CBPF – Centro Brasileiro de Pesquisas Físicas

Dissertação de Mestrado

Uma análise de Deep Learning aplicada à morfologia de galáxias no Southern Photometric Local Universe Survey

Ulisses Ribeiro

Orientador
Dr. Clécio Roque De Bom

Rio de Janeiro, RJ
2023

“UMA ANÁLISE DE DEEP LEARNING APLICADO À MORFOLOGIA DE GALÁXIAS NO SOUTHERN PHOTOMETRIC LOCAL UNIVERSE SURVEY”

ULISSES RIBEIRO DA SILVA

Dissertação de Mestrado em Física apresentada no Centro Brasileiro de Pesquisas Físicas do Ministério da Ciência Tecnologia e Inovação. Fazendo parte da banca examinadora os seguintes professores:

Documento assinado digitalmente
gov.br
CLECIO ROQUE DE BOM
Data: 04/10/2023 13:02:49-0300
Verifique em https://validar.iti.gov.br

Clécio Roque de Bom – Orientador/CBPF

Documento assinado digitalmente
gov.br
MARIANA PENNA LIMA VITENTI
Data: 30/08/2023 09:50:15-0300
Verifique em https://validar.iti.gov.br

Mariana Penna Lima Vitenti - UnB

Documento assinado digitalmente
gov.br
KARIN MENENDEZ DELMESTRE
Data: 30/08/2023 14:15:09-0300
Verifique em https://validar.iti.gov.br

Karín Menéndez-Delmestre – OV/UFRJ

Rio de Janeiro, 22 de agosto de 2023.

Ulisses Ribeiro

Uma análise de Deep Learning aplicada à morfologia de galáxias no Southern Photometric Local Universe Survey

Trabalho apresentado ao Programa de Pós-Graduação no Centro Brasileiro de Pesquisas Físicas como requisito parcial para obtenção do grau de Mestre em Física.

CBPF – Centro Brasileiro de Pesquisas Físicas

Orientador: Dr. Clécio Roque De Bom
Coorientador: Dra. Arianna Cortesi

Rio de Janeiro, RJ
2023

Ribeiro, Ulisses
Uma análise de Deep Learning aplicada à morfologia de galáxias no Southern Photometric Local Universe Survey/ Ulisses Ribeiro. - 2023
61 f. : il.

Dissertação de Mestrado – CBPF – Centro Brasileiro de Pesquisas Físicas , Rio de Janeiro, RJ, 2023.
Orientador: Dr. Clécio Roque De Bom

1. (Listar palavras-chave) I. Título
CDU 02:141:005.7

Dedico este trabalho
à minha avó (*in memorian*).

Agradecimentos

Primeiramente eu agradeço ao meu orientador Clécio De Bom e coorientadora Arianna Cortesi pela oportunidade de fazer parte deste trabalho, pela paciência que tiveram comigo e pelo conhecimento que compartilharam durante todo esse período. Nunca me senti desconfortável ou deslocado das atividades e boa parte disso se dá pelo caráter bem humorado e acolhedor de ambos.

Agradeço a meus pais Rosangela Ribeiro e Jefferson Ribeiro, juntamente da minha irmã Julia Ribeiro por sempre me incentivarem a seguir com o curso de Física. Vocês nunca me pressionaram a fazer outra coisa e muito pelo contrário sempre comemoraram minhas grandes e pequenas vitórias.

Agradeço a meus primos, tios, tias e avós por sempre serem tão gentis e acolhedores com a minha pessoa. Dias em família, com um bom churrasco e momentos descontraídos sempre foram refúgios para meus dias mais solitários.

Agradeço aos meus amigos que sempre trouxeram momentos leves para minha vida. Aqui vale destacar meu xará Ulisses Viana e minha amiga de longa distância Gabriela Barcelos por sempre me ajudarem nas questões filosóficas da vida, além de me ajudarem a enxergar melhor a pessoa que sou.

E finalmente, agradeço minha avó Augusta Rosa Ribeiro da Silva pelo seu esforço inicial em dar ao meu pai o que ele precisava para que hoje ele possa me dar o mesmo. Desde que nasci sempre foi para mim uma segunda mãe, sendo uma inspiração e exemplo de humildade em minha vida.

"O maior inimigo do conhecimento não é a ignorância, é a ilusão do conhecimento."

– Stephen Hawking

Resumo

O uso de redes neurais para a solução de diversos tipos de problemas tem ganhado espaço dentro da comunidade cientifica, visto o seu potencial em aprender, reconhecer e processar padrões gerando respostas precisas. Em especial, muitos estudos mostram uma alta performance em problemas de classificação, assim como este em que apresento neste trabalho. Ainda assim o processo de concepção dessas redes pode ser um pouco frustrante com a etapa de treinamento envolvendo várias medidas para que ocorra de forma suave. Em geral, não existem garantias de que a rede irá aprender o que se espera que ela aprenda, sendo assim o treinamento se torna um processo de muitas tentativas e investigações usando métricas para avaliar o desempenho tanto do treino, quanto da rede já treinada. Nesta tese discuto alguns procedimentos e decisões tomadas na concepção de uma rede neural capaz de classificar galáxias baseado em sua morfologia. Um trabalho que resultou em 164314 galáxias sendo classificadas usando o Data Release 3 do S-SPLUS. Eu também discuto as bases das redes neurais e mecanismos pelos quais essas redes são capazes de aprender. Espero que essa análise das técnicas de deep learning alerte para os cuidados necessários na sua implementação, ao mesmo tempo que destaque seu poder na resolução de problemas variados.

Palavras-Chave: Aprendizado Profundo, Inteligência Artificial, EfficientNet B2, Astronomia, Astrofísica, Morfologia de galáxias,

Abstract

The use of neural networks to solve different types of problems has gained space within the scientific community, given its potential to learn, recognize and process patterns generating accurate answers. In particular, many studies have shown a high performance in classification problems, such as the one I present in this work. Even so, the process of designing these networks can be a bit frustrating with the training step involving several cares so that it occurs smoothly. In general there are no guarantees that the network will learn what it is expected to learn, so the training can quickly turn into a process of many trials and investigations using metrics to evaluate the performance of both the training and the already trained network. In this thesis I discuss some procedures and decisions taken in the conception of a neural network capable of classifying galaxies based on their morphology. A work that resulted in 164314 galaxies being classified in the Data Release 3 of S-PLUS. I also discuss the bases of neural networks and the mechanisms by which these networks are able to learn. I hope this analysis of deep learning techniques highlights the cares needed for its implementation and present its power concerning problem solving.

Key-Words: Deep Learning, Artificial Intelligence, EfficientNet B2, Astronomy, Astrophysics, Galaxy Morphology

Lista de ilustrações

Lista de abreviaturas e siglas

CBPF	Centro Brasileiro de Pesquisas Físicas
IA	Inteligência Artificial
AM	Aprendizado de Máquina
RNA	Rede Neural Artificial
RNC	Rede Neural Convolucional
S-PLUS	Southern Photometric Local Universe Survey
ETG	Early Type Galaxy
LTG	Late Type Galaxy

Sumário

Capítulo 1

Introdução

A primeira revolução industrial trouxe as máquinas a vapor [2, 3] que transformaram os meios de produção gerando enorme automatização. Essas máquinas não eram capazes de pensar como os operários que elas substituíram, mas realizavam o trabalho braçal deles com alto desempenho. Essa tendência não parou desde então e hoje temos industrias inteiras focadas na fabricação de bens materiais onde boa parte dos trabalhadores são operários com a responsabilidade de garantir uma manutenção dessas máquinas.

Até este ponto as máquina apenas cobrem fazeres mecânicos do ser humano, pois elas apenas repetem de forma precisa rotinas do dia a dia de um trabalhador. Acontece que essa automação não se restringe ao campo mecânico, onde os computadores também são capazes de realizar tarefas que demandam certa atividade mental. Uma bem simples é a capacidade de fazer contas, onde hoje temos calculadoras capazes de resolver até equações diferenciais. Ainda assim, é importante notar que até este nível ainda existe uma grande diferença entre os humanos e as máquinas. O que temos é um programa que por meio de um conjunto de regras chamadas de **algoritmo** consegue executar tarefas logicas. Porém, por trás desse algoritmo, ainda existe uma mente pensante responsável pelo tratamento do problema de forma analítica e crítica.

A questão está no fato de que qualquer problema não é resolvido unicamente por um conjunto de regras passadas para o computador executar, tendo sempre um respaldo humano na sua criação ou execução. O ramo da astronomia é um desses casos que não dispensa recursos humanos, onde o universo e seus objetos são estudados com ajuda de imagens tiradas de telescópios e satélites [4]. A identificação do tipo desses objetos para, em seguida, estudar suas propriedades, envolve uma análise crítica dessas imagens e uma mente capaz de reconhecer padrões para classifica-las. Por outro lado, hoje contamos com vários levantamentos como o Legacy Survey [5], Sloan Digital Sky Survey (SDSS) [6, 7], Dark Energy Survey (DES) [8, 9], Southern Photometric Local Universe Survey (S-PLUS) [10] e vários outros projetos cobrindo extensas regiões do universo, o que aumenta

a demanda por técnicas mais automáticas que possam fazer essa identificação de forma rápida e robusta.

Na astronomia o processamento dessas imagens pode ser feito pelo computador, mas ainda existem problemas clássicos que são feitos utilizando recursos humanos. Um desses problemas é a classificação de galáxias, objetos extensos no universo que são muito dependentes das condições do meio interestelar, distância de observação, brilho e outros fatores que prejudicam uma classificação feita visualmente. Embora existam métodos que se proponham a classificar a estrutura das galáxias computacionalmente pelo cálculo de alguns parâmetros como assimetria , concentração da luz [11, 12], uma classificação puramente morfológica se apresenta como um desafio computacional. Por outro lado temos o aprendizado de máquinas que, através de modelos matemáticos, são capazes de analisar grandes volumes de dados, aprender e tomar decisões coerentes a partir deles, tornando-se úteis para problemas de classificação [13, 14]. Mais especificamente, tratando-se de imagens, temos as Redes Neurais Convolucionais (RNC) que, além de acelerar o processo de classificação, também apresentam um grande potencial para entregar resultados melhores que os dos seres humanos [15, 16, 17].

O trabalho desenvolvido nessa dissertação é uma extensão de uma primeira iniciativa que rendeu um artigo Bom et al. 2021 [18], no qual foi construída uma rede neural capaz de classificar galáxias entre Espirais (Late type) e Elípticas (Early type) para o projeto Southern Photometric Local Universe Survey (S-PLUS). Nessa dissertação analisaremos passo-a-passo quais foram as decisões necessárias para a construção do catálogo de morfologia de galáxias usando o Data Release 3 do S-PLUS, projeto apresentado em [19]. O projeto em questão também cataloga objetos de baixo brilho que podem se apresentar como um desafio, mesmo para classificações feitas visualmente. Portanto, analisar também as dificuldades na construção dessa rede neural, assim como seus futuros desafios se torna essencial para a sua constante melhora.

O texto está organizado da seguinte forma. No capítulo 2 apresento as bases mais gerais da rede neural, procurando analisar de forma analítica como o algoritmo consegue simular o aprendizado humano por meio de neurônios artificiais e reconhecer padrões dentro de uma imagem. O capítulo 3 se responsabiliza por trazer algumas bases de astronomia que são necessárias para tratar o problema de classificação, além de apresentar um pequeno histórico da importância e do estudo da morfologia das galáxias. Já no capítulo 4 o texto foca no uso de RNC para a classificação das galáxias, apontando as vantagens e dificuldades na concepção da rede. Este capítulo se destaca por discutir as tomadas de decisões que foram necessárias para tornar o modelo mais robusto. Passado pela arquitetura do modelo, o capítulo 5 mostra alguns resultados referentes ao seu desempenho no treinamento e performance final na classificação. Finalizo esta dissertação no capítulo 6, onde entrego uma conclusão para todo o trabalho e aponto possíveis caminhos futuros a serem seguidos.

Capítulo 2

Redes Neurais: uma história de automação

Nesta sessão trabalharemos alguns conceitos básicos que envolvem o aprendizado de máquinas com foco nas redes neurais. Introduzo dando algumas motivações para seu uso, para depois tratar de seus componentes principais como arquitetura, conjunto de treino e métricas de avaliação. Finalizo o capítulo tratando as Redes Neurais Convolucionais (RNC), que desempenham resultados melhores em problemas de classificação de imagem.

2.1 Motivação para o uso de redes neurais

O ser humano ao analisar um problema conta com seus conhecimentos e ferramentas disponíveis para fundamentar uma solução. Dentre essas ferramentas temos a matemática como uma opção que viabiliza uma solução mais analítica e o computador que, em associação com a matemática, estende essas capacidades de forma a facilitar tanto os cálculos numéricos, quanto a execução de tarefas lógicas a partir da programação. No entanto existem limites para o quanto uma solução pode ser dada de forma analítica ou numérica. Alguns problemas são muito complexos ou envolvem muitas variáveis, o que dificulta uma solução analítica como aqueles envolvendo sistemas caóticos ou equações diferencias não lineares [20, 21]. Outros envolvem muitos processos que são difíceis de serem traduzidos para um algoritmo, por serem muito dependentes de capacidades humanas como interpretar e tomar decisões críticas.

As redes neurais artificiais (RNA) são ferramentas adequadas para problemas sem solução analítica explícita e nos quais as regras de inferência não possam ser codificadas de maneira simples. As RNA apresentam um mescla entre o poder computacional de um computador, modelos matemáticos e conhecimentos estatísticos, fazendo com que essas regras possam ser aprendidas a partir de padrões em um conjunto de dados [22].

Um bom exemplo de seu uso está na área de processamento e classificação de imagens, tarefa que demanda certa interpretação e reconhecimento de padrões por parte dos seres humanos. Por essa linha de pensamento, tarefas simples para os seres humanos como a de classificar dígitos escritos em folha de papel se tornam complexas se tentarmos estritamente escrever um algoritmo para executá-la. No entanto, pelo ponto de vista computacional, uma imagem é interpretada pelo computador como um conjunto de píxeis que por sua vez são representados por números, então neste problema deve ser possível relacionar esse conjunto com a classificação que também é um número. De fato existem vários trabalhos usando redes neurais artificiais em um conjunto de dados conhecido como MNIST que executam tal tarefa [23, 24, 25]. Analogamente, alterações em uma imagem são transformações que levam um conjunto de píxeis em outro conjunto de píxeis. Assim é possível criar uma rede neural que aprende o estilo artístico de uma imagem e o tranfere para outra [26, 27, 28]. As redes neurais artificias são modelos matemáticos capazes de fazer esse intermédio entre um conjunto de dados iniciais e um conjunto de dados de resposta.

2.2 Princípios de uma Rede Neural Artificial

Para entendermos melhor o que são as RNAs não podemos deixar de falar de inteligência. Essa palavra é geralmente endereçada aos seres humanos, mas é bom ter em mente que este conceito existe em níveis diferentes em vários seres e eventos no nosso dia a dia. inteligência é um tópico de ampla discussão, porém a concepção mais comum deste conceito está ligada à capacidade de aprender. Nas RNAs esse aprendizado é feito de forma supervisionada, utilizando um conjunto de dados que chamamos de treino. Essencialmente o que se espera dessas redes é que elas consigam achar um caminho que leve os dados de um ponto inicial para um ponto final, processando eles durante o percurso. Na etapa de treino a RNA aprende a tratar esses dados sem a necessidade de alguém explicitamente programá-la para executar tal tarefa. Devido a essas características as redes neurais artificias se encontram dentro do grande tópico de inteligência Artificial.

Uma rede neural artificial é um algoritmo matemático que modela e representa a atividade neuronal, inspirando-se assim no funcionamento e na resposta a estímulos do cérebro [29]. Sua arquitetura reflete a complexidade da rede determinando como esses neurônios se interligam e processam a informação que passa através deles. Para melhor entender essas características vamos tomar uma rede simples que recebe 5 estímulos principais em uma arquitetura conhecida como **Perceptron** [30]. Além disso, para melhor se adequar à linguagem utilizada no meio, chamaremos os estímulos de entradas e as respostas de saídas. Toda rede neural é constituída de uma camada de neurônios de entrada em seu início e uma camada de neurônios de saída em seu fim. As camadas intermediárias são chamadas de camadas ocultas, sendo elas responsáveis por processar, transformar e pas-

sar a informação adiante até a camada de saída. O Perceptron é a rede neural mais simples possível, pois não contém camadas ocultas. Vamos chamar essa arquitetura de **Modelo-A**.

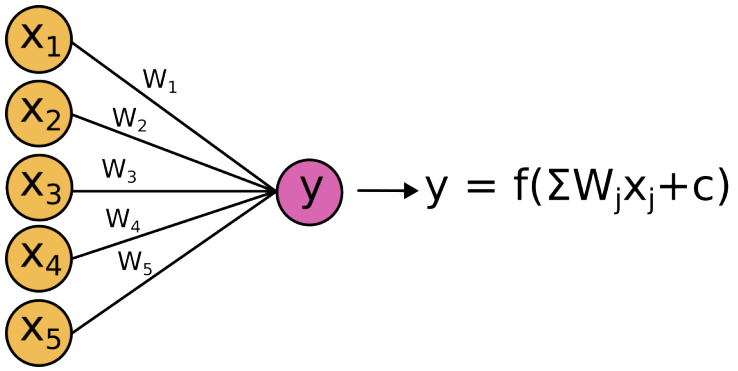

Figura 2.1 – Representação do modelo-A. Nessa arquitetura do tipo Perceptron o neurônio de saída y tem o valor de sua saída dada pela funçaõ de ativação f, que toma como argumento os valores dos neurônios de entrada mediados pelos pesos W_j.

Nas RNAs, a saída dos neurônios é dada por uma **função de ativação** que toma os neurônios imediatamente anteriores como variáveis. Existem vários tipos de função de ativação, onde cada uma trata esse conjunto de entradas de uma forma diferente. A escolha de cada uma está ligada ao tipo de problema, custo computacional e o tipo de saída que se espera da rede. A função de ativação do neurônio y não recebe apenas o somatório das entradas x_i mediadas pelos pesos, mas também vem somada de um bias c que também serve como um fator de ajuste para os sinais recebidos em cada neurônio. Um exemplo típico de função de ativação é a **ReLU - Unidade Linear Retificada** [31, 32]

$$\text{ReLU} \equiv f(x) = \max(0, x). \tag{2.1}$$

Nossa arquitetura do Modelo-A com essa função de ativação tem suas saídas positivas dadas por

$$y = O(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{5} W_j x_j + c, \tag{2.2}$$

onde definimos $O(x_1, x_2, \ldots, x_n) := \sum_{j=1}^{5} W_j x_j + c$ como a combinação linear dos pesos com as entradas de uma camada, somada por um bias. Veja que a ReLU contém algumas características, por exemplo, ela não é capaz de gerar um valor negativo. Isso pode parecer uma limitação, mas é um interessante recurso para filtrar valores indesejáveis, como será mostrado na seção (2.3.2).

Além disso, a ReLU contém um conjunto imagem ilimitada, podendo ter valores tão grandes quanto a intensidade de suas entradas. Sendo assim, se o objetivo for achar uma quantia que se limita a um certo intervalo talvez ela não seja a melhor escolha. Para

esse tipo de problema uma função de ativação adequada é a **Sigmoid** [33] que tem um conjunto imagem limitado ao intervalo $[0, 1]$. Esta função é muito utilizada nos neurônios de saída para problemas de classificação binária, como será discutido no capítulo (4). O valor de sua saída pode ser associado à probabilidade de um certo objeto pertencer a uma certa classe. Devido ao seu comportamento não linear essa função de ativação também pode ser útil para moldar problemas não lineares [34].

$$\text{Sigmoid: } \sigma(x) = \frac{1}{1 + e^{-x}}. \tag{2.3}$$

A escolha de uma função de ativação para os neurônios é uma etapa importante na construção da arquitetura do modelo e pode influenciar em sua convergência e aprendizado nas etapas de treino [33, 35].

O perceptron do Modelo-A ainda é bem limitado em seu potencial de processamento, pois ele apenas recebe um conjunto de dados inicias em seus neurônios sem efetuar muitas tranformações até enviá-los à saída. Por esse ponto de vista a Rede-A é simples e não tem muita capacidade de se adaptar a variados tipos de dados. Essa maior capacidade de ajuste para reconhecimento de padrões mais complexos ocorre se aumentarmos o número de parâmetros livres incluindo mais camadas ocultas, fomando um Perceptron de Múltiplas Camadas (PMC) [30]. Para explorar essa ideia de forma simples, vamos incluir apenas uma camada oculta contendo 2 neurônios como mostrado na figura (2.2). Para essa arquitetura daremos o nome de **Modelo-B**.

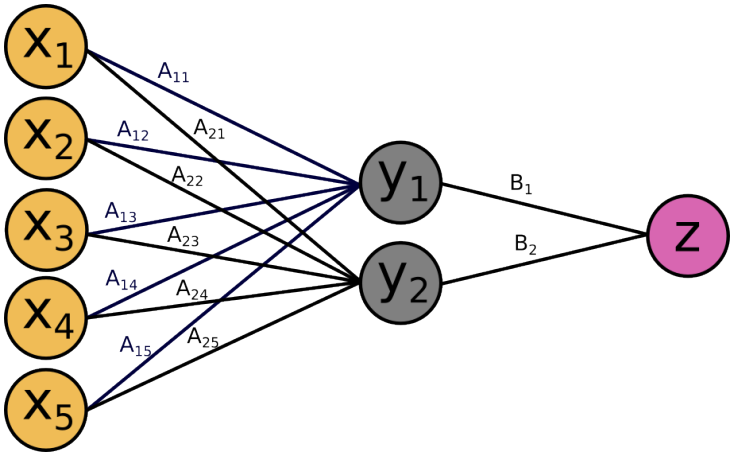

Figura 2.2 – Representação do modelo B. Nessa arquitetura do tipo Perceptron de Multicamadas, temos uma camada oculta entre os neurônios de entrada e os de saída. Cada neurônio, a partir da camada de entrada, passa informação adiante através de suas respectivas funções de ativação.

Perceba que o neurônio z representa a saída, os neurônios $\{y_1, y_2\}$ representam a camada oculta e os 5 neurônios $\{x_1, x_2, x_3, x_4, x_5\}$ representam a camada de entrada. Cada conexão de um dado neurônio com os seus imediatamente anteriores é feita como se fosse um Perceptron isolado. Quando temos uma rede em que todos os neurônio são interligados com os imediatamente anteriores dizemos que ela é **densamente conectada**.

Por simplicidade, vamos tomar a função de ativação de todos os neurônios como sendo a ReLU o que faz com que a saída da rede possa ser escrita da forma

$$z = O_2(y_1, y_2) = \sum_{i=1}^{2} B_i y_i + b. \tag{2.4}$$

Porém, cada neurônio y_i pode ser interpretado como um Perceptron dos neuronios $\{x_j\}$

$$y_i = \sum_{j=1}^{5} A_{ij} x_j + a, \tag{2.5}$$

onde A_{ij} é uma **matriz de pesos** que reuni em cada linha i os 5 pesos das conexões entre os neurônios de entrada $\{x_j\}$ com o neurônio y_i. Juntando as duas equações temos que o neurônio de saída da rede pode ser escrito da forma

$$z = \sum_{i=1}^{2} \sum_{j=1}^{5} B_i A_{ij} x_j + a \sum_{i=1}^{2} B_i + b. \tag{2.6}$$

Suponha que o Modelo-A tenha sido treinado para reconhecer um certo tipo específico de padrão ou característica. Nesta nova rede se tomarmos $A_{1j} = W_j$ e $a = c$ o primeiro neurônio oculto y_1 estará funcionando exatamente da mesma forma que o modelo-A, deixando livre para o neurônio y_2 o reconhecimento de uma característica adicional. A equação (2.4) simplesmente mostra que a saída da rede está sendo pesada por uma combinação dessas duas características capturadas em y_1 e y_2. Enquanto isso a equação (2.6) mostra que essa combinação de características só está sendo possível devido a uma maior quantidade de arranjos de pesos com os neurônios de entrada, o que destaca a importância de termos camadas ocultas na rede. Em geral, quanto maior o número de **hiperparâmetros**, ou seja, o número de camadas ocultas e de neurônios em cada camada, a rede ganha mais graus de liberdade se tornando mais capaz de correlacionar padrões complexos. No entanto, para resultados muito similares pode haver diferentes configurações de hiperparâmetros o que traz uma degenerecencia para o problema. Com uma maior quantidade de configurações possíveis para uma mesma tarefa, a rede tem mais dificuldade de achar sua configuração ideial.

2.2.1 Etapa de treinamento

O treinamento supervisionado das RNAs é o processo de otimização dos parâmetros livres da RNA a partir de um conjunto de treinamento que contém saídas desejadas da rede. No treino, esses dados de entrada passam por toda a rede produzindo saídas preditivas que são comparadas com as saídas alvo no gabarito por meio de uma **função custo** [22]. Um exemplo simples de função custo é o erro quadrático médio

$$E = \frac{1}{2} \sum_{i=1}^{n_s} ||y_i - \hat{y}_i||^2, \tag{2.7}$$

onde n_s é o número de neurônios de saída, y_i é a saída predita no neurônio i e $\hat{y}_i$ é a saída verdadeira. A partir de funções como essa, o modelo calcula o quão longe as predições estão de seus valores [36, 37] e atualiza os pesos para minimizar a função custo. Esse processo de atualização dos pesos ocorre das últimas camadas até as camadas iniciais e é conhecido como **retropropagação** [38, 39, 40]. Ao tentar relacionar um conjunto de entradas com suas respectivas saídas, o modelo procura durante o treino um conjunto ótimo de pesos que otimize esse processo. Essa otimização é feita de forma iterativa, onde em cada iteração o modelo percorre todos os conjuntos de dados e atualiza todos os pesos.

O algoritmo de retropropagação visa procurar um mínimo global ou suficientemente satisfatório da função custo para o desempenho do modelo. A técnica mais comum para realizar esse processo é a do gradiente descendente, onde os pesos W são atualizados na direção contrária do gradiente da função custo para cada camada

$$W_{k+1} = W_k - \eta \frac{\partial E}{\partial W}. \tag{2.8}$$

Aqui W_{k+1} são os pesos atualizados, k é o número da iteração e η é a taxa de aprendizagem [41]. Esta taxa determina o tamanho do passo que será dado na direção de otimização dos pesos. Se a taxa for muito alta os pesos sofrem mudanças muito drásticas a cada iteração, o que dificulta a convergência para seus valores ótimos. Por outro lado, se a taxa for muito pequena, o modelo irá precisar de muitas iterações para chegar a um mínimo da função custo [42]. Essa taxa é um parâmetro que deve ser ajustado para cada tipo de problema. Além disso, a técnica do gradiente descendente usa derivadas de primeira ordem e tipicamente não apresenta um bom desempenho quando o espaço de erros da função custo contém muitos pontos de sela, podendo ficar preso em mínimos locais. Para lidar com esses problemas existem técnicas de otimização que utilizam derivadas de segunda ordem para a atualização dos pesos [43].

É importante notar que para calcular a função custo levamos em consideração apenas o que acontece nos neurônios de saída. No entanto, as predições da rede $\hat{y}_i$ são função das entradas e determinadas por todos os parâmetros da rede. Nós conseguimos perceber isso na equação (2.6), onde o neurônio de saída z já contempla maiores combinações de pesos para a decisão de seu valor. Claramente este valor também depende da função de ativação adotada, mas o número de parâmetros que essa função vai adotar como argumento está intrinsecamente ligado à arquitetura da rede.

Outro fator que temos que levar em consideração na etapa de treinamento é o conjunto de treino. Primeiro é esperado que tenhamos uma quantidade considerável de amostras, da ordem de milhares ou mais no contexto das redes profundas discutidas nesta dissertação, para ser possível que o algoritmo relacione as entradas com as saídas. Segundo, é importante garantir uma diversidade entre essas amostras sejam representativas se comparadas ao dataset de teste. Para cada amostra i, vamos chamar de $\{x\}_i$ o conjunto de entradas fornecidas por ela e de $\{y\}_i$ o conjunto de saídas verdadeiras, que utilizaremos

para o treino de uma rede neural. Sendo assim, podemos definir uma tabela-verdade T que associa em cada linha os dados de entrada da amostra i com suas respectivas saídas dadas por um gabarito.

$$T_i := \{x\}_i \to \{y\}_i. \tag{2.9}$$

A partir deste ponto iremos considerar redes neurais aplicadas a **problemas de classificação**, visto que este é o foco do trabalho. Vamos supor que estejamos tentando classificar se uma imagem é de um cachorro ou não. Entraremos em mais detalhes de como podemos fazer isso na seção (2.3). Para essa abordagem qualitativa o que nos importa no momento é saber que dispomos de 600 amostras onde metade é um cachorro e a outra metade é de um outro animal qualquer. É importante que a divisão seja feita dessa forma para evitar qualquer preferência da rede em escolher uma das categorias. Se a grande maioria das amostras de treino forem de cachorros, a rede pode acabar adotando a estratégia de classificar todas as imagens como cachorro, visto que isso aumentaria seu número de acertos e certamente diminuiria a função custo.

No caso deste problema, para todas as 600 amostras da tabela-verdade a forma da saída sempre será binária e pode ser escrita da forma

$$T_i := \{x\}_i | \{0, 1\}$$

onde definimos que 1 é uma imagem de um cachorro e 0 é uma imagem de qualquer outro animal. No processo de treino, o algoritmo de otimização dos pesos passará pelos dados das amostras e quando ele cobrir todas elas completamos uma época, ou de iteração pelo ponto de vista matemático. A cada época o modelo tenta se aproximar cada vez mais do comportamento apresentado para ele na tabela-verdade. Completado todas as épocas temos o nosso modelo completamente treinado. Caso o treino seja satisfatório e a função custo tenha sido minimizada, o que se obtém até este ponto é um conjunto de pesos w_i que mapeia as entradas nas saídas com menor erro para qualquer amostra dentro do conjunto de treino. Naturalmente, se tivéssemos usado outras amostras, no final do treino teríamos um conjunto w_i' de pesos que não necessariamente são iguais a w_i. Pelo ponto de vista prático o que faz com que o conjunto w_i seja melhor que o conjunto w_i' é a sua capacidade de generalização do problema [22, 44]. Por exemplo, vamos supor que as 300 amostras usadas para ensinar à rede neural o que é um cachorro foram selecionadas com apenas cachorros da raça Rottweiler. Isso fará a rede ser muito boa em reconhecer Rottweilers, mas não tão boa em reconhecer cachorros de modo geral. Esse é um dos motivos de acontecer um dos principais problemas no treinamento das redes que é o **sobreajuste** [44]. Ele ocorre quando o modelo fica bem definido em reconhecer alguns padrões específicos, perdendo seu potencial de generalização. Por isso, outra medida importante é garantir uma maior diversidade dentro das amostras escolhidas para não ser introduzido algum viés na seleção.

2.2.2 Métricas de avaliação

Sabemos que um conjunto de treino com pouca diversidade pode causar um sobreajuste (overfitting) na rede, o que nos atenta para os cuidados necessários na hora de selecionar as amostras para compor esse conjunto. Para evitar esse problema contamos com algumas métricas responsáveis por ditar a eficiência do nosso modelo quando aplicado em conjuntos fora do nosso treino. Sendo assim, quando estamos na etapa de treino é geralmente separado mais dois **conjuntos avaliativos** chamados de **validação** e **teste** [22], nos quais essas métricas podem ser aplicadas para nos fornecer um melhor julgamento do desempenho da rede. Todas essas métricas são trabalhadas com mais detalhes em [22, 44].

- Validação: esse conjunto é separado para atuar durante o processo de treino e ele constitui uma tabela-verdade com entradas e gabaritos da maneira como foi definido em (2.9). A cada época, depois dos ajustes dos pesos, o modelo é aplicado nessas entradas e por meio de certas métricas uma avaliação da rede é fornecida. É importante notar que este conjunto não é usado no processo de atualização dos pesos, ele apenas serve para calcular um desempenho por época de treinamento.

- Teste: esse conjunto é separado para atuar depois do treinamento da rede e, da mesma forma que a validação, também constituí uma tabela-verdade. Em essência, se o conjunto de validação é usado para nos dar um julgamento contínuo durante o processo de treino o conjunto de teste é usado para um julgamento final após o treino.

Função Custo

A própria **função custo** pode ser usada como uma métrica avaliativa que é aplicada no conjunto de validação. Como discutido no início da seção (2.2.1), essa função determina uma distância entre as classificações previstas pelo modelo e a classificação real do objeto. Como veremos em (2.3), é possivel ajustar a arquitetura da rede de forma que suas saídas representem um probabilidade de pertencimento a uma certa categoria. Como estamos considerando problemas de classificação e, mais especificamente, classificações binárias, uma ótima função custo é a **entropia binária cruzada** [45, 44].

$$E = \frac{1}{N}\sum_{i=1}^{N}\{-[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]\} \tag{2.10}$$

Essa função custo em específico calcula a diferença entre a distribuição de probabilidade predita e a distribuição de probabilidade real. O logaritmo aqui é importante para penalizar mais as grandes diferenças do que as pequenas. Geralmente esta métrica é aplicada no conjunto de validação e, juntamente com a própria função custo do processo de treino, é possível construir um gráfico mostrando como essa função evolui por época

de treinamento, tanto para o conjunto de treino, quanto para o de validação. Sobreajustes normalmente são identificados quando a função custo do treino diminui, mas a função custo da validação aumenta, indicando que o modelo não está ganhando poder de generalização.

Acurácia

Outra métrica utilizada em problemas de classificação é a **Acurácia**. No final de cada época, o modelo é aplicado no conjunto de validação fazendo uma previsão para cada amostra. Essa métrica é dada por uma razão entre o número de previsões corretas e o número total de amostras dentro do conjunto de validação. Assim, com esse cálculo simples, nós temos uma porcentagem de acertos nas previsões da rede para cada época que pode ser disposta em um gráfico a fim de mostrar o desempenho do modelo conforme o treino ocorre. É importante destacar que essa métrica pode ser aplicada no conjunto de teste também, gerando uma avaliação final da rede.

As próximas métricas que abordaremos tentam olhar para o desempenho separado de uma das classificações, sendo assim, ao utilizá-las, implicitamente estaremos escolhendo uma classe de interesse. Se estivermos lidando com uma classificação binária, é possível dividir as previsões feitas pelo modelo em 4 tipos.

- Positivo Verdadeiro (PV): quando a previsão do modelo aponta que o objeto pertence à nossa classe de interesse e acerta em sua previsão.
- Positivo Falso (PF): quando a previsão do modelo aponta que o objeto pertence à nossa classe de interesse, mas erra em sua previsão.
- Negativo Verdadeiro (NV): quando a previsão do modelo aponta que o objeto não pertence à nossa classe de interesse e acerta em sua previsão.
- Negativo Falso (NF): quando a previsão do modelo aponta que o objeto não pertence à nossa classe de interesse, mas erra em sua previsão.

Geralmente o que mais nos interessa não é julgar uma previsão isolada, mas saber o numero total de PV's, PF's, NV's e NF's. Sendo assim, vamos tomar cada uma dessas abreviações como o número total de previsões que caem nelas. Também devemos lembrar que dependendo da função de ativação que selecionamos para os neurônios de saída o modelo nos retorna uma probabilidade de pertencimento a uma determinada categoria. A classificação em si é feita em respeito a um **Threshold** definido anteriormente. Basicamente um objeto pertence a uma certa classe se a probabilidade de pertencimento conferida a ele pelo modelo for maior que o threshold para esta classe. Sendo assim, a quantidade de PF, PV e os demais tipos de previsão dependem do threshold fixado.

No capítulo (5) mostramos o treshold escolhido para a classificação do nosso modelo e apontamos maneiras de selecioná-lo baseando-se nessas métricas.

Em termos dessas quantidades a Acurácia pode ser dada pela equação:

$$Acurácia = \frac{PV + NV}{PV + PF + NV + NF}. \tag{2.11}$$

Perceba que ela não prioriza nenhuma classificação, seu objetivo é simplesmente contabilizar a porcentagem de previsões corretas. Já se quisermos definir uma classe de interesse podemos contar com outras duas métricas.

Precisão

Dentre todas as previsões que o modelo fez apontando pertencer à nossa classe de interesse, esta métrica contabiliza quantas delas são Verdadeiros Positivos

$$Precisão = \frac{PV}{PV + PF}. \tag{2.12}$$

Perceba que a Precisão aumenta se diminuirmos o número de Positivos Falsos da rede, mas ela não se importa com os Negativos Falsos. Vamos supor um modelo que foi treinado para julgar ações no mercado financeiro, reconhecendo bons investimentos. O que precisamos é que esse modelo seja preciso em sua classificação e que possamos confiar nos investimentos que ele julgar rentável. Nessa tarefa os Positivos Falsos são muito prejudiciais, pois se o modelo julgar como rentável um investimento ruim isso pode causar uma enorme perda de dinheiro. No entanto, se o modelo deixar passar um investimento que era rentável (um Negativo Falso) apenas teríamos perdido uma chance de lucrar, mas ainda manteríamos nosso dinheiro. Para esse exemplo, manter a precisão alta significa se arriscar menos a perder dinheiro.

Recall (Completeza)

Dentre todos os objetos que de fato pertencem à nossa classe de interesse, esta métrica contabiliza quantos deles foram previstos pelo modelo corretamente.

$$Recall = \frac{PV}{PV + NF}. \tag{2.13}$$

Analogamente, perceba que o Recall aumenta se diminuirmos o número de Negativos Falsos da rede, mas ele não se importa com os Positivos Falsos. Essa métrica é interessante em uma situação que não possamos deixar passar Negativos Falsos. Se criarmos um modelo que identifica nos hospitais pessoas que estejam doentes, o que precisamos é que todos os pacientes doentes sejam pegos pelo modelo, mesmo que ele eventualmente acabe classificando algumas pessoas saudáveis como doentes (PF). Do ponto de vista da saúde do indivíduo o maior problema se dá se uma pessoa doente sair do hospital do que se uma pessoa saudável permanecer nele um pouco mais.

Matriz de confusão

Uma outra forma de dispor essas previsões é usando uma matriz de confusão. Ela é usada para comparar as previsões com as classificações verdadeiras dos objetos, de forma que tanto os tipos de previsões quanto essas métricas acima podem ser visualizadas com facilidade (Figura 2.3 e 2.4) se estivermos em um problema de classificação binária. Esse tipo de métrica geralmente é feita depois do processo de treino, utilizando o conjunto de teste separado.

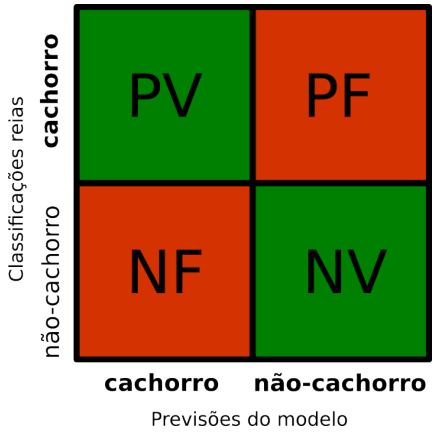

Figura 2.3 – Representação de uma matriz de confusão para um problema de classificação binária. Nela podemos ver bem onde se localizam os tipos de previsão. Para um problema com maior número de classificações essa matriz aumenta, mas a diagonal principal sempre representará o numero de previsões corretas.

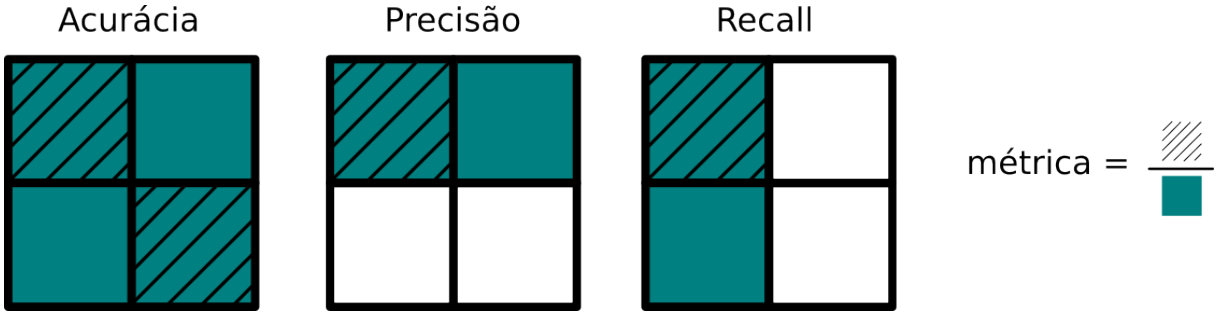

Figura 2.4 – Essa figura mostra como as Métricas estabelecidas acima podem ser calculadas a partir da matriz de confusão para um problema de classificação binária.

O algoritmo sempre atualizará o pesos de forma a tentar diminuir a função custo para as amostras dentro do conjunto de treino, porém essas métricas são avaliadas em conjuntos fora do treino. Sendo assim, se o comportamento dessas métricas for diferente no conjunto de validação isso é indicativo de problemas no processo de treino.

2.3 Redes Neurais Convolucionais (CNN)

As Redes Neurais Convolucionas (RNC) apresentam grande capacidade em processar e reconhecer padrões em imagens, o que as torna adequadas para problemas de classificação

de imagens [46, 22]. Elas em essência funcionam da mesma forma que as redes neurais convencionais, com o diferencial de que as imagens são processadas por meio de camadas convolucionais que possuem filtros com pesos ajustáveis que realizam um processo de convolução matricial no volume de entrada da camada. Antes de entrar em detalhes com esses processos, vamos primeiro discutir como o computador processa uma imagem.

2.3.1 Processamento de imagens

As cores em uma Imagem digital são definidas a separado por uma combinação linear de 3 filtros que representam cores primárias: o vermelho(R), verde(G) e o azul(B).

No padrão de imagens digitais tais como jpeg, png é convencionado uma escala de 8 bits, isto é 2^8 possíveis valores para cada uma das intensidades das cores primárias de 0 até 255, de forma que o vetor $(0, 0, 0)$ representa a cor preta e o vetor $(255, 255, 255)$ representa a cor branca. A fim de criar uma linguagem que nos permita tratar as imagens de forma numérica também definimos o píxel, que serve como a unidade mais básica que carregará essas três cores primárias dentro de si. Assim, matematicamente, podemos definir uma imagem no computador como uma matriz de píxeis $(N \times M)$. Ou seja, cada entrada dessa matriz carrega um vetor com 3 valores para definir a cor na formatação RGB para aquela entrada. A figura (2.5) exemplifica uma matriz de píxeis.

(0, 0, 128)	(175, 0, 141)	(175, 0, 141)	(175, 0, 141)	(175, 0, 141)	(175, 0, 141)	(0, 0, 128)
(206, 138, 100)	(142, 0, 0)	(142, 0, 0)	(142, 0, 0)	(142, 0, 0)	(142, 0, 0)	(0, 181, 170)
(206, 138, 100)	(142, 0, 0)	(0, 180, 0)	(0, 180, 0)	(0, 180, 0)	(142, 0, 0)	(0, 181, 170)
(206, 138, 100)	(142, 0, 0)	(0, 180, 0)	(0, 0, 128)	(0, 180, 0)	(142, 0, 0)	(0, 181, 170)
(206, 138, 100)	(142, 0, 0)	(0, 180, 0)	(0, 180, 0)	(0, 180, 0)	(142, 0, 0)	(0, 181, 170)
(206, 138, 100)	(142, 0, 0)	(142, 0, 0)	(142, 0, 0)	(142, 0, 0)	(142, 0, 0)	(0, 181, 170)
(0, 0, 128)	(176, 216, 0)	(176, 216, 0)	(176, 216, 0)	(176, 216, 0)	(176, 216, 0)	(0, 0, 128)

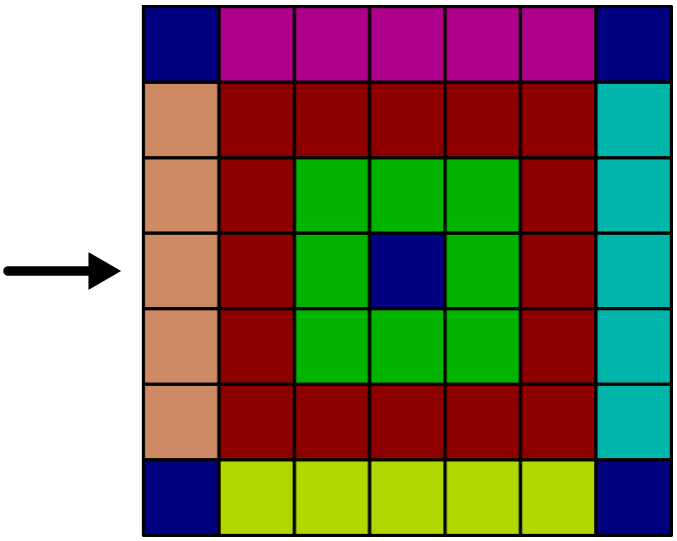

Figura 2.5 – Imagem representativa da matriz de píxeis. No lado esquerdo mostramos a matriz de píxeis da forma como o computador a enxerga. Já no lado direito temos a imagem colorida criada a partir da matriz de píxeis

2.3.2 Convolução Matricial e Pooling

Esses processos se dão quando temos uma imagem que chamaremos de matriz primária e tratamos ela mediante um filtro que chamamos de **kernel**. Em geral, o kernel é uma

matriz, sendo utilizado para realçar ou detectar certas características na imagem. Suponha um filtro simples de tamanho (3×3) onde chamaremos sua entrada central na posição $(2, 2)$ de **pivô**. A **convolução** ocorre quando, dada uma entrada base na matriz primária, sobrepomos o filtro de forma que seu pivô fique em cima desta **entrada base** [22, 44]. Em seguida multiplicamos todas as entradas sobrepostas para no final somar todas as multiplicações. O resultado desta operação é o valor da entrada base de mesma posição na matriz resultante (mapa de características).

É importante notar que da forma como essa operação foi definida é possível encontrarmos alguns problemas se tomarmos a borda da matriz primária como entradas base. Isso ocorre porque, após a sobreposição do kernel, algumas entradas do filtro não terão correspondentes com a matriz primária. Uma possível solução para o problema é cercar a matriz primária com zeros o suficiente para que o filtro sempre encontre correspondentes. Esses zeros não causaram muitas modificações visto que os passos da convolução envolvem multiplicações seguidas de somas. Esta é uma solução interessante, pois preserva o tamanho original da matriz primária apenas causando modificações. Um outra solução seria não permitir que a borda seja usada como entrada base, o que acarretará em uma diminuição no tamanho da matriz resultante. Essa diminuição não é um problema se o objetivo da convolução for o de compactar a informação e se os píxeis do centro forem mais importantes que os da borda.

Em geral, utilizamos convoluções quando queremos destacar importantes características da imagem, como é mostrado na figura (2.6). Existe um determinado tipo de filtro responsável por detectar determinada característica. No processo de convolução matricial, apenas passamos esses filtros em regiões da imagem para que ele destaque de cada região suas propriedades. No contexto de uma CNN cada elemento do kernel que representa o filtro são compostos por pesos e, como no caso das redes neurais sequenciais, esses pesos são definidos e ajustados pelo processo de treinamento. O tamanho dessas regiões depende do tamanho do kernel e não necessariamnte precisamos mover seu pivô de píxel a píxel. Esse número de passos que o pivô tem que dar para encontrar sua próxima entrada base é o que chamamos de **stride**. Quanto maior o stride maior a distância entre os pivôs dessas regiões.

$$\text{Kernel para reconhecer contornos: } \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

De fato, o processo de convolução pode ser feito de maneira a tanto diminuir o tamanho da imagem original quanto a diminuir sua dimensão, visto que strides maiores do que 1 não estariam utilizando todos os píxeis da imagem como entrada base. No entanto, quando falamos de compactar e resumir informação, podemos utilizar a técnica de **Pooling**. Essa técnica aplicada em cada região escolhe um valor representativo. Esta seleção pode ser a

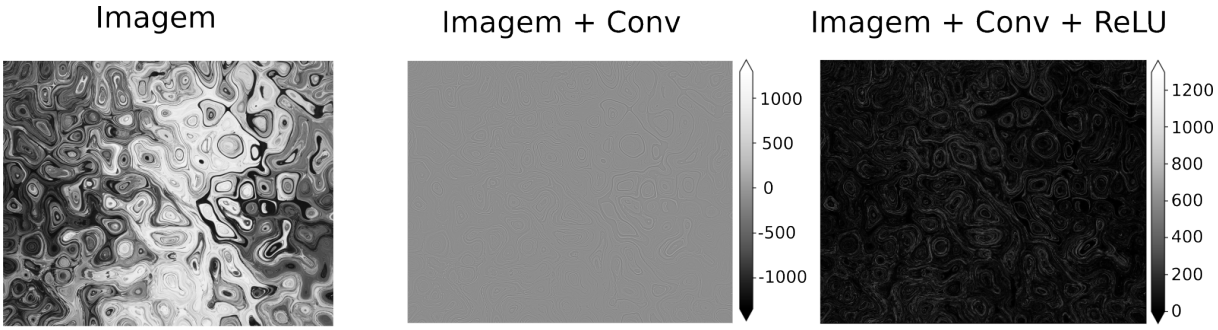

Figura 2.6 – Essa figura apresenta as mudanças na imagem após passá-las pelo filtro para reconhecer contornos seguido da função de ativação ReLU. Na imagem do meio após a convolução os píxeis que constituem bordas são destacados ficando com valores positivos, enquanto as regiões entre bordas ficam com valores zerados ou negativos. Já na terceira imagem a ReLU é aplicada, fazendo com que todos os valores negativos sejam jogados para zero. Dessa forma zeramos os píxeis que não apresentam as propriedades que o filtro está procurando, ao mesmo tempo que não permitimos valores negativos na matriz de píxeis.

entrada de maior valor com o Max-Pooling, a média entre as entradas com o Mean-Pooling e assim por diante.

Suponha que o kernel utilizado para um pooling foi de tamanho (5×5) e que não haja intersecção entre as regiões selecionadas. Dessa forma, para cada região que o kernel atuar 25 entradas serão substituídas por 1 valor representativo. Mais especificamente, se em um quadrado de 5 píxeis na horizontal e 5 na vertical todos se reduzirem a 1 píxel, o tamanho da nossa imagem será dividida por 5 nas duas direções, o que resulta em uma redução de 96% na área. Esse processo é interessante, pois ele consegue preservar, em certa medida, as características globais da imagem. De fato, por mais que menor, boa parte de sua assinatura se mantém como podemos ver na figura (2.7). Se dermos um zoom e colocarmos todas com o mesmo tamanho, fica claro que a imagem perdeu resolução, mas não sua forma. Claramente ao tentar resumir parte da informação é perdida, então esta é uma ferramente que deve ser usada com certa sabedoria.

Em geral essas ferramentas são utilizadas em conjunto nas redes neurais por propiciarem um melhor desempenho na etapa de treinamento. Assim surgem as Redes Neurais Convolucionais (CNN: Convolutional Neural Networks).

2.3.3 Arquitetura de uma CNN e suas vantagens

É importante destacar que não precisamos necessariamente de uma arquitetura que incorpore convoluções para utilizarmos redes neurais em classificações de imagens. Vamos supor uma imagem de $(N \times M)$ píxeis, onde apenas um dos canais do RGB está sendo utilizado. Em outras palavras, cada entrada da matriz de píxeis é dada por apenas um número, fazendo com que a imagem possa ser representada em tons de cinza. Para ali-

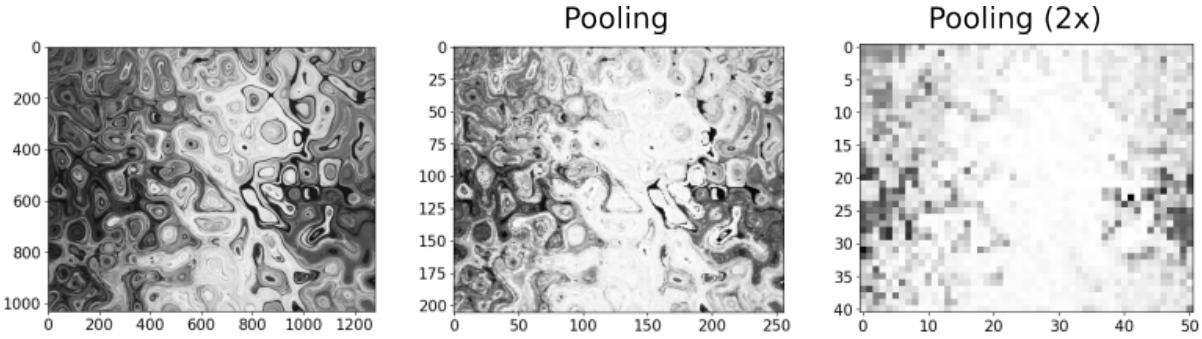

Figura 2.7 – Esta figura apresenta o que acontece com a imagem conforme aplicamos o pooling nela. Os eixos indicam a posição de cada pixel na imagem, o que mostra que conforme aplicamos o pooling a imagem diminui de tamanho. Nessa figura, as imagens que passaram pelo pooling foram ampliadas para ficar do mesmo tamanho da imagem original. Assim podemos perceber que a imagem perde detalhes, mas consegue manter suas características globais.

mentar a RNA com essa imagem de forma que ela possa servir como treino para uma certa classificação basta criarmos uma arquitetura que tenha $N.M$ neurônios de entrada, onde cada um receba um dos píxeis. Assim, depois de algumas camadas ocultas, fazemos com que todos os neurônios convertam para dois neurônios de saída, onde um é responsável por dizer se a imagem pertence a uma primeira classe e o outro responsável por dizer se ela pertence a uma segunda classe. Vale destacar que este é um problema de classificação binária, e por isso estamos utilizando apenas dois neurônios de saída. Outro fator importante nessa arquitetura está em suas funções de ativação. Nas camadas ocultas tipicamente é usado a função de ativação ReLU, seu comportamento linear costuma facilitar o processo de otimização da rede [44]. Porém, especialmente na camada de saída, a melhor escolha é utilizar uma função de ativação do tipo da Sigmoid. Isso nos permite obter, em cada neurônio de saída, a probabilidade da imagem pertencer à classe correspondente a este neurônio, o que nos dá uma classificação mais contínua.

Apesar dessa abordagem ser funcional em redes mais simples, como no caso das redes primeiramente propostas para solucionar a classificação do dataset MNIST [47, 48], ela apresenta algumas desvantagens:

- D1: Nesse exemplo utilizamos imagens em tons de cinza, mas se tivéssemos utilizado uma imagem colorida precisaríamos triplicar a quantidade de neurônios de entrada, pois cada um dos píxeis carregariam 3 números. Além disso, se mantivéssemos as mesmas camadas ocultas, teríamos também o triplo de ligações entre os neurônios de entrada e os neurônios da primeira camada oculta, o que aumenta a quantidade de pesos necessários para mediar todas essas ligações. Com mais parâmetros livres, mais o algoritmo demora para convergir para o conjunto ideal deles aumentando também a chance de causar um sobreajuste (2.2.1). Esse problema se agrava a medida que aumentamos o tamanho da imagem, se tornando computacionalmente

exigente. Nas CNN, por outro lado, o número de pesos necessários é menor a medida que os filtros são aplicados localmente.

- D2: Outro problema está no fato de a rede utilizar todos os píxeis da imagem para a coleta de informações, quando o questionamento principal é se isso realmente é necessário. Suponha uma situação em que tentamos classificar se a imagem é de uma pessoa ou não. Claramente existem vários contextos diferentes em que uma pessoa possa estar inserida dentro da imagem. Podemos ter uma pessoa na rua, dentro de uma casa, na floresta, na praia, enfim, o fundo da imagem não é importante. No entanto, a rede ainda está enxergando os píxeis do fundo para a classificação.

As redes neurais convolucionais são adequadas para evitar esses problemas citados acima. Sua arquitetura geralmente começa com camadas de convolução seguidas de pooling de forma a dar um primeiro tratamento às imagens e resumir o seu conteúdo informacional. Essas duas ferramentas funcionam muito bem juntas, pois uma prepara o terreno para que a outra atue [44]. Primeiro as convoluções destacam as principais características da imagem. Como essas ferramentas também representam camadas na arquitetura, também existe uma função de ativação que dita o comportamento de suas respostas suavizando a desvantagem D2, podemos verificar isso no exemplo apresentado na figura (2.6).

A partir daqui, a camada de pooling desempenha realçando as características selecionadas pela convolução. Pense novamente na figura (2.6), o kernel que utilizamos destacou os contornos da imagem e deixou o espaço entre eles zerado depois da ReLU. Quando aplicarmos o max-pooling e o pivô passar por um píxel que está destacado, ele vai gerar outro píxel destacado na matriz resultante, pois se ele está destacado, então no mínimo o píxel resultante tem o mesmo valor dele. Porém, se o pivô encontrar um píxel não destacado em uma região do kernel que tem píxeis destacados, o píxel representativo na matriz resultante também fica destacado. Ou seja, todos os píxeis próximos àqueles que eram destacados se tornam destacados também durante o pooling. Além disso, com as camadas de Pooling, essas características são simplificadas, o que compacta a informação em objetos menores e diminua o custo computacional quando elas forem passadas para os neurônios ocultos, mitigando a desvantagem D1. A figura (2.8) mostra o resultado da aplicação de uma convolução seguida de um Pooling.

A questão crítica por trás do funcionamento das camadas convolucionais está no ajuste dos pesos que definem cada elemento dos filtros durante o treinamento, possibilitando que as camadas convolucionais extraiam informaões relevantes para o problema. Uma vantagem geral das CNNs, se comparadas às RNAs, está no seu poder em operar de forma menos pontual e mais local nas imagens. Como dito antes, podemos alimentar os neurônios de entrada com cada um dos píxeis que formam a imagem. Poŕém, selecionando regiões ao redor dos píxeis as CNNs têm maior capacidade de reconhecer mudanças próximas à

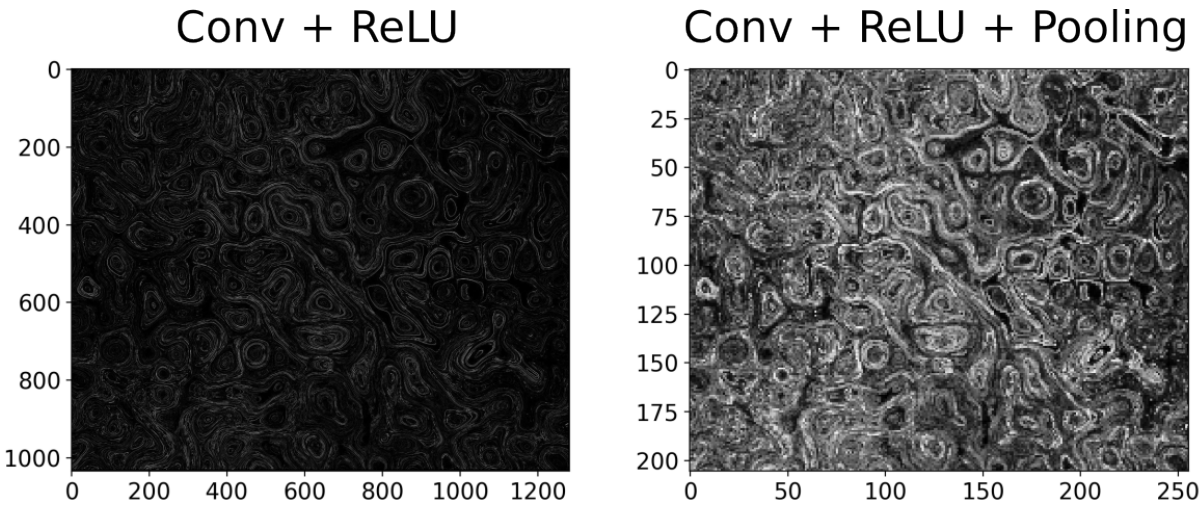

Figura 2.8 – Aqui apresentamos a atuação da convolução junto do pooling. Com o kernel da convolução os contornos foram destacados, deixando para o pooling compactar essa informação ao mesmo tem que realça esses contornos.

escala dos píxeis (locais). As vezes esses padrões em que a rede está procurando estão escondidos na imagem, seja pela câmera ser de baixa resolução ou algum efeito causado por um ruído. Devido a esta localidade da CNN, mesmo que atenuado por esses fatores acima, a rede ainda é capaz de reconhecer essa informação se treinada de forma adequada. Além disso, translações do conteúdo da imagem não afetam o julgamento da rede, visto que esses operadores não atuam em um pixel específico, mas procuram características em todo o seu espaço [44].

Capítulo 3

Classificação de Galáxias

3.1 Introdução

As galáxias podem ser definidas como um conjunto de estrelas, gás, poeira, e matéria escura gravitacionalmente ligados em uma estrutura coesa. São como grandes tijolos luminosos que constituem uma enorme teia cósmica, sendo tópico de grande estudo na astrofísica moderna. Inicialmente, esses objetos astronômicos foram classificados como nebulosas, pelo fato de não existirem métodos robustos para determinar a distância deles, e assim determinar se pertenciam a Via Láctea. Importante notar que, apesar do equívoco na classificação, varias dessas galáxias já haviam sido catalogadas. Podemos citar como exemplo o catálogo de Messier fundamentado pelo astrônomo francês Charles Messier por volta de 1781 [49]. Esse catálogo consta com 110 objetos onde o M31 é a então chamada Nebulosa de Andrômeda. Posteriormente, em 1888, o astrônomo John Dreyer compila um catálogo ainda maior com 7840 objetos [50] onde Andrômeda é rotulada de NGC 224, porém ela persistia sendo tratada como uma nebulosa.

Foi apenas em 1929 que Edwin Hubble foi capaz de demostrar que Andrômeda não fazia parte da nossa galáxia [51], utilizando o trabalho da henrietta Levitt sobre Cefeidas como vela padrão [52]. As Cefeidas são estrelas que ocupam regiões de instabilidade no diagrama de Hertzsprung-Russell [53], e representam períodos de transição entre estágios de evolução. Estrelas na faixa de instabilidade pulsam [54, 55], em outras palavras, o brilho varia em um espaço de tempo bem determinado. Esse período de variação é proporcional a sua Luminosidade que, por sua vez, é proporcional ao inverso do quadrado da distância [52, 56]. Assim, Hubble descobriu uma Cefeida na galáxia de Andrômeda e usando-a como vela padrão, foi capaz de determinar que a distância dela para o nosso sistema solar era de mais de 1 milhão de anos-luz, o que ultrapassava o tamanho da nossa galáxia. Com esse método, muitos objetos anteriormente tidos como nebulosas pertencente à Via Láctea se configuraram como galáxias. Hubble ainda em 1926 já havia publicado uma

classificação para uma lista de nebulosas extragaláticas [57], separando esses objetos em 3 tipos principais: espirais, elípticas e irregulares. Vale ressaltar que essa classificação não leva em consideração tamanho ou composição da galáxia, sendo simplesmente morfológica, como mostrado na figura (3.1), conhecida como **Diagrama de Hubble**. As principais características dessas três classes podem ser resumidas como:

- Espirais: as galáxias espirais são caracterizadas pela presença de braços espirais, podendo ser separada entre espirais e espirais barradas. Em geral, contêm muito gás o que propicia regiões de formação estelar. Elas podem ser dividas em duas regiões, uma mais externa onde se encontram os braços espirais, e outra mais interna, onde se encontra o bojo. Sua classificação simbólica começa com a letra S, seguida de uma letra a,b ou c que dita o quanto os braços espirais se sobressaem em relação ao bojo.

- Elípticas: as galáxias elípticas geralmente têm um perfil luminoso mais uniforme e elíptico, como o nome já sugere. Em geral elas têm carência de gás o que não proporciona alta formação estelar. Sua classificação simbólica começa com a letra E seguida de um número que vai de 0 a 7, onde esse número está relacionado com a excetricidade da elipse.

- Irregulares: As galáxias irregulares são aquelas que não tem uma forma definida, apresentando uma estrutura morfologica desordenada, caótica. Geralmente também apresentam uma quantidade considerável de gás.

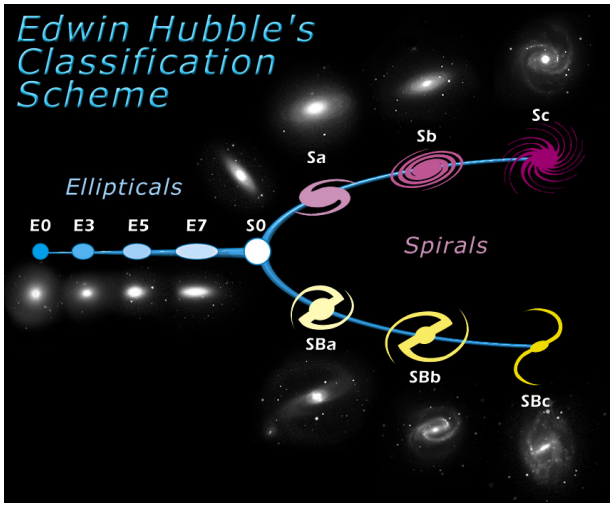

Figura 3.1 – Nesta figura mostramos uma representação do Diagrama de Hubble, com exemplos de galáxias que se encaixam em cada classificação. Créditos: NASA & ESA

No centro do **Diagrama de Hubble**, onde os dois ramos das galáxias espiral e o ramo das elípticas se juntam, se encontram as galáxias $S0$, também conhecidas como lenticulares. Estas galáxias têm um disco como as galáxias espirais, porém sem braços. Além

disso, elas também apresentam uma população estelar velha, como as galáxias elípticas. Em geral, sua identificação pode ser desafiadora visto que contemplam características dessas duas classes. Por exemplo, galáxias espirais edge-on, cujo disco está inclinado de 90 graus em relação à linha de visão, podem se confundir com galáxias lenticulares. Esses objetos intermediário entre as classificações, de fato, podem ser considerados como fases de transição na evolução das morfologias das galáxias. Vários trabalhos apontam a possibilidade das galáxias lenticulares terem sido galáxias espirais no passado que por algum mecanismo cessaram sua formação estelar e, como consequência, perderam sua estrutura espiral [58, 59, 60]. Na literatura comumente encontramos uma segunda forma de separar as galáxias, chamando as Elípticas e Lenticulares de Galáxias do Tipo Anterior (Early Type Galaxy - ETG) e as Espirais e Irregulares de Galáxias do Tipo Tardio (Late type Galaxy - LTG).

3.2 Morfologia das Galáxias e Fotometria

A morfologia das galáxias é um tópico ainda amplamente estudado e vai além da classificação em diferentes tipos de galáxias. Isso acontece, pois a morfologia pode trazer informações sobre a formação e evolução das galáxias, conexões com o ambiente circundante, entre outras [11]. Além disso, fatores como cor, taxa de formação de estrelas e massa estelar apresentam correlações com a morfologia o que estarei apresentando no decorrer dessa seção.

A cor de uma galáxia, definida como a diferença de sua magnitude em duas bandas fotométricas, veja o capitulo 3.2.1, é relacionada com sua taxa de formação de estrelas [61]. Especificamente, as regiões da galáxia com alta taxa de formação estelar são geralmente mais azuis[1]. De fato, as estrelas mais jovens nessa região emitem radiação em frequências da cor azul, com mais intensidade, dominando a luz avermelhada que vem das mais velhas. Com o tempo as estrelas vão emitindo em faixas mais próximas do vermelho [63]. Até este ponto a cor está bem relacionada com a idade das estrelas, a temperatura e no que se refere às galáxias, ligada à taxa de formação estelar. Já em 1958 Holmberg [64] descobriu que a maioria das galáxias elípticas são mais avermelhadas e apresentam baixas taxas de formação estelar, enquanto que as espirais, em sua maioria, são mais azuladas com uma alta formação de estrelas. Por mais que existam espirais vermelhas e elípticas azuis [65], vale destacar que o resultado de Holmberg apresenta uma tendência estatística, comprovando uma correlação, mesmo que com alta dispersão, entre a cor e a morfologia das galáxias. Além disso, Holmberg também mostrou outra tendência estatística no que se

[1] A existência de poeira nas galáxias também afeta a atividade de formação de estrelas, aumentando a taxa de formação molecular em duas ordens de grandeza em comparação com o caso sem poeira e resfriando eficientemente o meio interestelar. Por outro lado, a luz estelar, em particular em comprimentos de onda mais curtos, é absorvida pela poeira e reemitida como uma emissão térmica de infravermelho distante da poeira [62].

refere à massa estelar das galáxias. Ele encontrou que as elípticas têm tendência a serem mais massivas que as espirais, o que novamente nos chama a atenção para a possibilidade de algum mecanismo físico responsável por essa tendência.

Ainda no que tange essas características estatísticas, foi encontrado que galáxias elípticas habitam regiões de maior densidade do universo local (para $z < 1$), enquanto que as espirais costumam habitar regiões de mais baixa densidade [66, 67, 68]. Estudos envolvendo o uso de simulações numéricas, mostram que a fusão de duas galáxias tem grande potencial de modificar sua estrutura [69, 70, 71], dependendo da razão de massa dos objetos envolvidos. É aceito na literatura que grandes fusões entre galáxias espirais de massa equivalente têm alta probabilidade de produzir uma galáxia elíptica como resultante, o que age em boa concordância com a tendência de se encontrar mais galáxias elípticas em ambientes mais densos, visto que nessas regiões acontecem mais fusões de galáxias. No entanto, os aglomerados de galáxias apresentam um meio intergaláctico quente que pode causar a remoção do gás de uma galáxia espiral quando essa cai no potencial do aglomerado. A remoção do gás causa a cessação da formação de novas estrelas e a perda dos braços espirais transformando-as em galáxias lenticulares [72, 73, 74]. Essas ocorrências colocam em questão o quão importante é o ambiente onde se situam as galáxias para a sua morfologia e evolução.

Portanto, a morfologia das galáxias é um importante tópico da astronomia, visto que pode trazer informação sobre a formação e evolução desses objetos, além de estar bem correlacionada com outros parâmetros como massa estelar, formação de estrelas, conteúdo gasoso, entre outros. Assim, classificá-las adequadamente também torna-se um trabalho de grande importância. O **Diagrama de Hubble** é um esquema de classificação desenvolvido tomando como base características visuais das galáxias do universo local [11]. De fato, as galáxias em alto redshift mostram morfologias mais irregulares [75]. Além disso, a capacidade de recuperar morfologias de galáxias é limitada pela profundidade e resolução dos dados usados na classificação. Ela fica comprometida a medida em que se estuda galáxias de menor luminosidade ou tamanho, comprometendo particularmente a classificação de galáxias em médio e alto redshift ($z > 0.5$).

De fato, vários estudos discutem as diferenças na morfologia intrínseca e detectada quando aumentamos artificialmente redshift das galáxias. Vários autores simularam como as galáxias do universo local apareceriam em um redshift mais alto, considerando as características de levantamentos diferentes [76, 77], para calcular o redshift máximo no qual as estimativas morfométricas estão corretas. Por exemplo, no caso de imagens capturadas pelo Telescópio Espacial Hubble (HST), os índices de assimetria e concentração são reprodutíveis até $z = 1.5$, enquanto para dados do tipo SDSS, a classificação morfológica é válida apenas até $z = 0.2$ [78].

Além disso, por conta do redshift galáxias mais distantes são mais avermelhadas do que a redshift zero, o que impõe uma necessidade de filtros mais na faixa do infravermelho para

a obervação das galáxias. A resolução da câmera do instrumento assim como a profundeza das observações no telescópio é importante na obtenção de uma imagem com detalhe o suficiente para uma classificação robusta. O advento do telescópio espacial JWST está abrindo uma nova era para as morfologias das galáxias, estendendo nossa capacidade de estudar suas propriedades até $z \leq 8$ [79].

3.2.1 Filtros e índice de cor

Não temos como falar de astronomia observacional sem antes falar sobre fotometria. Objetos astronômicos, como estrelas, galáxias, quasares etc. emitem radiação eletromagnética em vários comprimentos de onda, dependendo do objeto em estudo. Vale destacar que o(s) pico(s) dessa radiação não precisa estar necessariamente na região do visível, o que mostra a importância de termos telescópios capazes de cobrir uma extensa região desse espectro. Para estudar a radiação emitida, o espectro eletromagnético é divididos em diferentes regiões, através do uso de um conjunto de bandas passantes (ou filtros) bem definidas, com uma sensibilidade conhecida à radiação incidente, onde cada banda abrange um determinado comprimento de onda central, com uma certa largura. Os telescópios trabalham com instrumentos, por exemplo câmeras ópticas, que possuem detectores chamados CCDs (Charge coupled devices [80]) especializados em capturar energia eletromagnética e traduzi-la para grandezas que possam ser interpretadas pelo computador utilizando o efeito fotoéletrico.

Esses detectores são câmaras fotossensíveis sobre as quais a luz que o telescópio coleta é focada. Os fotons atingem uma superfície semi-condutora dividida em pequenos elementos de área contendo aproximados 15×15 μm^2. Sendo assim, essas câmaras constituem uma matriz de elementos sensoriais, onde cada elemento representa um píxel. O processo de detecção nos telescópios depende de muitas variáveis, entre elas tempo de exposição, tratamento de ruído e outros fatores [4]. Além disso, os telescópios são equipados com diferentes filtros, que permitem obter imagens em diferentes faixas de comprimentos de onda. Juntando estas imagens é possível criar uma imagem a cores, que mostra, em primeira ordem, qual população estelar habita diferentes partes da galáxia. Por exemplo, a região central, onde domina a luz do bojo, é caracterizada por uma cor amarela/vermelha, típica de uma população estelar antiga. Por outro lado, os braços espirais, onde a formação estelar está ocorrendo ativamente, são azuis. Para criar uma imagem colorida, muitas vezes é usado o formalismo RGB, mesmo se esses 3 canais não forem necessariamente feitos com todos estes filtros. De fato, muitas imagens que temos do universo são coloridas artificialmente, o que significa escolher 3 bandas, ou mais, para compor esses 3 canais. A figura (3.2) mostra uma imagem da galáxia de Andrômeda, onde eu separo os 3 canais dela.

Em estudos fotométricos, as imagens produzidas são utilizadas para extrair informações quantitativas dos objetos em estudo [81, 82]. O fluxo de energia integrado em todos os

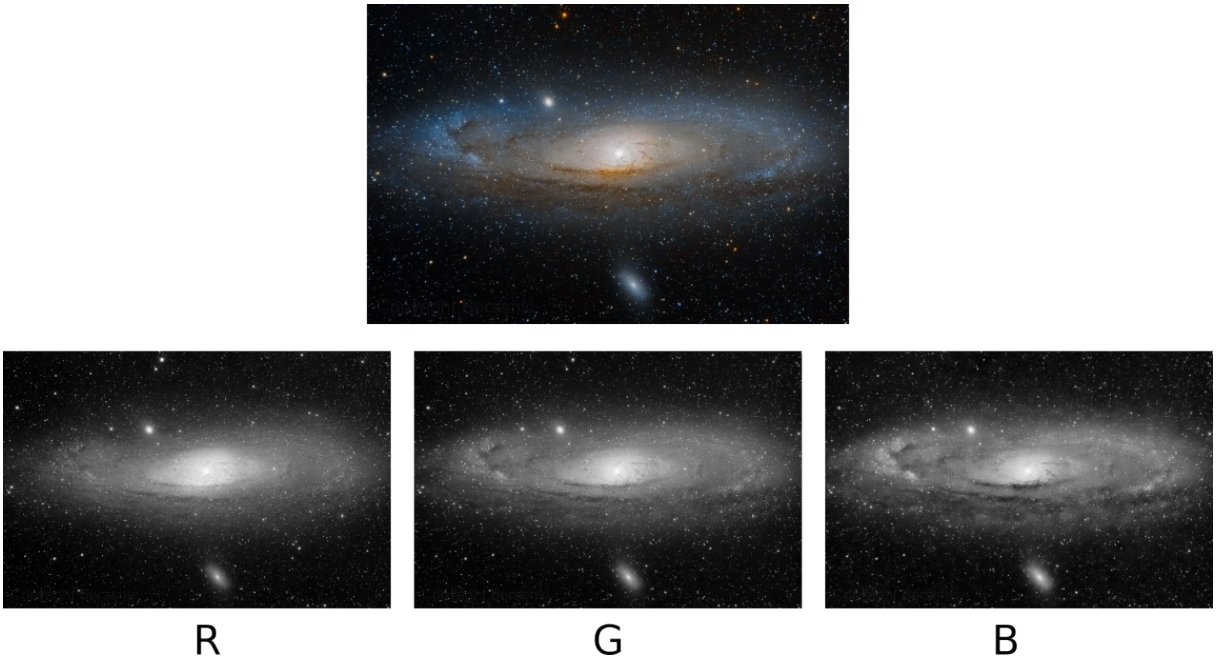

Figura 3.2 – Aqui é mostrado a galáxia de Andrômeda com cada canal do RGB colorido em escala de cinza. É possível verificar que parte da estrutura se modifica se vista por outra região do espectro luminoso. Créditos: Andrew Burwell

pixels associados a um objeto astronômico é chamado de magnitude aparente e podemos calculá-lo para um determinado filtro X usando a formula:

$$m_X = -2,5 \log_{10}(f_X) + \text{const}, \tag{3.1}$$

onde f_X é o fluxo de energia eletromagnética proveniente do objeto estudado e a constante de calibração. É importante dizer que a magnitude aparente depende da distância do observador, pois a intensidade luminosa cai com o inverso do quadrado da distância. Para podermos comparar objetos a distâncias diferentes definimos a **magnitude absoluta** M que é o valor da magnitude aparente quando se encontra a uma distância fixa de 10 pc:

$$m_X - M_X = 5 \log_{10}(d) - 5, \tag{3.2}$$

onde d é a distância medida em parsecs. Outra grandeza importante é obtida calculando a diferença entre duas magnitudes, tipicamente chamada de **índice de cor**. Dado duas bandas X e Y o índice de cor associado a elas é:

$$m_X - m_Y = -2,5 \log_{10} \left(\frac{f_X}{f_Y} \right). \tag{3.3}$$

O primeiro sistema fotométrico foi padronizado por volta de 1950 pelos astrônomos americanos Lester Johnson e William Morgan, ficando conhecido como sistema Johnson-Morgan ou sistema UBV. Esse sistema é composto pelas bandas U carregando faixas na região do ultravioleta, B na região do azul e V na região da luz visível. Por volta de 1965 esse sistema seria atualizado para o sistema Johnson-Cousins (UBVRI), onde o R carrega faixas na região do vermelho e o I na região do infravermelho.

Filtro	Ponto médio do comprimento de onda efetivo	Largura total Meio Máximo
U	365 nm	66 nm
B	445 nm	94 nm
V	551 nm	88 nm
R	638 nm	138 nm
I	797 nm	149 nm

Os índices de cor geralmente estão bem relacionados com características físicas do objeto astronômico. Por exemplo, no sistema UBVRI, o índice de cor associado às bandas (B−V) traz informação sobre a temperatura e a cor das estrelas: para aquelas mais frias e avermelhadas a magnitude na banda B é maior do que na banda V, o que faz com que o índice de cor seja um valor positivo. Porém, se a estrela for mais quente e azulada a situação se inverte tornando o índice de cor negativo. Como as galáxias, considerando unicamente a componente bariônica, são aglomerados de estrelas, gás e poeira, a cor integrada delas é a soma de vários fatores, entre eles a taxa de formação de estrelas. Estatisticamente as galáxias espirais tendem a ser mais azuis e as elípticas mais vermelhas como discutido anteriormente [83], no entanto isso não justifica uma classificação delas meramente baseada em índices de cor [84, 85]. Um diagrama frequentemente usado para estudar as propriedades das galáxias, é o diagrama índice de cor vs magnitude absoluta [86, 87, 88]. Neste diagrama, as galáxias elípticas estão em uma faixa quase horizontal, chamada de sequência vermelha, enquanto as espirais habitam o que é chamado de nuvem azul. Além disso, no diagrama cor-magnitude, existe uma região conhecida como **vale verde**, entre a sequência vermelha e a nuvem azul, em que as classificações morfológicas se misturam [89, 90]. Finalmente, os índices de cor precisam ser corrigidos, especialmente nos comprimentos de onda mais azuis, devido a absorções pelo meio interestelar e intergaláctico, além do avermelhamento causado pelo redshift.

Hoje contamos com vários levantamentos (surveys) *all-sky* que capturam imagens de milhares de galáxias pelo universo. Alguns, como o LEGACY survey, tem alta taxa de sinal-ruído, proporcionando imagens com uma alta profundidade e resolução, em bandas largas (g,r,i,z). No entanto, também temos levantamentos que utilizam uma combinação de filtros largos e estreitos, com tempos de integração menores, gerando imagens de menor profundidade e resolução. Acerca destes últimos, S-PLUS [10] é um levantamento que conta com um total de 12 bandas (5 largas e 7 estreitas). Cobrindo uma área de $\simeq$ 9300 deg^2, S-PLUS lançará milhões de imagens de galáxias em 12 bandas, tornando fundamental o desenvolvimento de um método capaz de classificar galáxias de forma rápida e robusta. Na próxima seção vamos discutir as vantagens práticas no uso de Deep Learning para esse problema, visto que o ganho não é apenas no tempo de execução.

Capítulo 4

Classificação automática de galáxias usando CNN

O trabalho de análise morfoçógica de galáxias vendo sendo aperfeiçoado, com vários métodos que procuram aumentar a eficiência se comparado com a classificação visual. O Galaxy Zoo [83] utilizando uma abordagem que aproxima a população da ciência, servindo instruções para que voluntários possam efetuar essa classificação. Existem métodos paramétricos que estimam parâmetros morfológicos traçando um perfil luminoso da galaxia [91, 92, 93], assim como métodos não paramétricos para mensurar parâmetros estruturais das galáxias através de sua concentração (C), assimetria (A) e aglomeração (S) dentro do sistema CAS [77, 94]. A medida que grandes levantamento de dados são construídos, o número de galáxias a serem classificadas cresce rapidamente o que destaca a importância desses métodos.

O interessante é que não precisamos dispensar a análise morfológica se temos a intenção de diminuir a necessidade de recursos humanos. Como discutido no capítulo 2, por mais que reconhecimento de padrões seja algo típico do dominio humano, esses padrões são aprendidos durante o treinamento de uma rede neural se tornando capazes de lidar com classificações morfológicas ou geométricas. Em particular, as Redes Neurais Convolucionais são as mais adequadas para este tipo de problema, podendo desempenhar até melhor do que o ser humano, visto que elas não sofrem com cansaço e interpretam imagens de maneira diferente [95, 15]. Como discutido na seção (3.2.1), algumas bandas carregam mais informação sobre uma componente morfológica (i.e. disco, bojo, braços espirais) do que outras. Emissões na região do ultravioleta geralmente são boas para capturar os braços espirais, mas se o brilho da galáxia for muito baixo talvez a intensidade dessa emissão seja muito pequena se comparada com as demais, de forma que no macro elas quase não se manifestem na imagem de cor. Outro problema é o fato de que emissões de pouca intensidade facilmente podem ser confundidas com o ruído natural do processo de

captura dessas imagens. Porém, vimos na seção (2.3.3) que as CNNs são boas em analisar características locais na imagem, de forma que a intensidade de uma banda pode ser pequena, mas a rede neural ainda vai detectar essa informação. Essa vantagem é ilustrada na seção de resultados.

É importante lembrar que quando se trata de redes neurais precisamos estabelecer uma arquitetura e um conjunto de treino adequado, de forma que ele seja diversificado o suficiente para englobar vários casos e diminuir o sobreajuste (overfitting). Assim discutiremos em detalhes as nunciais da arquitetura utilizada e as dificuldades obtidas durante o processo de treino da rede utilizada para a classificação das galáxias.

4.1 Arquitetura da CNN

O processo de construir uma certa arquitetura para resolver um problema pode ser uma tarefa difícil que envolve muita intuição, tentativa e erro. Os neurônios de entrada e os de saída são bem conhecidos, mas os neurônios ocultos são aqueles que trazem o real desafio para esta tarefa. Se tratando do problema de classificação de galáxias, precisamos de algumas camadas convolucionais no início para retirar as características principais e, em seguida, algumas camadas de pooling para dar uma visão mais global para a rede neural. Contudo, a pergunta principal é: quantas dessas camadas são necessárias?

Se colocarmos muitas camadas de convolução e pooling no início boa parte da informação contida localmente nas bandas é perdida, o que pode dificultar a detecção dos braços espirais se estivermos com galaxias de baixo brilho. Porém, se colocarmos poucas, o modelo não ganha tanto poder em reconhecer padrões morfológicos mais globais na imagem. Além disso, ainda teríamos que decidir a quantidade de neurônios ocultos e como eles se arranjam entre si. Isso mostra como encontrar a configuração ideal para a arquitetura da rede neural pode ser uma tarefa difícil. Ainda assim, existem vários estudos relacionados às redes neurais e com o tempo algumas arquiteturas ficaram famosas por apresentarem um alto desempenho para determinados problemas. Para o caso deste trabalho, as EfficientNets propostas por Tan & Le [96] se mostraram muito eficazes. Em seu paper Tan & Le apresentam 8 versões rotuladas por $\{B0, B1, ..., B6, B7\}$, onde o número está relacionado à quantidade de hiperparâmetros que cada uma usa. Com mais hiperparâmetros mais pesos devem ser ajustados durante o treino, o que aumenta o tempo de convergência do modelo para a sua configuração ótima. Como discutido na seção (2.2.1), aumentar o número de hiperparâmetros nem sempre está relacionado a uma melhor desempenho do modelo. Um modelo com número grande de parâmetros tem, por exemplo, mais propensão ao sobreajuste além de menor desempenho em termos de tempo computacional dado que o excesso de parâmetros sobrecarrega o processo de treino com conexões irrelevantes, possivelmente criando mais mínimos locais.

Para este trabalho utilizamos como base a EfficientNet B2, que também foi utilizada

nos dois trabalhos que substanciaram esta dissertação [18, 19]. Para adaptá-la ao problema de classificação de galáxias realizamos algumas alterações. A primeira alteração se refere à ajustar sua primeira camada de convolução de forma a receber o tamanho correto do recorte de imagem da galáxia, ou *stamp* como input. Do mesmo jeito precisamos alterar a saída para contemplar as duas categorias pelas quais estaremos classificando as galaxias: Late-Type Galaxy (LTG) ou Early-Type Galaxy (ETG). Porém, ainda existe um problema prático nesse tipo de classificação, visto que o S-PLUS não consta apenas com galaxias nos seus catálogos. As imagens capturadas podem conter outros objetos astronômicos que não são de interesse para esta classificação. Se passarmos, por exemplo, uma estrela para o modelo, em geral, ele não conseguirá entender que o objeto em questão nem faz parte do grupo em que queremos classificar, forçando assim uma classificação de galaxias para uma estrela. Vamos passar por três possiveis soluções para este **problema de classificação forçada**, onde a terceira foi a escolhida para este trabalho.

S_1:Ajustando o threshold

A definição de um threshold é muito importante em um problema de classificação, especialmente quando temos uma classificação binária. Quando as duas classes são mutuamente exclusivas as probabilidades de pertencimento se complementam e um primeiro threshold que imaginamos para as duas classes é o de 50%. Dessa forma, se uma das classes tiver uma probabilidade de pertencimento acima de 1/2, a outra será abaixo de 1/2 caracterizando o comportamento mutuamente exclusivo da classificação. Isso ocorre devido a certas decisões tomadas na saída da rede neural como o número de neurônios ou a função de ativação. Em geral, para um problema de classificação binária mutualmente exclusiva, apenas precisamos de um neurônio de saída responsável por identificar a probabilidade de pertencimento de uma das classes. A probabilidade de pertencimento da outra classe seria calculada tomando o complementar da primeira, visto que elas são excludentes. Mas se estivermos utilizando dois neurônicos de saída ainda podemos ter esse efeito se utilizamos a função de ativação $softmax$ [33] para esses dois neurônios, ela garante que as probabilidades se mantenham complementares entre si. Com a $softmax$ cada neurônio tem sua saída normalizada pelos outros neurônios da mesma camada, como podemos perceber na equação (4.1).

$$f(x_i) = \frac{e^{x_i}}{\sum_j^N e^{x_j}}, \tag{4.1}$$

onde x_i são as entradas recebidas no neurônio x_i e N é o número total de neurônios na camada [22]. Caso isso não seja desejado, podemos então utilizar a função de ativação $sigmoid$ que permite que as probabilidades das classes sejam calculadas de forma independente.

É importante notar que as probabilidades calculadas serem complementares está ligado à arquitetura adotada no modelo, mas a classificação final ser complementar está ligado

à escolha do threshold definido para cada classe. Como dito acima o threshold de 50% para as duas é a escolha mais simples e direta para esse tipo de classificação, mas pode não ser a ideal se a detecção de uma das classes for mais importante do que a detecção da outra. Se estamos priorizando uma das categorias então temos uma classe de interesse e como discutido na seção (2.2.2), alterar o threshold afeta a quantidade de falsos positivos, verdadeiros positivos e dos outros verificadores em respeito a esta classe de interesse. Suponha que tenhamos definido os thresholds de 70% e 30% correspondentes à primeira e a segunda classe. Dessa forma estamos aumentando a precisão do modelo na classificação da primeira classe, mas consequentemente diminuindo a precisão da segunda classe. Essa decisão costuma aumentar a métrica precision (2.12), pois se o modelo necessita achar mais padrões em comum com a classe a ser designada isso reduz o número de falsos positivos. Por outro lado, se precisamos de uma alta probabilidade para que a primeira classe seja escolhida objetos que compartilham de características das duas classes provavelmente serão levados para a segunda, o que aumenta a quantidade de falsos negativos diminui a métrica recall (2.13). O interessante está no fato de não precisarmos manter os thresholds complementares entre si, por mais que as probabilidades dadas pelo modelo possam ser.

Se exigirmos que o threshold seja maior que 50% para ambas categorias estamos aumentando a precisão de ambas classificações. Isso pode servir como uma solução para o problema de classificação forçada, visto que uma classificação só será endereçada se o objeto compartilhar de suficiente características para ser colocado em uma das duas classes, caso contrário ele não pertencerá a nenhuma delas. Isso configuraria uma forma indireta de ter uma terceira classificação, destinada aos objetos que parecem não ter muitas similaridades com os grupos em estudo. O problema dessa abordagem é que a classificação de galáxias contém objetos que podem ter características das duas classificações. A existência de galaxias lenticulares faz com que a classificação não se torne tão mutuamente exclusiva, pelo menos pelo ponto de vista prático. Para esse tipo de galáxia o modelo poderia determinar uma probabilidade de pertencimento próxima de 50% para ambas as classes o que acabaria fazendo a classificação final não designá-la a nenhuma classe devido ao alto threshold não complementar.

S_2:Criando classes adicionais

Vimos acima que podemos ajustar o threshold de forma a criar indiretamente uma terceira classe, mas fazer isso diretamente pode ser uma alternativa. Ajustar isso na arquitetura não parece ser difícil, visto que bastaria acrescentar mais neurônios de saída correspondentes às novas classes. Dessa forma poderíamos separar melhor os tipos de objetos que encontramos no catálogo do S-PLUS. É uma solução que deve ser feita com cuidado, visto que a separação do conjunto de treino teria que ser feita de forma ainda mais rigorosa para gerar uma boa diversidade dentre todas as classes. Além disso, a classificação deixaria de ser binária de forma que poderíamos explorar melhor o fato dela

não ser mutualmente exclusiva. Poderíamos ter uma classe separada para as stamps que foram dominadas por um estrela e uma classe separada para as galáxias lenticulares que na classificação binária poderiam causar confusão para o modelo.

O problema desta abordagem está no fato dela não ser tão simples quanto parece. Na prática, aumentar o número de classes pode requerer maior poder de generalização por conta da rede, pois agora existem mais padrões que ela deve aprender. Em outras palavras, não há garantias que aumentar o número de classes não acabe gerando uma necessidade de mudança na arquitetura, principalmente na quantidade de neurônios ocultos. Lembre-se que as camadas ocultas são aquelas responsáveis por fazer as correlações entre os dados recebidos nos neurônios de entrada e as classes correspondentes aos neurônios de saída. Naturalmente existe uma quantidade ótima de neurônios que consegue fazer essas correlações de forma ideal ajustando os pesos no processo de treinamento. Porém, se aumentarmos o número de classes, estaremos pedindo que a mesma configuração e quantidade de pesos se ajustem a uma quantidade maior de classes. Fazer isso sem que ela perca precisão em cada uma das classes pode ser uma tarefa bem difícil se não aumentarmos junto o número de hiperparâmetros. Pode ser uma abordagem que funcione bem utilizando uma versão com mais parâmetros da EfficientNet, mas não é viável se o objetivo é simplesmente o de catalogar as galáxias entre early e late type. O que precisamos é de um método robusto que nos permita filtrar os objetos de interesse do catálogo, para depois classificá-los.

S_3:Utilizando duas redes neurais

Ao invés de criar mais classes para um mesmo modelo possivelmente exigindo maior poder de processamento, basta dividir esse trabalho em dois modelos. Dessa forma, cada um fica responsável por parte do processo, não exigindo uma arquitetura única e mais complexa que faça as duas coisas. Essa segunda rede também trabalha em um esquema de classificação binária, mas apenas de forma a filtrar os objetos de interesse para a nossa classificação. Em vez de separar entre classe A e classe B ela separa entre Reliable (confiável) e Não Reliable (não confiável). Dessa forma, durante o treino, definimos como Reliable as stamps que contêm uma galáxia centralizada na imagem situada em ambientes não muito populados por outros objetos, enquanto colocamos nas Não Reliable o conjunto de objetos que não atendem aos critérios para ser Reliable.

É importante destacar que criar uma rede para identificar se um objeto pertence a uma classe A ou B é diferente de criar uma rede para identificar se um objeto pertence ou não a uma determinada classe. Ambos os casos são uma classificação binária, mas o propósito e a forma como eles são tratados durante o treino é diferente. Para o primeiro caso temos que ensinar ao modelo o conjunto de características que fazem um objeto pertencer à classe A e o conjunto de características que o fazem pertencer à classe B. Isso é importante para incorporar o caso em que a rede recebe um objeto que não contemple

características de nenhuma das duas classes. Já para o segundo caso só precisamos ensinar à rede as características dos objetos que pertencem à classe de interesse, pois qualquer outro objeto que não contemple essas características automaticamente será alocado na outra categoria. Para muitos problemas isso deve ajudar na convergência do modelo, mas isso não é uma regra e pode depender muito da intenção desejada com a rede. Mais do que filtrar stamps que não sejam galáxias, esse modelo também deve separar variações indesejadas da classe Reliable. Por exemplo, em uma stamp que tenha uma estrela comprometendo a magnitude estimada da galáxia e/ou sua forma, o modelo ainda pode encontrar as características que o fariam classificar como Reliable. No entanto, esse tipo de stamp é indesejável e deve ser classificado como Não Reliable. Por situações como esta, a convergência do modelo no segundo caso pode ser mais dificultosa se comparado com o primeiro e isso se apresentou no treinamento da nossa rede como podemos ver na figura (5.2) no capítulo de resultados.

Em especial este trabalho é a continuação de um trabalho anterior publicado em 2021 [18] que tinha o mesmo propósito de classificar galáxias do levantamento S-PLUS em early e late type. O trabalho atual adotou a solução S_3 para o problema de classificação forçada, garantindo que os objetos a serem classificados estejam em boas condições para tal. A arquitetura de ambos os modelos está representada na figura (4.1), onde ambos têm a EfficientNet B2 como base. Além disso, algumas mudanças na arquitetura foram feitas para aumentar a confiança na classificação. A mais importante está no fato da função de ativação dos neurônios de saída ter mudado de Softmax para a Sigmoid [33], o que permite que as probabilidades das classes não sejam calculadas de forma complementar. A predição é feita de forma independente, sendo assim existe a possibilidade de encontrarmos objetos classificados tanto como LTG quanto ETG. Dessa forma, teremos aqueles objetos que o modelo não terá duvidas sobre a sua classificação e outros objetos que o modelo dará uma alta probabilidade de pertencimento para as duas classes. É uma forma indireta de criar uma terceira classe de objetos, que foram chamados de ambíguos.

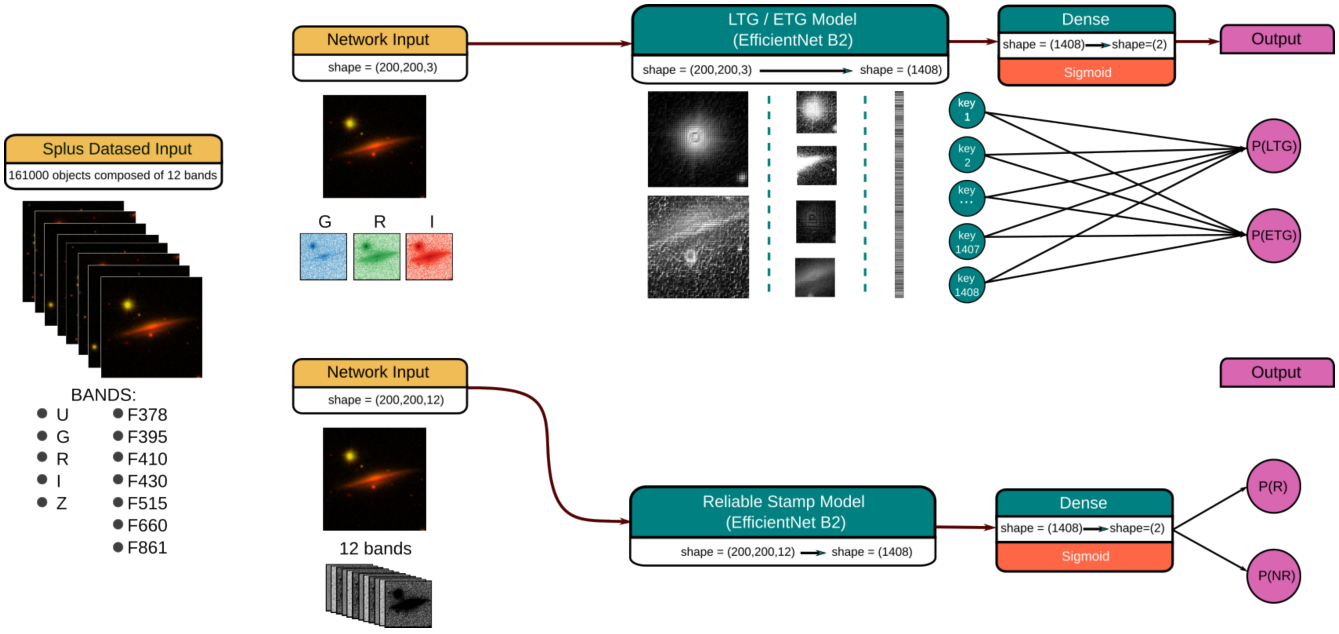

Figura 4.1 – Esta figura mostra de forma esquemática como ambas arquiteturas tratam a imagem, gerando no final delas as probabilidades de pertencimento. As primeiras camadas convolucionais e de pooling são responsáveis por reconhecer e compactar os padrões necessários para a classificação. Em seguida passamos essa informação por uma camada densa que novamente compacta toda a informação em um vetor com 1408 entradas, representada pelo código de barras. Quanto mais próximo esse vetor for daqueles gerados durante o treino para um certa classe, maior a probabilidade associada à essa classe. Ambas as redes funcionam da mesma forma, a diferença está na quantidade de bandas utilizadas.

4.2 Processo de Treino

Antes de qualquer coisa é importante conhecer as características dos dados de entrada. Os objetos utilizados para o treinamento da rede foram retirados do **S-PLUS** [10], um levantamento que atua catalogando e calculando grandezas de objetos astronômicos no hemisfério sul utilizando o telescópio T80S no Chile. O S-PLUS está equipado com um sistema de filtragem de sete filtros de banda estreita (J0378, J0395, J0410, J0430, J0515, J0660, J0861) e cinco de banda larga (u, g, r, i, z), resultando em catálogos com as magnitudes dos objetos nos 12 filtros, o que nos possibilita bastante informação tanto para estudo científico, quanto para a observação de objetos particulares, inclusive de baixo brilho

Filtro	Ponto médio do comprimento de onda efetivo (Å)	Largura total Meio Máximo (Å)
u	3563	66 nm
J0378	3770	151
J0395	3940	103
J0410	4094	201
J0430	4292	201
g	4751	1545
J0515	5133	207
r	6258	1465
J0660	6614	174
i	7690	1506
J00861	8611	408
z	8831	1182

Em especial, as galáxias constituem pequenas porções de todo o campo capturado pela câmera do T80, que cobre $\simeq 2\ deg^2$. A maioria delas podem ser identificadas fazendo cortes conhecidos como **stamps** de 200×200 pixels^2 em torno de seu centro. Em geral esse corte pode acabar compromentendo a classificação de galáxias mais extensas que invadam os limites do recorte, o que aumenta a necessidade de operar em associação com um segundo modelo que determina as stamps que não atendem às condições adequadas para a classificação.

Outra característica a se pensar em relação ao treino é como seria a configuração de entrada dos dados. As redes neurais convolucionais essencialmente recebem os dados como um volume de dados. No capítulo 2 discutimos que o computador visualiza uma imagem a parir de uma matriz de píxeis, separando ela em 3 canais comumente associados ao R (vermelho), G (verde) e B (azul). Fazemos isso devido ao fato das cores no espectro vísivel poderem ser decompostas em termos dessas 3 cores primárias, mas isso é apenas uma limitação humana no que tange a forma como enxergamos luz. Naturalmente um objeto astronômico cobre várias regiões do espectro luminoso e o S-PLUS com seus 12 filtros é capaz de detectar comprimentos de onda que vão desde o ultravioleta até o infravermelho. Para cada um desses filtros existe uma matriz de píxeis correpondente que o computador interpreta como um dos canais da imagem. As redes neurais não têm a limitação humana de trabalhar com apenas 3 canais, podemos passar para ela todo um volume de dados contendo todas as 12 bandas do S-PLUS de informação. Nessa sentido é importante separar as bandas que mais contribuem para aquilo que se quer encontrar na imagem, pois utilizar todas as bandas disponíveis pode acarretar o mesmo problema discutido na seção (2.2.1) no que se refere a entregar para a rede mais do que ela precisa. Além disso, as bandas mais azuis têm taxas de sinal-ruído mais baixas podendo se tornar uma fonte de ruído no estudo morfológico. A configuração que gerou os melhores

resultados foi utilizar 3 bandas largas (g, r, i) com pesos pré-treinados em um conjunto de dados conhecido por ImageNet [18].

O primeiro trabalho do nosso grupo [18] focava em classificar as galáxias disponíveis no Data Release 1 (DR1) do S-PLUS. O Galaxy Zoo 1 project [88, 83] conseguiu classificar aproximadamente 900 000 imagens de galaxias fornecidas pelo SDSS (Sloan Digital Sky Survey), utilizando ciência cidadã por dois anos. Dessas galaxias, 4232 pertenciam ao campo conhecido como Stripe-82, que o S-PLUS também cobre. Essas galáxias classificadas pelo Galaxy Zoo 1 no Stripe-82 foram utilizadas para o treino da rede no trabalho anterior e neste também, com a diferença que o atual utiliza o data release iDR3 do S-PLUS, que incorpora o DR1. Escolhido as amostras para o treino e a configuração da arquitetura, ainda existe a possibilidade do treino não apresentar uma boa convergência. Com isso, foi utilizado pesos já treinados de um conjunto de dados conhecido como ImageNet [97], mesmo eles vindo de outro tipo de problema, estes pesos devem compartilhar características semelhantes na forma de buscar informação para a classificação.

Outro problema que temos que lidar é com a seleção inteligente das amostras de treino, e nesse caso o conjunto é constituído de boas variações de cada classe. As LTGs selecionadas estão dispostas em vários ângulos e tamanhos diferentes como podemos ver na figura (4.2). Os braços espirais das galaxias somem se ela for vista de lado, i.e. edge-on, logo o modelo também tem que conseguir classificar corretamente essa situação. As figuras mostram a mesma galáxia no S-PLUS e Legacy Survey, onde o segundo consegue obter imagens com maior profundidade e resolução. É importante notar que em algumas delas os braços espirais ficam bem menos nítidos no S-PLUS o que dificultaria uma classificação feita visualmente, mas essas amostras são importantes justamente para que o modelo procure por padrões dentro da imagem que não sejam visiveis a olho nu. Já quanto às ETGs, é importante que o conjunto de treino seja composto de galaxias de diferentes excentricidades, como podemos ver na figura (4.3). Além disso, galáxias com alta excentricidade podem ser confundidas com LTGs edge-on, o que também não é desejado.

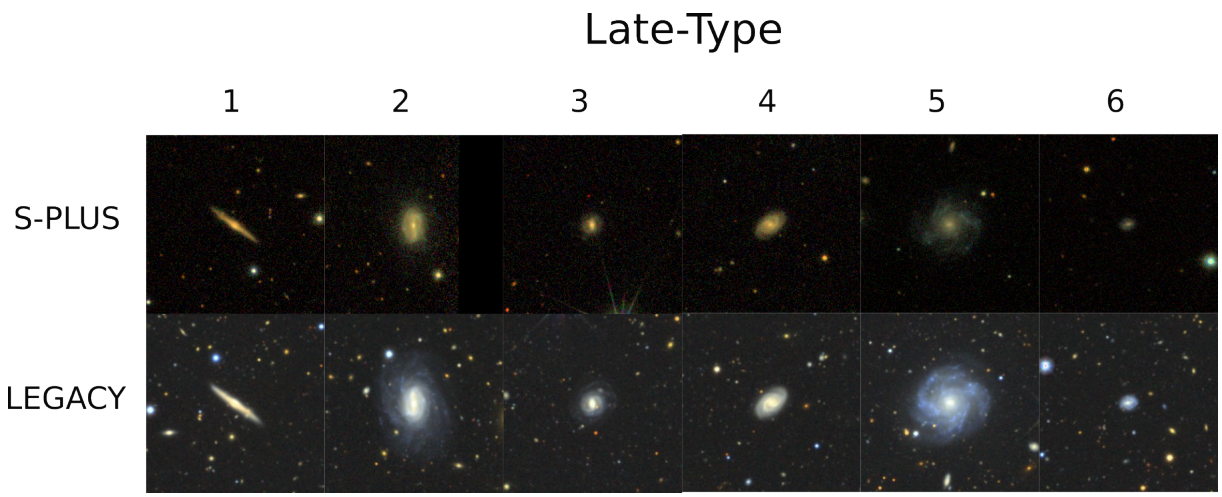

Figura 4.2 – Essa figura mostra alguns exemplos de galáxias LTG servidas como treino para rede, com o destaque para a variedade entre elas.

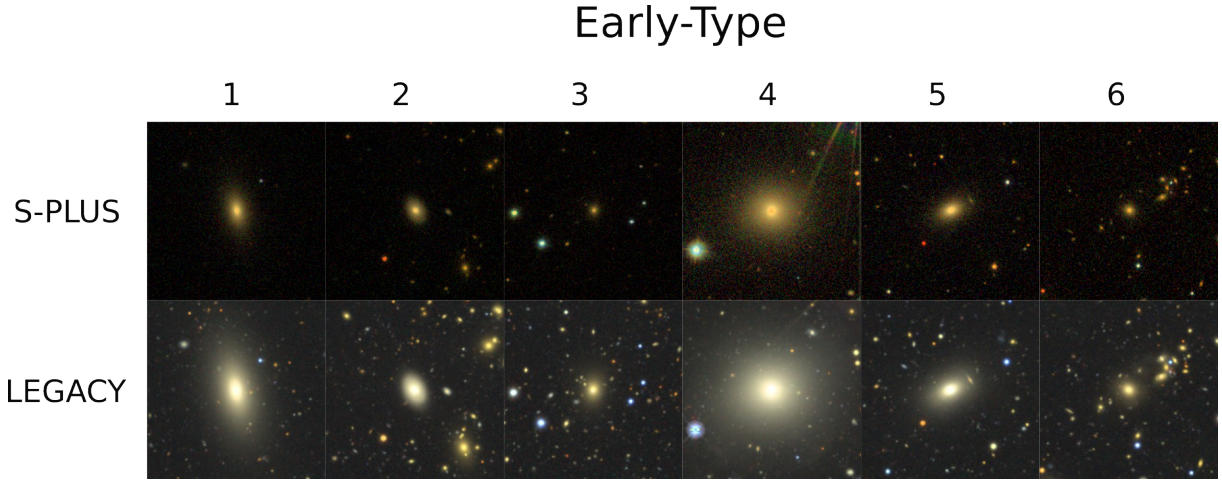

Figura 4.3 – Essa figura mostra alguns exemplos de galáxias ETG servidas como treino para rede, com o destaque para a variedade entre elas.

As amostras separadas costumam ser dividas em treino, validação e teste, sendo que o conjunto de validação é aquele utilizado para gerar métricas em respeito ao processo de treino (2.2.2). Essa própria avaliação da rede não vai ser muito precisa se este conjunto também não tiver ampla diversidade. Usualmente quando se trata de deep learning precisamos servir uma quantidade considerável de dados para o treinamento, dependendo do problema a ser tratado. Nosso treino continha um total de 4192 amostras e inspecionar uma por uma de forma a garantir diversidade pode ser uma tarefa trabalhosa. Se pegarmos uma parte aleatória do conjunto e designá-la para validação, podemos acabar tendo o azar de selecionar amostras muito parecidas entre si. O mesmo serve para as amostras designadas para o treino. Sendo assim, existem técnicas que ajudam a evitar esses efeitos de seleção, o que possibilita uma avaliação mais confiável do modelo. Uma delas é a **validação cruzada** que basicamente cria sessões onde o conjunto de validação se alterna com o de treinamento sem criar intersecções. Entre essas técnicas a validação

cruzada $k-fold$ [98] foi a utilizada; ela consiste em separar k pastas (folds), onde em cada uma dessas pastas ocorre o processo descrito acima. As pastas são criadas de forma que não haja intersecção entre a validação separada para cada uma, ver figura (4.4). Aliado a isso, para cada pasta, o treino muda levemente de forma a permitir outras configurações. Para este trabalho foram utilizados 7 folds. É importante notar que o conjunto de teste

Figura 4.4 – Aqui mostro uma imagem que ilustra a forma como foi feita a validação cruzada k-fold do modelo LTG-ETG. Para ele foram utilizadas 7 folds, onde cada fold altera tanto o conjunto de treino quanto o de validação, de forma a não haver intersecção entre as validações em cada fold.

se manteve fixo e para cada pasta j o modelo foi treinado obtendo um conjunto ótimo de pesos W_j. Assim, podemos julgar melhor tanto a etapa de treino, quanto a performance final da rede com um conjunto de teste comum a todos os folds.

Capítulo 5

Resultados e discussões

Nesta sessão apresentarei alguns resultados quanto a performance da rede durante e posteriormente ao treino. Também apresentarei algumas informações sobre o catálogo que foi possível ser feito com esse modelo utilizando o DR3 do S-PLUS e suas implementação no contexto de evolução de galáxias.

5.1 Performance do treino

Este trabalho foi feito utilizando as bibliotecas keras e tensorflow do Python [99, 100], onde com elas foram implementados todos os métodos discutidos nos capítulos anteriores. O processo de treino se deu criando uma arquitetura para cada uma das folds dadas pelo método de validação cruzada $k - fold$. Assim fomos capazes de obter uma métrica para cada fold e com elas julgar o treinamento da rede para cada conjunto de validação e treino separado. Um gráfico importante que nos ajuda a verificar isso é a curva **precision-recall**. Como discutido na seção (2.2.2), essas métricas são calculadas levando em consideração as taxas de acertos e erros: Positivo Verdadeiro, Positivo Falso, Negativo Verdadeiro e Negativo Falso. No entanto, essas taxas dependem do threshold escolhido para a classificação final, visto que o modelo em si apenas calcula as probabilidades de pertencimento. Sendo assim, cada ponto desse gráfico (Fig: 5.1) é feito calculando o precision e o recall dados nas equações (2.12)(2.13) para um conjunto suficientemente pequeno de valores discretos de threshold contido no intervalo [0,1]. Depois, para cada valor de threshold, foi calculado a média entre esses avaliadores, utilizando o desvio padrão em cada fold como uma aproximação para o erro nessas métricas. A intenção com esses resultados é aumentar o precision e também o recall, então essas curvas, além de serem avaliadores da etapa de treino, também podem ser usadas para estimar um valor ótimo de threashold para a classificação. Uma boa maneira de fazer isso é selecionando o ponto tal que a multiplicação do precision com o recall seja o mais próximo de 1. Com técnicas como

essa selecionamos o valor aproximado de 60% para o threshold do modelo LTG-ETG e de 54% para o modelo R-NR (Reliabble - Não Reliable). É importante destacar que o catálogo final consta com as probabilidades calculadas por ambos os modelos, sendo assim qualquer um pode selecionar um threshold que melhor atende às suas necessidades.

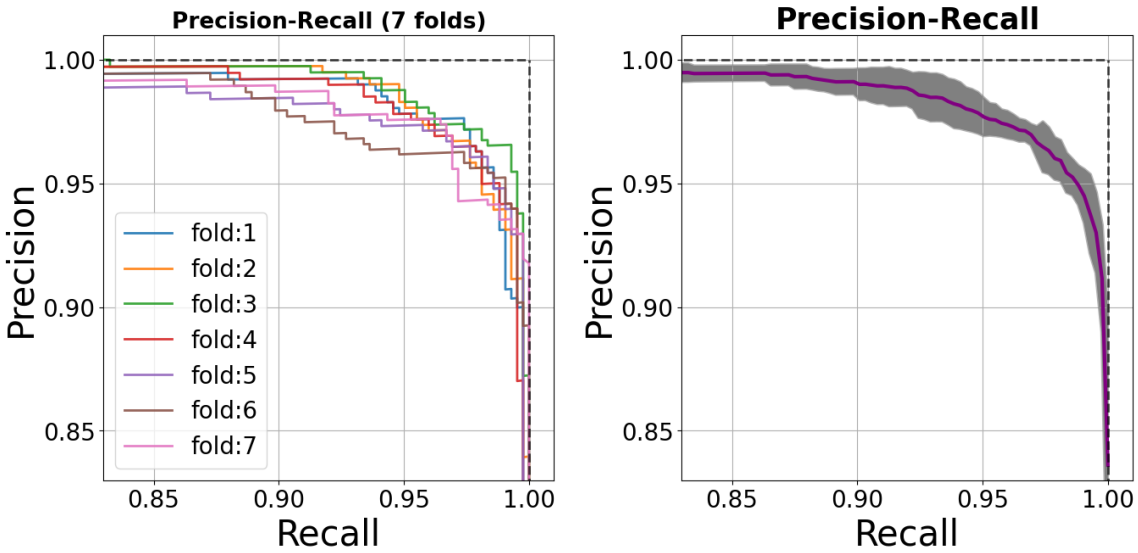

Figura 5.1 – Nesta figura mostro as curvas Precision-Recall do treinamento do modelo LTG-ETG. A imagem da esquerda mostra essa curva para o treinamento em cada uma das folds, enquanto que a imagem da direita mostra a curva média entre os folds em roxo, com a área em cinza sendo calculada usando o desvio padrão em relação a todas as folds.

A mesma abordagem pode ser feita para cada uma das métricas discutidas em (2.2.2). Ainda no que se refere ao treinamento, o modelo apresentou resultados mostrando que ele obteve ótimos níveis de generalização, o modelo final opera utilizando a fold que obteve melhores resultados com a área em baixo da curva precision-recall chegando a 0.996. O caso perfeito acontece quando a área resulta em 1, indicando a existência de um threshold capaz de separar completamente as duas classificações sem falsos positivos e falsos negativos. Na figura (5.2), podemos verificar que a função custo (Loss) de ambos os modelos diminui a cada época de treinamento. Isso ocorre tanto para o conjunto de treino, quanto para o de validação o que é um grande indicativo de que a rede não apresentou um overfitting durante o seu treinamento. Além disso, o fato desse comportamento se apresentar em todas as folds diminui a chance de termos simplesmente utilizado um conjunto de validação privilegiado que distorceu os resultados.

Perceba que o modelo LTG-ETG apresentou uma convergência mais suave do que o modelo R-NR. Como discutido na solução $S3$ da seção (4.1), isso pode ser explicado devido ao fato de ambas as classes do R-NR conterem uma galáxia em seu centro. Quando treinamos uma rede esperamos que ela ganhe generalidade no que foi ensinado, porém o objetivo da classe NR também é de filtrar pequenas variações indesejáveis da classe R, o que pode ir na direção oposta da generalização. Isso faz com que o modelo tenha que ser

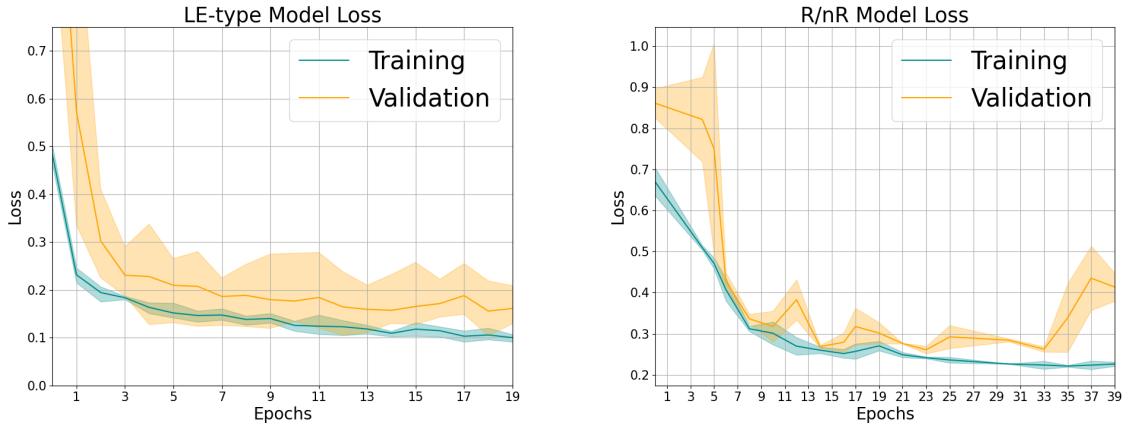

Figura 5.2 – Esta figura mostra a função custo (Loss) do treinamento de ambos os modelos. O método para fazer esse gráfico é o mesmo da figura acima, com a linha mais grossa indicando a Loss média e a região pintada sendo feita como um desvio calculado a partir de todas as folds. A curva azulada representa essa métrica aplicada no conjunto de treino, enquanto que na curva laranja a métrica foi aplicada no conjunto de validação

mais específico no conjunto de características que fazem um objeto ser de uma classe e não de outra.

Tendo selecionado um threshold ideal para a classificação, podemos utilizar o conjunto de teste para avaliar o desempenho da rede depois do treinamento. É importante lembrar que não existem amostras em comum entre o conjunto de treino, validação e teste. A figura (5.3) mostra a matriz de confusão da média entre as folds junto do precision e do recall calculado no conjunto de teste para ambos os modelos. Obtivemos uma taxa de acerto de aproximadamente 95% para o LTG-ETG e 90% para o R-NR.

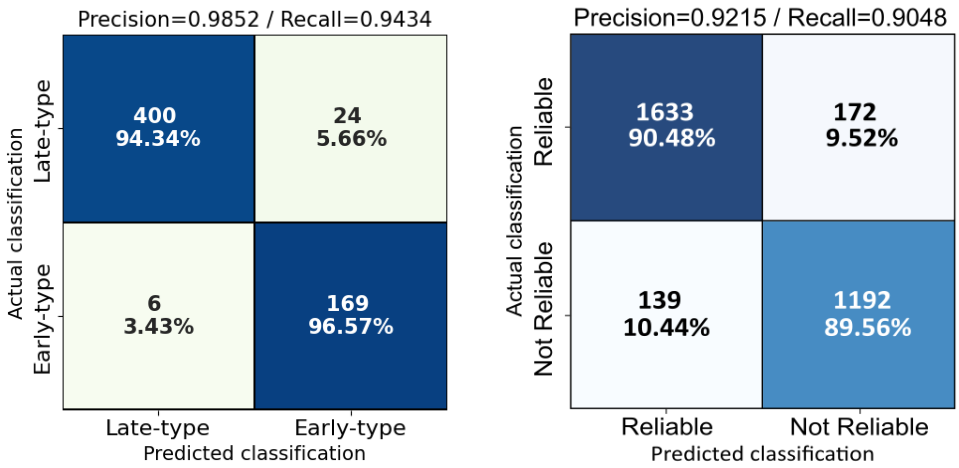

Figura 5.3 – Essa figura mostra a matriz de confusão, junto do Precision e Recall calculados no conjunto de teste para o fold de melhor desempenho. Em ambos os modelos essa métrica mostra uma alta precisão na sua classificação.

5.2 Produtos da classifcação

Com o modelo já treinado, utilizamos o DR3 do S-PLUS excluindo as galáxias que foram utilizadas para treino, validação e teste, para compor um novo catálogo com classificações totalmente geradas pelo modelo. Com isso fomos capazes de produzir um catálogo com 164314 objetos, onde constam as probabilidades determinadas por ambos modelos. Em alguns exemplos podemos atestar o potencial que essas redes têm para reconhecer padrões dentro da imagem. Na figura (5.4) está nítido no Legacy Survey que as duas imagens apresentam galáxias espirais (LTG), porém no S-PLUS os braços espirais não são visíveis devido ao fato do survey ser menos profundo. Ainda assim o modelo foi capaz de conferir uma alta probabilidade para a classe LTG e isso ocorre devido ao fato das redes neurais conseguirem encontrar a estrutura de uma espiral através da configuração dos píxeis. Parte do ruído é gerado por características físicas do meio interestelar, se todas as imagens servidas no conjunto de treino contemplam o mesmo tipo de ruído é bem possível que indiretamente a rede neural tenha aprendido a olhar através dele.

Figura 5.4 – Esta figura mostra duas galáxias primeiro no Legacy Survey depois no S-PLUS, junto das probabilidades geradas pelo modelo. Podemos ver que os braços espirais são bem reduzidos no S-PLUS, mas ainda assim o modelo conferiu uma alta probabilidade de pertencimento à classe LTG.

A seguir as imagens (5.5) e (5.6) mostram algumas classificações do modelo para a classe LTG e ETG. É possível observar que o modelo é capaz de servir uma boa classificação para diferentes perfis de galáxias dentro de uma mesma classe. Já a imagem (5.7) mostra aquelas galáxias que receberam uma alta probabilidade de ser LTG e ETG, caindo assim em um grupo especial que nomeamos de ambíguas. Essa ambiguidade serve como uma forma de o modelo apresentar incerteza quanto a certa classificação. Na galáxia

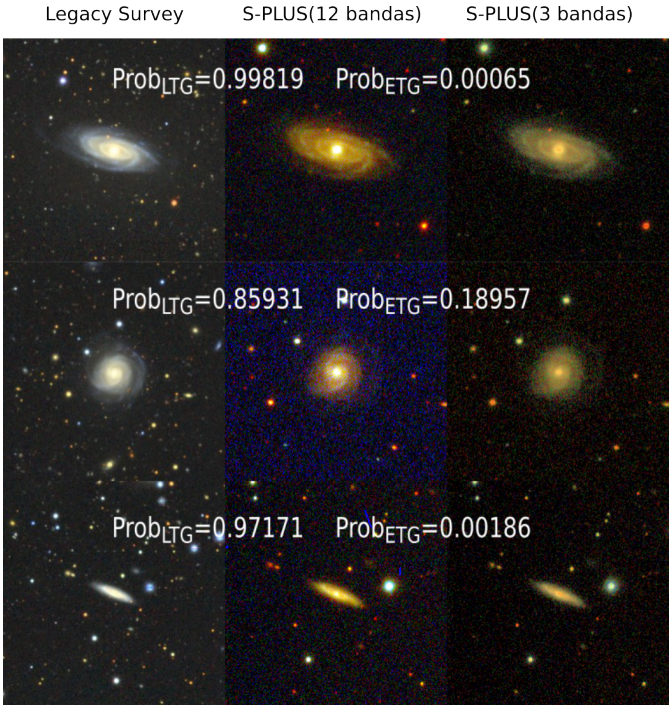

Figura 5.5 – Esta figura mostra 3 galáxias espirais em perfis diferentes classificadas pelo modelo. Em cada linha temos a mesma galáxia primeiro em Legacy Survey depois em S-PLUS usando todas as 12 bandas e finalmente em S-PLUS usando apenas as 3 bandas usadas no treino g, r e i.

da linha central novamente os braços espirais desaparecem no S-PLUS, o que justifica o porquê do modelo ter ficado indeciso. Ainda assim o modelo calculou uma probabilidade maior de ser LTG, se considerarmos isso como um critério de desempate. A classificação ficou ambígua, pois o threshold definido para ela é de aproximadamente 60% e as probabilidades geradas pelo modelo para ambas as classes estão acima desse valor. Vale lembrar que essa foi uma decisão consciente em relação a arquitetura da rede que evita o problema de classificação forçada discutido na seção (4.1). As probabilidades não têm a obrigação de somar 1 o que faz com que o modelo efetue essas previsões de forma mais independente. Já a galáxia de cima e de baixo tiveram uma maior probabilidade de ser ETG, mas também acabaram deixando o modelo indeciso devido ao fato de galáxias elípticas de alta excentricidade poderem ser confundidas com Espirais vistas de um certo angulo. Em geral essas galáxias ambíguas são objetos interessantes que pedem por uma segunda inspeção. No catálogo final 733 galáxias foram classificadas como ambíguas e Reliable, o que representa uma porcentagem muito pequena de todo o conjunto.

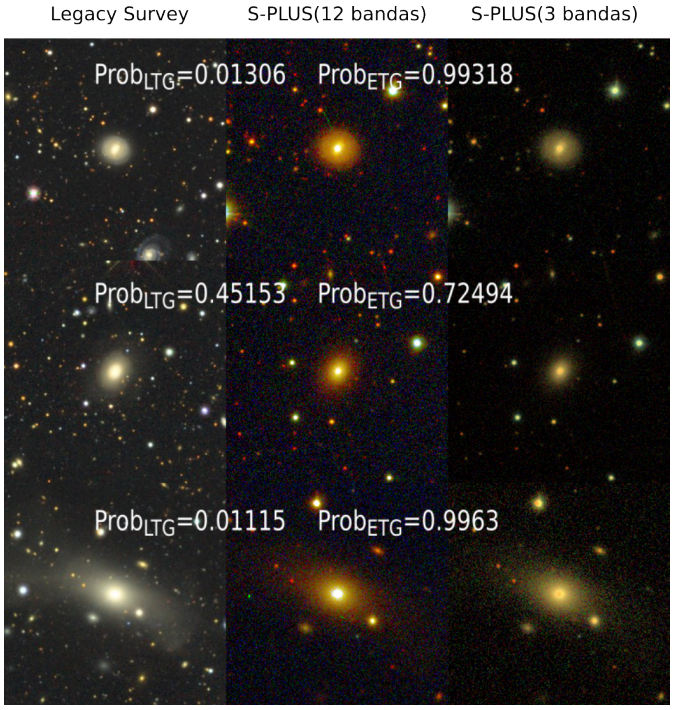

Figura 5.6 – Esta figura mostra 3 galáxias elípticas em perfis diferentes classificadas pelo modelo. Em cada linha temos a mesma galáxia primeiro em Legacy Survey depois em S-PLUS usando todas as 12 bandas e finalmente em S-PLUS usando apenas as 3 bandas usadas no treino g, r e i.

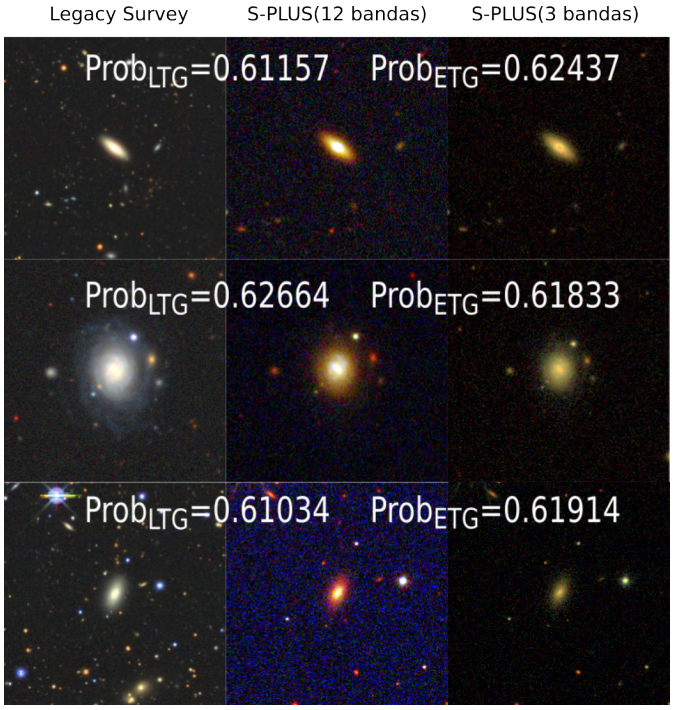

Figura 5.7 – Esta figura mostra 3 galáxias com classificações ambíguas geradas pelo modelo. Para todas as três a probabilidade associada a cada classe ultrapassa 60%, que foi o threshold selecionado para a classificação . Em cada linha temos a mesma galáxia primeiro em Legacy Survey depois em S-PLUS usando todas as 12 bandas e finalmente em S-PLUS usando apenas as 3 bandas usadas no treino g, r e i.

O modelo R-NR também apresentou resultados interessantes, conseguindo separar stamps com objetos que poderiam comprometer a classificação em 3 tipos. Na figura (5.8) podemos ver alguns exemplos de stamps NR separadas por esses tipos. Este é um modelo bom para filtrar objetos indesejados e que nos possibilitou aumentar a pureza do catálogo.

Figura 5.8 – Nesta figura mostro alguns exemplos de stamps que foram classificadas como Não Reliable, junto do provavel motivo delas terem caído nessa classificação.

Um diagrama muito utilizado para estudar objetos astronômicos, é o diagrama cor-magnitude, que é ligado a evolução estelar de um sistema [101]. Na figura (5.9) mostramos a mudança no diagrama $(g-r)$ vs M_r depois de desconsiderar os objetos Não Reliable. É possível ver que a dispersão diminui depois de selecionarmos apenas os objetos Reliable. Além disso, agora de posse do catálogo com as galáxias classificadas pelo modelo, em uma colaboração com Vitor Silva (Observatório Nacional) utilizamos as medidas dele com o k-vizinhos-próximos (k-nearest-neighbors) [102] para estimar a densidade dessas galáxias no ambiente que habitam. Com isso fomos capazes de recuperar as relações entre morfologia e densidade [66, 67, 68] no qual as ETGs tendem a habitar zonas mais densas do universo, enquanto que as LTGs são mais espalhadas no campo, ver figura (5.10). Para esse gráfico foram utilizadas apenas as galáxias Reliable e com magnitude na banda r menores que 17. O valor de $k = 4$ foi escolhido para estudar ambientes mais locais, o metodo utiliza os k vizinhos para estimar a densidade, logo se aumentamos muito o valor de k estarmos selecionando galaxias mais distantes.

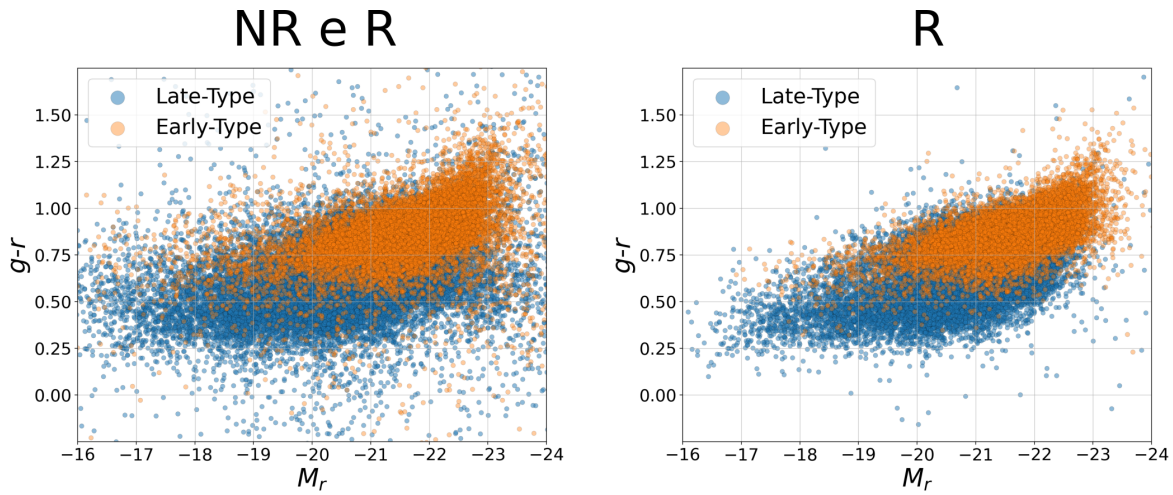

Figura 5.9 – Nesta figura mostro o diagrama cor-magnitude das amostras classificadas pelo modelo. A da esquerda são as amostras sem nenhum tipo de seleção. Já na segunda selecionamos apenas aquelas classificadas como Reliable pelo modelo R-NR.

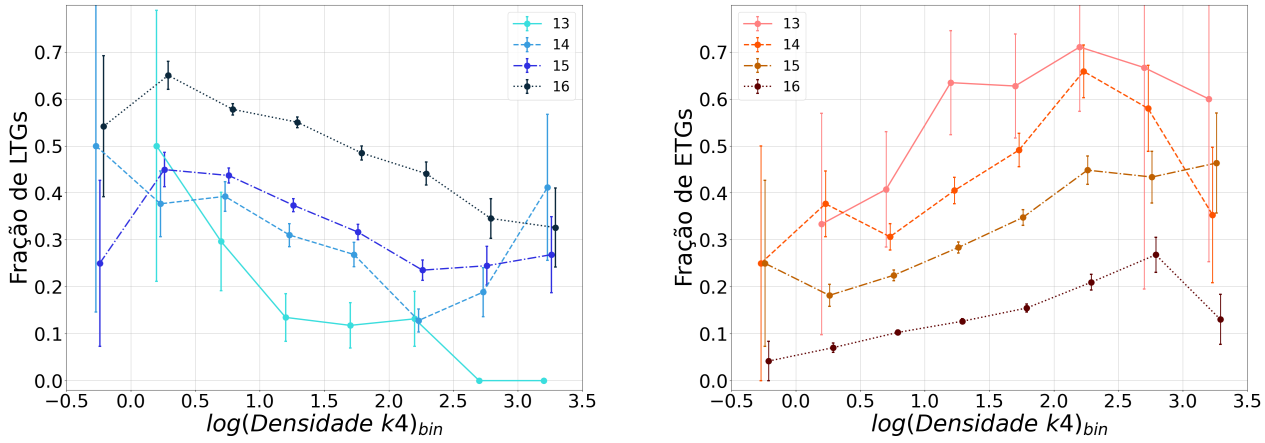

Figura 5.10 – Esta figura mostra a fração de galáxias que se encontram em um certo bin (intervalo) de densidade. Cada tipo de tracejado aliado à uma graduação na cor seleciona também o intervalo em M_r utilizado para fazer o gráfico. A figura da esquerda mostra o comportamento das galáxias late type, enquanto a da direita mostra as galáxias early type. É possível ver que o comportamento geral para todos os intervalos de magnitude é que a fração de LTGs diminui, enquanto que a fração de ETGs aumenta conforme aumentamos a densidade.

É importante notar que a classificação utiliza as informações contidas nas bandas g,r e i de forma independente, sendo ela explicitamente morfológica e não utilizando nenhum tipo de cor nos cálculos. Como consequência disso vale destacar que fomos capazes de recuperar galáxias Espirais vermelhas e Elípticas azuis na classificação, como mostra a figura (5.11).

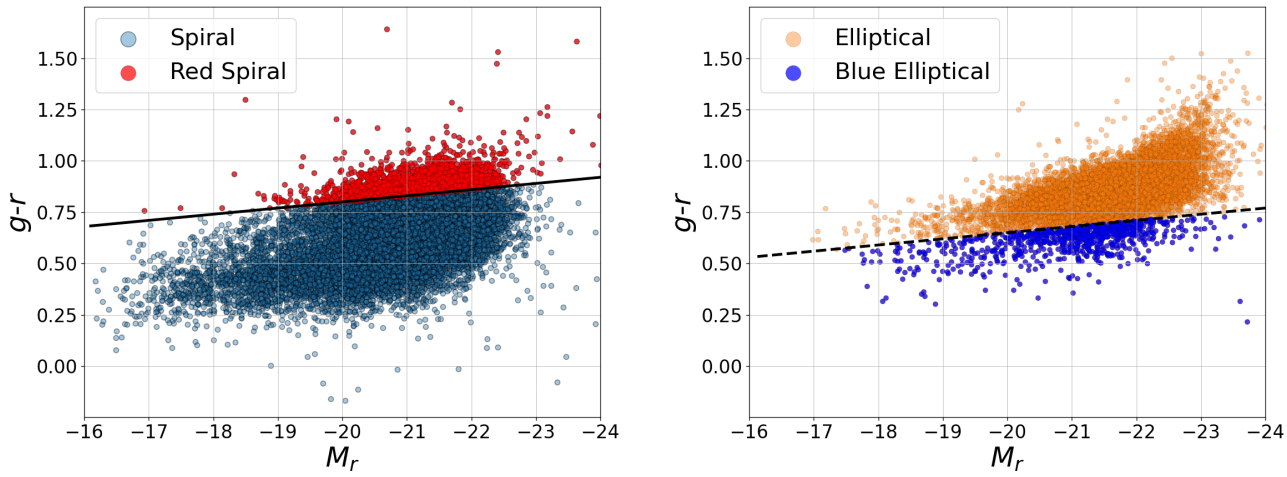

Figura 5.11 – Nesta figura mostro o diagrama cor-magnitude das amostras classificadas pelo modelo, com uma separação na cor delas. Amostras acima da linha continua são de galáxias mais avermelhadas enquanto que galáxias abaixo da linha tracejada são mais azuladas. Essas linhas foram calculadas utilizando o trabalho de Dhiwar et al. [1]

Em resumo, utilizamos as galáxias classificadas no Galaxy Zoo 1 no campo conhecido como STRIPE-82 como treino para a nossa rede neural. Modificamos a arquitetura da rede que atuou no DR1 do S-PLUS para que ela calcule as probabilidades de um galáxia pertencer à classe Late Type Galaxy ou Early Type Galaxy de forma independente, evitando assim problemas de classificação forçada. O modelo atua utilizando recortes de 200×200 pixeis^2, com uma galáxia em seu centro e para melhorar a confiança na classificação treinamos também um segundo modelo capaz de julgar se o recorte é Reliable (confiável) ou Não Reliable (não confiável) para a classificação. Utilizamos o método de validação cruzada $k - fold$ para minimizar efeitos de seleção no treinamento dos dois modelos e, ao mesmo tempo, ajudar a analisar o desempenho geral da arquitetura da rede por meio de algumas métricas como precisão (pureza) e recall (completeza). Selecionando o modelo cuja fold obteve as melhores métricas, efetuamos a classificação dos dois modelos em 164314 objetos do DR3 dividindo o catalogo final em um corte na magnitude da banda r. Como o treino foi feito utilizando galáxias de $mag_r < 17$, a melhor precisão é atingida classificando galáxias que estejam nesse mesmo intervalo. Na figura (5.12) mostro algumas informações em respeito ao catalogo final.

Catalogo	Seleção	Nº elementos	Fração
r_petro < 17	all	46763	-
	all_reliable	32073	0,69
	blue	11203	0,35
	red	8420	0,26
	Amb	452	0,01
	Spiral	18940	0,59
	Sred	2255	0,12
	Elliptical	12681	0,4
	Eblue	789	0,06
17 < r_petro < 18	all	114872	-
	all_reliable	22415	0,2
	blue	11616	0,52
	red	4357	0,19
	Amb	281	0,01
	Spiral	18868	0,84
	Sred	2254	0,12
	Elliptical	3266	0,15
	Eblue	266	0,08

Figura 5.12 – Nesta figura mostro informações gerais do catálogo final com as classificações efetuadas pelos modelos.

Capítulo 6

Considerações Finais e Trabalhos Futuros

A área de inteligencia artificial é bem ampla e vem ganhando cada vez mais espaço nesses últimos anos. Ainda há muito o que se estudar no que se refere às capacidades das redes neurais e a forma de utilizá-las. Hoje contamos com arquiteturas bem conhecidas para a resolução de determinados problemas como a EfficientNet utilizada neste trabalho [96], mas caso o leitor queira se aventurar em desenvolver uma arquitetura para lidar com algum problema específico irá se deparar com um longo processo que envolve muita tentativa e erro. Mesmo em arquiteturas já estabelecidas o processo de treino ainda envolve muita tentativa, erro e investigação dos possíveis motivos. Isso ocorre em boa parte porque não sabemos exatamente o que a rede neural aprendeu e como ela aprendeu um determinado conceito. Investigar mais a fundo o mecanismo por trás do aprendizado das redes, além do peso na introdução de camadas adicionas na arquitetura para a performance final do modelo é um dos meus objetivos para futuros trabalhos. Com esperança isso pode trazer mais entendimento do que se passa em cada pedaço da arquitetura, contribuindo para um entendimento mais profundo e que precise de menos tentativas para a concepção de uma rede. Tendo um bom conhecimento do que as diferentes camadas de convolução e pooling fazem, além dos neurônios ocultos, a grande pergunta é se não é possível criar uma arquitetura em que cada pedaço trate a imagem de uma maneira planejada, de forma que o treino possa ser melhor guiado à solução do problema.

Aliado a isso o DR4 do S-PLUS cria boas oportunidades para mais uma extensão deste trabalho. Na rede atual, com a existência dos objetos ambíguos, pudemos verificar que algumas galáxias lenticulares caíram nesse grupo. Esse acontecimento faz sentido devido à forma como as lenticulares são definidas no Diagrama de Hubble, mas é um evento que merece uma investigação mais minuciosa. Além disso, o modelo Reliable - Não Reliable também acabou separando objetos muito interessantes que nomeamos de ENR (Extra-

ordinary Not Reliable), veja figura (6.1) para alguns exemplos. Eles foram classificados como NR por uma série de motivos, dentre eles está o fato do objeto cobrir a maior parte do stamp, ultrapassar as dimensões do stamp, serem galáxias irregulares, entre outros. Mas o que chama a atenção é o fato do modelo ter atribuído na maioria dos casos, uma probabilidade baixíssima de ser Reliable e isso se dá por eles se distanciarem muito do que consideramos Reliable na etapa de treino.

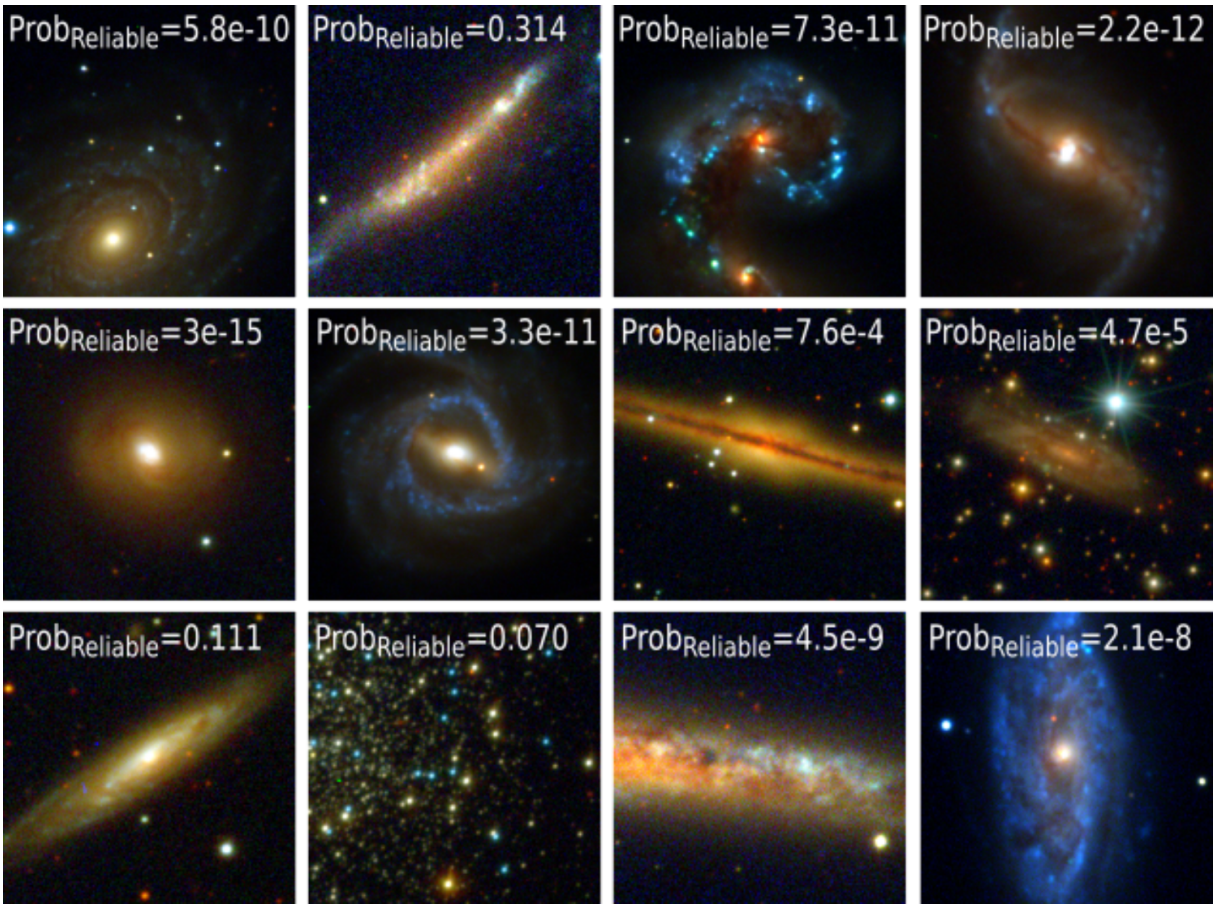

Figura 6.1 – Exemplo de galáxias que foram classificadas como Not-Reliable, mas que merecem certa atenção. Por esse motivo, nomeamos esses objetos de Extra-ordinary Not Reliable.

Conseguir selecionar as galáxias lenticulares das classificações ambíguas e os objetos ENR dos Not-Reliable também é um dos objetivos futuros para uma arquitetura que vai trabalhar no DR4. Em conclusão, visamos desenvolver uma arquitetura para o levantamento DR4 capaz de identificar galáxias lenticulares entre as classificadas ambíguas e ENR entre os Não Reliable stamps, gerando assim duas classes adicionais que o próprio modelo vai servir com suas respectivas probabilidades.

Referências

[1] Suraj Dhiwar, Kanak Saha, Avishai Dekel, Abhishek Paswan, Divya Pandey, Arianna Cortesi, and Mahadev Pandge. Witnessing the star formation quenching in l* ellipticals. Monthly Notices of the Royal Astronomical Society, 518(4):4943–4960, 2023.

[2] Thomas Southcliffe Ashton et al. The industrial revolution 1760-1830. OUP Catalogue, 1997.

[3] Phyllis M Deane. The first industrial revolution. Cambridge University Press, 1979.

[4] William A Baum. Photosensitive detectors. Annual Review of Astronomy and Astrophysics, 2(1):165–184, 1964.

[5] Arjun Dey, David J Schlegel, Dustin Lang, Robert Blum, Kaylan Burleigh, Xiaohui Fan, Joseph R Findlay, Doug Finkbeiner, David Herrera, Stéphanie Juneau, et al. Overview of the desi legacy imaging surveys. The Astronomical Journal, 157(5):168, 2019.

[6] Kyle S Dawson, Jean-Paul Kneib, Will J Percival, Shadab Alam, Franco D Albareti, Scott F Anderson, Eric Armengaud, Éric Aubourg, Stephen Bailey, Julian E Bautista, et al. The sdss-iv extended baryon oscillation spectroscopic survey: overview and early data. The Astronomical Journal, 151(2):44, 2016.

[7] Kyle B Westfall, Michele Cappellari, Matthew A Bershady, Kevin Bundy, Francesco Belfiore, Xihan Ji, David R Law, Adam Schaefer, Shravan Shetty, Christy A Tremonti, et al. The data analysis pipeline for the sdss-iv manga ifu galaxy survey: overview. The Astronomical Journal, 158(6):231, 2019.

[8] Dark Energy Survey Collaboration et al. The dark energy survey. arXiv preprint astro-ph/0510346, 2005.

[9] Dark Energy Survey Collaboration:, T Abbott, FB Abdalla, J Aleksić, S Allam, A Amara, D Bacon, E Balbinot, M Banerji, K Bechtol, et al. The dark

energy survey: more than dark energy–an overview. Monthly Notices of the Royal Astronomical Society, 460(2):1270–1299, 2016.

[10] Cláudia Mendes de Oliveira, T Ribeiro, William Schoenell, A Kanaan, RA Overzier, Alberto Molino, Laura Sampedro, P Coelho, Carlos Eduardo Barbosa, Arianna Cortesi, et al. The southern photometric local universe survey (s-plus): improved seds, morphologies, and redshifts with 12 optical filters. Monthly Notices of the Royal Astronomical Society, 489(1):241–267, 2019.

[11] Christopher J Conselice. The evolution of galaxy structure over cosmic time. Annual Review of Astronomy and Astrophysics, 52:291–337, 2014.

[12] Christopher J Conselice. The symmetry, color, and morphology of galaxies. Publications of the Astronomical Society of the Pacific, 109(741):1251, 1997.

[13] Shan Suthaharan. Big data classification: Problems and challenges in network intrusion prediction with machine learning. ACM SIGMETRICS Performance Evaluation Review, 41(4):70–73, 2014.

[14] Aized Amin Soofi and Arshad Awan. Classification techniques in machine learning: applications and issues. Journal of Basic & Applied Sciences, 13(1):459–465, 2017.

[15] R Benton Metcalf, MASSIMO Meneghetti, Camille Avestruz, Fabio Bellagamba, Clécio R Bom, Emmanuel Bertin, Rémi Cabanac, F Courbin, Andrew Davies, Etienne Decencière, et al. The strong gravitational lens finding challenge. Astronomy & Astrophysics, 625:A119, 2019.

[16] Humberto Farias, Daniel Ortiz, Guillermo Damke, M Jaque Arancibia, and Mauricio Solar. Mask galaxy: Morphological segmentation of galaxies. Astronomy and Computing, 33:100420, 2020.

[17] Ryan Hausen and Brant E Robertson. Morpheus: A deep learning framework for the pixel-level analysis of astronomical image data. The Astrophysical Journal Supplement Series, 248(1):20, 2020.

[18] Clecio R Bom, A Cortesi, G Lucatelli, Luciana Olivia Dias, P Schubert, GB Oliveira Schwarz, NM Cardoso, EVR Lima, C Mendes de Oliveira, L Sodre Jr, et al. Deep learning assessment of galaxy morphology in s-plus data release 1. Monthly Notices of the Royal Astronomical Society, 507(2):1937–1955, 2021.

[19] C. R. Bom, A. Cortesi, U. Ribeiro, L. O. Dias, K. Kelkar, A. V. Smith Castelli, L. Santana-Silva, V. Silva, T. S. Gonçalves, L. R. Abramo, E. V. R. Lima, F. Almeida-Fernandes, L. Espinosa, L. Li, M. L. Buzzo, C. Mendes de Oliveira,

Jr. Sodré, L., A. Alvarez-Candal, M. Grossi, E. Telles, S. Torres-Flores, S. V. Werner, A. Kanaan, T. Ribeiro, and W. Schoenell. An Extended Catalogue of galaxy morphology using Deep Learning in Southern Photometric Local Universe Survey Data Release 3. arXiv e-prints, page arXiv:2306.08684, June 2023.

[20] SS Motsa, P Dlamini, and M Khumalo. A new multistage spectral relaxation method for solving chaotic initial value systems. Nonlinear Dynamics, 72:265–283, 2013.

[21] MR Akbari, DD Ganji, A Majidian, and AR Ahmadi. Solving nonlinear differential equations of vanderpol, rayleigh and duffing by agm. Frontiers of Mechanical Engineering, 9:177–190, 2014.

[22] Francois Chollet. Deep learning with Python. Simon and Schuster, 2021.

[23] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. Neural computation, 22(12):3207–3220, 2010.

[24] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE signal processing magazine, 29(6):141–142, 2012.

[25] Alejandro Baldominos, Yago Saez, and Pedro Isasi. A survey of handwritten character recognition with mnist and emnist. Applied Sciences, 9(15):3169, 2019.

[26] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576, 2015.

[27] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4990–4998, 2017.

[28] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. arXiv preprint arXiv:1705.01088, 2017.

[29] Simon Haykin. Redes neurais: princípios e prática. Bookman Editora, 2001.

[30] Marvin Minsky and Seymour Papert. Perceptron: an introduction to computational geometry, 1969.

[31] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. arXiv preprint arXiv:1611.01491, 2016.

[32] Abien Fred Agarap. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375, 2018.

[33] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. Towards Data Sci, 6(12):310–316, 2017.

[34] Sridhar Narayan. The generalized sigmoid activation function: Competitive supervised learning. Information sciences, 99(1-2):69–82, 1997.

[35] Tomasz Szandała. Review and comparison of commonly used activation functions for deep neural networks. Bio-inspired neurocomputing, pages 203–224, 2021.

[36] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. arXiv preprint arXiv:1702.05659, 2017.

[37] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. IEEE Transactions on computational imaging, 3(1):47–57, 2016.

[38] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. nature, 323(6088):533–536, 1986.

[39] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In Neural networks for perception, pages 65–93. Elsevier, 1992.

[40] Randall J Erb. Introduction to backpropagation neural network computation. Pharmaceutical research, 10:165–170, 1993.

[41] Rafael Izbicki and Tiago Mendonça dos Santos. Aprendizado de máquina: uma abordagem estatística. Rafael Izbicki, 2020.

[42] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In Neural networks: Tricks of the trade, pages 9–50. Springer, 2002.

[43] Hong Hui Tan and King Hann Lim. Review of second-order optimization techniques in artificial neural networks backpropagation. In IOP conference series: materials science and engineering, volume 495, page 012003. IOP Publishing, 2019.

[44] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

[45] Usha Ruby and Vamsidhar Yendapalli. Binary cross entropy with deep learning technique for image classification. Int. J. Adv. Trends Comput. Sci. Eng, 9(10), 2020.

[46] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4):541–551, 1989.

[47] Ernst Kussul, Tetyana Baidyk, L Kasatkina, and V Lukovich. Rosenblatt perceptrons for handwritten digit recognition. In IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222), volume 2, pages 1516–1520. IEEE, 2001.

[48] Ernst Kussul and Tatiana Baidyk. Improved method of handwritten digit recognition tested on mnist database. Image and Vision Computing, 22(12):971–981, 2004.

[49] Charles Messier. Catalogue des nébuleuses et des amas d'étoiles (catalog of nebulae and star clusters). Connoissance des Temps ou des Mouvements Célestes, pages 227–267, 1781.

[50] John Louis Emil Dreyer. A new general catalogue of nebulæ and clusters of stars, being the catalogue of the late sir john fw herschel, bart, revised, corrected, and enlarged. Memoirs of the Royal Astronomical Society, 49:1, 1888.

[51] Edwin P Hubble. A spiral nebula as a stellar system, messier 31. The Astrophysical Journal, 69, 1929.

[52] Henrietta S Leavitt. 1777 variables in the magellanic clouds. Annals of Harvard College Observatory, 60(4):87–103, 1908.

[53] H. C. Arp. The Hertzsprung-Russell Diagram, pages 75–133. Springer Berlin Heidelberg, Berlin, Heidelberg, 1958.

[54] J. D. Fernie. The Structure of the Cepheid Instability Strip. , 354:295, May 1990.

[55] C. Chiosi, P. Wood, G. Bertelli, and A. Bressan. On the Instability Strip of the Cepheid Stars. , 387:320, March 1992.

[56] Henrietta S Leavitt and Edward C Pickering. Periods of 25 variable stars in the small magellanic cloud. Harvard College Observatory Circular, vol. 173, pp. 1-3, 173:1–3, 1912.

[57] Edwin P Hubble. Extragalactic nebulae. Astrophysical Journal, 64, 321-369 (1926), 64, 1926.

[58] Alan Dressler, Augustus Oemler Jr, Warrick J Couch, Ian Smail, Richard S Ellis, Amy Barger, Harvey Butcher, Bianca M Poggianti, and Ray M Sharples. Evolution since z= 0.5 of the morphology-density relation for clusters of galaxies. The Astrophysical Journal, 490(2):577, 1997.

[59] Giovanni Fasano, Bianca M Poggianti, Warrick J Couch, Daniela Bettoni, Per Kjærgaard, and Mariano Moles. The evolution of the galactic morphological types in clusters. The Astrophysical Journal, 542(2):673, 2000.

[60] V Desai, JJ Dalcanton, A Aragón-Salamanca, P Jablonka, B Poggianti, SM Gogarten, L Simard, B Milvang-Jensen, G Rudnick, D Zaritsky, et al. The morphological content of 10 ediscs clusters at 0.5¡ z¡ 0.8. The Astrophysical Journal, 660(2):1151, 2007.

[61] Samuel Boissier. Star Formation in Galaxies, pages 141–181. Springer Netherlands, Dordrecht, 2013.

[62] Ryosuke S. Asano, Tsutomu T. Takeuchi, Hiroyuki Hirashita, and Akio K. Inoue. Dust formation history of galaxies: A critical role of metallicity for the dust mass growth by accreting materials in the interstellar medium. Earth, Planets and Space, 65(3):213–222, March 2013.

[63] A Dictionary of Physics. Oxford University Press, 2009.

[64] Erik Holmberg. A photographic photometry of extragalactic nebulae. Meddelanden fran Lunds Astronomiska Observatorium Serie II, 136:1, 1958.

[65] Steven P Bamford, Robert C Nichol, Ivan K Baldry, Kate Land, Chris J Lintott, Kevin Schawinski, Anže Slosar, Alexander S Szalay, Daniel Thomas, Mehri Torki, et al. Galaxy zoo: the dependence of morphology and colour on environment. Monthly Notices of the Royal Astronomical Society, 393(4):1324–1352, 2009.

[66] Alan Dressler. The evolution of galaxies in clusters. Annual review of astronomy and astrophysics, 22(1):185–222, 1984.

[67] Percy L Gomez, Robert C Nichol, Christopher J Miller, Michael L Balogh, Tomotsugu Goto, Ann I Zabludoff, A Kathy Romer, Mariangela Bernardi, Ravi Sheth, Andrew M Hopkins, et al. Galaxy star formation as a function of environment in the early data release of the sloan digital sky survey. The Astrophysical Journal, 584(1):210, 2003.

[68] Michael R Blanton and John Moustakas. Physical properties and environments of nearby galaxies. Annual Review of Astronomy and Astrophysics, 47:159–210, 2009.

[69] Alar Toomre, BM Tinsley, and RB Larson. Evolution of galaxies and stellar populations, 1977.

[70] Jennifer M Lotz, Patrik Jonsson, TJ Cox, and Joel R Primack. Galaxy merger morphologies and time-scales from simulations of equal-mass gas-rich disc mergers. Monthly Notices of the Royal Astronomical Society, 391(3):1137–1162, 2008.

[71] Jennifer M Lotz, Patrik Jonsson, TJ Cox, Darren Croton, Joel R Primack, Rachel S Somerville, and Kyle Stewart. The major and minor galaxy merger rates at z¡ 1.5. The Astrophysical Journal, 742(2):103, 2011.

[72] Y. L. Jaffé, R. Smith, G. N. Candlish, B. M. Poggianti, Y.-K. Sheen, and M. A. W. Verheijen. BUDHIES II: a phase-space view of H I gas stripping and star formation quenching in cluster galaxies. , 448:1715–1728, April 2015.

[73] Lodovico Coccato, Yara L. Jaffé, Arianna Cortesi, Michael Merrifield, Evelyn Johnston, Bruno Rodríguez del Pino, Boris Haeussler, Ana L. Chies-Santos, Claudia L. Mendes de Oliveira, Yun-Kyeong Sheen, and Karín Menéndez-Delmestre. Formation of S0s in extreme environments I: clues from kinematics and stellar populations. , 492(2):2955–2972, February 2020.

[74] Evelyn J. Johnston, Alfonso Aragón-Salamanca, Amelia Fraser-McKelvie, Michael Merrifield, Boris Häußler, Lodovico Coccato, Yara Jaffé, Ariana Cortesi, Ana Chies-Santos, Bruno Rodríguez Del Pino, and Yun-Kyeong Sheen. Formation of S0s in extreme environments II: The star-formation histories of bulges, discs, and lenses. , 500(3):4193–4212, January 2021.

[75] Clár-Bríd Tohill, Steven Bamford, and Christopher Conselice. Exploring the Morphologies of High Redshift Galaxies with Machine Learning. arXiv e-prints, page arXiv:2302.11482, February 2023.

[76] Mauro Giavalisco, Mario Livio, Ralph C Bohlin, F Duccio Macchetto, and Theodore P Stecher. On the morphology of the hst faint galaxies. The Astronomical Journal, 112:369, 1996.

[77] Christopher J Conselice. The relationship between stellar light distributions of galaxies and their formation histories. The Astrophysical Journal Supplement Series, 147(1):1, 2003.

[78] Leonardo de Albernaz Ferreira and Fabricio Ferrari. The impact of redshift on galaxy morphometric classification: case studies for SDSS, DES, LSST and HST with morfometryka. Monthly Notices of the Royal Astronomical Society, 473(2):2701–2713, 09 2017.

[79] Clár-Bríd Tohill, Steven Bamford, Christopher Conselice, Leonardo Ferreira, Thomas Harvey, Nathan Adams, and Duncan Austin. A robust study of high-redshift galaxies: Unsupervised machine learning for characterising morphology with jwst up to z 8, 2023.

[80] Steve B. Howell. Handbook of CCD Astronomy. Cambridge Observing Handbooks for Research Astronomers. Cambridge University Press, 2 edition, 2006.

[81] Arne A Henden and Ronald H Kaitchuck. Astronomical photometry. New York: Van Nostrand Reinhold, 1982.

[82] W Romanishin. An introduction to astronomical photometry using ccds, 2002.

[83] Chris Lintott, Kevin Schawinski, Steven Bamford, Anže Slosar, Kate Land, Daniel Thomas, Edd Edmondson, Karen Masters, Robert C Nichol, M Jordan Raddick, et al. Galaxy zoo 1: data release of morphological classifications for nearly 900 000 galaxies. Monthly Notices of the Royal Astronomical Society, 410(1):166–178, 2011.

[84] Iskra Strateva, Željko Ivezić, Gillian R. Knapp, Vijay K. Narayanan, Michael A. Strauss, James E. Gunn, Robert H. Lupton, David Schlegel, Neta A. Bahcall, Jon Brinkmann, Robert J. Brunner, Tamás Budavári, István Csabai, Francisco Javier Castander, Mamoru Doi, Masataka Fukugita, Zsuzsanna Győry, Masaru Hamabe, Greg Hennessy, Takashi Ichikawa, Peter Z. Kunszt, Don Q. Lamb, Timothy A. McKay, Sadanori Okamura, Judith Racusin, Maki Sekiguchi, Donald P. Schneider, Kazuhiro Shimasaku, and Donald York. Color Separation of Galaxy Types in the Sloan Digital Sky Survey Imaging Data. , 122(4):1861–1874, October 2001.

[85] Suraj Dhiwar, Kanak Saha, Avishai Dekel, Abhishek Paswan, Divya Pandey, Arianna Cortesi, and Mahadev Pandge. Witnessing the star-formation quenching in L_* ellipticals. , November 2022.

[86] Craig Chester and Morton S. Roberts. Properties of Galaxies: color-magnitude diagram. , 69:635, October 1964.

[87] Eric F. Bell, Daniel H. McIntosh, Neal Katz, and Martin D. Weinberg. The Optical and Near-Infrared Properties of Galaxies. I. Luminosity and Stellar Mass Functions. , 149(2):289–312, December 2003.

[88] Chris J Lintott, Kevin Schawinski, Anže Slosar, Kate Land, Steven Bamford, Daniel Thomas, M Jordan Raddick, Robert C Nichol, Alex Szalay, Dan Andreescu, et al. Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey. Monthly Notices of the Royal Astronomical Society, 389(3):1179–1189, 2008.

[89] Stefano Zibetti, Brice Ménard, Daniel B. Nestor, Anna M. Quider, Sandhya M. Rao, and David A. Turnshek. Optical Properties and Spatial Distribution of Mg II Absorbers from SDSS Image Stacking. , 658(1):161–184, March 2007.

[90] R. J. Smethurst, C. J. Lintott, B. D. Simmons, K. Schawinski, P. J. Marshall, S. Bamford, L. Fortson, S. Kaviraj, K. L. Masters, T. Melvin, R. C. Nichol, R. A. Skibba, and K. W. Willett. Galaxy Zoo: evidence for diverse star formation histories through the green valley. , 450(1):435–453, June 2015.

[91] JL Sérsic. Influence of the atmospheric and instrumental dispersion on the brightness distribution in a galaxy. Boletin de la Asociacion Argentina de Astronomia La Plata Argentina, 6:41–43, 1963.

[92] Chien Y Peng, Luis C Ho, Chris D Impey, and Hans-Walter Rix. Detailed structural decomposition of galaxy images. The Astronomical Journal, 124(1):266, 2002.

[93] Luc Simard, J Trevor Mendel, David R Patton, Sara L Ellison, and Alan W McConnachie. A catalog of bulge+ disk decompositions and updated photometry for 1.12 million galaxies in the sloan digital sky survey. The Astrophysical Journal Supplement Series, 196(1):11, 2011.

[94] PE Freeman, R Izbicki, AB Lee, JA Newman, CJ Conselice, AM Koekemoer, JM Lotz, and M Mozena. New image statistics for detecting disturbed galaxy morphologies at high redshift. Monthly Notices of the Royal Astronomical Society, 434(1):282–295, 2013.

[95] Jakub Olczak, Niklas Fahlberg, Atsuto Maki, Ali Sharif Razavian, Anthony Jilert, André Stark, Olof Sköldenberg, and Max Gordon. Artificial intelligence for analyzing orthopedic trauma radiographs: deep learning algorithms—are they on par with humans for diagnosing fractures? Acta orthopaedica, 88(6):581–586, 2017.

[96] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning, pages 6105–6114. PMLR, 2019.

[97] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[98] Jose García Moreno-Torres, José A Sáez, and Francisco Herrera. Study on the impact of partition-induced dataset shift on k-fold cross-validation. IEEE Transactions on Neural Networks and Learning Systems, 23(8):1304–1312, 2012.

[99] TensorFlow Developers. Tensorflow. Zenodo, 2022.

[100] Bo Pang, Erik Nijkamp, and Ying Nian Wu. Deep learning with tensorflow: A review. Journal of Educational and Behavioral Statistics, 45(2):227–248, 2020.

[101] D Saumon and Mark S Marley. The evolution of l and t dwarfs in color-magnitude diagrams. The Astrophysical Journal, 689(2):1327, 2008.

[102] Oliver Kramer and Oliver Kramer. K-nearest neighbors. Dimensionality reduction with unsupervised nearest neighbors, pages 13–23, 2013.