121, ResNet50, AlexNet, VGG19 e VGG16. Além disso, para aprimorar os resultados, foram empregadas técnicas adicionais, como a classificação hierárquica e o *ensemble learning*. A análise foi realizada para classificar 21.785 amostras da base Sloan Digital Sky Survey em 10 classes morfológicas distintas. Por meio da aplicação de técnicas de *transfer learning*, a arquitetura EfficientNetV2-M apresentou uma acurácia de 82,90% nos testes. Já com a mesma técnica, combinada com o uso de *data augmentation* para balanceamento dos dados, a ResNet50 destacou-se, atingindo uma acurácia de 88,20% na base de teste. No entanto, a estratégia de *ensemble learning* superou esses resultados, alcançando uma acurácia de 85,53% com os dados desbalanceados e 91,75% com o balanceamento dos dados. Isso evidencia o *ensemble learning* como uma técnica promissora para aprimorar a classificação morfológica de galáxias, oferecendo uma abordagem eficaz e robusta para esse desafio complexo.

Palavras-chave: Rede Neural Convolucional. Transferência de Aprendizado. Classificação morfológica de galáxias. Aprendizagem Profunda. *Ensemble Learning*.

Abstract

The morphological classification of galaxies plays a crucial role in understanding the origin and evolution of the universe. Traditionally conducted by experts, the automation of this process through deep learning techniques emerges as a promising alternative. In this study, the performance of a new architecture proposed by [Gharat e Dandawate \(2022\)](#) was compared with six different architectures of convolutional neural networks: EfficientNetV2M, DenseNet121, ResNet50, AlexNet, VGG19, and VGG16. Additionally, to enhance the results, additional techniques such as hierarchical classification and ensemble learning were employed. The analysis aimed to classify 21,785 samples from the Sloan Digital Sky Survey into 10 distinct morphological classes. Through the application of transfer learning techniques, the EfficientNetV2-M architecture achieved an accuracy of 82.90% in the tests. With the same technique, combined with the use of data augmentation for data balancing, ResNet50 stood out, reaching an accuracy of 88.20% on the test dataset. However, the ensemble learning strategy surpassed these results, achieving an accuracy of 85.53% with unbalanced data and 91.75% with balanced data. This highlights ensemble learning as a promising technique to improve the morphological classification of galaxies, providing an effective and robust approach to this complex challenge.

Keywords: Convolutional Neural Network. Transfer Learning. Morphological Classification of Galaxies. Deep Learning. Ensemble Learning.

Lista de Ilustrações

Figura 1 – Estrutura de um neurônio biológico.	5
Figura 2 – Diagrama de um neurônio artificial.	6
Figura 3 – Rede neural artificial de múltiplas camadas.	7
Figura 4 – Backpropagation: propagação e retropropagação.	8
Figura 5 – Estrutura de uma rede convolucional.	10
Figura 6 – Diagrama de funcionamento da convolução.	11
Figura 7 – Hierarquia de padrões aprendidos em uma rede neural profunda.	11
Figura 8 – Operação de pooling com média (<i>mean pooling</i>) e com máximo (<i>max pooling</i>).	12
Figura 9 – Operação de <i>flattening</i>	13
Figura 10 – Estrutura e operações da camada <i>fully-connected</i>	13
Figura 11 – Classificadores locais por nó.	15
Figura 12 – Classificadores locais por nó pai.	16
Figura 13 – Classificadores locais por nível.	16
Figura 14 – Diagramas de blocos de <i>ensemble learning</i> sequencial e paralelo.	17
Figura 15 – Exemplo de matriz de confusão.	18
Figura 16 – Forquilha original do diapasão de Hubble publicado em Hubble (1936).	22
Figura 17 – Forquilha original do diapasão de Hubble estendida.	22
Figura 18 – Exemplos de galáxias elípticas.	23
Figura 19 – Partes de uma galaxia espiral.	24
Figura 20 – Exemplos de galáxias espirais: fotos de galáxias obtidas por Jim Wray, no McDonald Observatory.	24
Figura 21 – Exemplos de galáxias espirais e espirais barradas, respectivamente.	25
Figura 22 – Exemplo de galáxias espirais barradas	25
Figura 23 – Fotografias de galáxias à esquerda SO e à direita SBO.	26
Figura 24 – Foto das galáxias irregulares Grande Nuvem de Magalhães e Pequena Nuvem de Magalhães, obtida por Wei-Hao Wang.	26
Figura 25 – Representação da precisão do teste usando o gráfico de barras.	28
Figura 26 – Imagens de cada classe de galáxias do Galaxy10 Dataset	29
Figura 27 – Tabelas de comparação das métricas de precisão, <i>recall</i> e F1-score, respectivamente, para as classes geradas por cada um dos métodos.	33
Figura 28 – Galaxy Zoo <i>Decision Tree</i>	37
Figura 29 – Early Stopping	40
Figura 30 – Arquitetura da rede neural convolucional de Gharat e Dandawate (2022).	42
Figura 31 – Arquitetura do Classificador	43
Figura 33 – Fluxo do <i>ensemble learning</i>	44
Figura 32 – Hierarquia das classes.	45

Figura 34 – Matriz de confusão da rede EfficientNetV2M.	50
Figura 35 – Matriz de confusão da rede ResNet50,	52
Figura 36 – Matriz de confusão da rede ResNet50	54

Lista de Tabelas

Tabela 2.1 – Matriz de Confusão - Classificação Multiclasse	19
Tabela 2.2 – Sub-divisões das galáxias espirais	24
Tabela 2.3 – Propriedades das Galáxias Espirais, Elípticas e Irregulares	27
Tabela 2.4 – Sumarização dos Trabalhos Correlatos, Dados Utilizados e Resultados	34
Tabela 3.1 – Galaxy10 dataset	38
Tabela 4.1 – Esta tabela apresenta os resultados dos modelos avaliados em uma base de dados com classes de amostras desbalanceadas e com alteração no valor de <i>Patience</i> do <i>Early Stopping</i>	47
Tabela 4.2 – Esta tabela apresenta os resultados dos modelos avaliados em uma base de dados balanceada, utilizando a técnica de <i>oversampling</i> número 5 citado na Seção 3.2, e com alteração no valor de <i>Patience</i> do <i>Early Stopping</i>	48
Tabela 4.3 – Resultados dos modelos avaliados em uma base de dados com classes de amostras desbalanceadas e com <i>early stopping</i>	49
Tabela 4.4 – Acurácia dos Modelos por Classe.	49
Tabela 4.5 – Resultados dos modelos avaliados na base de dados com classes de amostras desbalanceadas, porém com aumento das amostras da classe <i>Disk</i> , <i>Edge-on</i> , <i>Boxy Bulge</i> para 204 e com <i>early stopping</i>	50
Tabela 4.6 – Acurácia dos modelos por classe.	51
Tabela 4.7 – Resultados dos modelos avaliados na base de dados com as amostras das classes balanceadas em 17 amostras (<i>undersampling</i>).	52
Tabela 4.8 – Resultados dos modelos avaliados na base de dados com aumento das amostras da classe <i>Disk</i> , <i>Edge-on</i> , <i>Boxy Bulge</i> para 204, e em seguida um balanceadas em 204 amostras nas demais classes (<i>undersampling</i>).	53
Tabela 4.9 – Resultados das métricas do experimento com balanceamento dos dados. Esta tabela apresenta os resultados dos modelos avaliados na base de dados balanceada com <i>oversampling</i> , onde todas as classes possuem 6.997 amostras.	53
Tabela 4.10–Resultados da classificação hierárquica de acordo com a arquitetura utilizada.	55
Tabela 4.11–Resultados do aprendizado por agrupamento (<i>ensemble learning</i>) das redes anteriormente treinadas no Experimento II com os dados sem alteração. Arquiteturas utilizadas: EfficientNet-V2-M, ResNet50, DenseNet121, VGG16 e VGG19.	56
Tabela 4.12–Resultados do aprendizado por agrupamento (<i>ensemble learning</i>) das redes anteriormente treinadas no Experimento II com os dados balanceados. Arquiteturas utilizadas: EfficientNet-V2-M, ResNet50 e DenseNet1219.	56
Tabela 4.13–Comparação dos resultados obtidos com outros trabalhos na literatura.	59

Lista de Abreviaturas e Siglas

ACC	<i>Accuracy (Acurácia).</i>
AGN	<i>Active Galactic Nuclei.</i>
ANN	<i>Artificial Neural Network (Redes Neurais Artificiais).</i>
ABC	<i>Artificial Bee Colony.</i>
CNN	<i>Convolutional Neural Network (Redes Neurais Convolucionais).</i>
GWO	<i>Grey Wolf Optimizer.</i>
GZ2	<i>Galaxy Zoo 2.</i>
IA	Inteligência Artificial.
MCP	McCulloch-Pitts.
ML	<i>Machine Learning (Aprendizado de máquina).</i>
MLP	<i>Multilayer perceptron.</i>
PSO	<i>Particle Swarm Optimization.</i>
RNA	Rede Neural Artificial.
SDSS	<i>Sloan Digital Sky Survey.</i>
TL	<i>Transfer Learning.</i>
ViT	<i>Vision Transformer.</i>

Sumário

1	Introdução	1
1.1	Justificativa	2
1.2	Objetivos	3
1.3	Organização do Trabalho	3
2	Revisão Bibliográfica	4
2.1	Fundamentação Teórica	4
2.1.1	Deep Learning	4
2.1.2	Redes Neurais Artificiais	4
2.1.2.1	Neurônio Artificial	5
2.1.2.2	Perceptron	6
2.1.2.3	<i>Backpropagation</i>	8
2.1.3	Redes Neurais Convolucionais	10
2.1.3.1	Camada Convolutiva	10
2.1.3.2	Camada de Pooling	12
2.1.3.3	Camada Totalmente Conectada	12
2.1.4	Transfer Learning (TL)	14
2.1.5	Classificação Hierárquica	14
2.1.5.1	Classificação Plana	14
2.1.5.2	Classificadores Locais Estruturados Hierarquicamente	15
2.1.6	Ensamble Learning	16
2.1.7	Métricas de Avaliação	18
2.1.7.1	Matriz de Confusão	18
2.1.7.2	Acurácia	19
2.1.7.3	<i>Recall</i>	20
2.1.7.4	Precisão	20
2.1.7.5	<i>F1-Score</i>	20
2.1.8	<i>Hugging Face</i>	20
2.1.9	Sistema de Hubble de classificação morfológica de galáxias	21
2.1.9.1	Elípticas	23
2.1.9.2	Espirais	23
2.1.9.3	Espirais Barradas	24
2.1.9.4	Lenticulares	25
2.1.9.5	Irregulares	26
2.1.9.6	As Classes	26
2.2	Trabalhos Relacionados	27
2.2.1	<i>Convolved Cosmos: Classifying Galaxy Images Using Deep Learning</i>	27

2.2.2	<i>Galaxy Classification: A deep learning approach for classifying Sloan Digital Sky Survey images</i>	28
2.2.3	<i>Morphological classification of galaxies with deep learning: comparing 3-way and 4-way CNNs</i>	29
2.2.4	<i>A Hyperparameter Optimization for Galaxy Classification</i>	30
2.2.5	<i>Galaxy Morphological Classification with Efficient Vision Transformer</i>	31
2.2.6	<i>The Classification of Optical Galaxy Morphology Using Unsupervised Learning Techniques</i>	31
2.2.7	Considerações Sobre os Trabalhos Relacionados	33
3	Desenvolvimento	36
3.1	Base de dados	36
3.2	Pré-processamento dos dados	38
3.3	Caracterização da base de dados	39
3.4	Treinamento do modelo	40
3.5	Avaliação	40
3.6	Etapa Experimental	41
3.6.1	Experimento I	41
3.6.2	Experimento II	42
3.6.3	Experimento III	43
3.6.4	Experimento IV	44
4	Resultados	46
4.1	Ferramentas e técnicas utilizadas	46
4.2	Experimento I	47
4.3	Experimento II	48
4.4	Experimento III	54
4.5	Experimento IV	55
4.6	Discussão dos resultados	56
4.7	Disponibilização do Modelo	58
4.8	Comparação com Trabalho Relacionados	58
5	Considerações Finais	60
5.1	Conclusão	60
5.2	Trabalhos Futuros	61
	Referências	62

1 Introdução

Desde a descoberta das galáxias em 1923 por Edwin Powell Hubble (1889-1953) (Oliveira, 2013), os estudos desses corpos têm grande relevância para a busca da resposta da origem do universo. As galáxias apresentam variadas formas, e com isso é possível obter informações importantes como sua formação até sua evolução (Silva; Ventura, 2019).

A morfologia das galáxias, moldada por sua formação, interações ambientais, influências internas como Núcleos Galácticos Ativos (do inglês, *Active Galactic Nuclei*, ou AGNs) e matéria escura, juntamente com suas variadas histórias de formação estelar, é um produto multifacetado (Buta, 2013). Segundo Gauthier, Jain e Noordeh (2016), este aspecto é observável nas centenas de bilhões de galáxias dispersas pelo cosmos, que desempenham um papel fundamental na definição da estrutura do universo em suas maiores escalas. A distribuição das propriedades físicas dessas galáxias oferece pistas cruciais para a compreensão do passado, presente e futuro do universo, sendo essenciais para informar os modelos teóricos dos astrônomos e cosmologistas, e fornecer *insights* valiosos para novas descobertas científicas.

Atualmente, existe um vasto volume de registros astronômicos disponíveis para pesquisa, tais dados tentem a aumentar cada vez mais. Por exemplo, projetos como *Galaxy Zoo*, criado em 2007, permitiu que os milhões de participantes do mundo todo acessassem imagens do *Sloan Digital Sky Survey* SDSS (York *et al.*, 2000) para análise e classificação de galáxias (Fortson *et al.*, 2011), resultando em uma quantidade significativa de dados classificados. Esses registros são fundamentais no estudo das galáxias.

No entanto, a classificação é baseada na inspeção visual humana e pode estar sujeita a vieses devido à subjetividade da observação humana (Silva; Ventura, 2019). Por esse motivo, a criação de classificadores automatizados utilizando técnicas computacionais tem ganhado importância.

Ludermir (2021) afirma que na IA Generalizada, também conhecida como IA Forte, os algoritmos desenvolvidos alcançam um nível de habilidade semelhante ao dos humanos em várias tarefas. Geralmente, esses algoritmos utilizam técnicas de aprendizado de máquina como ferramentas para atingir esse objetivo. Em determinadas tarefas, como visão computacional, os algoritmos podem obter desempenho comparável ao dos humanos. Dito isso, a utilização de aprendizado de máquina em conjunto com visão computacional pode ser utilizada para identificação automatizada de diferentes tipos de galáxias com base em suas características morfológicas.

A aplicação de algoritmos de aprendizado profundo e redes neurais convolucionais (do inglês, *Convolutional Neural Networks* - CNNs), por exemplo, torna possível a criação de modelos capazes de aprender tarefas a partir de vastos conjuntos de dados astronômicos previamente

classificados. Isso é evidenciado por [Montes, Lattari e Coelho \(2021\)](#) que, utilizando os dados do projeto [Dark Energy Survey](#), desenvolveram um modelo de CNN para discernir entre Supernovas e não Supernovas. Há também o trabalho de [Wu e Boada \(2019\)](#) que se concentra na previsão da metalicidade de galáxias a partir de imagens obtidas da SDSS ([York et al., 2000](#)).

Outro exemplo notável é apresentado por [Bialopetravičius e Narbutis \(2020\)](#), no qual é proposto um modelo de CNN capaz de determinar as idades, massas e tamanhos de aglomerados estelares utilizando imagens provenientes do *Panchromatic Hubble Andromeda Treasury* (PHAT) survey ([Dalcanton et al., 2012](#)). Além disso, a pesquisa conduzida por ([Gharat; Dandawate, 2022](#)) aborda a classificação morfológica de galáxias por meio de um modelo que utiliza o SDSS ([York et al., 2000](#)).

Dentro desse contexto, o escopo deste projeto de pesquisa consiste em explorar técnicas de aprendizado de máquina, com ênfase no aprendizado profundo, a fim de realizar a tarefa de classificação morfológica de galáxias. A implementação dessas técnicas permitirá uma abordagem automatizada que tem o potencial de superar as limitações associadas à classificação visual humana, como problema de precisão e principalmente o enorme volume de dados que torna impraticável a realização da tarefa em tempo hábil. Conseqüentemente, essa abordagem automatizada promoverá uma maior objetividade e precisão no processo de classificação.

1.1 Justificativa

Ainda que as galáxias possam apresentar semelhanças entre si, suas características distintas as agrupam com base em sua aparência visual. A classificação das galáxias por parâmetros morfológicos é crucial para compreender sua formação e evolução ([Silva; Ventura, 2019](#)). Além disso, a morfologia das galáxias desempenha um papel fundamental na compreensão da evolução do universo. Ao analisar a aparência e a estrutura das galáxias, astrônomos podem inferir sua idade relativa, padrões de formação estelar e distribuição no cosmos. Essas observações fornecem informações vitais sobre a história cósmica, auxiliando no mapeamento da evolução das galáxias ao longo do tempo e na validação de teorias fundamentais sobre a formação e estrutura do universo ([Buta, 2013](#)).

Tradicionalmente, a classificação de galáxias tem sido realizada de forma visual por especialistas na área ([Fortson et al., 2011](#)). No entanto, essa abordagem humana pode resultar em imprecisões, além de consumir considerável tempo, especialmente diante do crescente volume de dados astronômicos disponíveis, impulsionado pelo avanço da tecnologia espacial.

Este trabalho visa desenvolver uma ferramenta capaz de analisar e classificar automaticamente a morfologia das galáxias por meio de imagens astronômicas. Essa ferramenta busca superar as limitações da classificação visual, sobretudo lidando com o vasto volume de dados a serem classificados e minimizando erros de precisão associados à intervenção humana.

1.2 Objetivos

O principal objetivo deste trabalho é desenvolver um modelo de rede neural convolucional capaz de realizar a classificação morfológica de galáxias por meio de imagens astronômicas. Por meio da aplicação de técnicas de aprendizado de máquina, especialmente na área de aprendizado profundo, busca-se automatizar o processo de classificação, superando as limitações da classificação visual realizada por especialistas.

Para que o objetivo principal seja atingido, torna-se necessário a definição de objetivos específicos, os quais são definidos como:

- Seleção de um conjunto de dados adequado para o problema. Isso envolve a obtenção de imagens astronômicas de galáxias, juntamente com suas respectivas classificações morfológicas realizadas por especialistas. O conjunto de dados deve ser representativo e abrangente o suficiente para treinar e avaliar o modelo de forma adequada.
- Treinamento dos modelos de rede neural convolucional. Definir arquiteturas da rede neural convolucional, considerando camadas convolucionais, camadas de *pooling* e camadas totalmente conectadas, de acordo com as características das imagens astronômicas. Em seguida, realizar o treinamento dos modelos utilizando o conjunto de dados preparado, ajustando os parâmetros do modelo para otimizar seu desempenho.
- Avaliação do desempenho dos modelos. Avaliar a capacidade dos modelos de realizar a classificação morfológica de galáxias corretamente. Isso envolve a aplicação dos modelos em um conjunto de dados de teste separado, medindo métricas de desempenho, como acurácia, precisão, *recall* e *F1-score*.
- Analisar os resultados obtidos pelos modelos e comparar o desempenho entre eles, a fim de obter um bom modelo com boa eficácia e vantagens em relação aos outros. Além disso, realizar uma comparação com trabalhos da literatura que utilizam a mesma base de dados ou semelhante.

1.3 Organização do Trabalho

Este trabalho é distribuído em cinco capítulos, sendo o primeiro capítulo a introdução atual. No **Capítulo 2** está presente uma revisão bibliográfica acerca do tema do trabalho, sendo feita a apresentação de trabalhos relacionados ao tema de classificação morfológica de galáxias com aprendizado de máquina, e uma fundamentação teórica necessária para o entendimento do trabalho. O **Capítulo 3** refere-se ao desenvolvimento da pesquisa e como foi feito. Já o **Capítulo 4** apresenta os resultados obtidos através dos experimentos realizados durante o projeto e uma discussão acerca desses resultados. Por fim, o **Capítulo 5** traz as considerações finais e propõe atividades a serem executadas em trabalhos futuros.

2 Revisão Bibliográfica

Este capítulo apresenta uma base teórica essencial para compreender os conceitos explorados ao longo deste trabalho, visando oferecer ao leitor uma compreensão mais clara da metodologia empregada. Posteriormente, contextualiza o tema das classificações morfológicas de galáxias e deep learning por meio da exposição de trabalhos correlatos que exploram esses assuntos.

2.1 Fundamentação Teórica

Nesta seção são abordados os pilares teóricos essenciais para compreensão do presente trabalho, iniciando com uma apresentação da *Deep Learning*. Em sequência, são apresentados conceitos fundamentais dentro desse domínio, contemplando Redes Neurais Artificiais (RNAs) e, de maneira específica, Redes Neurais Convolucionais, em inglês *Convolutional Neural Networks* (CNNs). Na sequência, explora-se o conceito de *Transfer Learning*. Por fim, encerra-se a seção com uma exposição sobre o Sistema de Hubble, um sistema de classificação morfológica de galáxias amplamente reconhecido.

2.1.1 Deep Learning

Deep Learning, conhecido como aprendizado profundo, é uma abordagem que tem a capacidade de identificar padrões intrincados a partir de vastos conjuntos de dados. Esta forma de aprendizado de máquina possibilita que os computadores resolvam desafios perceptuais complexos, como a interpretação de imagens e a compreensão da fala. No âmbito do aprendizado profundo, métodos como as redes neurais artificiais profundas são empregados, os quais se utilizam de múltiplas camadas de processamento para revelar padrões e estruturas em conjuntos de dados de grande magnitude. Um aspecto distintivo é que o aprendizado profundo não demanda pré-processamento dos dados e, de forma automática, é capaz de extrair características relevantes (Rusk, 2016).

2.1.2 Redes Neurais Artificiais

A rede neural artificial (RNA) é um dos ramos da inteligência artificial, e é um modelo computacional inspirado no funcionamento do cérebro humano. “O cérebro é um computador (sistema de processamento de informação) altamente complexo, não linear e paralelo” (HAYKIN, 2001). Consiste em uma coleção interconectada de unidades de processamento chamadas neurônios artificiais. Esses neurônios são organizados em camadas e realizam operações matemáticas nas entradas recebidas, produzindo saídas que são transmitidas para a próxima camada.

O cérebro humano aprende com a “experiência” acumulada ao longo do tempo, a plasticidade cerebral, ou seja, a capacidade do cérebro de se adaptar e reorganizar ao longo do tempo em resposta a diferentes mudanças ambientais, permite que o sistema nervoso humano em desenvolvimento forme novas conexões entre os neurônios e reestruture sua organização funcional adaptando-se ao meio ambiente. Essa plasticidade é essencial para as redes neurais baseadas em neurônios artificiais (HAYKIN, 2001). Visto que, de acordo com HAYKIN (2001), uma rede neural, em sua forma mais geral, é uma máquina projetada para modelar a maneira como o cérebro executa uma tarefa específica ou função de interesse.

“Uma rede neural é um processador maciçamente e paralelamente distribuído, constituído de unidades de processamento simples, que têm a propensão natural de armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:

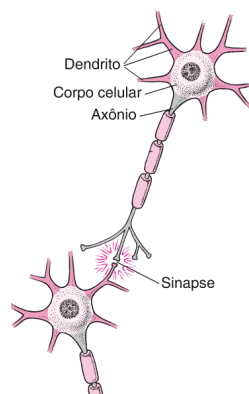
1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.
2. Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.” (HAYKIN, 2001)

2.1.2.1 Neurônio Artificial

Um neurônio é a unidade de processamento de informação fundamental para a operação de uma rede neural (HAYKIN, 2001). Um neurônio artificial é uma abstração do neurônio biológico.

A Figura 1 apresenta uma ilustração de um neurônio biológico. De forma resumida, o neurônio funciona recebendo informações de outros neurônios através dos dendritos. Com base nesses estímulos recebidos, ele gera um impulso que é transmitido para outros neurônios através do axônio. A transmissão ocorre por meio da sinapse, onde substâncias são liberadas pelo axônio do neurônio para estimular os receptores nos dendritos de outros neurônios. Esse processo possibilita a transmissão de impulsos nervosos de um neurônio para o outro (Cintra, 2016).

Figura 1 – Estrutura de um neurônio biológico.



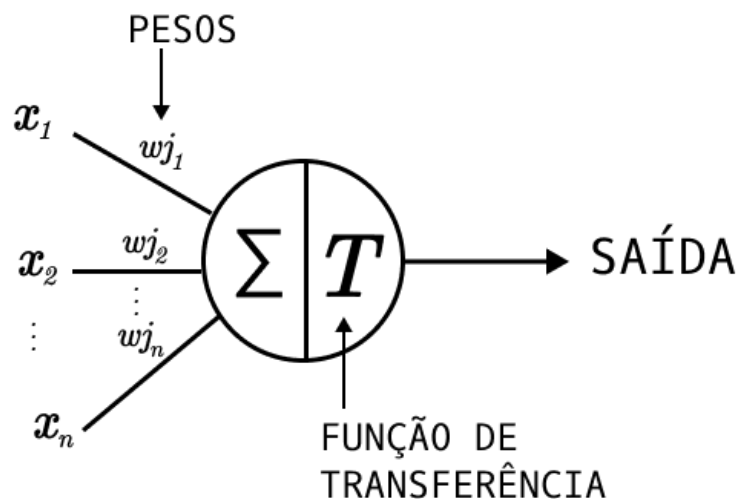
Fonte: Disponível em:

<<https://www.msmanuals.com/pt/casa/multimedia/figure/estrutura-típica-de-um-neurônio>>. Acesso em 24 de junho de 2023.

Seguindo a comparação, o neurônio biológico só dispara um impulso nervoso quando a soma dos impulsos de entrada ultrapassa o seu limiar de excitação. Já em um neurônio artificial, a soma dos impulsos é representada pela função de soma, onde há um mecanismo simples que faz a soma dos valores recebidos pelo neurônio (soma ponderada). O resultado é submetido à função de ativação que ativa ou não a saída, dependendo do valor da soma ponderada das suas entradas (Braga; Ludermir; Carvalho, 2007).

Em 1943, McCulloch e Pitts (McCulloch; Pitts, 1943) introduziram um modelo de neurônio artificial por meio de uma descrição matemática (Fleck *et al.*, 2016). “O modelo, denominado neurônio MCP (McCulloch-Pitts), é descrito por um conjunto de n entradas, as quais são multiplicadas por um determinado peso ($x * w$)” (Fleck *et al.*, 2016). A Figura 2 mostra uma representação do diagrama de um neurônio MCP (Braga; Ludermir; Carvalho, 2007).

Figura 2 – Diagrama de um neurônio artificial.



Fonte: Adaptação de: <<https://www.linkedin.com/pulse/neur%C3%B4nio-matem%C3%A1tico-conceitos-iniciais-com-exemplo-marcelo/>>. Acesso em 26 de maio de 2023.

Os sinais de entrada são representados por x , cada entrada é multiplicada pelo respectivo peso sináptico w_j do neurônio ao qual está conectada. No neurônio é realizada a soma ponderada de todos os sinais recebidos (função de soma) produzindo um determinado nível de atividade. Este nível de atividade é submetido à função de ativação, a unidade processadora produzirá uma determinada resposta como saída (Furtado, 2019). “Este modelo se apresenta constante para quase todas as Redes Neurais, variando somente a função de ativação” (Furtado, 2019).

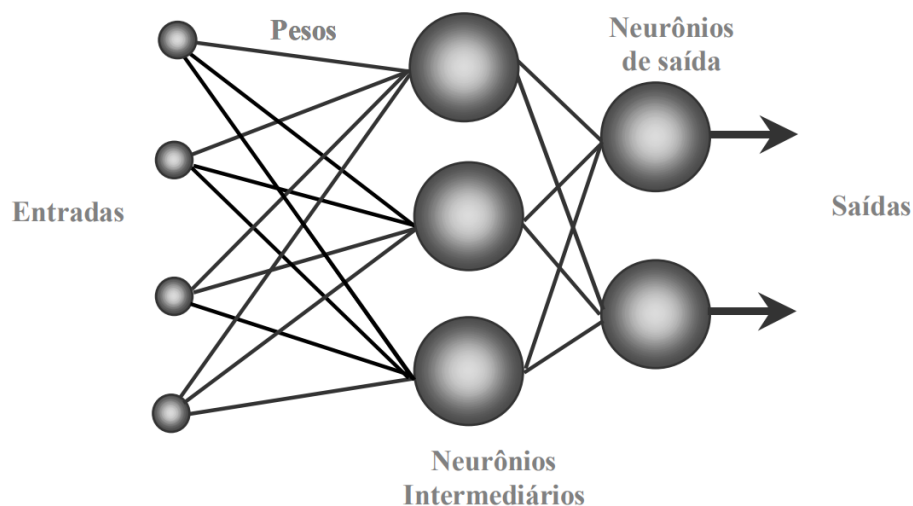
2.1.2.2 Perceptron

Frank Rosenblatt, em 1958, propôs umas das primeiras topologias de redes neurais chamada *perceptron* construída com neurônios artificiais acrescida de uma regra de aprendizagem. Sobre o modelo *perceptron* de Frank Rosenblatt, Braga, Ludermir e Carvalho (2007) afirmam:

“[...] era composta por unidades de entrada (retina), por um nível intermediário formado pelas unidades de associação e por um nível de saída formado pelas unidades de resposta. Embora esta topologia original possua três níveis, ela é conhecida como perceptron de uma única camada, já que somente o nível de saída (unidades de resposta) possui propriedades adaptativas.”

Entretanto, essa topologia resolvia apenas problemas linearmente separáveis, o que trouxe uma limitação para o uso de redes neurais artificiais na época. Para problemas não linearmente separáveis a solução foi o uso de redes com uma ou mais camadas intermediárias, ou escondidas (Braga; Ludermir; Carvalho, 2007). Esse tipo de RNAs foi chamado de *perceptron* multicamadas, ou MLP (*multilayer perceptron*) e está ilustrado na Figura 3.

Figura 3 – Rede neural artificial de múltiplas camadas.



Fonte: Furtado (2019).

Em relação ao aprendizado das redes, existe dois tipos de aprendizado: não-supervisionado e supervisionado. No treinamento não-supervisionado, como não há uma saída-alvo fornecida, a responsabilidade recai sobre a rede neural para extrair as características relevantes dos impulsos e agrupá-los em conjuntos pré-existentes (Furtado, 2019).

No treinamento supervisionado, um agente externo, denominado supervisor, é utilizado para instruir a rede sobre a resposta desejada para determinado padrão de entrada. Através da avaliação do erro, que representa a discrepância entre os valores esperados e os valores produzidos pela rede, o agente externo realiza ajustes nos parâmetros da rede. Essa atualização é continuada até que o erro seja reduzido ao mínimo, podendo ser eliminado completamente ou alcançar um nível considerado satisfatório (Furtado, 2019). Um método comumente utilizado nesse tipo de aprendizado é o chamado *backpropagation*.

Como Furtado (2019) destacou, durante esse processo, a rede tenta reduzir o erro entre suas saídas reais, ou seja, a ativação na camada final, e as saídas desejadas, chamadas de “alvo”.

Isso é feito ajustando os valores dos pesos da rede, conforme explicado por Makin (2006).

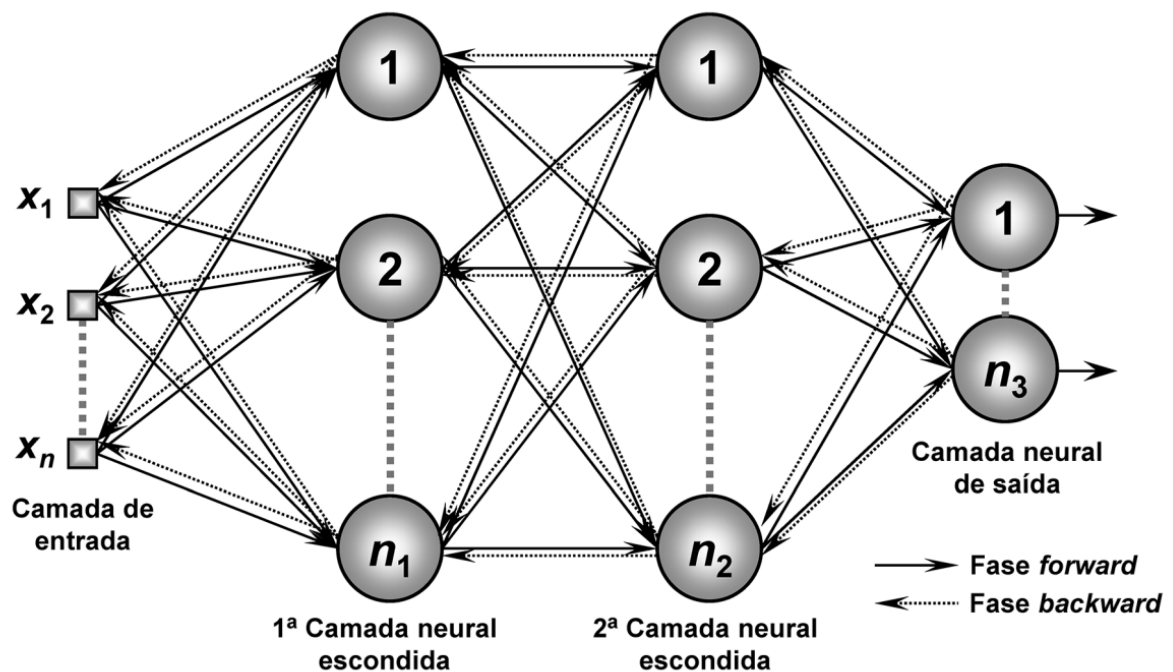
2.1.2.3 Backpropagation

Para atualizar os pesos de uma rede neural, a fim fazê-la aprender, é comumente utilizado o algoritmo de *backpropagation*, conhecido também como regra delta generalizada, ao decorrer do treinamento (Silva; Spatti; Flauzino, 2016). Essa é uma técnica de treinamento supervisionado utilizada para redes do tipo MLP proposto em 1986 por HAYKIN (2001). Esse algoritmo tem a capacidade de ajustar gradualmente os valores dos pesos, visando otimizar a saída da rede treinada (Furtado, 2019). O algoritmo consiste na minimização do erro decorrente do resultado obtido na saída e do resultado real.

O *backpropagation* é um algoritmo iterativo, ou seja, a mudança dos pesos ocorre de forma incremental. Os pesos são alterados proporcionalmente à sua influência no erro, quanto maior a influência do peso w_m , maior a mudança que o algoritmo deve fazer nesse peso (Makin, 2006).

O algoritmo é composto por duas fases: propagação adiante (*forward*) e retropropagação (propagação reversa - *backward*), em que cada fase percorre a rede em um sentido (Silva; Spatti; Flauzino, 2016), como ilustra a Figura 4.

Figura 4 – Backpropagation: propagação e retropropagação.



Fonte: Silva, Spatti e Flauzino (2016).

Na primeira fase, a *forward*, os sinais x_1, x_2, \dots, x_n da amostra do conjunto de treinamento são inseridos na entrada da rede e propagados camada a camada até a produção das

respectivas saídas, inicialmente com pesos iniciados de forma aleatória (Silva; Spatti; Flauzino, 2016). Logo, “a aplicação desta fase visa tão somente obter as respostas da rede, levando-se em consideração apenas valores atuais de pesos sinápticos e limiares de seus neurônios, os quais permanecerão inalterados durante cada execução desta fase” (Silva; Spatti; Flauzino, 2016).

Após alcançar a camada de saída da rede, as respostas produzidas pela rede durante esse processo serão comparadas com as respostas desejadas. Essa comparação é possível por se tratar de um aprendizado supervisionado (Silva; Spatti; Flauzino, 2016).

O erro é essencialmente a diferença entre a ativação obtida de um nó de saída (y_j) e a ativação desejada (t_j) para aquele nó específico. O erro total é composto pela soma dessas diferenças para cada nó de saída. Adicionalmente, para evitar que erros positivos e negativos se anulem mutuamente, elevamos essas diferenças ao quadrado antes de somá-las (Makin, 2006). Por fim, o resultado é ajustado por um fator de escala de $\frac{1}{2}$, calculado da seguinte forma:

$$E = \frac{1}{2} \sum_{j=1}^J (t_j - y_j)^2. \quad (2.1)$$

A Equação (2.1) se aplica somente quando j é uma camada de saída, pois é a única camada para a qual o erro é definido (Makin, 2006).

Em função do valor de erro aplica-se em seguida, a segunda fase, chamada de retropropagação, que ocorre na direção contrária a fase *forward*, ou seja, da camada de saída até a camada de entrada. Diferentemente da etapa anterior, os ajustes dos pesos sinápticos e limiares de todos os neurônios da rede são executadas no decorrer desta fase (Silva; Spatti; Flauzino, 2016).

A alteração do peso para um peso que conecta um nó na camada k a um nó na camada j é dada por:

$$\Delta w_{kj} = -\alpha \frac{\partial E}{\partial w_{kj}}, \quad (2.2)$$

onde ∂E é o somatório do erro quadrático médio total, apresentado na Equação (2.1) e α é um parâmetro livre chamado de “taxa de aprendizado” (em inglês, *Learning Rate*), esse valor é definido antes do treinamento. O sinal negativo indica que as mudanças nos pesos estão na direção de diminuição do erro (Makin, 2006).

Usando a regra da cadeia na Equação (2.2), podemos estabelecer que:

$$\Delta w_{kj} = -\alpha \frac{\partial E}{\partial w_{kj}} = -\alpha \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial x_j} \frac{\partial x_j}{\partial w_{kj}} = \alpha \cdot s_k \cdot e_j, \quad (2.3)$$

onde s_k é o valor de entrada recebido pela conexão k no neurônio j e e_j é o valor calculado do erro no neurônio j . Com isso, o valor do peso atualizado é definido pela seguinte equação:

$$w_{kj} \text{ (atualizado)} = w_{kj} \text{ (anterior)} + \Delta w_{kj} = w_{kj} \text{ (anterior)} + \alpha \cdot s_k \cdot e_j. \quad (2.4)$$

A repetição sequencial das etapas *forward* e *backward* provoca a automática adaptação dos pesos sinápticos e limiares dos neurônios a cada passo, resultando na gradual redução dos

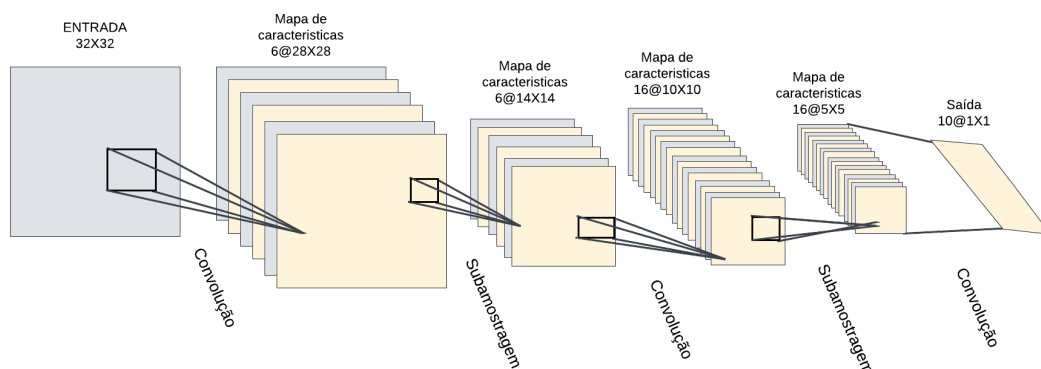
erros gerados pelas saídas da rede em comparação com as saídas desejadas (Silva; Spatti; Flauzino, 2016).

2.1.3 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (do inglês, *Convolutional Neural Networks*, ou CNNs) são modelos de redes neurais mais amplamente utilizados para processamento de imagens na atualidade. “A rede convolutiva é um *perceptron* de múltiplas camadas projetado especificamente para reconhecer formas bidimensionais com um alto grau de invariância quanto a translação, escalonamento, inclinação e outras formas de distorção” (HAYKIN, 2001).

Cada camada da CNN é responsável por uma sub-tarefa específica. As principais camadas que compõem uma Rede Neural Convolutiva são: camada de convolução, *pooling* (subamostragem) e camada totalmente conectada (HAYKIN, 2001). Essa estrutura é ilustrada na Figura 5 e cada camada é explicada nas seções seguintes.

Figura 5 – Estrutura de uma rede convolutiva.



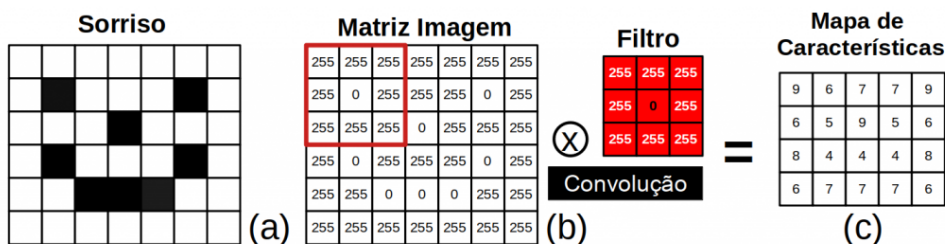
Fonte: Adaptado pela Autora de LeCun *et al.* (1998)

2.1.3.1 Camada Convolutiva

A Camada Convolutiva recebe esse nome pois é nela que ocorre a operação de convolução. A convolução é uma operação que aplica repetidamente um filtro (*kernel*) aos dados de entrada para gerar um mapa de características (*features*). O *kernel* normalmente possui dimensões reduzidas e representa padrões locais a serem detectados (Silva, 2022).

Essa operação utiliza uma janela deslizante que percorre a imagem de entrada. Em cada posição da janela, é realizado um produto entre os valores da região da imagem coberta pela janela e os pesos do filtro de convolução correspondente (Zhang *et al.*, 2021). Esses pesos são aprendidos durante o treinamento (Silva, 2022). Por fim, produz um valor que representa uma característica local da imagem, como é representado na Figura 6.

Figura 6 – Diagrama de funcionamento da convolução. (a) Imagem de entrada. (b) Filtro aplicado a matriz de pixels imagem. (c) Mapa de características gerado.



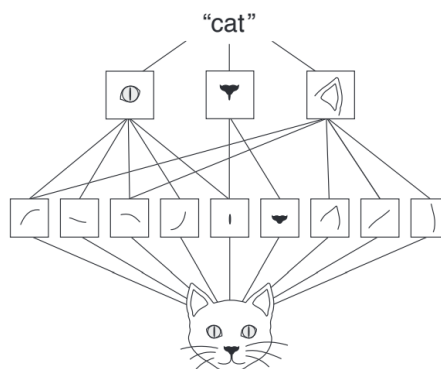
Fonte: Disponível em: <<http://www2.decom.ufop.br/imobilis/deep-learning-e-a-industria-4-0-introducao-as-redes-neurais-convolucionais/>>. Acesso em 28 de maio de 2023.

Há duas propriedades interessantes nas redes convolucionais:

1. Segundo *Zhang et al. (2021)*, os padrões aprendidos pela convolução são independentes de localização, o que implica que um padrão aprendido pode ser detectado em qualquer posição da imagem. Isso leva à conclusão de que os padrões aprendidos são invariantes à translação.

2. Essas redes podem produzir hierarquias espaciais de padrões, como apresentando na Figura 7. Ou seja, uma primeira camada convolucional pode aprender pequenos padrões locais, e uma segunda camada convolucional aprenderá padrões maiores compostos pelas característica aprendidas na primeira camada (*Chollet, 2021*).

Figura 7 – Hierarquia de padrões aprendidos em uma rede neural profunda.



Fonte: *Chollet (2021)*.

Para imagens RGB, ou seja, que possuem três canais de entrada: vermelho (*red*), verde (*green*) e azul (*blue*), a dimensão no eixo de profundidade do filtro é três, uma para cada canal de cor (*Chollet, 2021*).

Por fim, a camada convolucional em uma CNN desempenha um papel crucial na extração de características relevantes das imagens. Essa operação é essencial para a capacidade da CNN de reconhecer objetos, detectar bordas, texturas e outros elementos importantes em tarefas de visão computacional.

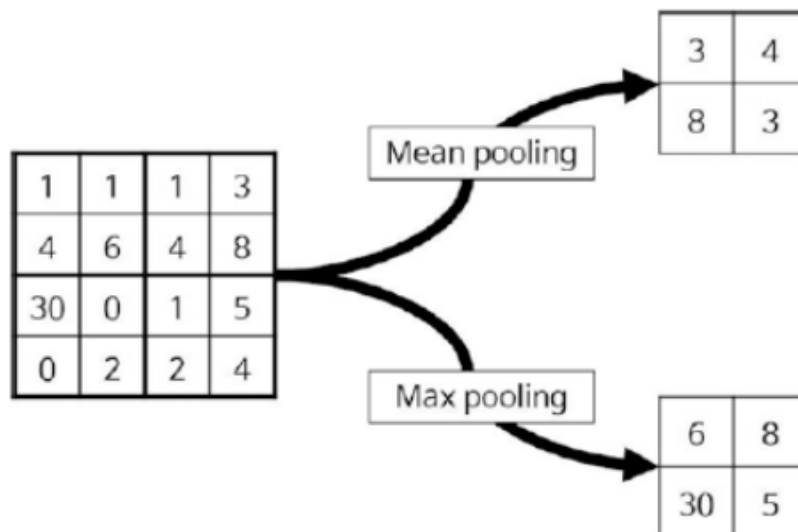
2.1.3.2 Camada de Pooling

A camada de *pooling*, também chamada de subamostragem, desempenha um papel importante nas redes neurais convolucionais ao permitir a redução da dimensionalidade dos mapas de características gerados pela camada convolucional, enquanto preserva as informações mais relevantes. Para realizar essa operação, a camada de *pooling* utiliza uma janela de formato fixo que desliza por todas as regiões e aplica os operadores determinísticos a essas regiões para obter um único valor representativo. As operações mais comuns de *pooling* são chamadas de *max pooling* e *average pooling* (Zhang *et al.*, 2021).

No *max pooling*, o valor máximo dentro de cada janela de *pooling* é selecionado como o valor de saída. No *average pooling*, é calculada a média dos valores dentro de cada janela de *pooling* (Zhang *et al.*, 2021).

Essas operações de *pooling* auxiliam no controle do número de parâmetros, reduzindo o custo computacional e tornando o processamento mais eficiente. A Figura 8 apresenta um exemplo de operação de *pooling* com *mean pooling* e *max pooling*, com janela de dimensão 2x2.

Figura 8 – Operação de *pooling* com média (*mean pooling*) e com máximo (*max pooling*).



Fonte: Kim (2017).

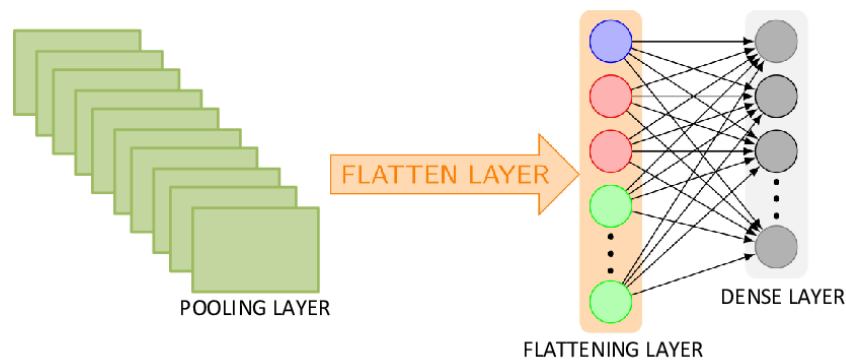
2.1.3.3 Camada Totalmente Conectada

Em redes neurais convolucionais (CNNs), é comumente utilizado camadas totalmente conectada, do inglês *fully connected*, após várias camadas convolucionais e *pooling*. Nesse tipo de camada, cada neurônio possui um peso associado a cada elemento do vetor de entrada (Ponti; Costa, 2018), e nela ocorre o aprendizado de forma convencional a uma MLP, como explicado na Seção 2.1.2.2. No entanto, para fazer a transição entre uma camada convolucional, que produz um tensor tridimensional, e uma camada totalmente conectada, é necessário vetorizar o tensor

(Ponti; Costa, 2018), esse processo é comumente chamado de *flattening* (Silva, 2022) e está ilustrado na Figura 9.

Ou seja, a camada totalmente conectada, recebe os recursos extraídos das camadas anteriores, convolucional e *pooling*, e os transforma em uma representação compacta, que é então alimentada para a camada de saída para a classificação. Essa camada também é conhecida como camada densa ou camada de classificação.

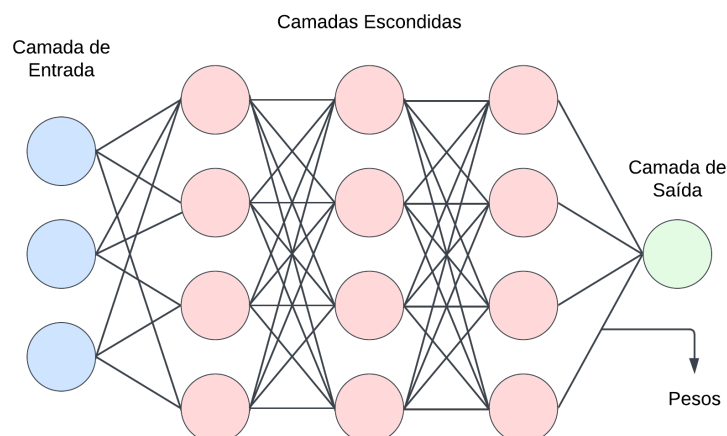
Figura 9 – Operação de *flattening*.



Fonte: Khan *et al.* (2022)

Sua estrutura básica é composta por três partes principais: entrada, intermediária (ou oculta) e saída. Na entrada, os dados são recebidos como um vetor numérico, passando então por operações de multiplicação de matrizes e aplicação de funções de ativação nas camadas intermediárias, onde as características dos dados são extraídas. Na camada de saída, os valores resultantes são interpretados para fornecer a saída desejada, como classificação ou regressão. Cada neurônio está conectado a todos os neurônios da camada anterior, permitindo à rede aprender representações complexas dos dados de entrada. Como ilustrado na Figura 10.

Figura 10 – Estrutura e operações da camada *fully-connected*.



Fonte: Goldberg (2021).

2.1.4 Transfer Learning (TL)

Segundo Hosna *et al.* (2022) apud Matt, é possível definir *Transfer Learning* (TL) como uma categoria dentro de *Machine Learning* (ML). De acordo com Hosna *et al.* (2022) apud Yoshua et al. define TL como uma técnica que treina modelos atuais com modelos anteriormente já treinados para tarefas relacionadas e similares e, por isso, Hosna *et al.* (2022) apud Jason escreve que TL é uma ferramenta de otimização que melhora o desempenho de novos modelos que a utilizam.

Há diferentes métodos para utilizar os modelos já pré-treinados em uma nova base de dados, e um desses métodos é o chamado *fine-tuning*. O processo de *fine-tuning* envolve essencialmente a continuação do treinamento a partir dos pesos iniciais do modelo, porém utilizando um subconjunto específico dos dados disponíveis. Como esse subconjunto de novos dados pode conter classes diferentes daquelas presentes na base original, se a intenção é desenvolver um novo classificador, faz-se necessário descartar a última camada de classificação da rede e acrescentar uma nova. Essa nova camada de saída deve corresponder ao número desejado de classes e requer um treinamento completo, partindo de uma inicialização aleatória (Vieira *et al.*, 2017).

Segundo Vieira *et al.* (2017), há várias abordagens para realizar *fine-tuning*, como:

1. Permitir que o algoritmo ajuste todos os pesos da rede com base nos novos dados.
2. Congelar algumas camadas e permitir o ajuste dos parâmetros de um subconjunto de camadas.
3. Criar novas camadas adicionais com configurações diferentes.

2.1.5 Classificação Hierárquica

Segundo os estudos de Silla e Freitas (2010), a classificação hierárquica pode ser considerada uma variante específica da classificação estruturada, na qual o resultado do algoritmo é determinado por meio de uma taxonomia de classes. Embora inicialmente direcionada a taxonomias em formato de árvore, essa definição pode ser estendida para taxonomias estruturadas em forma de grafo direcionado acíclico.

2.1.5.1 Classificação Plana

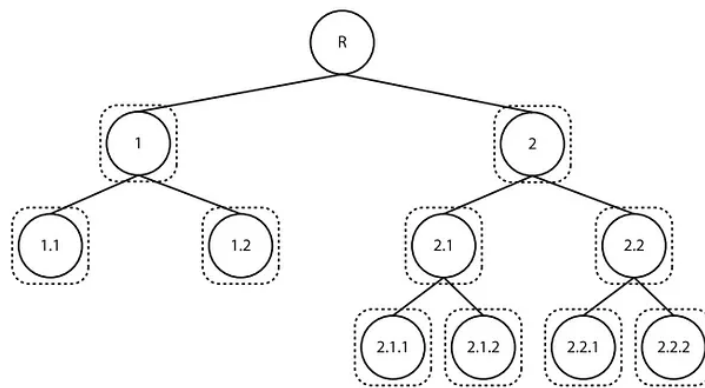
A abordagem mais simples e direta para a classificação taxonômica é conhecida como classificação plana. Nesse método, não se consideram as categorias-pai e os exemplos são diretamente classificados em seus rótulos finais, localizados no nível das folhas da hierarquia (Silla; Freitas, 2010).

2.1.5.2 Classificadores Locais Estruturados Hierarquicamente

Silla e Freitas (2010) definem essa abordagem como a mais difundida na literatura especializada. Ela utiliza uma taxonomia de dados predefinida para estabelecer uma hierarquia de classificadores. Tradicionalmente, existem três métodos empregados para alcançar esse objetivo:

- **Classificador Local por Nó:** Aqui, um classificador binário é treinado para cada nó presente na estrutura hierárquica. Este classificador busca determinar se um exemplo pertence ou não à classe específica associada ao nó. A Figura 11 ilustra essa configuração, onde cada nó tracejado representa a aplicação de um classificador para discernir a classe correspondente.

Figura 11 – Classificadores locais por nó.

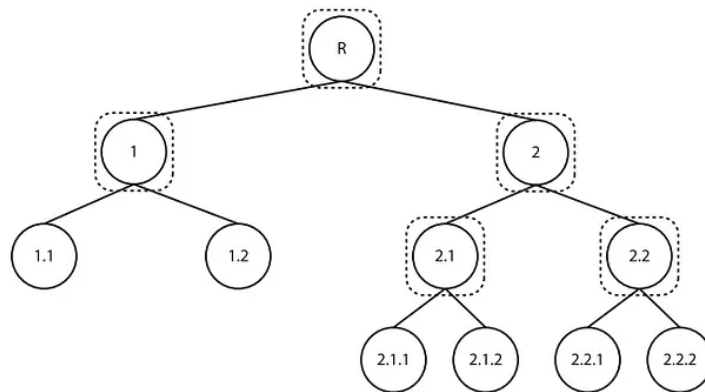


Fonte: Disponível em: <<https://medium.com/@anwarhermuche/um-guia-pratico-sobre-classificacao-hierarquica-e723e5680c15>>.

Acesso em 10 de janeiro de 2024.

- **Classificador Local por Nó Pai:** Este método implica o treinamento de um classificador multiclasse para cada nó pai na hierarquia, com o propósito de discernir entre suas subclasses. A Figura 12 ilustra essa configuração, onde um classificador multiclasse é aplicado em cada nó pai, representado por um tracejado.

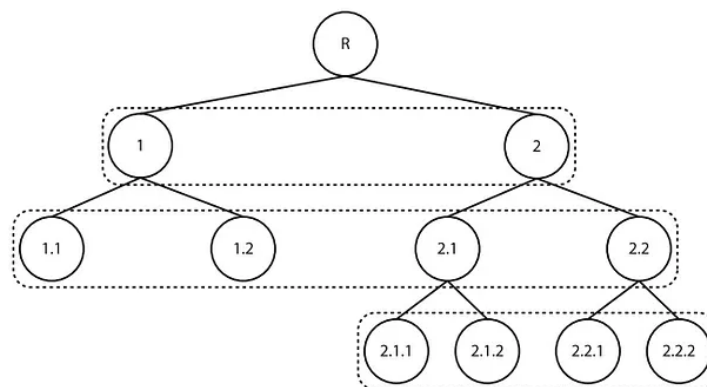
Figura 12 – Classificadores locais por nó pai.



Fonte: Disponível em: <<https://medium.com/@anwarhermuche/um-guia-pra-atico-sobre-classificacao-hierarquica-e723e5680c15>>. Acesso em 10 de janeiro de 2024.

- **Classificador Local por Nível:** Nesta abordagem, um classificador multiclasse é treinado para cada nível da hierarquia. Cada um desses classificadores aborda a distinção entre as diferentes classes existentes nesse nível. A Figura 13 ilustra como seria essa implementação, onde cada tracejado representa um nível e terá um classificador associado.

Figura 13 – Classificadores locais por nível.



Fonte: Disponível em: <<https://medium.com/@anwarhermuche/um-guia-pra-atico-sobre-classificacao-hierarquica-e723e5680c15>>. Acesso em 10 de janeiro de 2024.

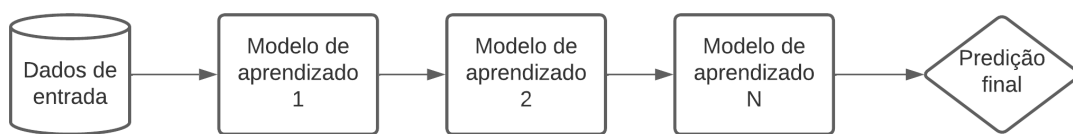
2.1.6 Ensemble Learning

De acordo com Mienye e Sun (2022), o uso do chamado *ensemble learning* envolve a combinação de diversos algoritmos de aprendizado de máquina para alcançar resultados melhores do que o uso de cada algoritmo separadamente. Em vez de depender apenas de um modelo, essa

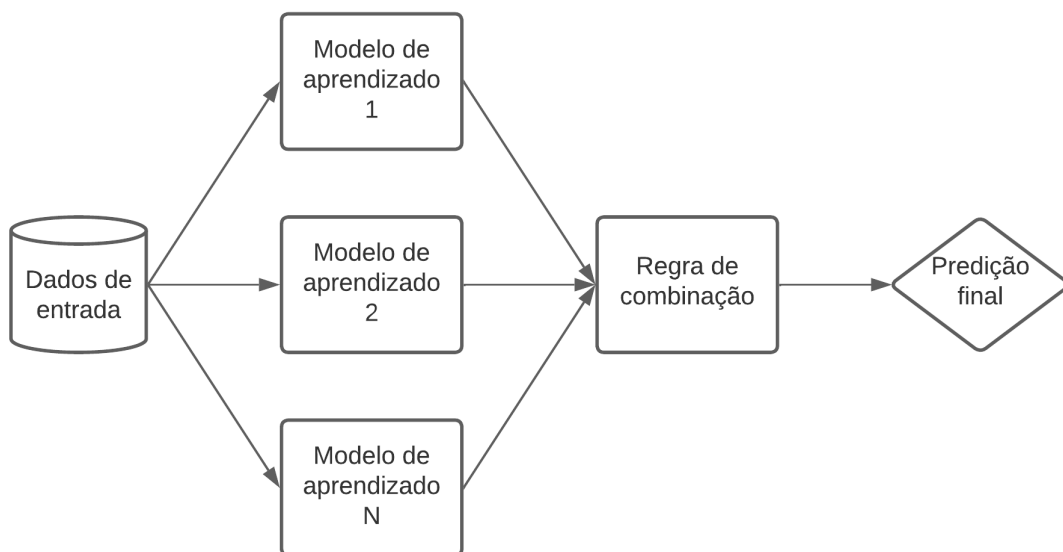
abordagem une as previsões de diferentes algoritmos usando uma técnica específica, resultando em previsões mais precisas.

Esses métodos podem ser divididos em conjuntos paralelos e sequenciais. Nos conjuntos paralelos, os algoritmos são treinados de forma independente e suas previsões são combinadas usando um método específico, tal método está ilustrado na Figura 14(b). Já os conjuntos sequenciais, ao contrário dos conjuntos paralelos, não treinam os modelos base independentemente, como ilustrado na Figura 14(a). Eles são treinados de forma iterativa, permitindo que cada modelo, em cada iteração, aprenda a corrigir os erros cometidos pelo modelo anterior (Mienye; Sun, 2022).

Figura 14 – Diagramas de blocos de *ensemble learning* sequencial e paralelo.



(a) Diagrama de blocos do *ensemble learning* sequencial.



(b) Diagrama de blocos do *ensemble learning* paralelo.

Fonte: Adaptado pela Autora de Mienye e Sun (2022).

Os conjuntos paralelos podem ser classificados como homogêneos ou heterogêneos, dependendo da semelhança dos modelos base. Os conjuntos homogêneos consistem em modelos construídos usando o mesmo algoritmo de ML, enquanto os conjuntos heterogêneos são compostos por modelos de algoritmos diferentes (Mienye; Sun, 2022).

Existem diversas regras de combinação no aprendizado de conjuntos, como o *bagging*, *boosting*, *hard voting*, *averaging/soft voting*, entre outros. Aqui, detalharemos as regras de votação: *hard voting* e *averaging/soft voting*, pois segundo Mienye e Sun (2022) é o mecanismo mais

comum para combinar modelos base de conjunto e é o que será utilizado durante o presente trabalho.

- **Hard Voting:** Os conjuntos de modelos de votação envolvem o treinamento de diferentes modelos de aprendizado de máquina. Ao combinar a saída desses modelos, busca-se uma previsão consolidada. A principal questão que surge é a estratégia para combinar efetivamente esses resultados. Uma das abordagens mais simples e confiáveis é inspirada na técnica clássica de votação. Na votação majoritária, a classe que recebe a maioria dos votos se torna a escolha preferencial. Geralmente, observa-se que essa classe resultante possui uma precisão maior do que qualquer modelo individual (Kumar; Mayank, 2020).
- **Averaging/Soft Voting:** *Averaging* oferece outra abordagem para combinar as saídas de diferentes classificadores. A distinção fundamental entre votação e *averaging* reside no processo: na média, são consideradas as probabilidades de previsão de cada classe individualmente, retirando-se a média dessas probabilidades resultantes (Kumar; Mayank, 2020).

2.1.7 Métricas de Avaliação

Quando se constrói um classificador utilizando *machine learning*, é importante avaliar o quão bem o modelo é capaz de fazer as classificações. Durante o treinamento, validação e teste do modelo de aprendizado, várias métricas podem ser utilizadas para realizar essa avaliação. As métricas utilizadas nesse projeto foram: acurácia, *recall*, precisão e *F1-score*.

2.1.7.1 Matriz de Confusão

Segundo Castro e Ferrari (2017), é possível explicar, de forma geral, a matriz de confusão como uma matriz de dimensões $C \times C$, onde C é o número de classes presente no problema de classificação em questão. Nessa matriz, as linhas indicam as classes verdadeiras (y), e as colunas indicam as classes previstas ($f(x)$). Cada célula da matriz é um contador que é incrementado com base no resultado da comparação entre $f(x)$ e y . As classes corretamente classificadas geram os valores que estão na diagonal principal da matriz de confusão.

Figura 15 – Exemplo de matriz de confusão.

		Classe Predita ($f(x)$)		
		Classe 1	Classe 2	Classe 3
Classe Esperada (y)	Classe 1	3	1	1
	Classe 2		3	1
	Classe 3			1

Fonte: Castro e Ferrari (2017).

A Figura 15 mostra um exemplo de matriz de confusão, onde podemos observar, segundo a definição de matriz de confusão, que a Classe 1 do exemplo foi classificada corretamente em três de suas amostras e ocorreu uma confusão com as Classes 2 e 3 com uma amostra cada.

Em uma matriz de confusão existem quatro tipo de valores presente nas células: Verdadeiro positivo (VP), Falso positivo (FP), Verdadeiro negativo (VN) e Falso negativo (FN) (Castro; Ferrari, 2017). Cada um deles possui um significado com base na comparação entre o resultado esperado e o resultado obtido. Castro e Ferrari (2017) explica cada um deles como:

- **Verdadeiro positivo (VP):** classificação correta na classe positiva, ou seja, pertence à classe positiva e foi classificado como pertencente à classe positiva.
- **Falso positivo (FP):** classificação incorreta na classe positiva, ou seja, pertence à classe negativa e foi classificado como pertencente à classe positiva .
- **Verdadeiro negativo (VN):** classificação correta na classe negativa, ou seja, pertence à classe negativa e foi classificado como pertencente à classe negativa.
- **Falso negativo (FN):** classificação incorreta na classe negativa, ou seja, pertence à classe positiva e foi classificado como pertencente à classe negativa.

Na Tabela 2.1, há uma representação dos valores previamente mencionados dispostos em uma matriz de confusão para classificação *multiclasse*.

Tabela 2.1 – Matriz de Confusão - Classificação Multiclasse

		Predição			
		Classe A	Classe B	Classe C	Classe D
Realidade	Classe A	VP(A)	FN(A)	FN(A)	FN(A)
	Classe B	FN(B)	VP(B)	FN(B)	FN(B)
	Classe C	FN(C)	FN(C)	VP(C)	FN(C)
	Classe D	FN(D)	FN(D)	FN(D)	VP(D)

Fonte: Criado pela Autora.

Para realizar uma avaliação abrangente dos diversos cenários possíveis, existem várias métricas de avaliação derivadas de uma matriz de confusão (Castro; Ferrari, 2017), como: acurácia, *recall*, precisão e *F1 score*.

2.1.7.2 Acurácia

A acurácia, do inglês *accuracy*, (ACC) avalia o percentual de classificações corretas (Harrison, 2019). Ela é calculada por:

$$\text{Acurácia} = \frac{\text{Previsões Corretas}}{\text{Total de Previsões}} \quad (2.5)$$

Fazendo uma relação com os valores presentes na matriz de confusão tem-se:

$$\text{Acurácia} = \frac{VP + VN}{VP + FP + VN + FN}. \quad (2.6)$$

2.1.7.3 Recall

Recall, também conhecido como sensibilidade (*sensitivity*), avalia a porcentagem de valores positivos classificados corretamente (Harrison, 2019). É a proporção de previsões positivas corretamente classificadas em relação ao total de exemplos positivos. O *Recall* é definido por:

$$\text{Recall} = \frac{\text{Previsões Positivas Corretas}}{\text{Exemplos Positivos}}. \quad (2.7)$$

Fazendo uma relação com os valores presentes na matriz de confusão temos:

$$\text{Recall} = \frac{VP}{VP + FN}. \quad (2.8)$$

2.1.7.4 Precisão

Precisão, do inglês *precision*, mede a proporção de predições positivos corretamente classificados em relação ao total de predições classificados como positivos. O cálculo é dado por:

$$\text{Precisão} = \frac{\text{Previsões Positivas Corretas}}{\text{Previsões Positivas}}. \quad (2.9)$$

Fazendo uma relação com os valores presentes na matriz de confusão temos:

$$\text{Precisão} = \frac{VP}{VP + FP}. \quad (2.10)$$

2.1.7.5 F1-Score

F1-Score, ou simplesmente F1, é a métrica que combina a precisão e o *recall* em uma única medida. Trata-se da média harmônica ponderada da precisão e do *recall* (Harrison, 2019). Ela é definida da seguinte forma:

$$F1\text{-score} = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}. \quad (2.11)$$

2.1.8 Hugging Face

Segundo Ben Lutkevich (2023), a plataforma *Hugging Face* foi desenvolvida pela empresa americana *Hugging Face Inc.* fundada em 2016. Inicialmente, a empresa desenvolveu um aplicativo de *chatbot* com o mesmo nome para adolescentes. A empresa mudou seu foco para ser

uma plataforma de aprendizado de máquina após disponibilizar como *open source* o modelo por trás do aplicativo de *chatbot*.

Atualmente *Hugging Face* é uma plataforma e comunidade de aprendizado de máquina e ciência de dados que auxilia os usuários na construção, implantação e treinamento de modelos de aprendizado de máquina (Ben Lutkevich, 2023).

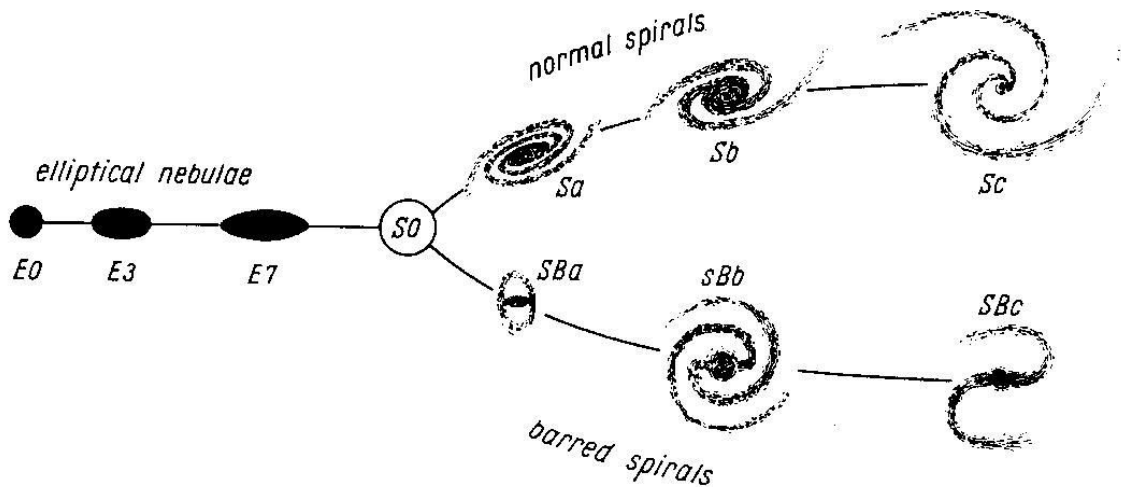
A plataforma do *Hugging Face* fornece a infraestrutura para demonstrar, executar e implantar inteligência artificial (IA) em aplicações em tempo real. Os usuários também podem navegar por modelos e conjuntos de dados que foram compartilhados por outras pessoas. *Hugging Face* é frequentemente comparado ao *GitHub* do aprendizado de máquina, pois permite que desenvolvedores compartilhem e testem abertamente seu trabalho (Ben Lutkevich, 2023).

2.1.9 Sistema de Hubble de classificação morfológica de galáxias

Antes da descoberta das galáxias, astrônomos haviam observado corpos extensos e difusos, aos quais denominaram “nebulosas” (Oliveira, 2013). Entretanto, em 1923, utilizando o telescópio de Monte Wilson, Edwin Powell Hubble (1889-1953), a fim de comprovar cientificamente a “hipótese dos universos-ilhas” do filósofo alemão Immanuel Kant (1724-1804), a qual implicava que as nebulosas eram sistemas estelares independentes, identificou estrelas individuais em uma grande “nebulosa” chamada de “Grande Nebulosa de Andrômeda”. Por meio dessas estrelas, Hubble calculou a distância até elas e, conseqüentemente, até a nebulosa em que estavam, a “Grande Nebulosa de Andrômeda”. Como resultado obteve que a distância até a nebulosa era muito maior que o tamanho da própria Via Láctea. Concluindo, então, que era na verdade um sistema estelar tão grandioso quanto a Via Láctea, esses passaram a ser chamados de galáxias (Soares, 2021).

Apesar de muito parecidas, as galáxias diferem entre si. Uma diferença estudada e observada é a morfologia, que se refere às formas das galáxias, sendo grande parte delas com formas espirais e elípticas. Aquelas que não têm forma definida são chamadas irregulares (Oliveira, 2013). Hubble, em seu livro de 1936 “The Realm of the Nebulae” (Hubble, 1936), publicou um diagrama conhecido como a “forquilha de Hubble” mostrada na Figura 16.

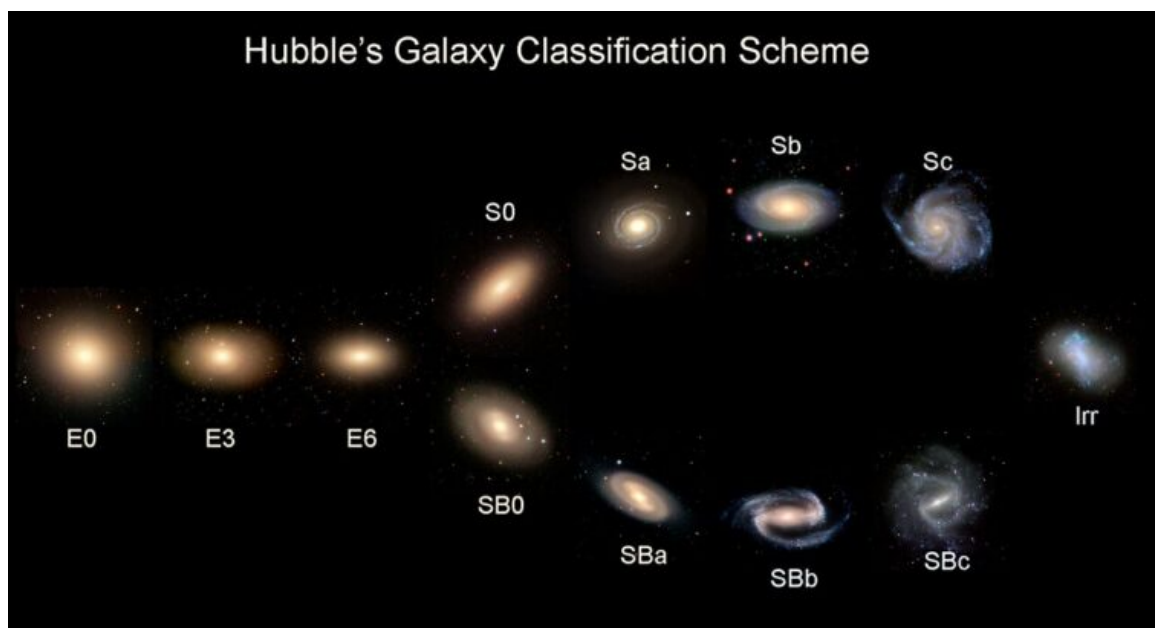
Figura 16 – Forquilha original do diapasão de Hubble publicado em Hubble (1936).



Fonte: Hubble (1936)

Analisando a Figura 16, observamos que as galáxias elípticas estão localizadas no cabo da forquilha, enquanto nos dois ramos da forquilha estão as galáxias espirais, tanto com quanto sem barra (Soares, 2021). No vértice da forquilha, encontramos a classe das galáxias lenticulares, que representam uma transição morfológica entre as galáxias elípticas e espirais (Soares, 2021). No esquema original, as galáxias irregulares não são representadas, já que essa categoria engloba todas aquelas que não se enquadram nas outras classificações de Hubble. No entanto, existem versões estendidas da forquilha que incluem essa classe, como mostrado na Figura 17.

Figura 17 – Forquilha original do diapasão de Hubble estendida.



Fonte: Valk et al. (2023)

A avaliação da morfologia das galáxias tem grande relevância para o estudo da sua formação e evolução. Esse estudo auxilia em pistas para a compreensão do surgimento do

universo (Calleja; Fuentes, 2004).

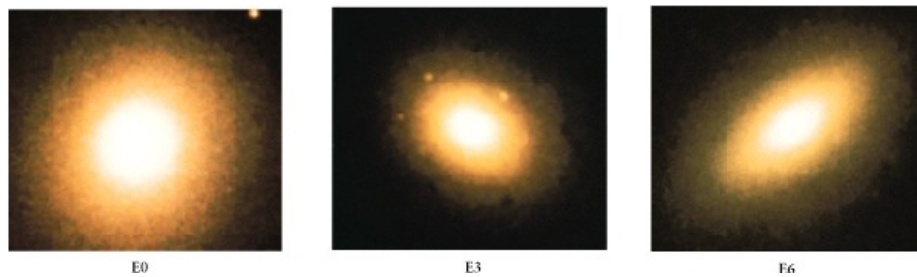
2.1.9.1 Elípticas

As galáxias elípticas possuem esse nome devido a sua forma esférica ou elipsoidal, ou seja, a forma de um círculo achatado (Soares, 2007; Saraiva, 2014).

Conforme apresentando na Figura 16, na extremidade estão as galáxias elípticas $E0$, essas não apresentam nenhum achatamento, seu corpo é circular. A medida que o achatamento aumenta, o algarismo colocado à direita da letra “E” também aumentava, logo, como mostrado ainda na Figura 16, “E7” possui um maior achatamento (Saraiva, 2014). Hubble subdividiu as elípticas da seguinte forma: “[...]classes de $E0$ a $E7$, de acordo com o seu grau de achatamento (n), sendo $n = \frac{10(ab)}{a}$; a é o eixo maior da elipse e b é o eixo menor da elipse.” (Saraiva, 2014).

A Figura 18 mostra exemplos de galáxias elípticas com a variação do achatamento.

Figura 18 – Exemplos de galáxias elípticas.

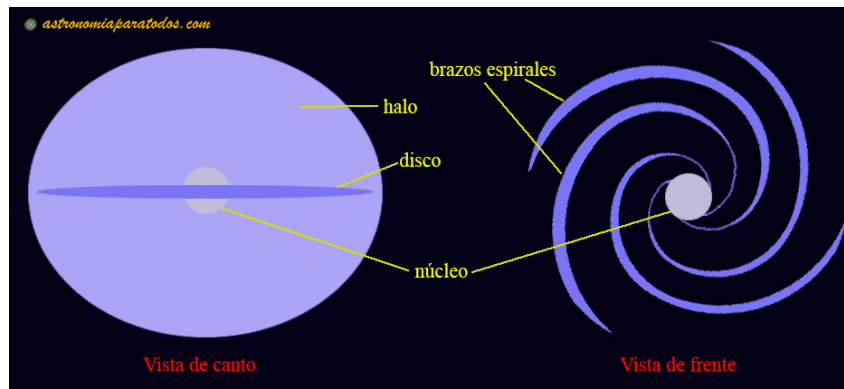


Fonte: Saraiva (2014).

2.1.9.2 Espirais

Galáxias espirais apresentam uma clara estrutura espiral quando vistas de frente. Exemplos de galáxia desse tipo é a Andrômeda (M31) e a própria galáxia Via Láctea (Saraiva, 2014). Também são chamadas por essa nome por possuírem os chamados “braços espirais”, essas estruturas espiralam em torno do núcleo central das galáxias (Soares, 2007). As galáxias espirais possuem em sua estrutura um núcleo, um disco, um halo, e braços espirais (Saraiva, 2014) como ilustrado na Figura 19.

Figura 19 – Partes de uma galáxia espiral.



Fonte: <https://astronomiaparatos.com/2020/03/03/tipos-de-galaxias/>.
Acesso em 24 de junho de 2023.

Na forquilha de galáxias de Hubble, apresentada na Figura 16, essas galáxias são representadas pela letra “S” seguida de “a”, “b” ou “c”. Essas três letras após o “S” descrevem a diferenciação das galáxias espirais em relação ao tamanho do núcleo e ao grau de desenvolvimento/enrolamento dos braços espirais (Saraiva, 2014) criando sub-divisões. A Tabela 2.2 mostra o que cada letra diz respeito a estrutura das galáxias espirais.

Tabela 2.2 – Sub-divisões das galáxias espirais

Classificação	Características
a	Núcleo maior, braços pequenos e bem enrolados
b	Núcleo e braços intermédios
c	Núcleo menor, braços grandes e mais abertos

Fonte: (Saraiva, 2014). Acesso em 24 de junho de 2023.

A Figura 20 mostra exemplos de galáxias espirais Sa, Sb e Sc respectivamente.

Figura 20 – Exemplos de galáxias espirais: fotos de galáxias obtidas por Jim Wray, no McDonald Observatory.



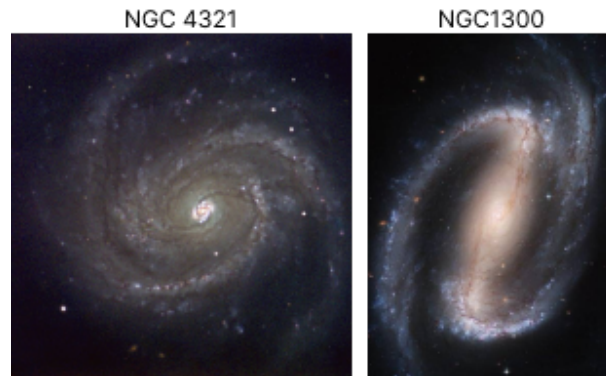
Fonte: Saraiva (2014).

2.1.9.3 Espirais Barradas

Hubble notou que havia dois tipos de galáxias espirais as galáxias espirais normais, onde os braços se destacam diretamente do núcleo da galáxia, como falado na seção anterior 2.1.9.2. Já o segundo tipo observado são as galáxias espirais “barradas”, nas quais os braços se

originam de uma estrutura semelhante a uma barra, que é uma estrutura central linear sobreposta ao núcleo. Essas galáxias barradas são representadas pelas letras “SB”, sendo a letra “S” referente a “espiral” (do inglês “*spiral*”) e a letra “B” significando “barrada” (Soares, 2007). Na Figura 21 é apresentado uma galáxia espiral e uma espiral barradas.

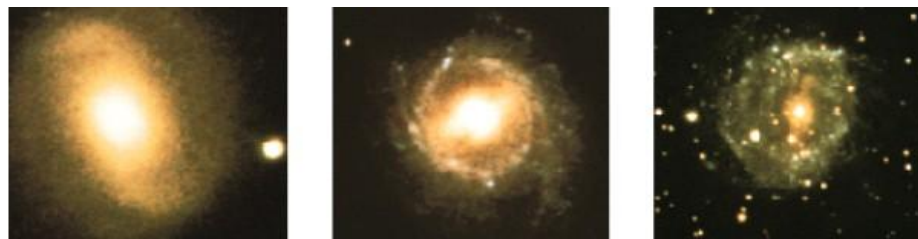
Figura 21 – Exemplos de galáxias espirais e espirais barradas, respectivamente.



Fonte: Saraiva (2014).

As galáxias espirais barradas também são sub-classificadas no esquema de Hubble por “a”, “b” e “c”, seguindo a mesma descrição das galáxias espirais não barradas. A Figura 22 mostra exemplos de galáxias espirais barradas de acordo com a forquilha de Hubble.

Figura 22 – Exemplo de galáxias espirais barradas



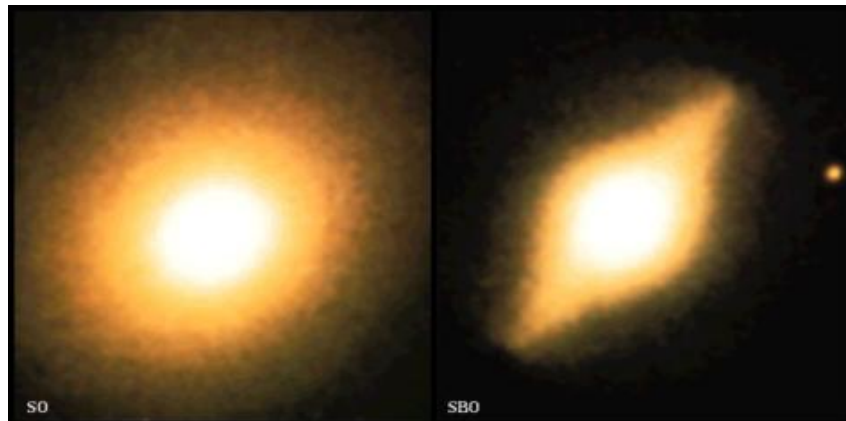
Fonte: Saraiva (2014).

2.1.9.4 Lenticulares

As galáxias lenticulares são parecidas com as espirais na sua estrutura possuindo núcleo, disco e halo, porém não possuem os braços espirais representando por Hubble como “S0”, o “0” do seu nome está relacionado com a sua forma, que é aproximadamente a de uma lente (Soares, 2007). Podem ser barradas “SB0” ou não barradas “S0”.

A Figura 23 mostra um exemplos de galáxias lenticulares “S0” e “SB0”.

Figura 23 – Fotografias de galáxias à esquerda SO e à direita SBO.



Fonte: Saraiva (2014).

2.1.9.5 Irregulares

Hubble classificou como irregulares todas as galáxias que não se enquadravam dentro dos tipos espiral e elíptico, ou seja, galáxias privadas de qualquer simetria circular ou rotacional (Saraiva, 2014).

A Figura 24 apresenta dois exemplos de galáxias irregulares mais conhecidas, são a Grande e a Pequena Nuvens de Magalhães, duas galáxias consideradas satélites da Via Láctea, visíveis a olho nu no Hemisfério Sul (Saraiva, 2014).

Figura 24 – Foto das galáxias irregulares Grande Nuvem de Magalhães e Pequena Nuvem de Magalhães, obtida por Wei-Hao Wang.



Fonte: Saraiva (2014).

2.1.9.6 As Classes

Além da diferenciação visual da forma das galáxias, cada classe também diz respeito a características de propriedades únicas de cada tipo de galáxia (Saraiva, 2014). A Tabela 2.3 resume as principais características das galáxias elípticas, espirais e irregulares (Saraiva, 2014).

Tabela 2.3 – Propriedades das Galáxias Espirais, Elípticas e Irregulares

Propriedade	Espirais	Elípticas	Irregulares
Massa (M)	10^9 a 10^{12}	10^5 a 10^{13}	10^8 a 10^{11}
Diâmetro (10^3 pc)	5 - 30	1 - 1000	1 - 10
Luminosidade (L)	10^8 a 10^{11}	10^6 a 10^{12}	10^7 a 2×10^9
População estelar	Velha e jovem	Velha	Velha e jovem
Tipo espectral	A a K	G a K	A a F
Gás	Bastante	Muito pouco	Bastante
Poeira	Bastante	Muito pouca	Varia
Cor	Azulada no disco Amarelada no bojo	Amarelada	Azulada
Estrelas mais velhas	10^{10} anos	10^{10} anos	10^{10} anos
Estrelas mais jovens	Recentes	10^{10} anos	Recentes

Fonte: Saraiva (2014).

2.2 Trabalhos Relacionados

Nesta seção são apresentados diversos trabalhos relacionados ao uso de técnicas de aprendizado profundo para a classificação de imagens de galáxias. Os estudos abordam diferentes conjuntos de dados e arquiteturas de redes neurais convolucionais.

No primeiro estudo, realizado por Misra *et al.* (2020), é proposta uma abordagem baseada em *Deep Learning* para classificar imagens de galáxias, assim como no segundo, Gharat e Dandawate (2022). Em um terceiro estudo, Cavanagh, Bekki e Groves (2021) focam na classificação morfológica de galáxias utilizando quatro arquiteturas diferentes de redes neurais. O quarto estudo apresentado, por Lin *et al.* (2022), propõe a utilização do *Vision Transformer* (ViT). Por fim, Fielding, Nyirenda e Vaccari (2022) implementam técnicas de aprendizado não supervisionado.

Ao final da seção é feita uma comparação dos resultados obtidos pelos métodos propostos em cada trabalho.

2.2.1 *Convoluting Cosmos: Classifying Galaxy Images Using Deep Learning*

O estudo conduzido por Misra *et al.* (2020) apresenta uma abordagem baseada em *deep learning* para classificar imagens de galáxias em três classes principais: elípticas, espirais e irregulares. A base de dados utilizada no projeto foi obtida do Kaggle (*Galaxy - Star separation*) e do site NASA Hubble-Space Gallery.

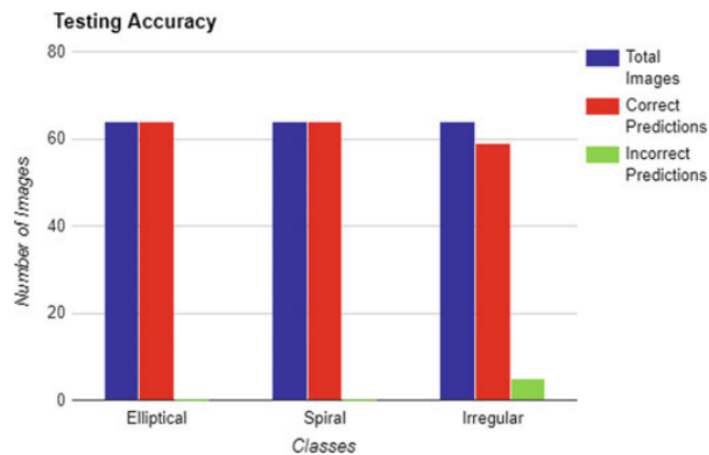
Devido ao desbalanceamento grande da base de dados utilizada, que têm apenas 11 amostra de imagens da classe de galáxias irregular, foi utilizado por Misra *et al.* (2020) *image augmentation* por meio do pacote *Augmentor* do Python, para gerar 1.615 novas imagens.

O modelo de CNN de Misra *et al.* (2020) é composta por uma camada de convolução seguida por quatro camadas ocultas e uma camada densa antes da camada de saída. Todas as

imagens foram redimensionadas para 256×256 pixels. O modelo foi treinado em 4.614 imagens por 200 épocas.

Analisando o gráfico da Figura 25 que mostra a precisão do teste de imagens de cada classe de galáxias, pode-se observar que a classe de galáxias irregulares obtém um pequeno valor maior de predições incorretas.

Figura 25 – Representação da precisão do teste usando o gráfico de barras.



Fonte: Misra *et al.* (2020).

O classificador teve sucesso na classificação das imagens com uma precisão de 97,3958% para as três classes.

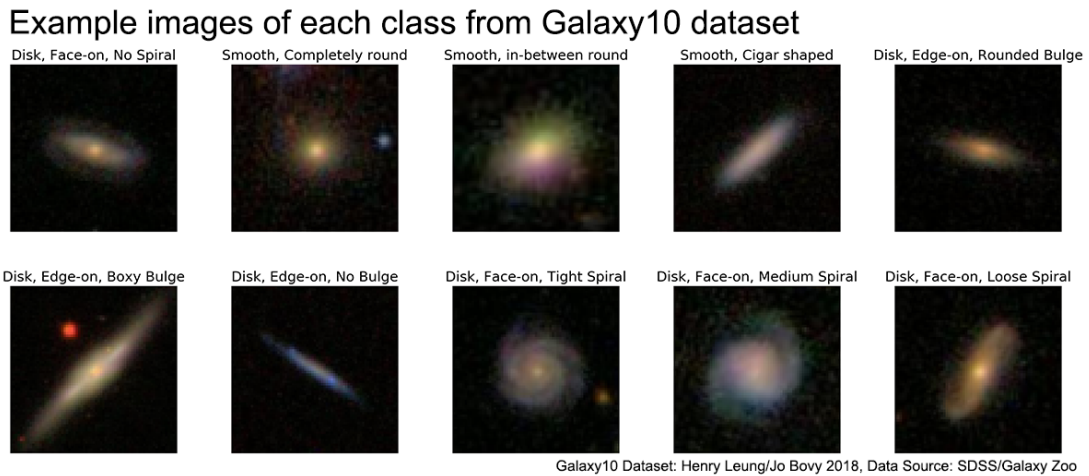
2.2.2 *Galaxy Classification: A deep learning approach for classifying Sloan Digital Sky Survey images*

Gharat e Dandawate (2022) propõem o uso de *deep learning* para classificar galáxias por meio de imagens do *Sloan Digital Sky Survey (SDSS)* com 21.785 amostras classificadas em 10 classes a partir do modelo estendido de “*Hubble Tuning Fork*”. Na Figura 26 há um exemplo e nome de cada uma das dez utilizadas no trabalho.

A base de dados utilizada - Galaxy10 Dataset, disponível em <<https://astronn.readthedocs.io/en/v1.0.0/galaxy10.html>> - no projeto implementado por Gharat e Dandawate (2022) já possui um pré-processamento nos dados em que seleciona apenas imagens em que mais de 55% das pessoas que a classificaram selecionaram a mesma classe, para contornar o problema de que, como o projeto Galaxy Zoo depende de voluntários, há chance de que esses cometam erros. As imagens possuem um desbalanceamento de dados não tratado.

O classificador desenvolvido possui cinco camadas de convolução com a função de ativação ReLU, e no final da extração das camadas de convolução, a saída é transformada em uma matriz 1D. No final, há duas camadas densas antes da camada de saída que usam a função

Figura 26 – Imagens de cada classe de galáxias do Galaxy10 Dataset



Fonte: Disponível em: <<https://astronn.readthedocs.io/en/v1.0.0/galaxy10.html>>. Acesso em 04 de julho de 2023.

de ativação ReLU. Na camada de saída, usou-se *Softmax* como função de ativação. Com essa descrição, pode-se concluir que o ajuste dos pesos da rede ocorreu do zero.

Ao final, com as 10 classes, o classificador apresentado no trabalho obteve uma acurácia geral nos testes de 84,78%.

2.2.3 Morphological classification of galaxies with deep learning: comparing 3-way and 4-way CNNs

Cavanagh, Bekki e Groves (2021) focam no treino e teste de quatro diferentes arquiteturas de redes neurais artificiais para classificação morfológica de galáxias em três classes: elípticas, lenticulares e espirais e quatro classes: elípticas, lenticulares espirais, irregulares/miscelânea) no conjunto de dados *Sloan Digital Sky Survey (SDSS) Data Release 4* que possui 14.034 imagens.

Foram utilizadas técnicas para aumento de dados, visto que a base de dados utilizada era relativamente pequena dada a complexidade da arquitetura. As técnicas de aumento de dados usadas foram: *Cropping*, *Rotation* e *Flipping/Mirroring*. Tais técnicas possibilitaram um aumento no conjunto de dados para 561.360 imagens.

A primeira arquitetura (C1) é a mais simples, possui um único bloco de duas camadas Conv2D sucessivas com MaxPool ao final, uma camada de *dropout*, e a camada densa antes da camada de saída. A segunda arquitetura (Diel) contém vários blocos convolucionais, começando com dois blocos de camadas alternadas de Conv2D e MaxPool, seguidos por um bloco de duas camadas Conv2D sucessivas com MaxPool ao final, há uma camada de *dropout*, e duas camadas densas antes da camada de saída. A terceira arquitetura (Alexnet) é baseada na AlexNet, uma rede pré-treinada e projetada para classificação geral de imagens. A quarta arquitetura (C2) foi projetada com uma CNN de três blocos - o primeiro com camadas de Conv2D e MaxPool, o

segundo duas camadas Conv2D sucessivas com MaxPool, e por fim camadas de Conv2D e MaxPool - uma única camada de *dropout* e duas camadas densas antes da camada de saída. Segundo os resultados do trabalho (Cavanagh; Bekki; Groves, 2021), a quarta rede, nomeada como C2, apresenta o melhor desempenho geral, uma vez que é a menos afetada pelo *overfitting*.

Os resultados do trabalho foram alcançados utilizando imagens de dimensões 100×100 pixels e 200×200 pixels. A melhor arquitetura, C2, obteve uma precisão de 83% e 81% na classificação de três e quatro classes, respectivamente. O trabalho destaca que esses resultados foram obtidos em imagens de tamanho 100×100 , enquanto em imagens com o dobro do tamanho, o modelo inspirado na AlexNet apresentou uma precisão um pouco superior, de 84% e 83%. No entanto, devido ao aumento do custo computacional em comparação com as outras arquiteturas, os dados de tamanho 100×100 foram considerados mais adequados.

2.2.4 A Hyperparameter Optimization for Galaxy Classification

A pesquisa de Şenel (2023) aborda a classificação morfológica de galáxias por meio de uma abordagem híbrida, considerando a importância desse processo para compreender a formação do universo.

A base de dados utilizada tem como base as imagens do *Sloan Digital Sky Survey (SDSS)* com um total de 21.785 amostras classificadas em 10 diferentes classes, sendo a mesma base de dados utilizada por Gharat e Dandawate (2022).

A pesquisa emprega três algoritmos meta-heurísticos: *Grey Wolf Optimizer (GWO)*, *Particle Swarm Optimization (PSO)* e *Artificial Bee Colony (ABC)* para otimizar a arquitetura da CNN, composta por nove camadas ocultas e duas camadas totalmente conectadas. O número de neurônios, coeficiente de aprendizado e tamanho do lote foram otimizados. A precisão de classificação alcançou 85%, sendo que o GWO obteve os melhores resultados, destacando-se na otimização manual da CNN. O estudo enfatiza o uso crescente da CNN na classificação, considerando-a uma técnica adequada para o problema de classificação de galáxias.

O GWO é detalhado como um algoritmo de otimização meta-heurística inspirado nas estratégias de sobrevivência e caça de lobos cinzentos na natureza. Ele segue uma estrutura hierárquica de lobos, onde os lobos alfa são os mais dominantes, seguidos pelos beta, delta e omega. Na busca pela solução ideal, os lobos alfa lideram e guiam a busca, enquanto os lobos beta os auxiliam na comunicação com outros lobos e os lobos omega são os de nível mais baixo, escolhidos pelos alfa. O GWO é aplicado para ajustar os parâmetros da CNN de modo a otimizar seu desempenho na tarefa de classificação de galáxias, mostrando eficácia na melhoria da precisão do modelo (Şenel, 2023).

A pesquisa compara o GWO com os algoritmos PSO e ABC, demonstrando que o GWO se destaca com uma precisão de treinamento de 93% e de teste de 85%. A abordagem híbrida utilizando GWO para otimização de hiperparâmetros em uma CNN se mostra eficaz na

classificação de galáxias. O estudo sugere que futuras pesquisas podem explorar abordagens híbridas utilizando conjuntos de dados homogêneos.

2.2.5 *Galaxy Morphological Classification with Efficient Vision Transformer*

Lin *et al.* (2022) propõem a utilização de *Vision Transformer* (ViT) - uma arquitetura de classificação desenvolvida recentemente pelo Google (Dosovitskiy *et al.*, 2020) - para a classificação morfológica de galáxias, a fim de apresentar seus resultados comparados com as CNNs.

O conjunto de dados utilizado em Lin *et al.* (2022) é baseado no *Galaxy Zoo 2* (Willett *et al.*, 2013). Por meio desse conjunto de dados, foram rotuladas oito classes distintas: *round elliptical*, *in-between elliptical*, *cigar-shaped elliptical*, *edge-on*, *barred spiral*, *unbarred spiral*, *irregular*, e *merger galaxies*, totalizando 155.951 imagens.

O modelo ViT utilizado foi o *Linformer*, que possui uma característica marcante de complexidade linear ($O(n)$, onde n é o tamanho da sequência do patch de entrada) (Lin *et al.*, 2022).

O melhor resultado geral obtido com o modelo proposto foi de 80,55% de precisão. Foram realizados treinamentos tanto com ajuste dos pesos quanto sem o ajuste aplicado nas funções de perda, e os melhores resultados foram alcançados com o primeiro método. Também foi utilizado um modelo de CNN, a *ResNet-50*, para fazer uma comparação com os modelos *Linformer* desenvolvidos no trabalho. A melhor precisão obtida pela *ResNet-50* foi de 85,12%, e embora os resultados tenham sido melhores, o *Vision Transformer* obtém resultados superiores em imagens de galáxias com maior nível de ruído, devido ao tamanho menor e brilho mais fraco das galáxias.

2.2.6 *The Classification of Optical Galaxy Morphology Using Unsupervised Learning Techniques*

Fielding, Nyirenda e Vaccari (2022) implementam técnicas de aprendizado não supervisionado - sem dados rotulados - para classificar morfológicamente galáxias por meio do conjunto de dados *Galaxy Zoo DECaLS*, que inclui 253.286 imagens.

O projeto utiliza um *convolutional autoencoder* como extrator de recursos (TensorFlow 2.5 e Python 3.8 foram utilizados para implementar este *autoencoder*). Também compara os resultados de três métodos de clusterização: *k-means*, *fuzzy c-means* e *agglomerative clustering* (agrupamento hierárquico aglomerativo).

Durante o treinamento, o *autoencoder* convolucional foi treinado em 100 épocas, e após o término do treinamento, os recursos foram salvos em um arquivo CSV para uso posterior nos modelos de clusterização. Os *clusters* gerados por esses métodos foram mapeados para as

classes reais já existentes e avaliados por meio de métricas. As classes avaliadas foram: *smooth or featured*, *disk edge on*, *has spiral arms*, *bar*, *bulge size*, *how rounded*, *edge on bulge*, *spiral winding*, *spiral arm count* e *merging*.

O trabalho não apresenta resultados gerais, apenas individuais para cada classe de galáxia e para cada método, como mostrado na Figura 27. Fazendo uma média geral dos resultados de precisão mostrados na Figura 27 tem-se: *K-means*: 0.527 (52,7%), *Fuzzy C-means*: 0.459 (45,9%) e *Agglomerative*: 0.564 (56,4%).

Os resultados apresentados no trabalho [Fielding, Nyirenda e Vaccari \(2022\)](#) mostram que, de modo geral, o *agglomerative clustering* obtém um melhor desempenho, porém requer um tempo de clusterização significativamente maior em comparação aos outros métodos.

Figura 27 – Tabelas de comparação das métricas de precisão, *recall* e F1-score, respectivamente, para as classes geradas por cada um dos métodos.

RECALL BY QUESTION

Question Name	Clustering Method		
	<i>K-means</i>	<i>Fuzzy C-means</i>	<i>Agglomerative</i>
smooth or featured	<u>0.395</u>	0.409	0.497
disk edge on	0.633	<u>0.504</u>	0.744
has spiral arms	0.702	<u>0.574</u>	0.772
bar	0.355	0.418	<u>0.281</u>
bulge size	0.235	0.384	<u>0.175</u>
how rounded	0.387	0.353	<u>0.297</u>
edge on bulge	0.427	0.407	<u>0.215</u>
spiral winding	<u>0.384</u>	0.410	0.470
spiral arm count	<u>0.197</u>	0.267	<u>0.189</u>
merging	0.262	0.295	<u>0.232</u>

PRECISION BY QUESTION

Question Name	Clustering Method		
	<i>K-means</i>	<i>Fuzzy C-means</i>	<i>Agglomerative</i>
smooth or featured	0.592	<u>0.509</u>	0.528
disk edge on	<u>0.654</u>	0.711	0.671
has spiral arms	0.728	<u>0.714</u>	0.736
bar	0.445	<u>0.385</u>	0.460
bulge size	0.377	<u>0.303</u>	<u>0.266</u>
how rounded	0.435	<u>0.422</u>	0.457
edge on bulge	0.608	<u>0.457</u>	0.687
spiral winding	0.427	<u>0.376</u>	0.433
spiral arm count	0.438	<u>0.250</u>	0.438
merging	0.752	<u>0.451</u>	0.756

F1-SCORE BY QUESTION

Question Name	Clustering Method		
	<i>K-means</i>	<i>Fuzzy C-means</i>	<i>Agglomerative</i>
smooth or featured	0.430	<u>0.414</u>	0.509
disk edge on	0.643	<u>0.545</u>	0.696
has spiral arms	0.714	<u>0.618</u>	0.749
bar	0.361	0.392	<u>0.250</u>
bulge size	0.251	0.333	<u>0.202</u>
how rounded	0.387	0.364	<u>0.298</u>
edge on bulge	0.443	0.400	<u>0.236</u>
spiral winding	<u>0.382</u>	0.384	0.442
spiral arm count	0.241	0.240	<u>0.236</u>
merging	0.329	0.332	<u>0.318</u>

Fonte: Fielding, Nyirenda e Vaccari (2022).

2.2.7 Considerações Sobre os Trabalhos Relacionados

Visando auxiliar na comparação entre este trabalho descritos anteriormente, a Tabela 2.4 foi construída, listando suas principais características.

Tabela 2.4 – Sumarização dos Trabalhos Correlatos, Dados Utilizados e Resultados

Trabalho	Base de dados	Amostras	Abordagem	Classes	Melhor Resultado
Misra <i>et al.</i> (2020)	Galaxy - Star separation (Kaggle)	4.614	CNN treinada do zero	3	97,39% de acurácia
Gharat e Dandawate (2022)	Sloan Digital Sky Survey (SDSS)	21.785	CNN treinada do zero	10	84,78% de acurácia
Cavanagh, Bekki e Groves (2021)	Sloan Digital Sky Survey (SDSS) Data Release 4	14.034	Testes em quatro arquiteturas de CNNs diferentes. Sendo a melhor uma arquitetura treinada do zero	3 e 4	83% (3 classes) de acurácia
Şenel (2023)	Sloan Digital Sky Survey (SDSS)	21.785	Utilização CNN com três algoritmos metaheurísticos GWO, PSO e ABC	10	85% de acurácia
Lin <i>et al.</i> (2022)	Galaxy Zoo 2	155.951	Utilização do modelo Vision Transformer (ViT) Linformer	9	80,55% de acurácia
Fielding, Nyirenda e Vaccari (2022)	Galaxy Zoo DECaLS	253.286	Empregou um <i>convolutional autoencoder</i> como extrator de recursos e modelos de clusterização com foco no treinamento não supervisionado: K-means, Fuzzy C-means e Agglomerative	10	56,4% de precisão

Fonte: Criado pela Autora.

Os trabalhos com os melhores resultados foram Misra *et al.* (2020), Gharat e Dandawate (2022), Gharat e Dandawate (2022) e Şenel (2023), fizeram a utilização de CNNs. Apenas dois deles utilizaram técnicas para tratar problemas de desbalanceamento das bases de dados, sendo eles Misra *et al.* (2020) e Cavanagh, Bekki e Groves (2021). As técnicas utilizadas para aumentar o conjunto de dados foram: recorte, rotação e espelhamento das imagens, tal ideia é chamada de *Data Augmentation*.

Os trabalhos Cavanagh, Bekki e Groves (2021) e Lin *et al.* (2022) realizaram testes com redes neurais convolucionais já pre-treinadas, AlexNet e ResNet-50, e obtiveram bons resultados finais após o treinamento, até melhores do que resultados obtidos no método destacado no próprio trabalho. Porém, os dois trabalhos pontuam que, apesar do bom resultado obtido com essas redes, o custo computacional era maior.

No estudo conduzido por Şenel (2023), aprimoramentos na CNN foram alcançados por meio da aplicação de algoritmos meta-heurísticos. Ao utilizar essa abordagem, o autor conseguiu otimizar a CNN, resultando em melhorias nos resultados desejados.

Nenhum dos trabalhos fez um pré-processamento direto nas imagens, a fim de reduzir ruídos e/ou melhorar sua qualidade.

Um observação a se fazer é em relação ao número de classes de galáxias classificadas, analisando os resultados é possível concluir que quanto maior o número de classes mais complexo é o processo de classificação, além de gerar uma maior confusões entre classes morfológicamente parecidas.

Portanto, observa-se que este trabalho compartilha semelhanças com os estudos mencionados anteriormente, ao utilizar *deep learning*, especialmente redes neurais convolucionais, para criar um modelo de classificação morfológica de galáxias. No entanto, destaca-se pela introdução de abordagens adicionais de *deep learning*, tais como *transfer learning*, que permite aproveitar o conhecimento de modelos pré-treinados em conjuntos de dados relacionados, a classificação hierárquica, que possibilita uma organização estruturada das classes de galáxias e o *ensemble learning*, que combina vários modelos para aprimorar a precisão e a generalização do sistema.

3 Desenvolvimento

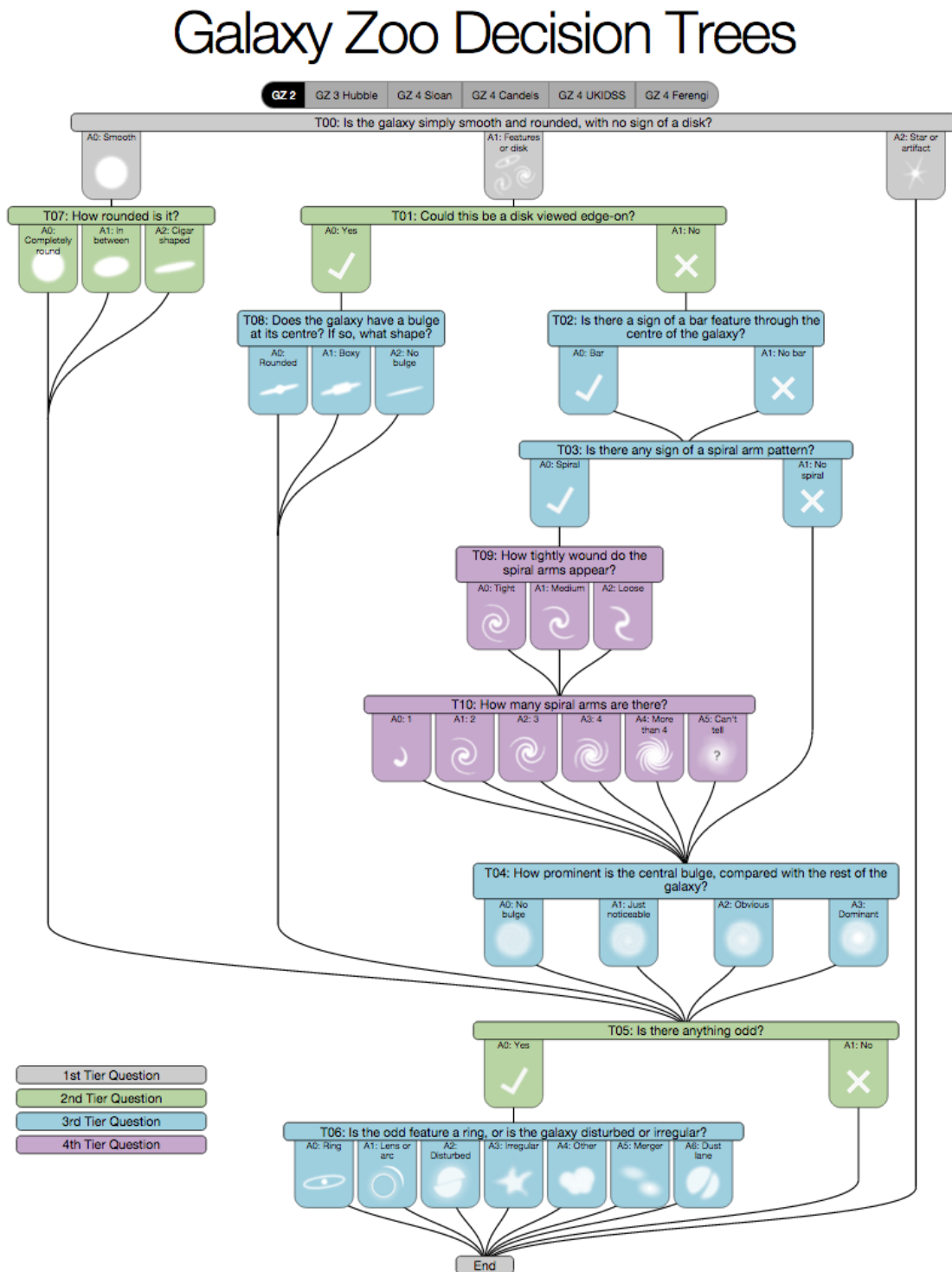
Como apresentado nos capítulos anteriores, existem diferentes estudos realizados para realizar a classificação morfológica de galáxias por meio de imagens, utilizando diversas técnicas para alcançar esse objetivo. Neste capítulo, pretende-se apresentar detalhadamente os processos dos experimentos realizados durante o desenvolvimento do trabalho. Esses experimentos têm como base, inicialmente, o projeto apresentado no trabalho de [Gharat e Dandawate \(2022\)](#), buscando obter resultados semelhantes ou até superiores, utilizando tanto a mesma estrutura de CNN apresentada no trabalho, quanto outras técnicas diferentes. Visto que, nesse trabalho, é pontuado a importância da classificação das galáxias em classes mais específicas, pois as diferentes propriedades das galáxias em cada classe tornam a classificação uma abordagem valiosa. Ao considerar minuciosamente essas características durante a classificação, torna-se mais acessível explorar detalhadamente tipos específicos de galáxias.

3.1 Base de dados

Na base de dados Galaxy10 SDSS Dataset, os rótulos de classificação são derivados do Galaxy Zoo, cujos detalhes são descritos em [Willett *et al.* \(2013\)](#). As imagens utilizadas nesse conjunto de dados são adquiridas do Sloan Digital Sky Survey (SDSS), conforme documentado em [York *et al.* \(2000\)](#). Antes de serem incorporadas ao conjunto de dados Galaxy10, essas imagens passam por um processo de pré-processamento, onde são selecionadas as imagens com melhor classificação da base original do Galaxy Zoo. Além disso, as imagens originais, com dimensões de 424 x 424 pixels, são recortadas para 207 x 207 pixels de forma centralizada e, em seguida, reduzidas para 69 x 69 pixels usando interpolação bilinear ([Gharat; Dandawate, 2022](#)).

Os rótulos de classificação atribuídos através do projeto *Galaxy Zoo* seguem a *Galaxy Zoo Decision Tree (GZ2)*, uma árvore de decisão elaborada para auxiliar os participantes na classificação sistemática de galáxias. A árvore de decisão GZ2 está disponível em https://data.galaxyzoo.org/gz_trees/gz_trees.html e pode ser visualizada na Figura 28.

Figura 28 – Galaxy Zoo Decision Tree.



Fonte: Disponível em <https://data.galaxyzoo.org/gz_trees/gz_trees.html>.

Acesso em 20 de dezembro de 2023.

Por fim, a Tabela 3.1 apresenta a relação de classe e quantidade de amostras presente na base de dados final, Galaxy10 SDSS Dataset, disponível.

Tabela 3.1 – Galaxy10 dataset

Número da Classe	Classe	Número de Imagens
0	Disk, Face-on, No Spiral	3461
1	Smooth, Completely round	6997
2	Smooth, in-between round	6292
3	Smooth, Cigar shaped	394
4	Disk, Edge-on, Rounded Bulge	1534
5	Disk, Edge-on, Boxy Bulge	17
6	Disk, Edge-on, No Bulge	589
7	Disk, Face-on, Tight Spiral	1121
8	Disk, Face-on, Medium Spiral	906
9	Disk, Face-on, Loose Spiral	519

Fonte: Disponível em: <<https://astronn.readthedocs.io/en/latest/galaxy10sdss.html>>.

Acesso em 24 de junho de 2023.

3.2 Pré-processamento dos dados

Na etapa de pré-processamento dos dados, é feito o aumento de dados (do inglês *data augmentation*) por meio de técnicas que realizam modificações nas imagens originais para gerar novas imagens. Essas modificações foram: espelhamento horizontal aleatório (do inglês *Random Horizontal Flip*), espelhamento vertical aleatório (do inglês *Random Vertical Flip*), corte aleatório de 60 pixels e rotação aleatória de até 90 graus (do inglês *Random Rotation*). Essa técnica foi utilizada a fim de amenizar o desbalanceamento do número de imagens nas classes que existem na base de dados, como mostra a Tabela 3.1.

Durante os experimentos os dados foram alterados para das configurações a seguir:

1. **Dados padrões da base de dados:** Nenhuma alteração foi feita na quantidade de amostras; a base de dados manteve suas proporções originais.
2. **Redução de amostras em todas as classes:** Todas as classes tiveram suas amostras reduzidas para a mesma quantidade da classe 5, ou seja, 17 amostras, técnica chamada de *undersampling*.
3. **Aumento de amostras na classe 5:** As amostras da classe 5 foram aumentadas para 204, enquanto as demais classes permaneceram inalteradas.
4. **Aumento na classe 5 e redução nas demais classes:** As amostras da classe 5 foram aumentadas para 204, e as demais classes foram reduzidas para a mesma quantidade de amostras, ou seja, 204.
5. **Aumento de amostras em todas as classes:** Todas as classes tiveram suas amostras aumentadas para a mesma quantidade da classe 1, ou seja, 6997 amostras, técnica chamada de *oversampling*.

É importante observar que nem todos os experimentos envolveram todas as alterações mencionadas.

Em todos os casos em que o processo de aumento dos dados foi realizado, foram utilizadas as seguintes técnicas: *Cropping* (recorte), *Rotation* (rotação) e *Flipping/Mirroring* (espelhamento). Isso significa que, para cada imagem, uma ou mais dessas técnicas foram aplicadas com parâmetros aleatórios. Tais técnicas foram adotadas com base no trabalho de [Cavanagh, Bekki e Groves \(2021\)](#), visando ampliar a diversidade e a robustez do conjunto de dados de treinamento.

3.3 Caracterização da base de dados

O conjunto de dados para aprendizado e avaliação de um modelo de CNN é geralmente composto por pares de imagens e suas respectivas classes. Além disso, é essencial dividir esses dados de forma adequada para um treinamento e validação eficientes. Vamos abordar a divisão dos dados em três partes: treino, validação e teste.

- **Treino:** O conjunto de dados de treinamento é usado para treinar a CNN, ajustando os pesos e parâmetros do modelo com base nas imagens e classes fornecidas. Geralmente, uma grande parte do conjunto de dados é alocada para o treinamento, pois é nessa fase que ocorre o aprendizado do modelo.
- **Validação:** O conjunto de validação permite avaliar o desempenho da CNN durante o treinamento e ajustar os hiperparâmetros do modelo para otimizá-lo. O conjunto de dados de validação é usado para avaliar a precisão e o desempenho geral do modelo em exemplos não vistos durante o treinamento.
- **Teste:** Uma vez que a CNN tenha sido treinada e ajustada usando os dados de treinamento e validação, é importante avaliar seu desempenho final em um conjunto de dados completamente independente, chamado de conjunto de teste.

A divisão dos dados em treinamento, validação e teste é essencial para evitar o *overfitting*, que ocorre quando um modelo se ajusta demais aos dados de treinamento e não consegue generalizar corretamente para os demais dados.

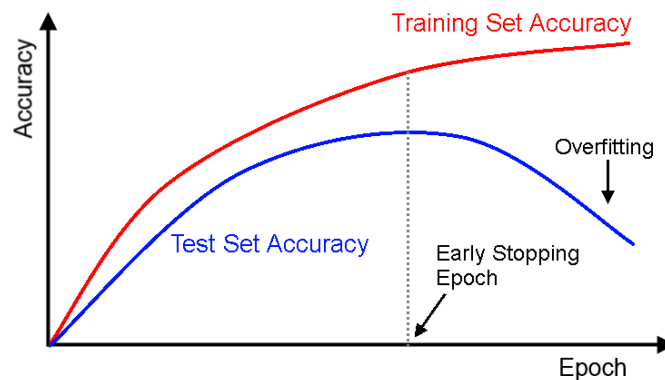
A divisão dos dados para treino, validação e teste segue uma proporção de 70%, 15% e 15% do conjunto total de imagens, respectivamente. Essa estratégia de divisão foi adotada em conformidade com as práticas estabelecidas por [Gharat e Dandawate \(2022\)](#), o qual serviu como referência para este estudo.

3.4 Treinamento do modelo

Na fase de treinamento, a rede neural é configurada com vários parâmetros, incluindo o número de épocas, que representa uma iteração completa em que todos os dados do conjunto de treinamento são passados pela rede. O tamanho do lote (*batch size*) é determinado para definir o número de amostras utilizadas para atualizar os pesos da rede a cada iteração. Além disso, são fornecidos os conjuntos de dados de treinamento, validação e teste. Neste trabalho, o número máximo de épocas foi definido como 50, seguindo a pesquisa de Gharat e Dandawate (2022). Quanto ao *batch size*, optou-se por um valor de 32 após experimentação empírica. Inicialmente, o processo começou com um *batch size* de 8 e foi multiplicado por dois até atingir 128. No entanto, observou-se que após 64, os resultados não apresentaram melhoria significativa, levando à escolha do valor 32 como o mais adequado para este cenário.

O treinamento também foi configurado com parada antecipada, também conhecido, do inglês, como *Early Stopping*, assim como no trabalho de Gharat e Dandawate (2022). Ao usar essa técnica significa que ao final de cada época, é calculado o valor de perda (*loss*) e comparado com o valor das épocas anteriores, e quando esse valor parar de diminuir, indicando que a rede está "aprendendo menos", o treinamento é finalizado. Essa técnica evita *overfitting* e pode reduzir o número de épocas necessárias para treinar. A Figura 29 ajuda a visualizar o conceito de *early stopping*.

Figura 29 – Early Stopping



Fonte: <<https://www.deeplearningbook.com.br/usando-early-stopping-para-definir-o-numero-de-epocas-de-treinamento/>>.
Acesso em 06 de julho de 2023.

3.5 Avaliação

Durante o treinamento são registradas informações sobre as métricas de desempenho ao longo do tempo para avaliação. Os modelos de CNN foram avaliados utilizando métricas como acurácia, *recall*, precisão e *F1-Score*. Essas métricas fornecem uma visão abrangente do

desempenho dos modelos e foram utilizadas para avaliar a eficácia dos modelos de CNN na tarefa de classificação das galáxias.

3.6 Etapa Experimental

Esta seção apresenta os quatro experimentos conduzidos neste trabalho. O Experimento I e II abordam o treinamento de arquiteturas de CNNs. O Experimento I, detalhado na Seção 3.6.1, consiste na construção e treinamento da estrutura a partir do zero, enquanto o Experimento II, descrito na Seção 3.6.2, emprega a técnica de *Transfer Learning* (TL).

No Experimento III, discutido na Seção 3.6.3, adota-se a abordagem de classificação hierárquica, estabelecendo uma hierarquia entre as classes e utilizando diferentes CNNs para treinar diferentes níveis dessa hierarquia.

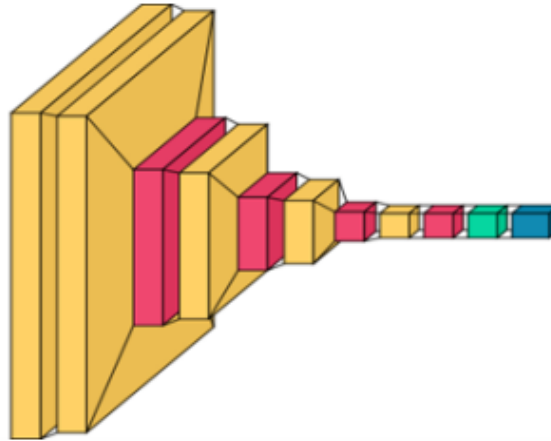
Por fim, o Experimento IV, apresentado na Seção 3.6.4, emprega a técnica de *ensemble learning*, combinando algumas das CNNs treinadas no Experimento I para aprimorar os resultados.

3.6.1 Experimento I

Este experimento é realizado com uma arquitetura de CNN que será treinada do zero. Toda a arquitetura de CNN desenvolvida é a mesma apresentada em [Gharat e Dandawate \(2022\)](#). Tal arquitetura é ilustrada na Figura 30, onde os blocos amarelos representam as camadas convolucionais, os vermelhos representam a camada de *pooling*, verde representa o *dropout* e, por fim, o azul representa o *flattening*. A primeira camada convolucional possui 32 filtros e as demais 64, além disso, todas elas possuem um kernel de dimensões 3x3 e como função de ativação é utilizada a ReLU.

Figura 30 – Arquitetura da rede neural convolucional de Gharat e Dandawate (2022).

Os blocos amarelos representam as camadas densas, enquanto os blocos vermelhos correspondem às camadas de *pooling*, já o bloco verde de azul são, respectivamente, camada de *droupout* e *flattening*.



Fonte: Gharat e Dandawate (2022).

Ao final do bloco convolucional, após o *flattening*, há duas camadas densas antes da camada de saída. A primeira camada densa possui 64 neurônios, enquanto a segunda camada possui 32 neurônios. Como o número de classes de saída é 10, o número de neurônios na camada de saída é 10. A função de ativação utilizada nas duas primeiras camadas densas é a ReLU, enquanto na camada de saída usamos o *Softmax* como função de ativação. Para evitar o *overfitting* - sobre-ajuste dos pesos do modelo aos dados de treinamento -, é utilizada a técnica de *dropout* em ambas as camadas densas. Um *dropout* de 0.2 é usado, onde 20% dos neurônios atuam como neurônios inativos (Gharat; Dandawate, 2022).

Durante esse experimento os dados não sofreram alteração no número de amostras.

3.6.2 Experimento II

Existem várias CNN pré-treinadas que são frutos do treinamento de dados da ImageNet (Deng *et al.*, 2009), que consiste em milhões de imagens de alta resolução pertencentes a milhares de categorias diferentes, e são com essas redes que é possível realizar a técnica de TL. Nesse trabalho, foram feitos experimentos em seis dessas redes, sendo elas: EfficientNetV2M (Tan; Le, 2021), DenseNet121 (Huang *et al.*, 2017), ResNet50 (He *et al.*, 2016), AlexNet (Krizhevsky; Sutskever; Hinton, 2012), VGG19 (Simonyan; Zisserman, 2015) e VGG16 (Simonyan; Zisserman, 2015). Cada uma delas possui uma estrutura diferente.

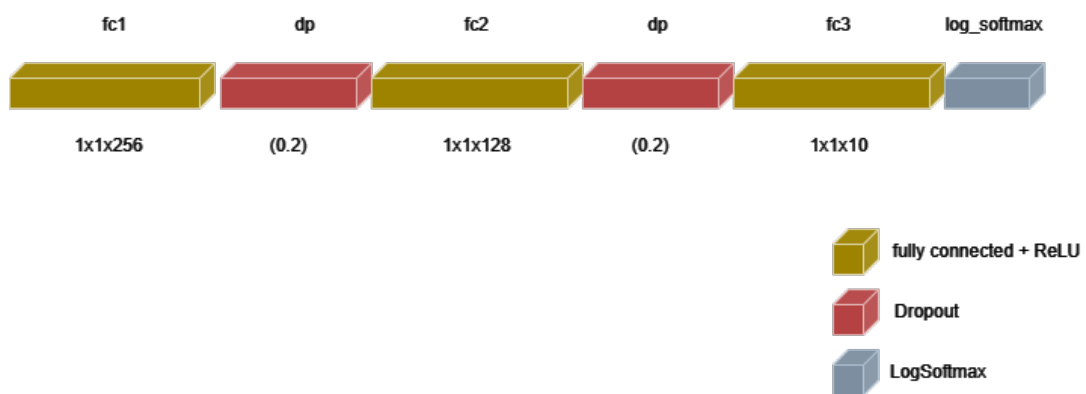
Esse experimento tem como objetivo estabelecer o desempenho inicial das arquiteturas e servirá tanto de base de comparação para os demais experimentos, quanto para utilização em outros experimentos. O processo de *fine-tuning* foi conduzido pela substituição das camadas de classificação em cada uma das arquiteturas, conforme ilustrado na Figura 31, Além disso,

todas os blocos foram “descongelados” para atualização dos pesos. Nessa nova configuração, os blocos amarelos representam camadas densas, com números específicos de neurônios: 256 nas duas primeiras e 128 na última. Por fim, a camada de saída é composta por 10 neurônios, correspondentes às classes de destino. Já os blocos vermelhos denotam camadas de *dropout*, ambas com taxa de 0.2.

Os valores dos neurônios foram inicialmente baseados na camada de saída mencionada no trabalho de [Gharat e Dandawate \(2022\)](#), citado no experimento anterior. O experimento iniciou empiricamente com 64 neurônios nas primeiras camadas e 32 na última, e esses valores foram progressivamente dobrados.

Figura 31 – Arquitetura do Classificador

Os blocos amarelos representam as camadas densas, enquanto os blocos vermelhos correspondem às camadas de *dropout*.



Fonte: Elaborado pela autora.

Ao final da estrutura, após a camada de saída, existe uma camada de *LogSoftmax*, que se trata de uma função usada para converter um vetor de números reais em probabilidades normalizadas. Nesse experimento foram realizadas diferentes análises com a diferentes alteração dos dados como apresentado na Seção 3.2.

3.6.3 Experimento III

Nesse experimento há uma abordagem de classificação hierárquica utilizando as mesmas arquiteturas de *transfer learning* do Experimento I. Isso implica usar modelos pré-treinados para a classificação em múltiplos níveis, onde as classes são organizadas hierarquicamente de acordo com a morfologia das galáxias.

Essa técnica envolve a classificação em um nível superior (categoria ampla) seguida por classificações mais específicas em níveis inferiores (subcategorias). Essa abordagem foi escolhida pois o problema tem classes que podem ser naturalmente organizadas em uma hierarquia.

A organização do classificador hierárquico segue a divisão estabelecida na base de dados deste estudo, utilizando como referência a Galaxy Zoo Decision Tree GZ2, apresentada ante-

riormente na Seção 3.1 e ilustrada na Figura 28. A estrutura hierárquica da árvore de decisão empregada neste trabalho é finalmente representada na Figura 32. Essa abordagem hierárquica possibilita uma classificação sistemática das galáxias, em conformidade com as diretrizes estabelecidas pelo *Galaxy Zoo*.

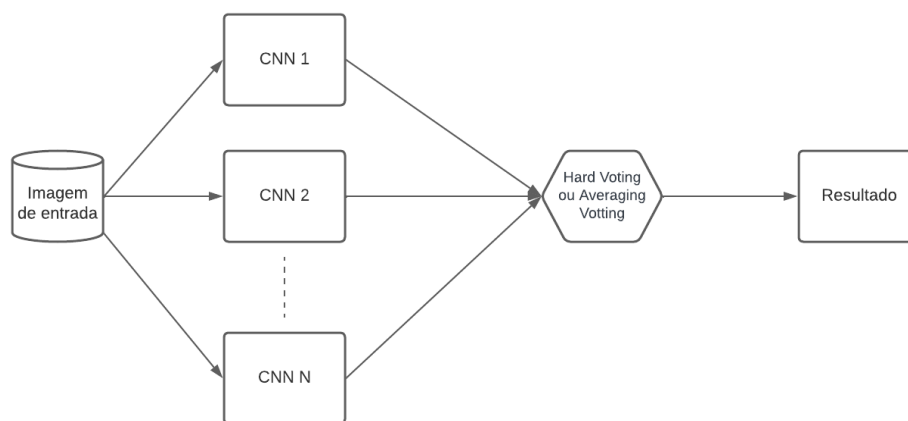
Cada retângulo tracejado na Figura 32 representa a aplicação de uma CNN específica na tarefa de classificação. No total, seis CNNs foram empregadas, as mesmas do Experimento II nesse processo final, cada uma responsável por uma etapa da classificação hierárquica.

Para cada uma das redes, os dados foram adaptados com rótulos conforme a estrutura hierárquica estabelecida na Figura 32. Isso permitiu a classificação em níveis superiores, representando categorias amplas, e em níveis inferiores, para subcategorias mais específicas. Essa estratégia de classificação hierárquica baseou-se na morfologia das galáxias, refletida na árvore de decisão do *Galaxy Zoo*.

3.6.4 Experimento IV

Com base nos resultados obtidos nos experimentos anteriores utilizando *transfer learning*, este experimento tem como objetivo a técnica de *ensemble learning* visando melhorar os resultados obtidos com as arquiteturas isoladas avaliadas no Experimento II. Nessa etapa de *ensemble learning* as regras de combinações utilizadas foram as de votações *hard* e média (do inglês, *averaging*). A Figura 33 ilustra como o fluxo de *ensemble learning* irá funcionar.

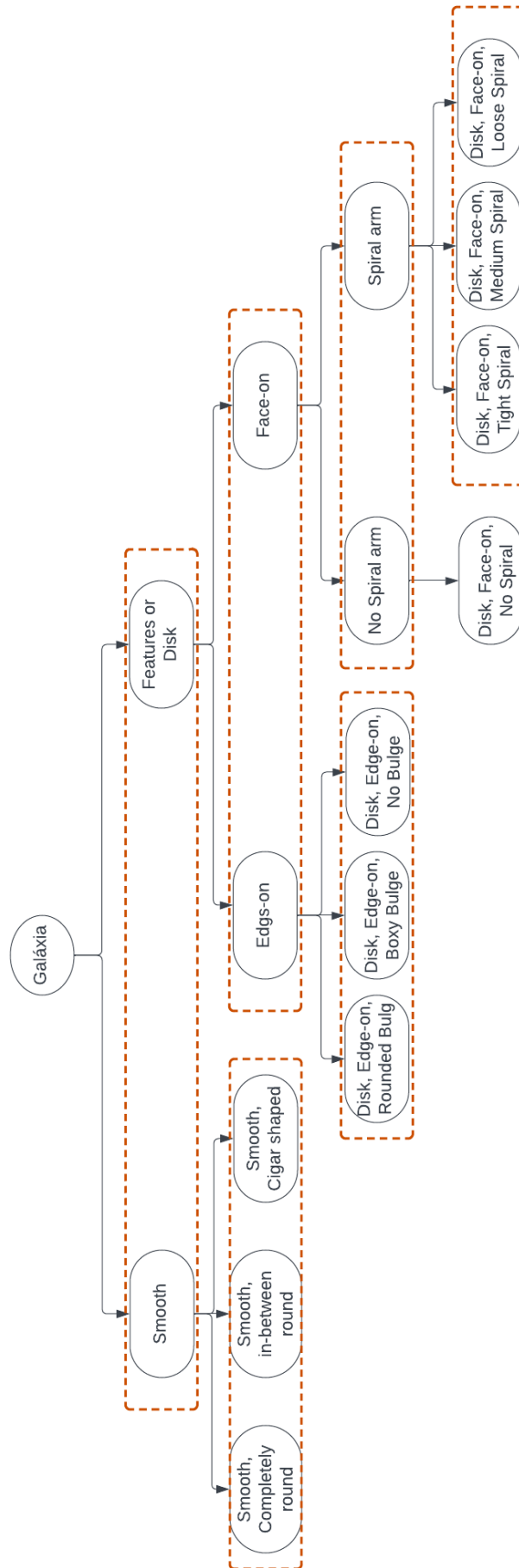
Figura 33 – Fluxo do *ensemble learning*.



Fonte: Elaborado pela autora.

Logo, esse experimento visa utilizar as redes, com bons resultados, treinadas no Experimento II para realização do método de *ensemble learning*.

Figura 32 – Hierarquia das classes.



Fonte: Elaborado pela autora.

4 Resultados

Neste capítulo, são apresentadas a interpretação e análise dos resultados obtidos ao longo deste estudo. A Seção 4.1 detalha as ferramentas e técnicas empregadas nos experimentos conduzidos. Em seguida, a Seção 4.2 apresenta os resultados obtidos com a aplicação da estrutura de Redes Neurais Convolucionais (CNN) proposta por [Gharat e Dandawate \(2022\)](#). Posteriormente, na Seção 4.3, são descritos os resultados alcançados ao utilizar a técnica de *Transfer Learning* (TL). A Seção 4.4 expõe os resultados derivados da técnica de classificação hierárquica. Adicionalmente, na Seção 4.5, são revelados os resultados provenientes da aplicação da técnica de *ensemble learning*, utilizando redes previamente treinadas em experimentos anteriores. Na conclusão da exposição dos resultados, na Seção 4.6, realiza-se uma discussão abrangente sobre os achados. A Seção 4.7 descreve o procedimento adotado para disponibilizar o modelo para acesso público. Por fim, na Seção 4.8, o melhor resultado obtido é confrontado com estudos relacionados que utilizaram a mesma base de dados, permitindo uma comparação direta. É importante ressaltar que ambos os experimentos foram conduzidos utilizando a mesma base de dados, a [Galaxy10 SDSS Dataset](#).

Durante os experimentos, a principal métrica de avaliação considerada é a acurácia. Além de facilitar a comparação com outros trabalhos relacionados, este estudo tem como objetivo analisar a proporção de exemplos que foram corretamente classificados pelo modelo em relação ao total de exemplos. A acurácia é uma medida amplamente utilizada que reflete a taxa de acertos do modelo, o que proporciona uma visão geral do seu desempenho na tarefa de classificação.

Ao longo desta seção, são discutidas as descobertas e análises obtidas a partir dos resultados, a fim de proporcionar uma compreensão completa do desempenho das abordagens utilizadas.

4.1 Ferramentas e técnicas utilizadas

Todo o desenvolvimento foi realizado utilizando os recursos oferecidos pelo Google Colab, sendo eles:

- Disco: 80 GB de memória
- GPU: NVIDIA Tesla T4 (16 GB de memória de vídeo)
- RAM: 12.7 GB de memória RAM

Para a realização da classificação morfológica das galáxias através de imagens, foi implementada a técnica de CNN treinada do zero exclusivamente para o problema, assim como

no trabalho Gharat e Dandawate (2022). Além disso, buscando otimizar o modelo de aprendizagem, foram utilizadas algumas técnicas sendo elas: *transfer learning*, classificação hierárquica e *ensemble learning*.

Durante todos os experimentos, foi utilizada a linguagem Python em conjunto com a biblioteca PyTorch 2.0.1+cu118 (2021). Facebook AI Research. Disponível em: <<https://pytorch.org/>>. Acesso em: 06 de maio de 2023.

4.2 Experimento I

Nesta seção será apresentado os resultados obtidos com a CNN proposta por Gharat e Dandawate (2022). A Tabela 4.1 apresenta os resultados do modelo de CNN avaliado em uma base de dados com classes de amostras desbalanceadas e com alteração no valor de *patience* do *early stopping*. A *patience* é um parâmetro usado no treinamento de modelos de aprendizado de máquina que controla o número de épocas em que a métrica de desempenho não melhora antes de interromper o treinamento. Neste experimento, foram testados dois valores diferentes de *patience*: 5 e 3, além disso, foi feito um experimento onde o *early stopping* não foi utilizado, ou seja, o modelo foi treinado com as 50 épocas, indicado por “-” na Tabela 4.1.

Tabela 4.1 – Esta tabela apresenta os resultados dos modelos avaliados em uma base de dados com classes de amostras desbalanceadas e com alteração no valor de *Patience* do *Early Stopping*.

<i>Patience</i>	Acurácia	Precisão	F1	<i>Recall</i>	<i>Loss</i>	Épocas
5	0,8061	0,7891	0,8061	0,7915	0,5614	31
3	0,7565	0,7431	0,7565	0,7386	0,6499	14
-	0,8256	0,8171	0,8256	0,8182	0,5098	50

Fonte: Criado pela Autora.

Os resultados da Tabela 4.1 sugerem que o uso do *early stopping* com uma *patience* maior (5) resultou em um desempenho melhor em relação às métricas do que o *patience* menor (3). No entanto, o modelo treinado sem *early stopping* obteve o melhor resultado geral, podendo indicar que, para esse conjunto de dados específico, um número maior de épocas pode ser necessário para alcançar o melhor desempenho.

A Tabela 4.2 exhibe os resultados obtidos com a base de dados balanceada, para afeitos de comparação, utilizando a técnica de *oversampling*, utilizando *early stopping* com *patience* de 5, e não utilizando *early stopping* deixando o modelo realizar a aprendizagem com as 50 épocas.

Tabela 4.2 – Esta tabela apresenta os resultados dos modelos avaliados em uma base de dados balanceada, utilizando a técnica de *oversampling* número 5 citado na Seção 3.2, e com alteração no valor de *Patience* do *Early Stopping*.

<i>Patience</i>	Acurácia	Precisão	F1	<i>Recall</i>	<i>Loss</i>	Épocas
5	0,7164	0,7072	0,7164	0,7083	0,7633	46
-	0,7576	0,7568	0,7576	0,7520	0,6686	50

Fonte: Criado pela Autora.

Os resultados podem indicar que a arquitetura da CNN não é robusta o suficiente para lidar com a quantidade de dados disponíveis, ou que a técnica de *oversampling* pode ter sido inadequada. Isso pode ser observado quando comparados com os resultados do experimento apresentado anteriormente, mesmo considerando que a quantidade de épocas para o treinamento foi aumentada. Esta análise sugere que outros aspectos da arquitetura ou do processo de treinamento devem ser considerados para melhorar o desempenho do modelo.

4.3 Experimento II

Na Seção 3.2, foram discutidas as alterações de pré-processamento em relação às amostras disponíveis na base de dados. As seguintes modificações foram abordadas:

1. Dados padrões da base de dados, sem alteração;
2. Redução de amostras em todas as classes, exceto a minoritária (Classe 5) ;
3. Aumento de amostras na classe 5;
4. Aumento na classe 5 e redução nas demais classes;
5. Aumento de amostras de todas as classes, exceto a majoritária (Classe 1) .

Ao longo desta seção são apresentados os resultados obtidos e os efeitos dessas mudanças no desempenho dos algoritmos de aprendizado de máquina aplicados aos dados processados. Analisamos os resultados obtidos para cada cenário e identificamos possíveis ganhos e desafios relacionados a cada abordagem de pré-processamento.

Na Tabela 4.3, os modelos foram avaliados em uma base de dados com classes de amostras desbalanceadas, sem alteração na base de dados. Pode-se observar que os modelos *EfficientNetV2-M*, *DenseNet121* e *ResNet50* apresentaram os melhores resultados, mostrando que esses modelos conseguiram lidar relativamente bem com o desbalanceamento dos dados. Por outro lado, o modelo *VGG16* obteve as piores métricas, o que pode indicar dificuldades em lidar com as classes minoritárias. Logo em seguida, na Tabela 4.4, é mostrado a acurácia de cada modelo em relação a

classificação de classe de forma individual. É possível observar que a classe cinco obteve acurácia de 0% em todas as arquiteturas.

Tabela 4.3 – Resultados dos modelos avaliados em uma base de dados com classes de amostras desbalanceadas e com *early stopping*.

Modelo	Acurácia	Precisão	F1	Recall	Loss	Épocas
EfficientNetV2-M	0,8290	0,8230	0,8203	0,8290	0,5508	06
DenseNet121	0,8198	0,8179	0,8148	0,8198	0,5597	09
ResNet50	0,8149	0,8350	0,8194	0,8149	0,5951	09
AlexNet	0,7336	0,7418	0,7238	0,7336	0,7262	09
VGG19	0,7929	0,7892	0,7810	0,7929	0,6163	16
VGG16	0,7210	0,6788	0,6798	0,7210	0,7699	06

Fonte: Criado pela Autora.

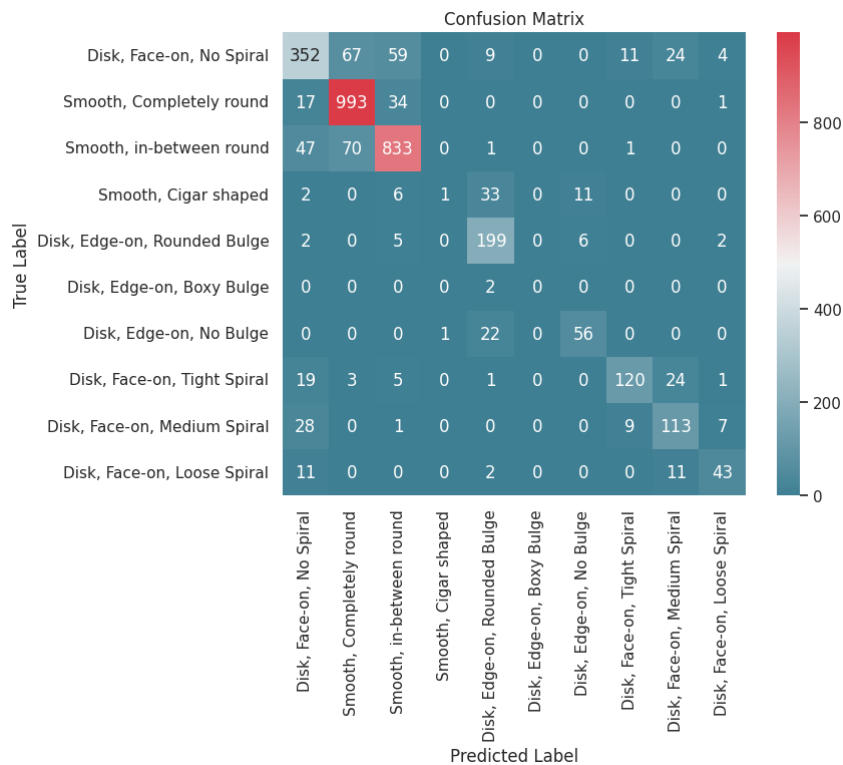
Tabela 4.4 – Acurácia dos Modelos por Classe.

Classe	VGG19	VGG16	EfficientNet-V2-M	DenseNet121	ResNet50	AlexNet
Classe 00	0,5399	0,4563	0,6692	0,6901	0,8023	0,5722
Classe 01	0,9646	0,9445	0,9502	0,9474	0,8727	0,8641
Classe 02	0,9128	0,8876	0,8750	0,8803	0,8498	0,8225
Classe 03	0,4906	0,0000	0,0189	0,4340	0,5849	0,1887
Classe 04	0,8832	0,9346	0,9299	0,8645	0,7757	0,9299
Classe 05	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Classe 06	0,6456	0,0000	0,7089	0,8861	0,8861	0,7595
Classe 07	0,2948	0,2601	0,6936	0,4277	0,6012	0,1098
Classe 08	0,5063	0,0759	0,7152	0,5443	0,7468	0,5633
Classe 09	0,5075	0,4179	0,6418	0,7612	0,4776	0,5075

Fonte: Criado pela Autora.

A Figura 34 apresenta o resultado da matriz de confusão do modelo EfficientNet-V2-M no presente experimento. Como essa arquitetura apresentou o melhor resultado é válido analisar como ela se comporta classificando as classes morfológicas das galáxias. Nela é possível observar comportamentos como: a dificuldade de classificar corretamente classes que têm menos amostras e uma confusão notável entre as classes *Disk*, *Face-on*, *Tight*, *Medium* e *Loose Spiral* com a classe *Disk*, *Face-on*, *No Spiral*.

Figura 34 – Matriz de confusão da rede EfficientNetV2M.



Fonte: Elaborado pela autora.

Na Tabela 4.5, os modelos foram avaliados em uma base de dados com amostras desbalanceadas, mas com aumento das amostras da classe *Disk, Edge-on, Boxy Bulge* para 204, é possível observar que o modelo *ResNet50* obteve os melhores resultados, o que pode indicar que o aumento das amostras da classe minoritária possa ter melhorado o desempenho desse modelo em relação aos resultados apresentados no experimento anterior. Em seguida, na Tabela 4.6 exibe a acurácia de cada modelo em relação a classificação de classe de forma individual.

Tabela 4.5 – Resultados dos modelos avaliados na base de dados com classes de amostras desbalanceadas, porém com aumento das amostras da classe *Disk, Edge-on, Boxy Bulge* para 204 e com *early stopping*.

Modelo	Acurácia	Precisão	F1	Recall	Loss	Épocas
EfficientNet-V2-M	0,7798	0,7799	0,7798	0,7731	0,7075	05
DenseNet121	0,8019	0,8025	0,8019	0,7979	0,6145	08
ResNet50	0,8320	0,8271	0,8320	0,8261	0,7724	11
AlexNet	0,7489	0,7350	0,7489	0,7291	0,7377	09
VGG19	0,7392	0,7386	0,7392	0,7159	0,7099	11
VGG16	0,7631	0,7548	0,7631	0,7512	0,6657	10

Fonte: Criado pela Autora.

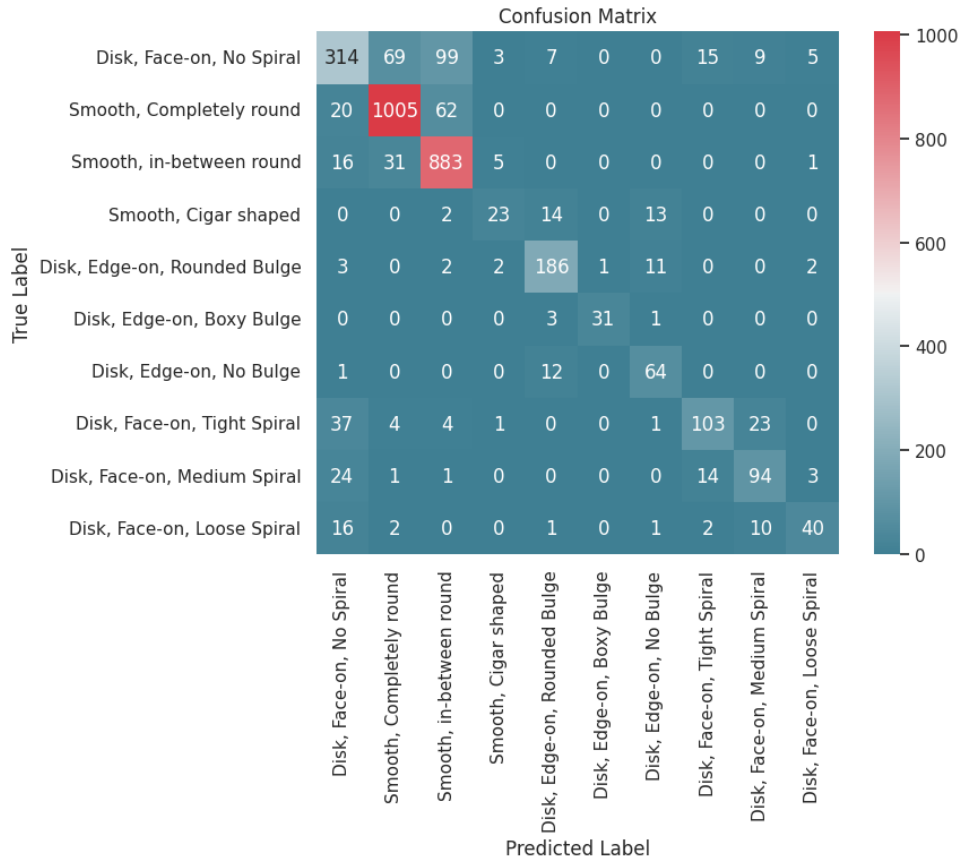
Tabela 4.6 – Acurácia dos modelos por classe.

Classe	VGG19	VGG16	EfficientNet-V2-M	DenseNet121	ResNet50	AlexNet
Classe 00	0,5451	0,4818	0,7006	0,5912	0,6027	0,5010
Classe 01	0,9181	0,9117	0,8188	0,8868	0,9246	0,8924
Classe 02	0,8910	0,8996	0,9391	0,9049	0,9434	0,9135
Classe 03	0,0000	0,0769	0,5000	0,6346	0,4423	0,2115
Classe 04	0,7729	0,9227	0,7923	0,8019	0,8986	0,8841
Classe 05	0,0000	0,0571	0,0000	0,8571	0,8857	0,0000
Classe 06	0,6623	0,5065	0,7922	0,6494	0,8312	0,6104
Classe 07	0,0347	0,5029	0,4971	0,6647	0,5954	0,6012
Classe 08	0,5985	0,5036	0,4745	0,6788	0,6861	0,1168
Classe 09	0,3056	0,5556	0,4861	0,5278	0,5556	0,3056

Fonte: Criado pela Autora

A Figura 35 exibe a matriz de confusão da arquitetura ResNet50 no experimento mencionado anteriormente. Essa representação visual é importante para identificação das classes que estão sendo classificadas de maneira precisa e daquelas que estão sendo frequentemente confundidas. É evidente a persistência da confusão entre as classes *Disk*, *Face-on*, *Tight*, *Medium* e *Loose Spiral* com a classe *Disk*, *Face-on*, *No Spiral*, como observado no experimento anterior com os dados desbalanceados. No entanto, é notável um desempenho aprimorado em classes que anteriormente possuíam poucas amostras e apresentaram poucos acertos, cenário também observado no experimento anterior.

Figura 35 – Matriz de confusão da rede ResNet50,



Fonte: Elaborado pela autora.

Na Tabela 4.7, os modelos foram avaliados em uma base de dados com as amostras das classes balanceadas através da quantidade de amostras da classe minoritária, ou seja, 17 amostras (*undersampling*). Aqui, os resultados mostram que os modelos tiveram um desempenho geral muito baixo em relação aos experimentos anteriores, com acurácias muito inferiores. A diminuição dos dados resultou em uma perda significativa, o que afetou negativamente a capacidade dos modelos classificarem as galáxias.

Tabela 4.7 – Resultados dos modelos avaliados na base de dados com as amostras das classes balanceadas em 17 amostras (*undersampling*).

Modelo	Acurácia	Precisão	F1	Recall	Loss	Épocas
EfficientNetV2	0,1852	0,1613	0,1484	0,1852	2.6714	03
DenseNet121	0,2593	0,4000	0,2775	0,2593	1.8130	06
ResNet50	0,2963	0,3679	0,2852	0,2963	6.7297	07
AlexNet	0,1852	0,1222	0,1023	0,1852	2.0304	08
VGG19	0,0741	0,0055	0,0102	0,0741	2.3324	04
VGG16	0,0000	0,0000	0,0000	0,0000	2.3527	06

Fonte: Criado pela Autora.

Na Tabela 4.8, os modelos foram avaliados em uma base de dados com aumento das

amostras da classe *Disk*, *Edge-on*, *Boxy Bulge* para 204 e, em seguida, balanceamento em 204 amostras nas demais classes (*undersampling*). Aqui, podemos ver que os resultados ainda são inferiores aos obtidos no experimento com amostras desbalanceadas e aumento, indicando também que a redução das amostras prejudicou o desempenho dos modelos.

Tabela 4.8 – Resultados dos modelos avaliados na base de dados com aumento das amostras da classe *Disk*, *Edge-on*, *Boxy Bulge* para 204, e em seguida um balanceadas em 204 amostras nas demais classes (*undersampling*).

Modelo	Acurácia	Precisão	F1	Recall	Loss	Épocas
EfficientNet-V2-M	0,6046	0,6154	0,6046	0,5958	1.1019	09
DenseNet121	0,5490	0,6050	0,5490	0,5492	1.3766	06
ResNet50	0,5261	0,5808	0,5261	0,4959	1.9682	07
AlexNet	0,5098	0,5593	0,5098	0,5155	1.2462	11
VGG19	0,0948	0,0090	0,0948	0,0164	2.3046	03
VGG16	0,0686	0,0047	0,0686	0,0088	2.3053	03

Fonte: Criado pela Autora.

Por fim, a Tabela 4.9 apresenta os resultados dos modelos avaliados em uma base de dados com a técnica de *oversampling*, onde os dados foram balanceados realizando o aumento do número de amostras das classes minoritárias a partir da classe majoritária, na base de dados consiste na classe 01 que possui 6.997 amostras, logo, ao final do processo, todas as classes possuem 6.997 amostras. Podemos ver que os resultados são melhores comparados aos modelos treinados com a base desbalanceada, e também aos modelos treinados com base balanceada com poucas amostras. É possível também destacar que os modelos *ResNet50*, *DenseNet121* e *EfficientNet-V2-M* apresentaram os melhores resultados, já *VGG19* e *VGG16* não trouxeram bons resultados. Essas duas arquiteturas classificaram todas as imagens em uma só classe, podendo indicar que um grande volume de dados prejudica o desempenho desses modelo.

Tabela 4.9 – Resultados das métricas do experimento com balanceamento dos dados. Esta tabela apresenta os resultados dos modelos avaliados na base de dados balanceada com *oversampling*, onde todas as classes possuem 6.997 amostras.

Modelo	Acurácia	Precisão	F1	Recall	Loss	Épocas
EfficientNet-V2-M	0,8708	0,8741	0,8708	0,8696	0,4184	07
DenseNet121	0,8796	0,8788	0,8796	0,8777	0,3620	14
ResNet50	0,8820	0,8832	0,8820	0,8816	0,3771	08
AlexNet	0,7712	0,7752	0,7712	0,7702	0,6864	11
VGG19	0,0957	0,0091	0,0957	0,0167	2.3028	06
VGG16	0,0994	0,0099	0,0180	0,0994	2.3027	05

Fonte: Criado pela Autora.

A Matriz de Confusão na Figura 36 ilustra o desempenho da arquitetura ResNet50 no experimento mencionado previamente. Ela oferece *insights* sobre como o modelo se comporta e

onde ocorrem as confusões durante o processo de classificação.

Figura 36 – Matriz de confusão da rede ResNet50



Fonte: Elaborado pela autora.

Em geral, os resultados destacam o desafio enfrentado ao lidar com classes desbalanceadas, e as estratégias de pré-processamento, como aumento de dados e balanceamento, têm um impacto significativo no desempenho dos modelos. O uso de modelos mais robustos, como o *ResNet50*, *EfficientNetV2-Me* o *DenseNet121*, mostra-se uma abordagem promissora para enfrentar esse desafio. No entanto, é importante reconhecer que a escassez de amostras pode afetar substancialmente os resultados finais dos modelos de classificação.

O balanceamento dos dados, combinado com o aumento das amostras das classes minoritárias, demonstrou resultados satisfatórios, especialmente em modelos com arquiteturas mais robustas. No entanto, é essencial considerar que o processo de *oversampling* pode ser computacionalmente custoso. Portanto, é necessário encontrar um equilíbrio entre o ganho de desempenho e o custo computacional ao escolher as estratégias de pré-processamento.

4.4 Experimento III

Neste experimento, foram conduzidos cinco treinamentos de diferentes CNNs, cada uma utilizando arquiteturas selecionadas, para cada um dos níveis da árvore representada na Figura

32. No total, foram empregadas as seguintes arquiteturas e seus respectivos resultados, conforme apresentado na Tabela 4.10 que exibe as métricas dos dados de teste obtidos com a classificação hierárquica seguindo a árvore da Figura 32.

Tabela 4.10 – Resultados da classificação hierárquica de acordo com a arquitetura utilizada.

Modelo	Acurácia	Precisão	F1	Recall
EfficientNet-V2-M	0,6568	0,6791	0,6372	0,6568
DenseNet	0,7020	0,6873	0,6347	0,7020
AlexNet	0,6990	0,7032	0,6337	0,6990
ResNet50	0,7320	0,6983	0,7037	0,7320
VGG16	0,2817	0,0794	0,1239	0,2817
VGG19	0,3090	0,0955	0,1459	0,3090

Fonte: Criado pela Autora.

Neste experimento, os dados foram utilizados sem qualquer alteração da base original. No entanto, devido à constatação de resultados não tão promissores, a decisão foi interromper o experimento. Diante dessa circunstância, optou-se por não prosseguir com essa abordagem e não realizar novos experimentos com outros pré-processamento dos dados.

Enquanto algumas redes foram capazes de alcançar níveis aceitáveis de acurácia, quando comparados com experimento anteriores, outras demonstraram uma capacidade inferior de discriminação entre as classes hierárquicas.

Durante os processos de treinamento das redes neurais, foi possível observar uma tendência consistente em relação à dificuldade de aprendizado para um grupo específico de classes. Isso foi avaliado mediante os resultados de acurácia durante o treinamento das redes, revelando uma clara disparidade. Em todos os experimentos realizados, notou-se que a rede destinada à classificação das classes referentes a ‘*Disk, Face-on, Tight Spiral*’, ‘*Disk, Face-on, Medium Spiral*’ e ‘*Disk, Face-on, Loose Spiral*’ apresentou um desempenho inferior em comparação com as demais categorias.

Enquanto as redes demonstraram uma acurácia na faixa de 80-100% para as outras classes, para esse grupo específico, os resultados se situaram entre 50-70%. Essa discrepância constante nos resultados evidencia uma maior complexidade ou dificuldade de discriminação entre essas subcategorias específicas, sugerindo que essas classes possam compartilhar características visuais ou estruturais que são mais desafiadoras de serem aprendidas pelas redes neurais.

4.5 Experimento IV

As arquiteturas treinadas no Experimento II foram agrupadas usando o método de aprendizado de conjunto (*ensemble learning*), empregando as regras de votação *hard* e média. Este experimento foi conduzido em dois conjuntos de dados: o conjunto de dados padrão e o con-

junto de dados balanceado com a técnica de *oversampling*, pois esses dois conjuntos de dados apresentaram bons resultados durante o Experimento II. Os resultados referentes aos dados sem balanceamento estão apresentados na Tabela 4.11, enquanto a Tabela 4.12 contém os resultados do conjunto de dados balanceado.

É importante ressaltar que, nos dados balanceados, nem todas as arquiteturas do Experimento II foram utilizadas. As arquiteturas VGG16 e VGG19 apresentaram desempenho significativamente inferior, sendo excluídas da análise. Portanto, apenas as redes EfficientNet-V2-M, ResNet50 e DenseNet121 foram consideradas nesse caso. Notavelmente, houve uma melhoria nas métricas avaliadas, especialmente com as redes treinadas nos dados balanceados, as quais demonstraram um desempenho superior, corroborando os resultados observados no Experimento II.

Tabela 4.11 – Resultados do aprendizado por agrupamento (*ensemble learning*) das redes anteriormente treinadas no Experimento II com os dados sem alteração. Arquiteturas utilizadas: EfficientNet-V2-M, ResNet50, DenseNet121, VGG16 e VGG19.

Arquiteturas	Acurácia	Precisão	F1	Recall	Loss
<i>Hard</i>	0,8506	0,8442	0,8506	0,8441	0,1275
<i>Averaging</i>	0,8653	0,8597	0,8653	0,8602	0,8292

Fonte: Criado pela Autora.

Tabela 4.12 – Resultados do aprendizado por agrupamento (*ensemble learning*) das redes anteriormente treinadas no Experimento II com os dados balanceados. Arquiteturas utilizadas: EfficientNet-V2-M, ResNet50 e DenseNet1219.

Arquiteturas	Acurácia	Precisão	F1	Recall	Loss
<i>Hard</i>	0,9071	0,9116	0,9071	0,9061	0,0681
<i>Averaging</i>	0,9175	0,9203	0,9175	0,9168	0,4963

Fonte: Criado pela Autora.

Com esse experimento, obteve-se métricas com os melhores resultados da pesquisa, alcançando uma acurácia de 91.75% nos testes, representando o melhor desempenho alcançado até o momento.

4.6 Discussão dos resultados

Os resultados observados nos experimentos, apresentados no capítulo anterior, foram diretamente afetados pelo desbalanceamento dos dados e pela falta de amostras em algumas classes. Observa-se que, a utilização de *undersampling* não se mostrou tão efetiva, o que pode estar relacionado com a baixa quantidade de amostras em algumas classes, evidenciando a complexidade significativa desse processo.

Quando as classes estão desbalanceadas, os melhores resultados em termos de acurácia foram 82.9% com *EfficientNet-V2-M* sem aumento nas amostras da classe 5 e 83.2% utilizando *ResNet50* com aumento das amostras da classe 5 para 204. Pode-se afirmar que o aumento para 204 amostras não resultou em uma melhora significativa na métrica geral dos modelos. No entanto, ao observar a acurácia da classificação individual por classe, os modelos *DenseNet121* e *ResNet50* alcançaram uma acurácia de mais de 85% para a classe 5, que antes do aumento era 0%. Esse fato também é evidenciado na matriz de confusão dos dois experimentos. Isso mostra que aumentar a quantidade de amostras para classes com poucas amostras pode resultar em melhorias na sua classificação.

Por outro lado, o balanceamento dos dados com a técnica de *oversampling* trouxe melhorias significativas nas redes mais robustas, com *EfficientNet-V2-M* alcançando 87.08% de acurácia, *DenseNet121* com 87.96% de acurácia e *ResNet50* com 88.20% de acurácia com a base de teste. Além disso, é válido considerar que, para alcançarem esses resultados, o número de épocas necessárias, devido ao uso de *early stopping*, foi relativamente baixo em comparação com o máximo da configuração, que é de 50 épocas.

Ao comparar os resultados dos experimentos com a CNN proposta por [Gharat e Dandawate \(2022\)](#) - Experimento I - com os experimentos utilizando TL - Experimento II - , é possível constatar que o treinamento ocorre de forma mais rápida em relação à quantidade de épocas necessárias para obter o resultado. No mesmo cenário, com *patience* igual a três e dados desbalanceados, temos o melhor resultado através da *ResNet50* com 83.2% de acurácia, enquanto a CNN apresenta 75.6% de acurácia, mostrando-se inferior. Já no cenário de balanceamento, fica evidente que a CNN é inferior em relação às arquiteturas mais robustas utilizadas com a técnica de TL.

A abordagem de classificação hierárquica ofereceu *insights* sobre um grupo específico de classes que as redes podem estar enfrentando maior dificuldade em classificar. No entanto, apesar dessa análise mais detalhada, os resultados obtidos não superaram os experimentos anteriores. O melhor desempenho alcançado foi uma acurácia de 73.20%. Este resultado, embora informativo, ficou abaixo das métricas de desempenho obtidas nos experimentos anteriores, ressaltando a dificuldade adicional que a classificação hierárquica pode apresentar em comparação com abordagens anteriormente apresentadas de classificação direta das classes.

Contrastando com os resultados anteriores, os experimentos envolvendo *ensemble learning* demonstraram um desempenho superior, revelando-se como uma técnica altamente promissora para a resolução do problema em questão. Ao aplicar essa abordagem aos dados não balanceados, alcançamos uma notável acurácia de 86.53%, representando um aumento significativo de 3.63% em relação ao melhor resultado obtido através do treinamento isolado das redes utilizando a técnica de Transfer Learning (TF) com esses dados.

Por outro lado, ao empregar o *ensemble learning* e lidar com dados balanceados, a acurácia atingiu 91.75%. Esta abordagem revelou uma diferença de precisão de aproximadamente 3.55%

em comparação com a melhor técnica identificada no Experimento II com dados balanceados. Essa sutil diferença destaca a consistência e o notável desempenho desse método, enfatizando sua relevância como estratégia primordial para otimização e aprimoramento dos resultados em análises e classificações de dados similares.

4.7 Disponibilização do Modelo

Após uma análise minuciosa dos resultados obtidos, ficou evidente que a abordagem mais satisfatória para a classificação das imagens do Galaxy10 SDSS Dataset foi a utilização da técnica de *ensemble learning*, combinada ao treinamento de redes com PyTorch e o emprego de *oversampling* para lidar com desequilíbrios nos dados.

Como resultado desses esforços, os modelos resultantes foram disponibilizados no Hugging Face Model Hub para acesso público. Eles podem ser acessados através do seguinte link: <https://huggingface.co/enyasantos/galaxy-classification-v02>.

Além disso, como a abordagem de *ensemble learning* com os dados desbalanceados também apresentou resultados satisfatórios, também foi disponibilizado no Hugging Face Model Hub e é acessível através do link: <https://huggingface.co/enyasantos/galaxy-classification-v01>.

A disponibilização desses modelos é um convite aberto à comunidade para explorar, utilizar e contribuir com o aprimoramento contínuo desses recursos, visando impactar positivamente pesquisas futuras e aplicações práticas.

4.8 Comparação com Trabalho Relacionados

Nesta seção, realiza-se um comparativo com os resultados mais significativos de dois estudos anteriores com os obtidos no nosso trabalho, todos utilizando a mesma base de dados para a tarefa de classificação. No estudo de [Gharat e Dandawate \(2022\)](#), optou-se pelo treinamento de uma rede neural convolucional do zero, resultando em uma acurácia de 84.78%. Em [Şenel \(2023\)](#), adotou-se a otimização de CNN com o algoritmo GWO, atingindo 85% de acurácia. Ambos os resultados indicaram desempenhos sólidos em suas abordagens.

No presente estudo, alcançou-se o melhor resultado por meio da aplicação de *fine tuning* em conjunto com o *ensemble learning*. Essa estratégia aproveitou as arquiteturas pré-treinadas da AlexNet, DenseNet121, EfficientNet-V2-M e ResNet50, treinadas em um conjunto diversificado de dados. Além disso, aplicamos a técnica de *oversampling* para mitigar o desequilíbrio entre as classes. Como resultado, alcançou-se uma acurácia de 91.75% na base de teste, superando os estudos mencionados anteriormente. Aplicando a técnica de *fine tuning* em conjunto com *ensemble learning* com as arquiteturas pré-treinadas AlexNet, DenseNet121, EfficientNet-V2-M, ResNet50, VGG16, e VGG19 e dados desbalanceados, também apresentam resultados satisfatórios com a acurácia de 86.53% na base de teste.

A Tabela 4.13 resume as principais diferenças entre esses estudos, destacando as abordagens adotadas e os resultados obtidos por cada um.

Tabela 4.13 – Comparação dos resultados obtidos com outros trabalhos na literatura.

Pesquisa	Abordagem	Acurácia em Teste	Estratégia	Equilíbrio de Classes
Gharat e Dandawate (2022)	Treinamento CNN do zero	84.78%	Baseada em construção do zero	Não mencionado
Şenel (2023)	CNN com algoritmo GWO	85%	Baseada em algoritmos de otimização de CNN	Não mencionado
Presente Estudo	<i>Ensemble learning</i> com TL	86.53%	Baseada aprendizado por conjunto	Dados desbalanceados
Presente Estudo	<i>Ensemble learning</i> com TL	91.75%	Baseada aprendizado por conjunto	Abordado com <i>oversampling</i>

Fonte: Criado pela Autora

Essa comparação destaca como diferentes abordagens podem levar a resultados distintos, demonstrando a eficácia do *ensemble learning* em conjunto com *transfer learning* e da estratégia de pré-processamento adotada neste estudo.

5 Considerações Finais

Neste capítulo são apresentadas as conclusões obtidas ao longo do trabalho, descrevendo os objetivos alcançados e os resultados obtidos durante os experimentos realizados. Por fim, o capítulo também apresenta propostas para a continuação deste estudo, levando em consideração o que foi concluído e alcançado através do presente trabalho.

5.1 Conclusão

Este trabalho apresentou o desenvolvimento de experimentos com redes neurais convolucionais para realizar a classificação morfológica de galáxias a partir de imagens astronômicas, seguindo a proposta de [Gharat e Dandawate \(2022\)](#). Como objetivos específicos têm-se a seleção e preparo de um conjunto de dados adequado para o problema, o treinamento de modelos de redes neurais, a avaliação do desempenho desses modelos treinados e a análise dos resultados obtidos, com o intuito de identificar possíveis desafios e limitações nesta técnica.

A base de dados utilizada durante os processos de treinamento, validação e teste foi a [Galaxy10 SDSS Dataset](#), disponibilizada através de uma biblioteca em Python, contendo 21.785 imagens no total, todas com dimensões de 69x69 pixels. A metodologia apresentada neste documento possibilitou o desenvolvimento de modelos de redes neurais convolucionais para a classificação morfológica das galáxias, permitindo a realização de experimentos para avaliar as melhores arquiteturas e diferentes configurações de bases de dados.

Os testes revelaram que o balanceamento de dados, neste caso utilizando *data augmentation* para alcançar esse objetivo, em conjunto com uma arquitetura de rede robusta, como a *ResNet50*, pode alcançar uma acurácia de teste de 88.20%. Para esse resultado, os dados foram aumentados para um total de 69.970 imagens, o que demandou um custo computacional considerável, mas foi compensado pelo uso do TF, permitindo atingir os resultados de forma mais rápida por conta do pré-treinamento.

A arquitetura proposta por [Gharat e Dandawate \(2022\)](#) obteve um resultado aproximado ao descrito no trabalho, que era de 84.82% de acurácia nos testes, alcançando um valor de 82.56% ao remover a técnica de Early Stopping. Entretanto, ao aumentar a quantidade de dados com o balanceamento totalizando 69.970 imagens, a arquitetura não apresentou resultados tão satisfatórios, atingindo no máximo 75.76% de acurácia nos testes, o que sugere que a rede pode não ser suficientemente robusta quando lida com uma grande quantidade de dados.

Os resultados alcançados com a técnica de classificação hierárquica destacam-se, oferecendo *insights* valiosos para a classificação morfológica de galáxias. No entanto, a abordagem hierárquica, embora tenha fornecido informações úteis sobre classes mais desafiadoras, não

superou os experimentos anteriores, atingindo uma acurácia máxima de 75.99%.

A abordagem de *ensemble learning* demonstrou-se particularmente eficaz, mostrando um ganho substancial de desempenho em relação às técnicas convencionais. Essa melhoria significativa ressalta a importância de explorar estratégias mais avançadas de combinação de modelos para otimizar a precisão e a capacidade de generalização em tarefas desafiadoras de classificação, como a morfologia de galáxias a partir de imagens astronômicas.

Por fim, é possível afirmar, com base nos resultados obtidos, que as implementações realizadas apresentam uma alternativa viável para a classificação morfológica de galáxias. Fornecendo *insights* valiosos sobre a eficácia de arquiteturas e modelos de CNN, bem como técnicas para ajuste dos dados. Os resultados sugerem que as arquiteturas *ResNet50*, *DenseNet121* e *EfficientNet-V2-M*, em conjunto com técnicas de TF, podem ser opções promissoras para a classificação morfológica de galáxias através de imagens astronômicas. Contudo, os resultados mostram uma melhoria substancial ao empregar técnicas de ensemble learning, superando a precisão das arquiteturas individuais, indicando um caminho promissor para futuras pesquisas na área.

5.2 Trabalhos Futuros

Há diversas oportunidades para aprimorar e expandir as capacidades do modelo de classificação morfológica de galáxias. Para otimizar a disponibilidade e utilidade do modelo, uma interface gráfica amigável para os usuários seria uma adição valiosa. Essa interface permitiria uma interação mais intuitiva e facilitaria o uso do modelo por usuários sem experiência técnica em programação, oferecendo funcionalidades para carregar imagens, realizar previsões e visualizar resultados de maneira compreensível.

A melhoria do pré-processamento de imagens é um ponto-chave para aprimorar a eficiência do modelo. Explorar diferentes técnicas, como normalização, aumento de contraste e remoção de ruído, poderia aperfeiçoar a qualidade dos dados de entrada, impactando diretamente na precisão das previsões.

Outro caminho interessante seria investigar como o tamanho das imagens afeta o desempenho do modelo. Testar diferentes resoluções poderia oferecer *insights* sobre a influência da qualidade da imagem na precisão das previsões, contribuindo para determinar a resolução ideal para contextos específicos de aplicação.

Referências

- Ben Lutkevich. **Hugging Face**. 2023. Disponível em: <<https://www.techtarget.com/whatis/definition/Hugging-Face#:~:text=Hugging%20Face%20is%20a%20machine,and%20train%20machine%20learning%20models.>>. Acesso em: 02 de janeiro 2024.
- BIALOPETRAVIČIUS, J.; NARBUTIS, D. Deriving star cluster parameters with convolutional neural networks-ii. extinction and cluster-background classification. **Astronomy & Astrophysics**, EDP Sciences, v. 633, p. A148, 2020.
- BRAGA, A. d. P.; LUDERMIR, T. B.; CARVALHO, A. C. P. d. L. F. **Redes neurais artificiais: teoria e aplicações**. [S.l.]: LTC, 2007.
- BUTA, R. J. **Galaxy Morphology**. 2013.
- CALLEJA, J. d. I.; FUENTES, O. Automated classification of galaxy images. In: SPRINGER. **International Conference on Knowledge-Based and Intelligent Information and Engineering Systems**. [S.l.], 2004. p. 411–418.
- CASTRO, L. de; FERRARI, D. **Introdução a mineração de dados**. Saraiva Educação S.A., 2017. ISBN 9788547200992. Disponível em: <<https://books.google.com.br/books?id=SSlrDwAAQBAJ>>.
- CAVANAGH, M. K.; BEKKI, K.; GROVES, B. A. Morphological classification of galaxies with deep learning: comparing 3-way and 4-way CNNs. **Monthly Notices of the Royal Astronomical Society**, v. 506, n. 1, p. 659–676, 06 2021. ISSN 0035-8711. Disponível em: <<https://doi.org/10.1093/mnras/stab1552>>.
- CHOLLET, F. **Deep Learning with Python, Second Edition**. Manning, 2021. ISBN 9781617296864. Disponível em: <<https://books.google.com.br/books?id=XHpKEAAQBAJ>>.
- CINTRA, R. **Introdução à Neurocomputação**. 2016. <http://www.inpe.br/elac2018/arquivos/ELAC2018_MC3_apostila.pdf>. (Accessed on 01/09/2024).
- DALCANTON, J. J. *et al.* The panchromatic hubble andromeda treasury. **The Astrophysical Journal Supplement Series**, IOP Publishing, v. 200, n. 2, p. 18, 2012.
- DENG, J. *et al.* Imagenet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2009. p. 248–255.
- DOSOVITSKIY, A. *et al.* An image is worth 16x16 words: Transformers for image recognition at scale. **arXiv preprint arXiv:2010.11929**, 2020.
- FIELDING, E.; NYIRENDA, C. N.; VACCARI, M. The classification of optical galaxy morphology using unsupervised learning techniques. In: **2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)**. [S.l.: s.n.], 2022. p. 1–6.
- FLECK, L. *et al.* Redes neurais artificiais: Princípios básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, v. 1, n. 13, p. 47–57, 2016.
- FORTSON, L. *et al.* Galaxy zoo: Morphological classification and citizen science. **Advances in Machine Learning and Data Mining for Astronomy**, 04 2011.

- FRANK, M. **Wastewater Renovation and Reuse: Proceedings of the International Conference on the Renovation and Reuse of Wastewater Through Aquatic and Terrestrial Systems**. [S.l.]: Marcel Dekker, 1977. v. 3.
- FURTADO, M. I. V. **Redes neurais artificiais: uma abordagem para sala de aula**. Ponta Grossa, PR. Atena Editora, p. 19, 2019.
- GAUTHIER, A.; JAIN, A.; NOORDEH, E. Galaxy morphology classification. **Lecture Notes, Stanford University**, v. 16, 2016.
- GHARAT, S.; DANDAWATE, Y. Galaxy classification: a deep learning approach for classifying Sloan Digital Sky Survey images. **Monthly Notices of the Royal Astronomical Society**, v. 511, n. 4, p. 5120–5124, 02 2022. ISSN 0035-8711. Disponível em: <<https://doi.org/10.1093/mnras/stac457>>.
- GOLDBARG, M. A. S. d. S. **Análise de técnicas de compressão em redes neurais profundas por poda em dataset de imagens**. Dissertação (B.S. thesis) — Universidade Federal do Rio Grande do Norte, 2021.
- HARRISON, M. **Machine Learning – Guia de Referência Rápida: Trabalhando com dados estruturados em Python**. Novatec Editora, 2019. ISBN 9788575228173. Disponível em: <<https://books.google.com.br/books?id=i-7CDwAAQBAJ>>.
- HAYKIN, S. **Redes Neurais - 2ed**. Bookman, 2001. ISBN 9788573077186. Disponível em: <<https://books.google.com.br/books?id=1Bp0X5qfyjUC>>.
- HE, K. *et al.* Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.
- HOSNA, A. *et al.* Transfer learning: a friendly introduction. **Journal of Big Data**, Springer, v. 9, n. 1, p. 102, 2022.
- HUANG, G. *et al.* Densely connected convolutional networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 4700–4708.
- HUBBLE, E. **The Realm of the Nebulae**. New Haven: Yale University Press, 1936.
- KHAN, A. A. *et al.* Crowd anomaly detection in video frames using fine-tuned alexnet model. **Electronics**, MDPI, v. 11, n. 19, p. 3105, 2022.
- KIM, P. **MATLAB Deep Learning With Machine Learning, Neural Networks and Artificial Intelligence**. [S.l.: s.n.], 2017.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 1097–1105.
- KUMAR, A.; MAYANK, J. **Ensemble learning for ai developers**. BApres: Berkeley, CA, USA, Springer, 2020.
- LECUN, Y. *et al.* Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Ieee, v. 86, n. 11, p. 2278–2324, 1998.
- LIN, J. Y.-Y. *et al.* **Galaxy Morphological Classification with Efficient Vision Transformer**. 2022.

- LUDERMIR, T. B. Inteligência artificial e aprendizado de máquina: estado atual e tendências. **Estudos Avançados**, SciELO Brasil, v. 35, p. 85–94, 2021.
- MAKIN, J. Backpropagation. **Cornell Computer Science Courses**, 2006.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, p. 115–133, 1943.
- MIENYE, I. D.; SUN, Y. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. **IEEE Access**, IEEE, v. 10, p. 99129–99149, 2022.
- MISRA, D. *et al.* Convolved cosmos: Classifying galaxy images using deep learning. In: SHARMA, N.; CHAKRABARTI, A.; BALAS, V. E. (Ed.). **Data Management, Analytics and Innovation**. Singapore: Springer Singapore, 2020. p. 569–579. ISBN 978-981-32-9949-8.
- MONTES, G.; LATTARI, L.; COELHO, A. Redes neurais convolucionais otimizadas para a detecção de supernovas. In: **Anais do XV Brazilian e-Science Workshop**. Porto Alegre, RS, Brasil: SBC, 2021. p. 1–8. ISSN 2763-8774. Disponível em: <<https://sol.sbc.org.br/index.php/bresci/article/view/15782>>.
- OLIVEIRA, M. d. F. S. Kepler de. **Astronomia e Astrofísica**. [S.l.: s.n.], 2013. 808 p. ISBN 978-8578611873.
- PONTI, M. A.; COSTA, G. B. P. D. Como funciona o deep learning. **arXiv preprint arXiv:1806.07908**, 2018.
- RUSK, N. Deep learning. **Nature Methods**, v. 13, n. 1, p. 35–35, Jan 2016. ISSN 1548-7105. Disponível em: <<https://doi.org/10.1038/nmeth.3707>>.
- SARAIVA, K. d. S. O. F. e. A. M. M. Maria de F. O. **Aula 23: Galáxias**. 2014. <<http://www.if.ufrgs.br/~fatima/fis2010/Aula23-132.pdf>>. (Accessed on 06/24/2023).
- ŞENEL, F. A. A hyperparameter optimization for galaxy classification. **Computers, Materials & Continua**, v. 74, n. 2, 2023.
- SILLA, C. N.; FREITAS, A. A. A survey of hierarchical classification across different application domains. **Data Mining and Knowledge Discovery**, v. 22, p. 31–72, 2010. Disponível em: <<https://api.semanticscholar.org/CorpusID:207113055>>.
- SILVA, I. N. d.; SPATTI, D. H.; FLAUZINO, R. A. **Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas. Fundamentos Teóricos e Aspectos Práticos**. [S.l.]: Artliber Editora, 2016.
- SILVA, M.; VENTURA, T. Classificação morfológica de galáxias por meio de redes neurais. In: **Anais da X Escola Regional de Informática de Mato Grosso**. Porto Alegre, RS, Brasil: SBC, 2019. p. 31–36. ISSN 2447-5386. Disponível em: <<https://sol.sbc.org.br/index.php/eri-mt/article/view/8590>>.
- SILVA, R. O. V. Josenildo C. da. Minicursos da ercemapi 2022 - capítulo 4: Introdução às redes neurais profundas com python. **Sociedade Brasileira de Computação**, 2022.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: **Proceedings of the international conference on learning representations**. [S.l.: s.n.], 2015.

- SOARES, D. **A forquilha de galáxias de Hubble**. 2007. <<http://lilith.fisica.ufmg.br/~dsoares/reino/forhub-p.pdf>>. (Accessed on 06/24/2023).
- SOARES, D. **O Reino das Galáxias**. [S.l.: s.n.], 2021. ISBN 978-65-00-34729-6.
- TAN, M.; LE, Q. V. Efficientnetv2: Smaller models and faster training. **arXiv preprint arXiv:2104.00298**, 2021.
- VALK, G. A. *et al.* Análise da anisotropia orbital de populações de galáxias em aglomerados. Universidade Federal de Santa Maria, 2023.
- VIEIRA, V. *et al.* Tópicos em gerenciamento de dados e informações: Minicursos do sbbd 2017. **Tópicos**, v. 2, p. 10, 2017.
- WILLETT, K. W. *et al.* Galaxy zoo 2: detailed morphological classifications for 304 122 galaxies from the sloan digital sky survey. **Monthly Notices of the Royal Astronomical Society**, Oxford University Press, v. 435, n. 4, p. 2835–2860, 2013.
- WU, J.; BOADA, S. Using convolutional neural networks to predict galaxy metallicity from three-colour images. **Monthly Notices of the Royal Astronomical Society**, v. 484, p. 4683–4694, 04 2019.
- YORK, D. G. *et al.* The sloan digital sky survey: Technical summary. **The Astronomical Journal**, v. 120, n. 3, p. 1579, sep 2000. Disponível em: <<https://dx.doi.org/10.1086/301513>>.
- ZHANG, A. *et al.* Dive into deep learning. **arXiv preprint arXiv:2106.11342**, 2021.