



CBPF – Centro Brasileiro de Pesquisas Físicas

Dissertação de Mestrado

Detecção de sistemas de Lentes Gravitacionais com
Deep Learning:
Aplicação em regime de poucos dados

Kayque Teles

Orientador
Dr. Clécio Roque De Bom

Rio de Janeiro, RJ
2023

Kayque Teles

**Detecção de sistemas de Lentes Gravitacionais com Deep
Learning:
Aplicação em regime de poucos dados**

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

Rio de Janeiro, RJ
2023

“DETECÇÃO DE SISTEMAS DE LENTES GRAVITACIONAIS COM DEEP-
LEARNING”

KAYQUE DA SILVA TELES

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:


Clécio Roque de Bom – Orientador/CBPF



Documento assinado digitalmente
CRISTINA FURLANETTO
Data: 26/01/2023 18:07:51-0300
Verifique em <https://verificador.iti.br>

Cristina Furlanetto – UFRGS



Bruno Azevedo Lemos Moraes - CBPF

Rio de Janeiro, 25 de janeiro de 2023.

Teles, Kayque da Silva
Detecção de sistemas de Lentes Gravitacionais com Deep Learning:
Aplicação em regime de poucos dados/ Kayque Teles. - 2023
96 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 meu pai (*in memoriam*).

AGRADECIMENTOS

Primeiro agradeço profundamente ao meu orientador Clécio e o coorientador Martín por todo o conhecimento compartilhado, confiança, paciência e auxílio no desenvolvimento deste trabalho. Tenho bastante orgulho das escolhas de orientação que fiz e fico honrado por ter sido amparado por vocês até o final deste processo.

Em seguida a CAPES, pelo financiamento deste trabalho, sem o qual a sua execução assim como a minha permanência no mestrado não teria sido possível.

Entre os amigos e colegas do mestrado que quero agradecer, duas menções especiais são Luciana e Renan. Vocês foram o meu porto seguro, sem o apoio e suporte de vocês, estou certo que esta trajetória também não teria sido possível. Vocês são incríveis!

Além destes, quero também agradecer a Elton, Nina, Jesus e João pelo auxílio prestado com as disciplinas e a Manuel e Patrick (*in memoriam*) pelo auxílio com a programação.

Fora da pós-graduação, agradeço imensamente a Adrielle, Patrícia, Marcos, Jean, Eduardo, Ivan, Roberta, Kayo, Felipe, Danilo, Maycon e Magno pelo privilégio que me deram de se fazer presentes em contextos necessários para desestressar. Graças aos momentos de diversão que tivemos, pude encontrar a disposição necessária para prosseguir com o trabalho durante a pandemia e o posterior.

Agradeço à minha mãe Mirian, meus irmãos Ezequiel e Micael por acreditarem em mim e me fornecerem indispensável suporte para terminar o mestrado.

Por fim, mas não menos importante, agradeço profundamente a pessoa que foi maior inspiração e incentivo ao longo de toda esta trajetória: meu namorado Roberto. A presença dele tornou todas as dificuldades que tive durante o período remoto muito mais suportáveis. A ideia de prosseguir com o projeto no intuito de podermos construir um futuro juntos foi indispensável para manter o ânimo. Você que acompanhou todo este processo remoto, então neste trabalho também tem muito de você.

”Quando acordei de manhã eu sabia quem eu era,
mas acho que já mudei muitas vezes desde então”
- Alice, de Alice no País das Maravilhas

RESUMO

O presente trabalho propõe o uso de redes neurais convolucionais para o estudo e detecção de arcos gravitacionais. O objetivo é analisar técnicas e métodos de aprendizagem profunda para classificação de dados em imagens astronômicas simuladas usando redes neurais, treinadas nos dados de duas edições de um do *Gravitational Lens Finding Challenge*. O intuito é encontrar bons métodos de aprendizagem de máquina que possam auxiliar na detecção do efeito forte de lente em suas várias formas, sobretudo em casos em que o número de dados de treinamento é reduzido. Esta preocupação se faz necessária dado ao número de registros de ocorrência do lenteamento gravitacional ainda estar em ordem de centenas. Inicialmente, intui-se observar a eficiência de uma rede neural do tipo residual (ResNet) na classificação de arcos, conforme o tamanho do conjunto de treino diminui, utilizando a base de dados gerados para a primeira edição do Gravitational Lens Finding Challenge. Dentre os 20.000 objetos disponibilizados para treino, o limite de poucos dados obtido é de que seriam necessários por volta de 100 candidatos de lente para que a arquitetura construída consiga obter uma AUC por volta de 0,8. Constatamos que, para este problema de classificação nesta base de dados, a aumento de dados se mostrou pouco eficiente na melhora da AUC, enquanto o pré-carregamento de pesos se mostrou bem eficaz. Em seguida, serão explicitadas algumas das estratégias adotadas a fim de tentar encontrar o limite de poucos dados para o catálogo da segunda edição do Gravitational Lens Finding Challenge, e aqui entre os 100.000 dados totais de treino, a exigência obtida foi de um conjunto de treino com 10.000 objetos. Por fim, constatamos ainda que a variação do critério da tabela verdade nos dados da segunda edição tem impacto significativo na performance do modelo. Espera-se que as arquiteturas construídas e as estratégias adotadas possam ser úteis em trabalhos futuros de busca de efeito forte de lente, e que os resultados aqui encontrados possam ser cada vez mais refinados.

Palavras Chaves: Lente Gravitational, Efeito Forte de Lente, Aprendizado de Máquina, Detecção, Gravitational Lens Finding Challenge, EfficientNet B2, Inteligência Artificial, ResNet50.

ABSTRACT

The present work proposes the use of convolutional neural networks (CNN) on the study and detection of gravitational arcs. The main objective is to analyze techniques and methods of deep learning on the classification of astronomical image data simulated using neural networks, once trained with data from two editions of the Gravitational Lens Finding Challenge. We expect to find good machine learning methods who could help the detection of strong lensing phenomena in many ways, mainly for cases in which the number of training data is low. This worry makes itself necessary due the number of records of strong lensing phenomena still be a few hundred. At first, we aim to observe the efficiency of a residual neural network on arc classifying, as the training data size shrinks, using the generated database for the first edition of Gravitational Lens Finding Challenge. Among the given 20.000 objects for training, the obtained result is that it should be required around 100 lens candidates in order to the built architecture achieve a AUC of 0,8. We found that, for this problem used to classify this database, data augmentation had not a positive impact to improve the AUC, while the use of preloaded weights has shown to be very effective. Then, some strategies were adopted towards finding the minimal possible value on the data from the second edition of Gravitational Lens Finding Challenge, and we found that around 10.000 training samples would be necessary, from the whole dataset of 100.000 samples. Last, we found that adjusting the criteria which defines the truth table has a significant impact on the model's performance. It is expected the built architectures could be proven useful for future works in the search of strong lensing effect, and that these results could be even more refined.

Key-Words: Gravitational Lens, Strong Lensing Effect, Machine Learning, Detection, Gravitational Lens Finding Challenge, EfficientNetB2, Artificial Intelligence, ResNet50.

LISTA DE ILUSTRAÇÕES

Figura 1.1 – Imagem do telescópio espacial Hubble conhecida como <i>Hubble Ultra Deep Field</i> (HUDF), mostrando algumas das galáxias mais distantes já imageadas no ótico. Apesar de haver mais de dez mil galáxias na imagem, a região do céu equivalente a foto é nada mais que cerca de um décimo do diâmetro da Lua observado a partir da Terra. Fonte: https://esahubble.org/images/heic0611b/ Créditos: NASA, ESA, e S. Beckwith (STScI) e o time HUDF	2
Figura 1.2 – Imagem feita pelo telescópio espacial Hubble do aglomerado de galáxias Abell 370, localizado à cerca de 4 bilhões de anos-luz. Linhas em formas de arcos de luz branca e azul podem ser discernidos como a luz de galáxias mais distantes atrás do aglomerado, distorcidas pelo efeito de lente gravitacional. Apesar da luz dessas galáxias distantes ser fraca demais para ser vista diretamente pelo Hubble, o aglomerado Abell 370 atua como uma lente magnificadora - um telescópio gravitacional - que permite ver estas galáxias mais distantes. Uma grande mancha contínua à noroeste próxima ao centro da imagem é conhecida como "O dragão", consistindo de uma sobreposição de várias imagens múltiplas de uma galáxia espiral longínqua sendo lenteada. Créditos: NASA, ESA, e J. Lotz e o time HFF (STScI).	3
Figura 2.1 – Desvio da luz de uma fonte S devido as proximidades de seu caminho óptico com o campo gravitacional gerado pela massa M . As escalas dos ângulos estão exageradas pelo teor didático.	10
Figura 3.1 – Um exemplo de rede neural densamente.	17
Figura 3.2 – Esquema das operações que ocorrem em cada neurônio de uma rede neural.	18
Figura 3.3 – Ilustração da convolução que um núcleo (ou filtro) K realiza por um tensor de imagem I , resultando no mapa de características $I * K$	22
Figura 3.4 – Esquema básico da arquitetura de uma RNC, constituída por uma camada de imagens de entrada, camadas convolucionais, de agrupamento e completamente conectadas.	23
Figura 3.5 – Representação esquemática da redução de dimensionalidade de um mapa de características através do agrupamento.	24

Figura 3.6 – Uma matrix de confusão 2x2 para um problema de classificação binária e suas possibilidades.	27
Figura 3.7 – Uma curva ROC, construída por uma série de pontos no espaço formado pelas TFPs com as TPVs.	29
Figura 3.8 – Representação esquemática de um bloco de conexão de atalho de uma ResNet. Os dados entram no ramo à esquerda, passam por uma camada convolucional de pesos e normalização do lote (<i>Conv2D</i> e <i>BatchNormalization</i> , respectivamente) seguida de uma ativação ReLU. Em seguida, passam por outra camada convolucional com normalização do lote e, antes de passarem pela próxima ativação ReLU, são somadas com uma cópia dos mesmos dados de duas convoluções atrás.	31
Figura 3.9 – Representação esquemática de um bloco de conexão de atalho de uma ResNet50. Diferentemente da ResNet de 34 camadas, aqui a conexão de atalho salta 3 camadas ao invés de duas, que por sua vez estão organizadas num design de gargalo.	32
Figura 3.10–Bloco Convolucional de Gargalo Invertido da EfficientNet. A principal característica é de que ele inverte as conexões de atalho de modo que, ao invés de conectar saídas de blocos convolucionais 1x1, ele conecta os próprios blocos convolucionais centrais.	35
Figura 3.11–Organização em blocos da EfficientNet B2. Há um bloco de entrada seguido de uma camada convolucional comum, após o qual há os blocos de Gargalo Móvel Invertido até a última camada.	36
Figura 4.1 – 25 exemplos de imagens combinadas das bandas R, G, e U da primeira edição do GLFC. Fica evidente o ruído e a poluição visual que se espera de imagens obtidas a partir de um telescópio com a poluição atmosférica.	42
Figura 5.1 – Gráfico de Acurácia da rede ResNet50 em toda a base de dados.	47
Figura 5.2 – Gráfico da Função Custo da rede ResNet50 em toda a base de dados.	48
Figura 5.3 – Gráfico da curva ROC da rede ResNet50 em toda a base de dados.	48
Figura 5.4 – Distribuição de AUCs de acordo o tamanho do conjunto de treinamento.	51
Figura 5.5 – Esquema representativo do uso de diferentes conjuntos de treino e de validação durante o treinamento de uma rede neural utilizando validação cruzada de <i>k folds</i> com 4 <i>folds</i>	53
Figura 5.6 – Distribuição de AUCs de acordo o tamanho do conjunto de treinamento e respectivas barras de erro.	54
Figura 5.7 – Comparação entre as AUCs com e sem pré-carregamento de pesos.	56
Figura 5.8 – Comparação entre as AUCs com e sem aumento de dados.	58
Figura 5.9 – Comparação de desempenhos das redes ResNet50 e EfficientNetB2.	59
Figura 5.10–Comparação de desempenhos da rede ResNet50, EfficientNetB2 e do ensemble entre elas.	61

Figura 5.11–Comparação de desempenhos de duas técnicas de ensemble pré-carregadas e do ensemble por média aritmética com o das redes individuais.	62
Figura 5.12–Dois exemplos de imagens em suas bandas H, J, Y, VIS e o resultado da combinação das três bandas HJY	65
Figura 5.13–Histograma de pixels das imagens nas bandas H, J, Y e VIS	67
Figura 5.14–Histograma final de pixels das imagens nas bandas H, J, Y e VIS	68
Figura 5.15–Acurácia do Treino da primeira pasta das redes EfficientNet B2 (à esquerda) e ResNet50 (à direita) no Ensemble de Redes sobre a segunda base de dados do GLFC.	69
Figura 5.16–Pontos da Função Custo do Treino da primeira pasta das redes EfficientNet B2 (à esquerda) e ResNet50 (à direita) no Ensemble de Redes sobre a segunda base de dados do GLFC.	70
Figura 5.17– Curva ROC e AUC de cada modelo ao final da primeira pasta das redes EfficientNet B2 (à esquerda) e ResNet50 (à direita) no Ensemble de Redes sobre a segunda base de dados do GLFC.	71
Figura 5.18– Curva ROC e AUC de cada modelo ao final da primeira pasta das redes EfficientNet B2 (à esquerda) e ResNet50 (à direita) no Ensemble de Redes sobre a segunda base de dados do GLFC, usando a banda VIS.	71
Figura 5.19–Desempenho da rede EfficientNet B2 nos dados da segunda edição do GLFC.	73
Figura 5.20–Desempenho das redes EfficientNet B2 concatenadas nos dados da segunda edição do GLFC, para diversos tamanhos de conjuntos de treino.	74

LISTA DE TABELAS

Tabela 1 – Comparação entre AUCs variando o critério de Lente Gravitacional . . . 76

LISTA DE ABREVIATURAS E SIGLAS

CBPF	Centro Brasileiro de Pesquisas Físicas
IA	Inteligência Artificial
AM	Aprendizado de Máquina
RNA	Rede Neural Artificial
RNC	Convolutional Neural Network
AUC	Área Sob a Curva
ROC	Característica de Operação do Receptor
LG	Lenteamento Gravitacional
AP	Aprendizagem Profunda
MC	Mapa de Características
ILSVRC	<i>ImageNet Large Scale Visual Recognition Challenge</i>
GLFC	<i>Gravitational Lens Finding Challenge</i>
ESA	<i>European Space Agency</i>
MECO	Método de Escalonamento Composto
SDSS	<i>Sloan Digital Sky Survey</i>
SLACS	<i>Sloan Lens ACS Survey</i>
DES	<i>Dark Energy Survey</i>
CFHTLS	<i>Canada-France-Hawaii Telescope Lens Survey</i>
DESI	<i>Dark Energy Spectroscopic Interferometer</i>

KiDS	<i>Kilo Degree Survey</i>
HST	<i>Hubble Space Telescope</i>
HDF	<i>Hubble Deep Field</i>
HUDF	<i>Hubble Ultra Deep Field</i>

SUMÁRIO

Lista de ilustrações	xiii
Lista de tabelas	xvii
Sumário	xxi
1	INTRODUÇÃO 1
2	FÍSICA DO LENTEAMENTO GRAVITACIONAL 7
2.1	Breve Histórico 7
2.2	Teoria do Lenteamento Gravitacional 9
2.2.1	Deflexão da Luz por Efeito de Lente e Fonte Pontuais 10
2.2.2	Deflexão da Luz por Lentes com Simetria Axial 12
2.2.3	Deflexão da Luz com Fontes Circulares 13
3	REDES NEURAIS ARTIFICIAIS 15
3.1	História e Introdução 15
3.2	Redes Neurais Artificiais (RNA) 17
3.2.1	Aspectos Gerais de uma RNA 17
3.2.2	Função de Ativação 18
3.2.3	Treinamento da RNA 19
3.2.3.1	Retropropagação 19
3.3	Rede Neural Convolucional (RNC) 21
3.3.1	Camadas Convolucionais 22
3.3.2	Esquema Geral de uma RNC 23
3.4	Métricas de Avaliação de Qualidade do Desempenho de uma Rede Neural 25
3.4.1	Acurácia 25
3.4.2	Função Custo 25
3.4.3	Matriz de Confusão 26
3.4.4	Curva Característica de Operação do Receptor (ROC) 28
3.4.5	Área sob a Curva (AUC) 29
3.5	Arquiteturas Utilizadas 30

3.5.1	ResNet50	30
3.5.2	EfficientNet B2	33
4	DETECÇÃO AUTOMATIZADA DE LENTES GRAVITACIONAIS COM REDES NEURAS ARTIFICIAIS	37
4.1	Histórico da Busca automatizada de Efeito Forte de Lente	37
4.2	As bases de dados do Gravitational Lens Finding Challenge (GLFC)	40
5	RESULTADOS	45
5.1	Problematização	45
5.2	Desempenho da ResNet50 nos dados da primeira edição do GLFC	46
5.2.1	A investigação de um limite de poucos dados	49
5.2.2	Multiplos conjuntos de treino/teste	52
5.2.3	Avaliação da influência do pré-carregamento de pesos	55
5.2.4	Avaliação da influência da aumento de dados	57
5.2.5	Uso do ensemble de redes nos dados da primeira edição do GLFC	58
5.3	Classificação com dados da Segunda Edição do GLFC	63
5.3.1	Pré-processamento dos dados da segunda edição do GLFC	64
5.3.2	Desempenho do Ensemble de redes nos dados da Segunda Edição do GLFC	68
5.3.3	Treino usando a concatenação de duas EfficientNet B2	72
5.3.4	Desempenho das Efficient Net B2 concatenadas na base de dados do GLFC variando o critério da tabela-verdade	75
5.4	Conclusões e Discussão	76
6	CONSIDERAÇÕES FINAIS	79
	REFERÊNCIAS	81

CAPÍTULO 1

INTRODUÇÃO

A busca pela compreensão do Universo pelo ser humano remonta de tradições orais de povos da antiguidade. Esse ímpeto levou a produções culturais e uma visão mítica acerca do Cosmos, através das observações do céu a olho nu. O desenvolvimento do método científico despertou a possibilidade de uma concepção científica do Universo, cuja iniciativa mais revolucionária data do século XV d.C por Galileu Galilei, com a invenção do telescópio [1]. O telescópio permitiu observar muito além do que o olho poderia enxergar, revelando corpos celestes dos quais não se tinha registro algum antes. Inicialmente foram descobertas quatro luas de Júpiter, retirando o ser humano do centro do universo e atribuindo ao Sol o caráter de apenas mais uma estrela entre todos os pontos do céu. Logo até mesmo a Via Láctea perdeu seu lugar especial, depois de ser concebida a ideia de que o universo era um zoológico de galáxias, bem mais vasto do que era possível observar, confirmada por Hubble no início do século XX. Vieram as inovações dos telescópios espaciais, como o telescópio espacial Hubble, que originou uma das primeiras imagens em alta definição do universo - a figura do *Hubble Ultra Deep Field* 1.1, revelando uma parte pequena do céu com galáxias jovens, algumas das mais distantes já observadas no ótico.

Muitos mapeamentos do céu foram feitos nos anos posteriores ao lançamento do Hubble, alavancados pela tecnologia dos instrumentos de observação disponíveis (para uma visão mais aprofundada, ver [2]). No entanto, a maior inovação veio com os levantamentos de imagens digitais. Entre alguns destes levantamentos estão

- O *Sloan Digital Sky Survey* (SDSS) [3, 4], que até então conta com 17 lançamentos de imagens de um telescópio em solo terrestre (chamados comumente por *ground-based*), em diversas bandas e filtros, cobrindo uma vasta área do céu do hemisfério norte [5];
- O *Kilo Degree Survey* (KiDS) [6], um levantamento a partir do telescópio terrestre

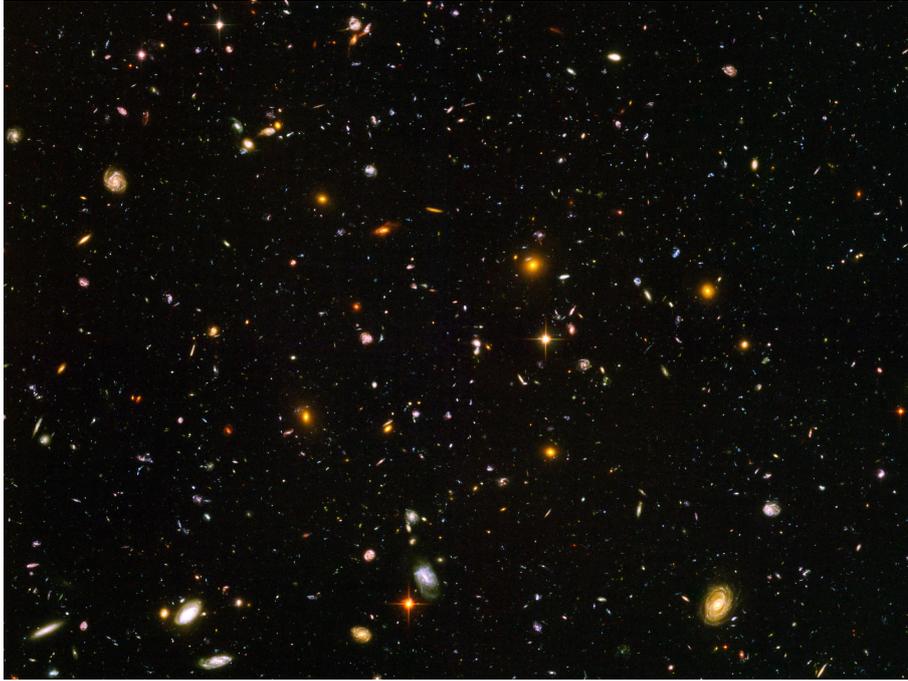


Figura 1.1 – Imagem do telescópio espacial Hubble conhecida como *Hubble Ultra Deep Field* (HUDF). Apesar de haver mais de dez mil galáxias na imagem, a região do céu equivalente a foto é nada mais que cerca de um décimo do diâmetro da Lua observado a partir da Terra.

Fonte: <https://esahubble.org/images/heic0611b/>

Créditos: NASA, ESA, e S. Beckwith (STScI) e o time HUDF

VLT (*Very Large Telescope*), situado no norte do Chile, que já teve 4 lançamentos [7];

- O *Dark Energy Survey* (DES) [8, 9] observou de 2013 a 2019, também *ground-based*, que já teve dois lançamentos de bases de dados [10] que cobrem grande parte do hemisfério sul.

Captando imagens em diversas bandas, o resultante de todos estes levantamentos outros que ainda estão por vir, como os do telescópio LSST (*Vera Rubin Observatory's Legacy Survey of Space and Time*[11]) e do telescópio Euclid [12] são petabytes de dados observacionais.

O surgimento de novos instrumentos de observação do céu ao longo do séc. XX e o imageamento digital permitiram uma miríade de descobertas que, mais tarde fomentaram os levantamentos de dados de grande área no início do século XXI. Esses dois grandes movimentos nos permitiram enxergar e construir imagens mais profundas e detalhadas do céu e também a observar fenômenos raros ou até então apenas hipotetizados. Dentre eles, o fenômeno de Lenteamento Gravitacional ou Efeito de Lente Gravitacional, que consiste na deflexão da luz de um objeto longínquo por outro mais próximo [13, 14]. Este efeito se deve ao desvio dos raios de luz ao passar próximo a um campo gravitacional não-homogêneo, percorrendo diferentes geodésicas em direção ao observador. Nesta perspectiva, comumente refere-se ao objeto que gera o campo gravitacional como “lente” e ao objeto mais distante cujos raios de luz são defletidos como “fonte” [13]. O resultado

desse efeito pode ser o alongamento da imagem da fonte ao longo do plano da lente, até a sua distorção em forma de arcos que podem resultar em um anel completo. Além disto, surgirão múltiplas imagens do mesmo objeto, que podem se apresentar como vários arcos. Esse efeito gera, ainda, diferenças temporais entre as imagens, que surgem em diferentes momentos.

O motivo pelo qual esse fenômeno atua como um “telescópio gravitacional” se deve por conta de as imagens das fontes serem bastante magnificadas, possibilitando tornar visível a um telescópio objetos que seriam, de outra forma, invisíveis. Tudo dependerá das dimensões da fonte e perfil de densidade de massa da lente e posições relativas de lente, fonte e observador. Estas distorções já foram observadas, como por exemplo, na imagem do aglomerado Abell 370 na figura 1.2, obtida pelo telescópio Hubble.

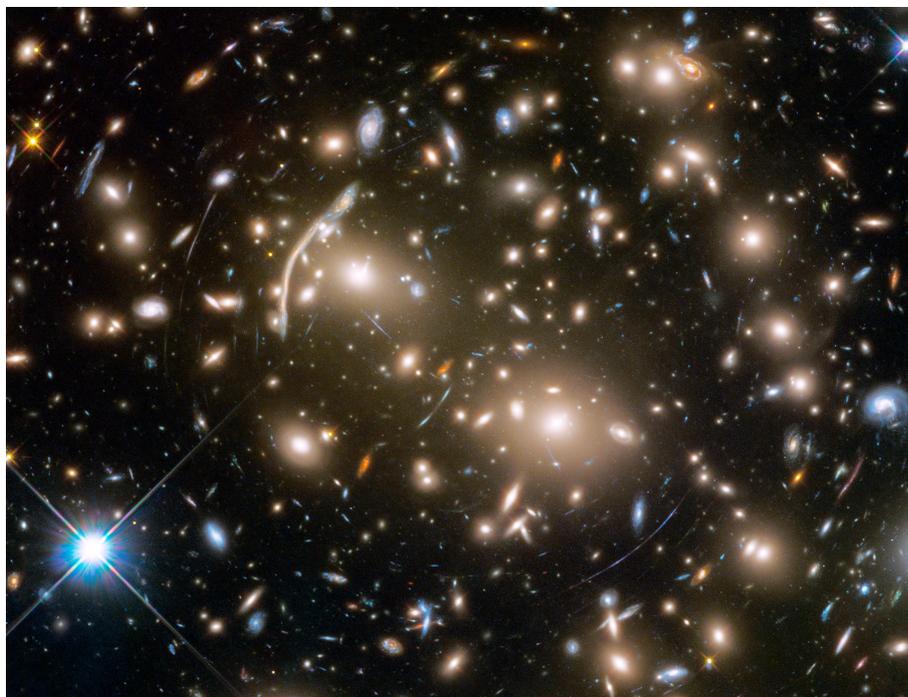


Figura 1.2 – Imagem feita pelo telescópio espacial Hubble do aglomerado de galáxias Abell 370, localizado à cerca de 4 bilhões de anos-luz. Linhas em formas de arcos de luz branca e azul podem ser discernidos como a luz de galáxias mais distantes atrás do aglomerado, distorcidas pelo efeito de lente gravitacional. Apesar da luz dessas galáxias distantes ser fraca demais para ser vista diretamente pelo Hubble, o aglomerado Abell 370 atua como uma lente magnificadora - um telescópio gravitacional - que permite ver estas galáxias mais distantes. Uma grande mancha contínua à noroeste próxima ao centro da imagem é conhecida como “O dragão”, consistindo de uma sobreposição de várias imagens múltiplas de uma galáxia espiral longínqua sendo lenteada.
Créditos: NASA, ESA, e J. Lotz e o time HFF (STScI).

Estas distorções são eventos raros, no entanto, o uso de diferentes técnicas automatizadas e inspeção visual resultaram em centenas de candidatos a eventos de lenteamento gravitacional [15, 16, 17, 18, 19]. O catálogo do *Master Lens Database*¹, por exemplo, que se propõe a registrar e catalogar os eventos de lenteamento gravitacional conhecidos, contém 675 eventos de lenteamento gravitacional catalogados, entre arcos, anéis e imagens múltiplas [20]. Enquanto telescópio gravitacional, o efeito de lentes já se provou ser

¹ <http://admin.masterlens.org/index.php>

bastante eficiente na observação de galáxias a valores de desvio pro vermelho maior que 6 [21, 22, 23, 24]. Adicionalmente, também foram observadas imagens de supernovas como a PS1-10afx [25] e a supernova Refstal [26].

Uma das principais aplicações para o estudo do efeito forte de lente é a reconstrução de perfis de densidade de massa das lentes [27] e a inferência de parâmetros cosmológicos. Sobretudo os casos em que é possível medir diferenças temporais, *time-delay*, entre as imagens, constituem de boas oportunidades para obter a constante de Hubble H_0 [28] e o parâmetro de densidade Ω_m , envolvida na expansão do universo [29]. Este feito foi primeiro idealizado por Refsdal em 1964 [30].

O lenteamento gravitacional também tem um papel importante na detecção da matéria escura. Keeton e Moustakas [31] mostraram que algumas subestruturas de matéria escura afetam a diferença de tempo de surgimento de imagens múltiplas de um objeto, de modo que seja possível reconstruir o perfil de densidade dessas subestruturas [31]. Outras detecções de matéria escura através da análise do seu efeito de lenteamento gravitacional também foram realizadas [32, 33].

Apesar da acurácia da da inspeção visual realizada em alguns trabalhos, o volume de dados de imagem geradas por um levantamento torna essa tarefa proibitiva. Neste contexto, a automatização do processo de detecção e localização, através do aprendizado de máquina, surge como consequência natural do aumento da capacidade de processamento dos computadores. O trabalho que seria realizado por uma equipe de cientistas pode ser reduzido a um conjunto de GPUs (*Graphics Processing Unit*), ganhando agilidade no processo de detecção. Alguns contextos de ocorrência do efeito forte de lente são evidentes a inspeção visual, sobretudo quando há grandes distorções, mas nem sempre é tão fácil identificar o fenômeno acontecendo em menores escalas, por exemplo. Uma vez que um algoritmo encontrar um possível candidato, este pode seguir para posterior inspeção visual e estudo de suas propriedades.

Desta forma, este trabalho busca oferecer uma contribuição no estudo de métodos do uso do aprendizado de máquina, no campo das redes neurais convolucionais, para a detecção do fenômeno do lenteamento gravitacional, com enfoque em duas bases de dados. Observando a raridade do fenômeno que ainda permanece, apesar de o número de registros já estar na casa das centenas, serão objetivadas estratégias que possam melhorar os resultados de classificação com poucos volumes de dados.

Esta dissertação está organizada da seguinte maneira. No segundo capítulo deste trabalho, apresentamos uma breve introdução teórica do fenômeno de lenteamento gravitacional e as distorções de imagens mais comuns. No terceiro, discutimos de forma geral as redes neurais artificiais, entrando no ramo das redes neurais convolucionais e as duas arquiteturas que foram utilizadas neste trabalho: ResNet50 e EfficientNet B2. No quarto, apresentamos um breve histórico da detecção de lentes gravitacionais, desde a inspeção visual até as redes neurais convolucionais; e também uma introdução aos detalhes técnicos

das bases de dados utilizadas. No quinto, descrevemos o processo investigativo adotado sobre as bases de dados, evidenciando as etapas de cada estudo feito e; no sexto, as considerações finais.

 CAPÍTULO 2

FÍSICA DO LENTEAMENTO GRAVITACIONAL

Neste capítulo, cabe realizar uma introdução ao fenômeno que inspirou a geração das imagens sobre as quais se debruçou o trabalho: o lenteamento gravitacional. Na primeira seção temos uma breve revisão histórica dos momentos mais importantes para a ciência do fenômeno de lente, culminando na sua descoberta. Em seguida, discutimos a teoria do lenteamento gravitacional abordando as soluções mais importantes para compreender a base das distorções mais comuns que podem ser encontradas.

2.1 Breve Histórico

A teoria da Gravitação Universal de Isaac Newton junto às leis de movimento revolucionaram o entendimento sobre o universo, publicadas em seu livro de 1686 *Philosophiæ Naturalis Principia Mathematica* (do latim, *Princípios Matemáticos da Filosofia Natural*). Em 1784, John Mitchell trouxe a ideia da luz como corpúsculos que são atraídos pela gravidade, ao estudar estrelas binárias. Após receber uma carta de Mitchell sobre este tema, Henry Cavendish calculou o ângulo de deflexão $\hat{\alpha}$ de um raio de luz ao passar por um campo gravitacional gerado por um objeto de massa M . Derivando as equações de movimento de uma partícula em velocidade c passando por um campo gravitacional produzido por uma massa pontual, temos

$$\hat{\alpha} = \frac{2GM}{c^2\xi}, \quad (2.1)$$

onde ξ é conhecido como *parâmetro de impacto*, constituindo a distância entre a massa M e o raio de luz no plano da lente. [34].

Uma novo entendimento da natureza da gravidade se deu com o advento da Teoria da Relatividade Geral em 1915 e 1917 por Einstein, em que foi obtido o ângulo de deflexão

nesta teoria como

$$\hat{\alpha}_E = \frac{4GM}{c^2\xi}; \quad (2.2)$$

em que o índice E denota o ângulo de deflexão de acordo com a teoria de Einstein. Nota-se que um dos ângulos de deflexão é múltiplo do outro, e essa divergência nas previsões permitiu testar a Relatividade Geral [34]. Uma oportunidade de verificar experimentalmente a validade de cada uma das teorias surgiu com a previsão de um eclipse solar em 1919. Conhecendo a posição e trajetória de estrelas do céu, bastaria medir o desvio sofrido pela luz de algumas estrelas próximas ao perímetro do Sol devido a proximidade deste, e verificar a qual das duas equações o ângulo de deflexão produzida pelo Sol correspondia. Duas expedições foram realizadas no intuito de medir esse desvio da luz em locais onde o eclipse seria total: a cidade de Sobral no estado do Ceará (Brasil), e a ilha de Príncipe (hoje, São Tomé e Príncipe) [35]. O modelo newtoniano previa um ângulo de reflexão de aproximadamente $0,875''$ [36], enquanto o modelo da Relatividade Geral previa o dobro. Ambos os times de cada expedição mediram para o desvio o valor de $1,61'' \pm 0,3''$, confirmando o resultado da Relatividade Geral [37].

Logo após, em 1924, O. Chwolson publicou uma pequena nota em um jornal científico acerca da possibilidade da produção de “estrelas falsas” devido o efeito de lenteamento gravitacional [38]. A partir de considerações das posições de um observador, lente e fonte que tivessem seus planos alinhados, poderia surgir uma estrutura de lenteamento em forma de anel. Essa estrutura ficou conhecida posteriormente como *anel de Einstein*. A ideia de que o lenteamento poderia gerar imagens múltiplas de um objeto também foi reforçada pelo próprio Einstein, em um artigo de 1936, onde compartilhava as equações da geração de imagens múltiplas através do lenteamento de uma estrela [39]. Para ele, no entanto, não haviam grandes chances de detectar diretamente este fenômeno por conta das escalas angulares necessárias para a sua observação serem muito pequenas para a época.

Já em 1937, Fritz Zwicky publica os resultados de suas considerações em que, caso os objetos envolvidos sejam não de escalas estelares mas, escalas galácticas, haveriam chances reais de se observar imagens múltiplas de uma galáxia através do lenteamento gravitacional [40]. De fato, algum tempo foi preciso para que o fenômeno pudesse ser observado, porém, em 1979, foi descoberto o que pôde ser confirmado como o primeiro sistema de imagens múltiplas por lenteamento: duas contrapartes ópticas do quasar Q0957+561 [41]. Não obstante, este quasar foi o primeiro objeto em que foi verificado não apenas a distorção de imagens mas, a diferença entre os tempos de ocorrência das imagens, de cerca de 420 dias [42].

A partir de então, o lenteamento gravitacional evoluiu para um dos mais importantes ramos da pesquisa em astrofísica e cosmologia. O fenômeno passou a ser detectado em diversos contextos, como nos levantamentos de imagens do céu. Foram descobertos arcos gravitacionais nos dados do SDSS (*Sloan Digital Sky Survey*) [43], CLASS (*Cosmic Lens All-Sky Survey*) [44], SQLS (*Sloan Digital Sky Survey for Quasar Lens Search*) [45], dentre

outros.

2.2 Teoria do Lenteamento Gravitacional

Conforme já abordado, é uma previsão da Relatividade Geral que a luz é defletida na presença de um campo gravitacional não-homogêneo. Para que seja possível visualizar este fenômeno, é preciso que haja um alinhamento ou quase alinhamento entre o objeto que gera o campo gravitacional que fará o lenteamento, o observador e a fonte de luz, cujos raios sofrerão deflexão ao atravessarem o campo gravitacional. Isto faz sentido porque cada raio de luz tem de percorrer diferentes geodésicas ao atravessar o campo gravitacional da lente, o que permite a formação de imagens múltiplas. É possível, como alternativa, estudar a distorção da luz através do princípio de Fermat, da óptica geométrica - segundo o qual, ondas de luz em uma dada frequência sempre percorrem o menor caminho entre dois pontos [46]. No entanto, o efeito de lente gravitacional é acromático e não depende, como no caso da óptica geométrica, da frequência da luz; e sim das dimensões da fonte e do perfil de densidade da lente, além de suas posições relativas. Os casos gerais serão abordados nas subseções seguintes.

Cabe ainda, antes de entrar na física e geometria das distorções, tratar dos diferentes regimes possíveis em que pode ocorrer o lenteamento. Dois tipos de critérios definem as possibilidades de distorção: quanto à intensidade e quanto à escala angular. Quanto às intensidades das distorções, eles podem ser divididos nos regimes de “forte” e “fraco”.

- Lenteamento Forte¹: este regime é caracterizado majoritariamente por grandes distorções nas imagens da fonte. Tais distorções podem aparecer na forma de imagens múltiplas, arcos ou anéis [47]. Um importante efeito que também pode ser verificado é um aumento do brilho nas imagens da fonte, de modo análogo à concentração de luz de uma lente convexa [48]. Isto possibilita que o fenômeno amplifique a imagem de um objeto fraco, permitindo o seu estudo. O lenteamento forte será o regime foco do presente trabalho.
- Lenteamento Fraco: é o regime no qual há pequena magnificação da fonte, seja por conta de a distância entre a lente e o observador ser pequena em relação a distância da fonte, ou dos ângulos de deflexão serem muito pequenos [47]. Geralmente as distorções nas imagens das fontes são bem pequenas, constituindo comumente de alongamentos da imagem. Esse efeito pode ser detectado apenas estatisticamente usando uma amostra de fontes.

Já quanto à escala angular, o efeito forte de lente gravitacionais pode ser classificado como macro, micro e mililenteamento.

¹ Este regime também pode ser abordado neste trabalho como “Efeito Forte de Lente”.

- Macrolenteamento: ocorre para objetos em escalas cosmológicas como galáxias, aglomerados ou superaglomerados, de modo que mesmo que as distâncias envolvidas sejam muito grandes, o fenômeno ainda pode ser detectado em escalas de segundos de arco [49]. Comumente ocorre lenteamento forte nestas escalas, por conta da intensidade do potencial gravitacional dessas estruturas. As imagens sofrem grandes distorções.
- Microlenteamento: ocorre, tipicamente, em casos no quais a magnificação pode ser observada, mas o efeito de distorção é frequentemente menor que a escala de resolução do telescópio usado para observá-lo, como por exemplo o lenteamento de uma estrela longínqua por outra estrela mais próxima [50]. O microlenteamento possui diversas aplicações na detecção de massa de estrelas, busca por exoplanetas e foi crucial na busca de objetos compactos como MACHOS (*Massive Compact Halo Objects*) [51].
- Mililenteamento: enquanto o microlenteamento diz respeito a ocorrências do fenômeno de lente em escalas de microssegundos de arco, o mililenteamento se refere a casos intermediários da ocorrência em milissegundos de arco [49]. Constitui principalmente do lenteamento de quasares ou estrelas dentro de galáxias.

2.2.1 Deflexão da Luz por Efeito de Lente e Fonte Pontuais

Para começar a estudar a deflexão da luz pelo lenteamento gravitacional, cabe iniciar a abordagem com o caso mais simples possível, de lente e fonte pontuais. Isto significa que tanto o objeto que servirá de lente quanto a fonte podem ser tratados como objetos pontuais. Situando a lente de massa M ao longo da linha de visada, a fonte S terá a sua imagem defletida de um ângulo $\hat{\alpha}$ tal que um observador O irá encontrá-la na posição I , de acordo a figura 2.1.

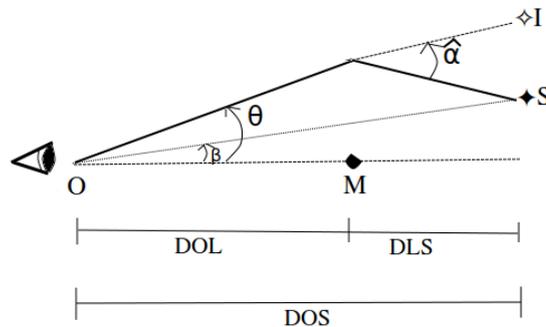


Figura 2.1 – Desvio da luz de uma fonte S devido as proximidades de seu caminho óptico com o campo gravitacional gerado pela massa M . As escalas dos ângulos estão exageradas pelo teor didático.

Desta forma, ao invés de parecer ocupar a posição angular β em relação à massa M , a fonte S parece ocupar a posição angular θ . Considera-se que as distâncias D_{OL}

(entre o observador e a lente), D_{LS} (entre lente e fonte) e D_{OS} (entre observador e a fonte) são muito maiores do que a representação esquemática parece evidenciar, e os ângulos envolvidos são bastante pequenos. Por conseguinte, da figura 2.1 pode-se extrair a seguinte relação:

$$\theta D_{OS} = \beta D_{OS} + \hat{\alpha} D_{LS}. \quad (2.3)$$

Definindo o ângulo de deflexão reduzido $\alpha = \hat{\alpha} \frac{D_{LS}}{D_{OS}}$, podemos reescrever 2.3 como

$$\beta = \theta - \alpha. \quad (2.4)$$

A equação 2.4 é conhecida como *equação da lente*, que descreve as relações entre as posições reais e aparentes da fonte [13]. Retomando a equação 2.2, podemos utilizar o ângulo de deflexão reduzido definido previamente, para escrever em 2.4 que

$$\alpha = \frac{D_{LS}}{D_{OS} D_{OL}} \frac{4GM}{c^2 \theta}, \quad (2.5)$$

onde utilizamos que, no presente esquema, o parâmetro de impacto ξ é tal que $\xi = \theta D_{OL}$. Substituindo o ângulo de deflexão encontrado em 2.5 em 2.4, podemos reescrever a equação da lente como

$$\theta^2 - \beta\theta - \theta_E^2 = 0; \quad (2.6)$$

em que o termo θ_E denota o que é chamado por *ângulo de Einstein* [13, 14] e é tal que

$$\theta_E = \sqrt{\frac{D_{LS}}{D_{OS} D_{OL}} \frac{4GM}{c^2}}. \quad (2.7)$$

Uma consequência direta da resolução para θ da equação 2.6 é que há duas soluções possíveis. Dado que θ é a posição aparente da imagem da fonte S, a possibilidade de duas soluções significa a possibilidade de haver duas imagens da fonte S. As imagens estarão situadas ao longo de uma reta que passa por elas e pela fonte. Além disso, nas ocasiões em que houver um perfeito alinhamento entre a fonte e a lente, teremos $\beta = 0$, e assim, a solução para θ da equação 2.6 será θ_E , em qualquer direção. Desta forma, a imagem resultante da fonte pelo efeito de lenteamento será um anel de raio θ_E , uma estrutura chamada de *anel de Einstein* [13, 14]. Apesar da transição de duas imagens para um anel ser bastante abrupta, cabe mencionar que isto se deve às considerações feitas quanto a lente e fonte serem pontuais. Portanto, é possível que este efeito ocorra predominantemente em pequenas escalas, como no lenteamento de estrelas [35]. Para efeitos mais realísticos em galáxias, cabe observar outras situações de efeito de lente.

Antes, cabe mencionar uma derivação de nosso interesse, definir a grandeza chamada por *magnificação*. A magnificação μ é a razão entre os ângulos sólidos da imagem da galáxia sem efeito de lente e a imagem aparente, de modo que [52]

$$\mu = \frac{\theta}{\beta} \frac{d\theta}{d\beta}. \quad (2.8)$$

A magnificação indica quantas vezes o brilho da fonte foi ampliado na imagem, um efeito também resultante do próprio efeito de lenteamento. Nisto reside um dos pontos de importância o estudo do efeito de lenteamento: a possibilidade de utilizar lentes como telescópios gravitacionais naturais. Alguns objetos longínquos difíceis de se observar podem se tornar mais visíveis e/ou detectáveis caso sofram um efeito de lente, por conta da conservação do brilho superficial da imagem. De fato, o lenteamento já permitiu a detecção de estrelas [53], galáxias e quasares [54, 55, 16, 56] e até mesmo supernovas [26, 25].

2.2.2 Deflexão da Luz por Lentes com Simetria Axial

Na subseção 2.2.1, vimos como são formadas as imagens no caso em que consideramos as lentes como objetos pontuais, bastante utilizadas no regime de microlenteamento para estrelas. Neste regime, as dimensões do objeto utilizado como lente podem ser ignoradas. No entanto, no caso de lentes mais extensas como uma galáxia ou um aglomerado será preciso adotar a forma vetorial da equação 2.4, de modo que teremos.

$$\vec{\beta} = \vec{\theta} - \vec{\alpha}. \quad (2.9)$$

Em particular, um caso de interesse para galáxias atuando como lentes é o caso em que há simetria axial, isto é, caso em que a lente é circularmente simétrica. Além disso, considerando a distribuição de matéria escura em halos que envolvem o plano galáctico para um pouco além deste e, assim, também contribuem no efeito de lenteamento, um caso bastante comum de ser abordado é a modelagem chamada por *Esfera Isotérmica Singular* (EIS). Muitas galáxias podem ter o seu perfil de densidade ρ em termos do raio r aproximado por [14]

$$\rho(r) = \frac{\sigma^2}{2\pi G r^2}, \quad (2.10)$$

onde σ é o parâmetro de dispersão de velocidades na direção radial. Integrando este perfil de densidade ao longo da linha de visada, obtém-se a densidade superficial

$$\Sigma(\theta) = \frac{\sigma^2}{2GD_{OL}\theta}. \quad (2.11)$$

Por conseguinte, a massa enclausurada num círculo de raio θ ao longo da linha de visada é obtida como

$$M(\theta) = \frac{\pi\sigma^2 D_{OL}\theta}{G}. \quad (2.12)$$

Substituindo a expressão 2.12 na expressão para a forma vetorial do ângulo de deflexão 2.2, teremos

$$\vec{\alpha} = 4\pi \left(\frac{\sigma}{c}\right)^2 \hat{\xi}, \quad (2.13)$$

que indica que o ângulo de deflexão é constante e aponta na direção radial. Neste caso, teremos para a equação da lente vetorial 2.9 que

$$\vec{\beta} = \vec{\theta} - \theta_E \frac{\vec{\theta}}{|\vec{\theta}|}; \quad (2.14)$$

em que o ângulo de Einstein θ_E assume a forma

$$\theta_E = 4G \frac{D_{LS}}{D_{OL}} \left(\frac{\sigma}{c} \right)^2. \quad (2.15)$$

No contexto do lenteamento que ocorre por uma lente extensa, pode ser notado que também são produzidas duas imagens, como solução de 2.14. No entanto, se $\beta > \theta$, uma das imagens desaparece. O fato do ângulo de deflexão ser constante indica que não há amplificação das imagens da fonte durante o lenteamento, e a magnificação se deve a deformação tangencial do ângulo sólido da imagem [13]. Este modelo de lente possui uma singularidade no ponto em que $\beta = 0$, de modo que a imagem resultante seja um anel de Einstein. Uma maneira de evitar a singularidade é definindo o perfil de densidade ρ como [46]

$$\rho = \frac{\sigma^2}{2\pi G(r^2 + r_c^2)}; \quad (2.16)$$

onde r_c é o raio do núcleo da lente. O modelo que obedece a este perfil de densidade é comumente chamado por *Esfera Isotérmica Suavizada*, por conta de fornecer as mesmas soluções e corrigir a singularidade resultando do modelo de EIS. A grandes distâncias do núcleo, verifica-se que ele retorna o comportamento da EIS.

Tendo em mãos estes dois modelos de lentes, já é possível reproduzir duas formas bem conhecidas em que o fenômeno do lenteamento se manifesta, que são as imagens duplicadas e o anel de Einstein. Há ainda outros modelos de distorções como a geração de imagens quadruplicadas da fonte, no entanto, realizar estas considerações exige o uso de outros perfis de densidade [57].

2.2.3 Deflexão da Luz com Fontes Circulares

Nas subseções anteriores, tem-se atribuído às diferentes distorções da imagem da fonte os diferentes perfis de densidade da lente. No entanto, é possível obter ainda diferentes distorções ao considerar que a fonte também pode ser extensa. Considerar ainda que a fonte é extensa abre a possibilidade de, em conjunto a diferentes perfis de densidade, formar não apenas réplicas pontuais das imagens, mas o que se chamam por *arcos gravitacionais*. Os arcos gravitacionais situam-se numa situação intermediária entre as imagens múltiplas e os anéis de Einstein, sendo resultado de distorções nas imagens até o ponto em que se tornem um anel completo.

Para obter as soluções com arcos, será abordado o caso particular de que as fontes sejam circulares, por simplificação. Considerando uma fonte circular de raio R situada à

uma distância radial r de uma lente, tome a posição angular do centro da fonte como ϕ_0 e a posição de um ponto qualquer no perímetro da fonte como ϕ . É possível demonstrar geometricamente que a posição de um ponto β na superfície da fonte admite as seguintes possibilidades, considerando a simetria axial do círculo [58]:

$$\beta^\pm = s \cos(\phi - \phi_0) \pm \sqrt{R^2 - r^2 \sin^2(\phi - \phi_0)}. \quad (2.17)$$

Como o segundo termo de 2.17 pode ser positivo ou negativo, para cada ϕ há dois valores de β possível. Considerando os valores possíveis para ϕ , as duas soluções se encontram no momento em que o segundo termo de 2.17 se anula. Para utilizar essa relação para β na equação de lente, primeiro tomemos a solução para θ da equação 2.14, que retorna:

$$\theta^\pm = \beta \pm \theta_E. \quad (2.18)$$

Cabe lembrar que esse resultado deriva da suposição de que a lente também é extensa e obedece o perfil de densidade de uma EIS. Já foi visto na seção 2.2.2 que, para este caso e uma fonte pontual, há uma solução dupla. Agora, ao utilizar 2.17 na solução 2.18, temos:

$$\theta^\pm = \theta_E \pm r \cos(\phi - \phi_0) \pm \sqrt{R^2 - r^2 \sin^2(\phi - \phi_0)}. \quad (2.19)$$

É possível notar de 2.19 que há quatro possibilidades de solução, para cada valor de ϕ . Desta maneira, a solução para a imagem da fonte θ tem quatro curvas possíveis. O resultado dessas quatro curvas são dois arcos gravitacionais! As soluções positivas formam o perímetro externo dos arcos enquanto as negativas foram o perímetro interno. De acordo os valores de r em relação a R , há diferentes distorções na forma dos arcos [58].

CAPÍTULO 3

REDES NEURAIS ARTIFICIAIS

3.1 História e Introdução

O neurônio é a célula que é a unidade fundamental do sistema nervoso e do cérebro, funcionando a partir de uma transmissão de pulsos elétricos de um para outro pelos contatos chamados por axônios. Existem estruturas nos neurônios chamados por dendritos que são capazes de receber os estímulos locais do ambiente e, depois desses estímulos serem transmitidos para o corpo celular do neurônio, respostas são transmitidas pelos axônios que estão ligados ao axônios de outros neurônios [59]. Desta maneira, um neurônio comumente está ligado a mais de um neurônio, de modo que o resultado seja uma rede de neurônios interligados que possam transmitir pulsos através de seus axônios. Esses pulsos elétricos devidamente processados carregam informações, e dessa interconexão resulta o conceito de uma rede neural. Os neurônios estão interligados de modo que, em diferentes partes do corpo, um neurônio está a apenas alguns pulsos de distância de outro. A ideia de construir modelos matemáticos para resposta de neurônios foi descrita por McCulloch e Pitts [60]. Este trabalho permitiu, posteriormente, a construção da noção de neurônio artificial conhecido como perceptron e o desenvolvimento das redes neurais artificiais.

Por sua vez, podemos conceber uma rede neural artificial (RNA) através de um algoritmo matemático que simula a conexão entre unidades básicas de processamento que realizam o papel dos neurônios, organizados em camadas. A primeira camada recebe os dados de entrada que, na analogia biológica consistiria dos estímulos que ativam os neurônios locais, que por sua vez irão passar a informação adiante até chegar ao cérebro que decidirá como processar a informação, como por exemplo, um estímulo de toque na pele. As camadas seguintes da RNA, portanto, na mesma analogia, consistem de conjuntos de neurônios que recebem a informação já processada. Estes sinais de saída seguem para outros neurônios que receberão as combinações de saídas de outros, e chegarão a uma camada final de saída, onde ocorrerá a resposta final acerca da variável em estudo [61].

O entendimento sobre como ocorria o processamento de imagens no sistema nervoso começou a avançar com os trabalhos de Hubel e Wiesel. Eles propuseram em 1959 que haviam dois tipos de células no córtex visual, responsáveis pelo reconhecimento de padrões visuais [62]. As do tipo mais simples seriam capazes de detectar contornos e barras em orientações e locais particulares de uma imagem, enquanto as mais complexas seriam capazes de realizar a mesma detecção em diferentes posições do campo visual - propriedade chamada por “invariância espacial” [63]. Em 1962, Hubel e Wiesel propuseram que essa propriedade das células complexas seria atingida através de uma soma de impulsos de saída de muitas outras células simples, de modo que a especialização das células simples em contornos de posições específicas seja essencial para realizar o reconhecimento de contornos em situações mais complexas [64]. Este conceito, de que a soma de detectores simples seria capaz de gerar um detector mais complexo é a ideia central das redes neurais convolucionais, que foram amplamente utilizadas neste trabalho.

Um dos primeiros modelos de redes neurais artificiais para o reconhecimento de imagens utilizando este conceito do processamento no córtex visual de vertebrados foi proposto por Fukushima em 1981, chamado por Neocognitron [65]. Neste modelo computacional, a detecção de diversos padrões geométricos simples nas imagens é utilizada para a formação de um padrão mais complexo, que possa resultar em um objeto claro. Em outras palavras, o detector era capaz de extrair informações das imagens sem a necessidade de instruções sobre o que procurar, baseando-se apenas em padrões geométricos [66, 67].

Já em 1995 foi proposto por Lecun et. al [68] um trabalho utilizando redes neurais convolucionais para a detecção de dígitos à mão. O trabalho foi realizado sobre uma base de dados pública conhecida como MNIST, cujos objetos são dígitos numéricos escritos à mão, imagens binárias escritas em diferentes grafias de algarismos de 0 à 9 [69]. O *dataset* completo consiste de mais de 60.000 objetos para treino, mas Lecun et. al [68] utilizaram 7.064 objetos para treino e pouco mais de 2.000 para teste. Nesse artigo foi realizada uma comparação entre classificadores para, logo em seguida, ser proposta a ideia de uma rede neural convolucional, numa arquitetura personalizada chamada por *LeNet*. Basicamente, a ideia da rede seria, ao invés de extrair características dos dados de entrada pela combinação de características locais, realizar um reconhecimento geral do objeto de modo que se extraia o que se chama de mapa de características, de modo a facilitar que o objeto seja encontrado em diferentes posições ou rotações. Em termos computacionais, ao invés de neurônios totalmente conectados, são definidos filtros com um tamanho fixo, e portanto um número de pesos reduzidos. Estes filtros são convoluídos com os dados de entrada. Isso faz com que um filtro que detecte uma característica forme um mapa de ativação sensível apenas localmente. Estes mapas de ativação são, por sua vez usados de entrada para a camada seguinte que terá um filtro ativo para um outro padrão treinável. Daí a ideia de que a rede neural convolucional consiste de um grande detector a partir de muitos detectores simples.

Alguns modelos de redes neurais convolucionais foram amplamente utilizados neste trabalho, e por isso iremos nos debruçar brevemente sobre o seu funcionamento.

3.2 Redes Neurais Artificiais (RNA)

Nesta seção serão enumerados alguns aspectos gerais de uma RNA de modo geral, no intuito de causar familiaridade com o funcionamento de RNA antes de prosseguir para as redes neurais convolucionais.

3.2.1 Aspectos Gerais de uma RNA

Para iniciar essa abordagem, iniciaremos com o esquema de RNA presente na figura 3.1.

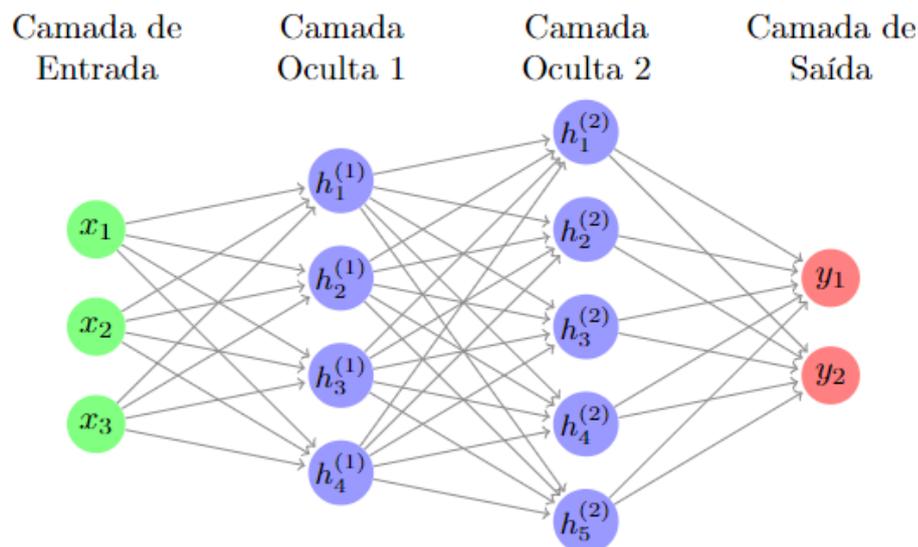


Figura 3.1 – Um exemplo de rede neural densamente conectada.

Nesta figura, cada círculo representa um *neurônio* ou *nó*, distribuídos em *camadas*. Os neurônios de cada camada estão conectados a todas as saídas da camada anterior e, à cada conexão é atribuído um parâmetro chamado *peso*. Os valores destes pesos podem flutuar livremente e é a atualização dos valores deles, de acordo o que for mais adequado para cada situação, que determina o quão eficiente foi o aprendizado de uma RNA.

A primeira camada da rede é chamada por “Camada de Entrada”, que constitui de contato da rede neural com a base de dados sobre a qual ocorrerá o aprendizado. É nessa camada que os objetos de estudo entram na rede neural, sejam eles imagens, tabelas ou caracteres. O tamanho desta camada está diretamente relacionado à forma do dado no qual a rede fará uma inferência única. As camadas subsequentes de uma rede totalmente conectada, também conhecidas como densas, com exceção da última, são chamadas por Camadas Ocultas. Essas camadas possuem pesos que, quando treinados, processam os

dados de entrada com objetivo de extrair informação relevante dos dados. Em cada neurônio ocorre uma operação, os dados de entrada são cada um multiplicados por um peso e entram em uma função de ativação, de acordo a figura 3.2.

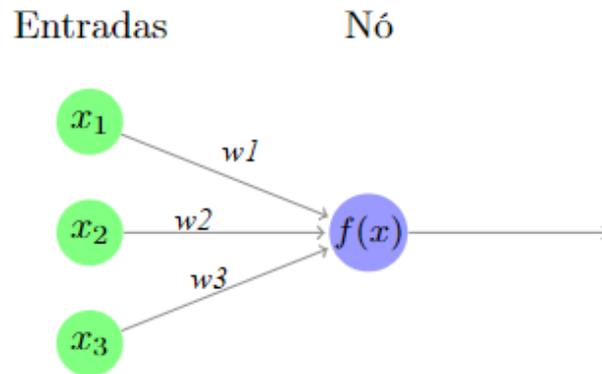


Figura 3.2 – Esquema das operações que ocorrem em cada neurônio de uma rede neural..

Mantendo o mesmo exemplo da figura 3.1, este neurônio está sendo alimentado por três entradas correspondentes ao lote de dados de entrada da rede. Cada uma das entradas será multiplicada por um peso e, em seguida, somadas. A seguir, a depender da arquitetura da rede, existirá uma função que realizará alguma operação sobre esta soma. Os pesos são determinados aleatoriamente no momento de geração do modelo, e são atualizados durante o processo de treinamento, conforme veremos na seção seguinte.

3.2.2 Função de Ativação

As funções que são comumente utilizadas em redes neurais neste passo de computar as entradas multiplicadas pelos pesos são chamadas por *função de ativação*. O objetivo da função de ativação é atribuir um valor de saída às somas dos dados de entrada multiplicados pelos pesos, transmitindo uma informação adiante aos neurônios da camada seguinte. Uma das funções de ativação mais amplamente utilizadas, por exemplo, é a função ReLU - Unidade Linear Retificada¹:

$$f(x) = \max \{0, x\}; \quad (3.1)$$

isto é, a função retorna o máximo do conjunto formado pelo algarismo 0 e o valor da soma das entradas multiplicadas pelo peso. Por conseguinte, há apenas dois valores possíveis de saída desta função de ativação: 0 ou x . Como a função retorna sempre o máximo entre esses dois, significa que caso x seja um valor negativo, a função retornará zero. Caso contrário, retorna o valor da soma. Assim, caso a soma dê um valor negativo a ReLU simplesmente desliga aquele neurônio, impedindo-o de transferir alguma informação além

¹ do inglês por tradução livre: *Rectified Linear Unit*

de zero para os neurônios da camada seguinte. Este cálculo é realizado em cada neurônio da rede, e os resultados são transmitidos para os neurônios da camada seguinte.

De certa forma, este é o comportamento geral das funções de ativação: determinar a relevância que as entradas recebidas terão nas próximas camadas da rede. Existem ainda algumas redes que possuem o que chamamos por *viés*, que tem a única função de adicionar uma constante à combinação linear de entradas e pesos do neurônio. O objetivo é transmitir alguma tendência à função de ativação, de modo a afetar o resultado como o próprio nome sugere: adicionando um viés que possa contribuir para a aprendizagem da rede. Outra técnica comumente utilizada é a *normalização dos lotes*², que subtrai a saída dos neurônios de cada camada pela média das saídas naquele neurônio e os divide pelo seu desvio padrão[70].

3.2.3 Treinamento da RNA

O treinamento da RNA é o processo em que ocorre, efetivamente, a aprendizagem da rede de se adaptar à base de dados de entrada. Isto ocorre através da atualização dos pesos dos neurônios, portanto, dizer que uma rede neural se mostra eficiente na detecção de um dado fenômeno em uma base de dados significa dizer que ela consegue facilmente adaptar os seus pesos para aquela base de dados. Esta otimização dos pesos é realizada através do processo de retropropagação.

3.2.3.1 Retropropagação

No processo de retropropagação³, ao final da propagação dos sinais de entrada da rede até a última camada, é realizada uma avaliação de desempenho através de uma função custo ou perda⁴. Nos casos em que existe uma tabulação clara atribuída aos dados do teste cego⁵, diz-se que ocorre uma aprendizagem supervisionada, enquanto no caso contrário ocorre a aprendizagem não-supervisionada [71]. No presente trabalho, foi utilizada apenas a abordagem do aprendizado supervisionado no regime de classificação. A classificação de uma base de dados consiste de tentar fazer o modelo de RNA aprender a discernir um objeto em questão entre outros, separar os objetos da base de dados em classes. Neste regime, após o teste em um conjunto de dados independente, i.e., um teste cego, é possível fazer uma estimativa do erro da rede neural que é calculada para o conjunto de treino e de validação.

Neste passo é realizada uma auto avaliação do modelo e, como consequência, os pesos da rede sofrem uma otimização em um processo chamado por *retropropagação* de acordo

² do inglês por tradução livre: *batch normalization*.

³ do inglês por tradução livre: *backpropagation*

⁴ da terminologia em inglês *loss function*.

⁵ No contexto de problemas de classificação utilizando redes neurais, diz-se que foi realizado um teste cego ao analisar a eficiência da classificação de um modelo em dados novos, isto é, nunca antes vistos pelo modelo.

a sua relevância para a minimização do erro geral da rede. Depois de otimizados os pesos, é hora de colocar mais um lote de dados dentro da rede. Depois dos neurônios realizarem a soma das entradas multiplicadas pelos pesos otimizados e passarem um sinal adiante até a última camada da rede, a função custo é calculada. De acordo o resultado, pode-se definir uma regra de otimização que minimize a função custo alterando os pesos na direção do gradiente desta função.

Quando chegam à última camada da rede neural, os neurônios fazem o mesmo processo de multiplicar os dados de entrada pelos pesos atribuídos às conexões ali presentes. O número de neurônios na última camada da rede neural depende do tipo de problema a ser abordado, algumas saídas podem ser binárias, valores únicos ou terem mais possibilidades. Não raramente a função de ativação utilizada nos neurônios desta camada pode ser diferente da utilizada ao longo das demais camadas da rede.

O processo de otimização ocorre em pequenos lotes de dados, chamados *batches*, quando o conjunto de lotes de dados utiliza todo o dataset de treino dizemos que a rede foi otimizada em uma época. Em geral, a otimização é realizada em diversas épocas até que o erro da rede no treino e na validação se estabilizem.

Se os pesos da rede são iniciados aleatoriamente, ao longo das primeiras interações, é esperado que este erro tenha um valor alto e, no intuito de minimizá-lo é que entra o algoritmo de retropropagação. Este, por sua vez, irá procurar mínimo nesta função, através do cálculo do seu gradiente. Um exemplo típico de função custo para os problemas de regressão é o erro quadrático médio [72]:

$$E = \frac{1}{2} \sum_{i=1}^{n_s} |p_i - t_i|^2; \quad (3.2)$$

onde n_s refere-se ao número de saídas da rede, p_i se refere ao valor previsto para cada objeto predito pela rede e t_i ao valor verdadeiro do objeto. As somas das diferenças entre os valores preditos pela rede e os valores tabulados dos objetos de teste, define, portanto, o erro da rede.

Em seguida, são calculadas as derivadas parciais da função erro em cada peso da rede, começando pelos das últimas camadas até os pesos das primeiras. Daí o nome do processo de “retropropagação”. Esta ordem se deve aos pesos das últimas camadas dependerem dos sinais das camadas anteriores. Cada passo do *backpropagation* consiste, portanto, de realizar a atualização nos valores dos pesos w_i de uma dada camada da rede através de derivadas parciais, de modo que

$$w_i \rightarrow w_i - \eta \frac{\partial E}{\partial w_i}, \quad (3.3)$$

onde E corresponde à função erro utilizada no algoritmo e η é uma grandeza chamada por *taxa de aprendizagem* [61]. A taxa de aprendizagem é um parâmetro definido na etapa de construção do modelo de rede neural e que pode ser ajustado de problema a problema.

A sua função é de suavizar a descida do gradiente. Se a taxa de aprendizado for muito alta, é possível que o ajuste atravesse o vale deste mínimo local e chegue ao lado oposto. Por outro lado, se a taxa for muito pequena, é possível que o gradiente precise de muitas etapas de treino para chegar ao mínimo local [71].

Este processo é repetido quantas vezes forem necessárias para que a necessidade de atualizar os pesos seja a mínima possível, ou até que o número de épocas de treino seja atingido.

3.3 Rede Neural Convolutacional (RNC)

As principais aplicações das RNCs é o uso na detecção de padrões em imagens, apesar de poder ser utilizada para processar dados tabulares, séries temporais ou qualquer dado estruturado. Enquanto as RNAs mais básicas têm dificuldades para lidar com dados na forma de imagens, as RNCs possuem design apropriado para o processamento de imagens. Por exemplo, para fazer um detector com a base de dados MNIST, cujas imagens são binárias e tem dimensões de 28 x 28 pixels, o neurônio da primeira camada oculta de uma RNA amplamente conectada teria 784 pesos para atualizar (isto é 28^2) e, no contexto de camadas posteriores totalmente conectadas, cada neurônio da camada seguinte precisa estar ligado com todos os neurônios da camada de entrada. Considerando bases de dados com imagens maiores, fica evidente o problema computacional resultante desta abordagem. Considerando todos os hiperparâmetros, a rede teria um número exorbitante de pesos para atualizar à cada retropropagação [73].

Um número tão grande de hiperparâmetros poderia levar a um fenômeno conhecido no ramo de aprendizado de máquina por sobreajuste⁶, em que o modelo se especializa demais nos dados de treino de modo a ser completamente incapaz de classificar novos dados [74]. Significa que, enquanto a classificação sobre os próprios dados de treino torna-se ótima, o modelo adquire uma grande dificuldade de executar um teste cego.

Mencionar a possibilidade de que haja um design apropriado para executar imagens significa que os neurônios das RNCs têm formato diferente, em comparação aos de uma rede neural artificial completamente conectada. As camadas de uma RNC consistem de neurônios tridimensionais, que possuem uma altura, largura e profundidade definidos. Ao contrário de uma RNA completamente conectada, os neurônios de uma RNC se conectarão a apenas uma parte pequena dos neurônios da camada anterior. Adicionando camadas de redução de dimensionalidade a tendência é, portanto, que a saída resultante em cada camada seja cada vez menor, de modo que sejam selecionadas as características da imagem importantes para o treinamento. Além disso, estas conexões com pequenas partes dos neurônios da rede anterior diminui bastante o número de parâmetros, tornando a rede computacionalmente mais eficiente. Adicionalmente, um grupo de conexões

⁶ do termo comumente usado em inglês *overfitting*.

pode compartilhar os mesmos pesos, reduzindo ainda mais a quantidade de parâmetros [75]. Por isso, verificaremos como estas conexões possibilitam a formação de um arranjo diferente de rede neural, em camadas convolucionais.

3.3.1 Camadas Convolucionais

Através das conexões locais, os neurônios podem extrair características visuais básicas do objeto na imagem, como cantos e contornos. Estas características são, por conseguinte, conectadas em camadas subsequentes no intuito de detectar características mais específicas do objeto [73]. O fluxo de dados que sai de uma camada convolucional são organizados em matrizes correspondendo a região da imagem analisada e são chamadas por *mapa de características* [73].

A unidade básica de uma camada convolucional são os filtros, cada filtro pode ser representado por uma matriz de pesos. O filtro é convoluído pela imagem ou pelo mapa de características das camadas anteriores. A operação é ilustrada na figura 3.3: o filtro K é sobreposto ao dado de entrada da camada (I). Nesta sobreposição, o filtro realiza um produto ponto a ponto com os valores da região da entrada sobre o qual o filtro está sobreposto. Depois, o filtro começa a se mover sobre o o dado de entrada, e a cada passo realizado esse produto escalar é realizado com as partes do dado de entrada [76]. Os resultados destes produtos compõem o novo mapa de características de saída.

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 4 & 3 & 4 & 1 \\ 1 & 2 & 4 & 3 & 3 \\ 1 & 2 & 3 & 4 & 1 \\ 1 & 3 & 3 & 1 & 1 \\ 3 & 3 & 1 & 1 & 0 \end{pmatrix}$$

I
 K
 $I * K$

Figura 3.3 – Ilustração da convolução que um núcleo (ou filtro) K realiza por um tensor de imagem I , resultando no mapa de características $I * K$.

No caso da figura 3.3, veja que as operações que resultaram no tensor $I * K$ foram realizadas com o filtro se movendo de uma em uma coluna e linha. Existe um parâmetro que indica a quantidade de linhas e colunas pelas quais o núcleo deve passar a cada produto escalar ou matricial, chamado por *passo*⁷. Ao aumentar o passo, menos cálculos serão realizados, o que indica que o mapa de características resultante terá um tamanho menor.

⁷ do inglês por tradução livre: *stride*.

Outro ajuste que pode ser realizado na convolução é a adoção do *preenchimento*⁸. Comumente o produto escalar ou matricial pode ser feito entre valores internos da tabela de dados, mas inevitavelmente ocorrerá uma redução de dimensionalidade da imagem para o mapa de características, qualquer que seja o passo do núcleo adotado. Durante algumas convoluções, no entanto, pode-se evitar esta perda de dimensionalidade, e esta constitui uma das funções do preenchimento. Sua adoção permite que o núcleo atravesse uma região do tensor de imagem que saia das bordas, através do aumento das dimensões do tensor de imagem original adicionando zeros. Desta forma, o núcleo pode continuar deslizando e realizando as convoluções em regiões que comumente estariam quase fora do tensor de imagens. Por conseguinte, é possível evitar a redução do tamanho do mapa de características resultante.

3.3.2 Esquema Geral de uma RNC

Tendo em mãos a operação de convolução e a geração dos mapas de características no processamento de imagens, é possível prosseguir para a estrutura completa de uma RNC, de modo análogo ao que foi feito na subseção 3.2.1. De modo geral, uma RNC é composta pelas camadas expostas no esquema da figura 3.4:

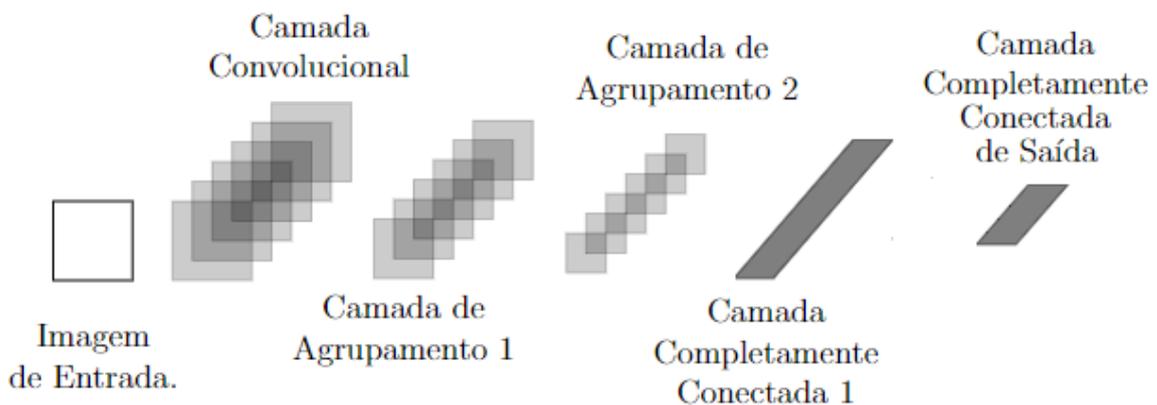


Figura 3.4 – Esquema básico da arquitetura de uma RNC, constituída por uma camada de imagens de entrada, camadas convolucionais, de agrupamento e completamente conectadas (Créditos: [77]).

Há quatro áreas básicas de funcionamento de uma RNC. Do mesmo modo que no esquema de uma RNA comum, a primeira camada consiste da entrada dos objetos de treinamento. No presente caso de RNCs, os objetos de treinamento são imagens. Portanto, na primeira área da rede teremos a camada de entrada com imagens.

Em seguida, temos a área da camada convolutiva que também utilizam funções de ativação. Serão responsáveis por fazer a extração de características dos objetos e determinar as saídas dos neurônios, depois do processo de calcular produtos matriciais ou

⁸ do inglês por tradução livre: *padding*.

escalares nos núcleos. Subsequentemente, temos o que se chama por camadas de *agrupamento*⁹. A função das camadas de agrupamento é reduzir ainda mais as saídas em mapa de características gerados, de modo a agrupar por exemplo, cada região 2×2 de um mapa de características em uma unidade 1×1 . Como diversos mapas de características saem da camada convolucional, o agrupamento é aplicado para cada mapa. Matematicamente, um mapa de características de altura a e largura l , submetido a um agrupamento de núcleo de dimensões $n \times n$ e passo p , terá suas dimensões alteradas de modo que:

$$a' = \frac{a - n + p}{p}; l' = \frac{l - n + p}{p}; \quad (3.4)$$

em que fica evidente a dependência com as dimensões do núcleo p . As duas maneiras mais comuns de se realizar o agrupamento são através da média ou do máximo. No agrupamento pela média, à cada região de tamanho a ser agrupada do mapa de características, é transmitido para um novo mapa apenas o valor médio daquela região. Analogamente, no agrupamento por máximo (que será o mais utilizado nas arquiteturas deste trabalho) o valor transmitido de cada região do mapa de características é o máximo. Uma ilustração do agrupamento está presente na figura 3.5.

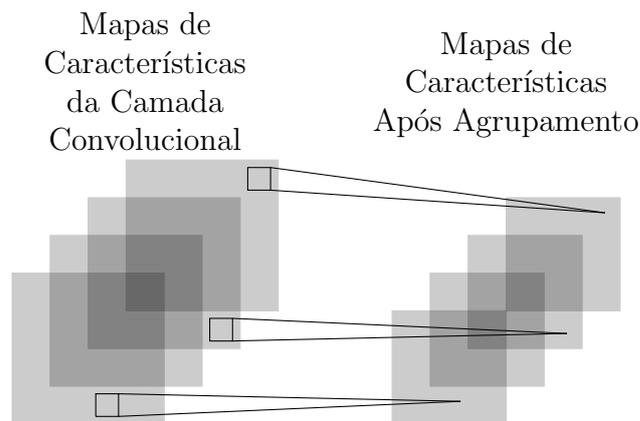


Figura 3.5 – Representação esquemática da redução de dimensionalidade de um mapa de características através do agrupamento.

O principal benefício do uso do agrupamento é o mesmo das reduções de dimensionalidade: reduzir o número de parâmetros da rede, acelerando os cálculos e o tempo de treinamento. Ambas as maneiras de se reduzir a dimensionalidade tem como consequência evidenciar aspectos mais específicos da imagem. Dependendo da complexidade das imagens, podem ser adicionadas camadas extras de agrupamento, no intuito de capturar estrutura das imagens de entrada [78]. Além disso, podem também haver mais uma camada convolucional acompanhada por uma camada de agrupamento, ou uma série destas.

Por fim, as últimas camadas da RNC são as *camadas completamente conectadas*. De modo análogo aos neurônios da RNA comum abordadas na subseção 3.2.1, estas camadas

⁹ do inglês por tradução livre: *pooling*.

possuem neurônios completamente (ou *densamente*) conectados aos neurônios de duas camadas adjacentes. Da mesma maneira, uma função de ativação vai julgar a informação a ser passada adiante, até chegar o estágio de realizar predições [74]. Ao final do treino da RNC serão realizadas predições que servirão de base para o cálculo de um erro, que irá induzir o mesmo processo de retropropagação e atualização dos pesos.

3.4 Métricas de Avaliação de Qualidade do Desempenho de uma Rede Neural

As predições do modelo ao fim das operações são essenciais para verificar o quão bem ocorreu o aprendizado da rede. Em particular, é em um conjunto independente que as métricas de performance devem ser avaliadas, isto é, realizando um teste cego. Restringindo a análise aos problemas de classificação de imagens, significa que ao fazer o modelo realizar uma predição com imagens que apareceram durante o treino, o processo de atualização dos pesos irá gerar uma espécie de “memória” que se recorda de quais eram as classes atribuídas àquelas imagens. A adição de novas imagens não utilizadas no treino, no entanto, é que permitirá uma avaliação confiável da generalização do modelo.

3.4.1 Acurácia

Durante o treino de uma rede neural, comumente é reservada uma pequena parcela desses dados para formar um conjunto chamado por *dados de validação*. Estes dados de validação são utilizados para fazer a avaliação do algoritmo de rede neural ao final de cada época, isto é, após todos os dados de treino passarem pela rede neural. O modelo observa os dados no conjunto de validação e busca fazer uma predição: determinar à que classe cada dado pertence (reiterando que estamos no contexto de problemas de classificação). Conseqüentemente, no contexto do aprendizado supervisionado, haverá um número de acertos nesta previsão e, conhecendo o tamanho do conjunto de validação, a razão entre estes dois valores multiplicado por cem fornece a primeira métrica de avaliação: a *acurácia*.

A acurácia, portanto, nada mais é do que a porcentagem de acertos de uma predição. É uma medida bastante direta da capacidade de predição de uma rede, por conta de ser um cálculo simples.

3.4.2 Função Custo

Outra maneira de se observar a capacidade de predição da rede é através da função erro, já citada na subseção 3.2.3. A função erro, aqui chamada por custo, constitui de uma medição da eficiência do aprendizado da rede através de uma soma das distâncias entre os valores preditos e os valores reais de uma classificação. Várias funções podem

fazer o papel de função de custo, como por exemplo a função erro quadrático médio e a entropia cruzada binária, amplamente utilizada neste trabalho.

A função da entropia cruzada binária é comparar as probabilidades previstas na saída da rede com a tabela verdade fornecida e, por conseguinte, calcular uma pontuação que penalizará as previsões com base na distância ao valor esperado. Como o próprio nome sugere, esta função é adequada ao regime de classificação binária, i.e, em que os valores da tabela verdade alternam entre 0 e 1. A distância entre os valores é calculada através de correções nas probabilidades de modo que as previsões indiquem a chance de pertencimento ou não à classe. Nos casos em que o valor esperado é 1, as probabilidades se mantêm iguais às previsões. Já nos casos em que o valor esperado é zero, as probabilidades são corrigidas pelo resultado de 1 subtraído em seu valor. O cálculo da média do negativo destes logaritmos, por sua vez, dá o valor do erro através da entropia binária cruzada. De modo mais sucinto, pode-se encontrar o custo através dessa função pela relação 3.5 em que [79]

$$Custo = \frac{1}{N} \sum_{i=1}^N \{-[y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]\} \quad (3.5)$$

sendo o primeiro termo dentro da soma correspondente às probabilidades sem correção (quando o valor esperado é 1), enquanto o segundo às probabilidades corrigidas. A variável y_i corresponde aos valores da tabela verdade, enquanto p_i a previsão feita pela rede sobre estes valores. A principal vantagem do uso desta estatística é o fato dos termos em logaritmo suavizarem a descida do gradiente, sem penalizar demasiadamente as previsões incorretas.

3.4.3 Matriz de Confusão

Uma maneira de se avaliar diversos casos possíveis de uma operação de classificação é usando o recurso da matriz de confusão. Retomemos o cenário do presente trabalho, em que lidamos com um problema de classificação e aprendizagem supervisionada. Especificando ainda mais, tomemos o caso de uma classificação binária, isto é, em que há duas possibilidades para classificar um objeto: ou ele pertence a uma certa classe, ou não. No momento em que o modelo realiza previsões de uma base de dados, seja os de validação ou de teste, será retornada uma probabilidade de que o objeto se aproxime de cada uma das possibilidades. Uma probabilidade próxima a 1 indica que o objeto pertence à classe de interesse, enquanto uma próxima a zero indica que o objeto não pertence a esta classe. Ao comparar a previsão do modelo com a tabela verdade dos objetos, surgem naturalmente uma multiplicidade de possibilidades acerca do desempenho do modelo:

- Positivo Verdadeiro (PV): Os casos de positivo verdadeiro constituem as ocorrências em que o modelo aponta uma alta probabilidade do objeto pertencer à classe de interesse e, a tabela verdade indica que o objeto também pertence à classe.

- Negativo Verdadeiro (NV): De modo análogo ao PV, constituem os casos em que o modelo apontou uma tendência do objeto **não** pertencer à classe de interesse e, há uma concordância com a tabela verdade que indica que o objeto não pertence à classe.
- Falso Positivo (FP): Ocorre quando o modelo indica uma tendência do objeto pertencer à classe mas, segundo a tabela, o objeto não pertence à classe.
- Falso Negativo (FN): Ocorre quando o modelo indica uma tendência do objeto não pertencer à classe mas, na verdade, o objeto pertence de acordo a tabela-verdade.

No caso de mais classes, a quantidade de casos possíveis aumenta com o quadrado do número de classes, de modo que a diagonal corresponda sempre aos casos verdadeiros de cada classe. No nosso presente cenário, a matriz de confusão referente às possibilidades evidenciadas teria a forma da figura 3.6.

	Predição Positiva	Predição Negativa
Positivo Tabelado	PV	FN
Negativo Tabelado	FP	NV

Figura 3.6 – Uma matrix de confusão 2x2 para um problema de classificação binária e suas possibilidades.

Portanto, ao final de uma predição e comparação com a tabela verdade, o número de ocorrências de cada caso poderão ser distribuídos nessa tabela, cuja soma deve resultar sempre no número total de objetos usados para teste. A principal motivação de separar cada caso em específico é a possibilidade de realizar estatística com estes índices. Conforme abordado na subseção 3.4.1, a acurácia é a proporção de classificações corretas dividida pelo número total de classificações realizadas, portanto, em termos destas possibilidades teríamos que

$$Acurácia = \frac{PV + NV}{PV + FN + FP + NV}. \quad (3.6)$$

Há muitas outras métricas possíveis de serem calculadas com estas combinações mas, no intuito de limitar-nos ao que será usado no nosso trabalho, cabe definir duas taxas que definem uma métrica a ser explicada na seção 3.4.4. A primeira delas é a taxa de falsos positivos (TFP), dada por

$$TFP = \frac{FP}{PV + FP}, \quad (3.7)$$

isto é, a proporção de casos negativos que foram incorretamente classificados como positivos, entre todos os casos positivos [80]. Outra taxa de bastante utilidade será a taxa de positivos verdadeiros (TPV), dada por

$$TPV = \frac{PV}{PV + FP}, \quad (3.8)$$

que indica a taxa de casos positivos corretamente designados positivos, entre todos os casos positivos [80]. Idealmente, um modelo de RNA poderá ser dito eficiente num trabalho de classificação caso a sua TFP seja um valor próximo de zero, enquanto a sua FPV seja o maior possível. Estas duas métricas são utilizadas para calcular outra que será amplamente utilizada neste trabalho: a curva característica de operação do receptor.

3.4.4 Curva Característica de Operação do Receptor (ROC)

A curva ROC¹⁰ é uma métrica de análise da discriminância de modelos estatísticos, bastante eficiente em mostrar como o modelo está realizando diagnósticos do fenômeno em estudo. Consiste do resultado da sobreposição de valores da taxa de falsos positivos sobre valores da taxa de positivos verdadeiros (respectivamente mencionadas como TFP e TPV na subseção 3.4.3), considerando certos limiares [80, 81]. Na referida seção foi mostrado que estes dois valores são dados a partir de um limiar, ou *threshold*.

No modelo de classificação binária, definimos a curva ROC considerando diferentes valores de limiar no intervalo de 1 a 0, serão observadas as listas com predições: todos os valores com pontuação de predição maior que aquele valor são considerados como pertencentes a classe 1 e os demais como pertencentes a classe 0, no cálculo de uma TFP e TPV. Cada valor de TFP e TPV definem assim um ponto no espaço da curva ROC. Realizando este procedimento para todos os valores de limiar definidos entre 1 e 0, teremos portanto uma série de valores de TFP e TPV que, agrupados no espaço TFP contra TPV definem a curva ROC. Um exemplo da curva resultante de uma classificação de valores está presente na figura 3.7.

A curva da figura 3.7 reproduz um caso em que a curva ROC mostra uma boa discriminância do modelo em questão. No caso contrário, em que o modelo demonstra pouca ou nenhuma habilidade de discernir entre as duas classes, é esperado que o gráfico da curva ROC se aproxime da linha pontilhada em destaque. Neste regime, a habilidade de predição do modelo é indiferente da escolha aleatória quanto ao pertencimento ou não do objeto à classe [82, 83].

Cada valor dentro da sequência de números de 1 a 0 é um *limiar* acima do qual os valores são aproximados como pertencem à classe. Os primeiros valores de limiar terão poucas predições acima deles e portanto, a TFP e a TPV serão bem baixas. Já mais tarde, próximo ao final da lista de limiares, quase todos os valores de predição estarão acima do limiar e portanto, indicarão que os objetos pertencem à classe mesmo que as probabilidades sejam muito baixas, o número de falsos positivos se tornará bem grande e assim, a TFP aumentará até se aproximar de 1,0. Neste contexto, o número de positivos verdadeiros também será grande e assim, a TPV também se aproximará de 1,0. Portanto,

¹⁰ do inglês por tradução livre *Receiver Operating Characteristics*, em que optamos por adotar a sigla em inglês correspondente.

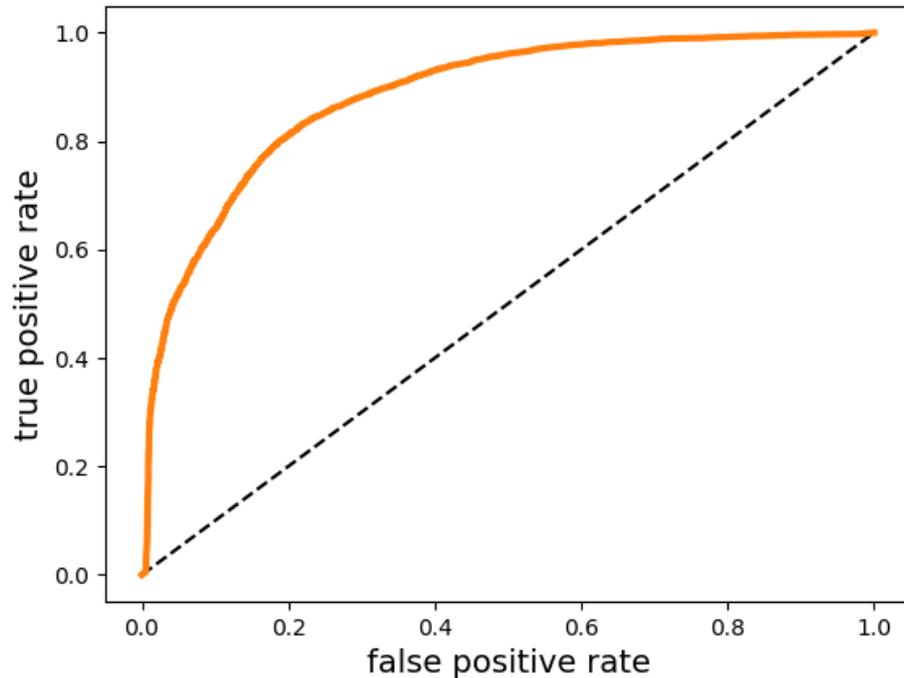


Figura 3.7 – Uma curva ROC, construída por uma série de pontos no espaço formado pelas TFPs com as TPVs.

nesta distribuição de probabilidades, a curva ROC deve sempre começar a partir do ponto $(0,0)$ e seguir até o ponto $(1,1)$ [83].

Nos casos em que o modelo mostrar boa habilidade de predição, conforme alguns poucos valores de probabilidade conseguirem ficar acima do limiar, a TPV começará a subir rapidamente, enquanto a TFP continuará bastante baixa. Com a repetição, em dado momento o limiar tomará um valor bastante baixo para que a TFP acabe aumentando o suficiente para aproximar a curva do nordeste do gráfico. Quanto mais a curva sobe e se aproxima do ponto $(1,0)$ à noroeste do gráfico mantendo um baixo valor de FPR, mais eficaz o modelo se mostrou na classificação [81, 82, 84].

3.4.5 Área sob a Curva (AUC)

Apesar de ser uma boa medidora da capacidade de discriminância de um modelo, a curva ROC por si só acaba sendo pouco prática em comparações mais específicas entre modelos. Por exemplo, para comparar o desempenho de dois modelos de classificação diferentes, seria necessário encontrar qual curva se aproxima mais do noroeste. Tendo em mente esta questão, uma escolha comum de se reduzir uma curva ROC a um único índice comparável é o uso do cálculo da área sob a curva (AUC¹¹, de agora em diante). Por ser uma integração sobre todos os pontos do gráfico da curva ROC, a AUC é um índice que resume a acurácia geral dos testes [84].

¹¹ do inglês por tradução livre *area under the curve*, em que optamos por adotar a sigla em inglês correspondente

Enquanto um cálculo de área, o máximo valor possível para a AUC de uma curva ROC é de 1 unidade ao quadrado, dado que o máximo possível para ambas os índices que servem de eixo é 1,0. De fato, de modo geral poderíamos resumir o caráter da avaliação observando os valores de AUC de acordo a seguinte proposição presente em [85]:

- AUC de 0,5 = nenhuma eficiência de discriminação, nenhuma diferença do regime aleatório em um classificador binário;
- AUC entre 0,5 e 0,7 = capacidade de discriminação baixa, pouco melhor que o regime aleatório;
- AUC entre 0,7 e 0,8 = capacidade de discriminação aceitável;
- AUC entre 0,8 e 0,9 = capacidade de discriminação excelente;
- AUC maior que 0,9 = capacidade de discriminação quase infalível;
- AUC de 1,0 = capacidade de discriminação perfeita.

Cabe evidenciar, no entanto, que não há referências universais na interpretação de índices de AUC; a presente lista nada mais serve como substrato para fornecer uma ideia de como interpretar estes valores [85, 86, 87]. Cada problema estatístico pode exigir intervalos mais ou menos rigorosos do que os supracitados, de acordo a aplicação. No presente trabalho, estaremos seguindo esta ideia buscando considerar sempre os desempenhos com AUC de 0,8 e acima como ótimos.

3.5 Arquiteturas Utilizadas

Neste trabalho foram utilizadas dois tipos de RNC a fim de resolver o problema de detecção de lentes gravitacionais. A arranjos de diferentes camadas e modos de realizar conexões de uma RNA, comumente são chamados no ramo do aprendizado de máquina por *arquiteturas*. Por isso, cabe enunciar cada uma das duas separadamente, uma vez que utilizam diferentes estratégias para realizar a tarefa de aprendizado a partir de uma base de dados.

3.5.1 ResNet50

A rede residual profunda de 50 camadas (*ResNet50*, de agora em diante) surgiu no contexto de classificação de imagens do *ImageNet Large Scale Visual Recognition Challenge* de 2015 (ILSVRC2015). O ILSVRC consistiu, em suas edições, de uma importante contribuição no avanço do desenvolvimento de técnicas de aprendizado de máquina. As entradas do desafio foram capazes de redefinir as principais métricas de estado da arte, na classificação de imagens da época [88]. A base de dados de treino usada no ILSVRC2015

consistia de um subconjunto com 1,2 milhões de dados da base de dados ImageNet, composta por cerca de 15 milhões de objetos. Havia três tarefas que poderiam ser atribuídas a um modelo de rede, a classificação, detecção e localização [88]. A ResNet exibiu um desempenho superior nas três categorias.

Conforme introduzido na seção 3.2, uma grande variedade de funções tem o potencial de ser aproximada por uma rede neural, desde que ela convirja no treino e a base de dados seja representativa o suficiente. No universo das redes neurais convolucionais, logo foi descoberto que aumentar o número de camadas (isto é, aumentar a profundidade) tipicamente demonstrava uma tendência de aumentar o desempenho nos problemas de classificação de imagens, apesar do custo computacional necessário e o sobre ajuste [89, 90]. Conseguir RNCs com melhor capacidade de discriminação seria, portanto, um problema de computação [90, 91]. No entanto, na contramão do aumento na profundidade da rede neural, surge o problema de gradientes que começam a desaparecer ou explodem. No processo de retropropagação, quando a rede é muito profunda, a multiplicação sucessiva de gradientes pode acabar fazendo com que nas primeiras camadas o gradiente já esteja tão pequeno que não faça diferença significativa no peso, ou grande o suficiente para desequilibrar totalmente os pesos da rede [92, 93]. Outro problema que se tornou comum ao desenvolver RNCs mais profundas é o problema da degradação da acurácia. Foi verificado por He et. al [94] que redes mais profundas geram maiores taxas de erro durante o treinamento do que redes mais rasas.

A ResNet surge com a inovação de inserir *conexões de atalho* entre camadas da rede. À cada certo número de camadas convolucionais, a saída de camadas anteriores é somada com a saída da camada atual, de acordo a figura 3.8

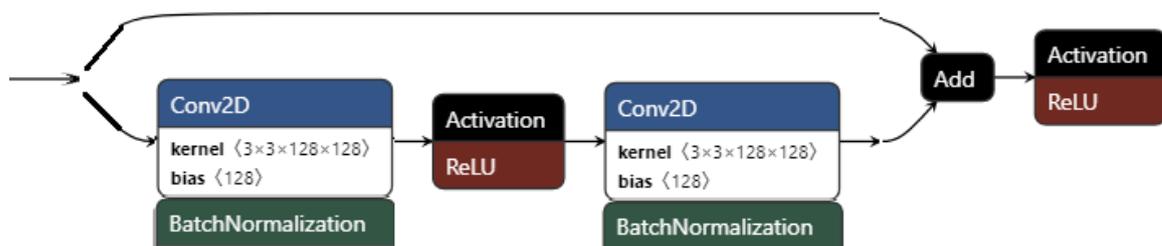


Figura 3.8 – Representação esquemática de um bloco de conexão de atalho de uma ResNet. Os dados entram no ramo à esquerda, passam por uma camada convolucional de pesos e normalização do lote (*Conv2D* e *BatchNormalization*, respectivamente) seguida de uma ativação ReLU. Em seguida, passam por outra camada convolucional com normalização do lote e, antes de passarem pela próxima ativação ReLU, são somadas com uma cópia dos mesmos dados de duas convoluções atrás.

Portanto, se na entrada é processado X por duas camadas, a saída será a sobreposição $X + F(X)$. A informação X da camada inicial é retomada em algumas camadas posteriores, a partir de uma soma matricial [90]. Esta operação não exige parâmetros adicionais,

e contribui para evitar o problema dos gradientes explosivos ou que desaparecem pois, ao estar sempre retomando entradas de momentos anteriores, os gradientes anteriores também são retomados. Além disso, a camada mais adiante tende, em algumas circunstâncias, a desempenhar melhor do que as camadas anteriores, uma vez que está sendo alimentada com o mapa de saída do processamento das camadas imediatamente acima e o mapa de algumas camadas atrás [95].

Há vários tipos de redes neurais residuais de acordo o tipo de camadas, mas cabe mencionar a que foi mais utilizada neste trabalho: a ResNet50. O índice 50 diz respeito à quantidade de camadas que a rede possui, quantidade um pouco maior do que a primeira rede residual proposta que possui 34 [90]. Além disso, uma outra mudança foi realizada para construir a ResNet50 e arquiteturas baseadas de número superior de camadas, como a ResNet100 e a ResNet152: a inserção de camadas intermediárias que possuam menos neurônios do que uma camada acima ou abaixo. Estas súbitas reduções no número de neurônios encorajam a rede neural a comprimir as representações de características aprendidas. Este recurso é chamado por *design de gargalo*¹² [90].

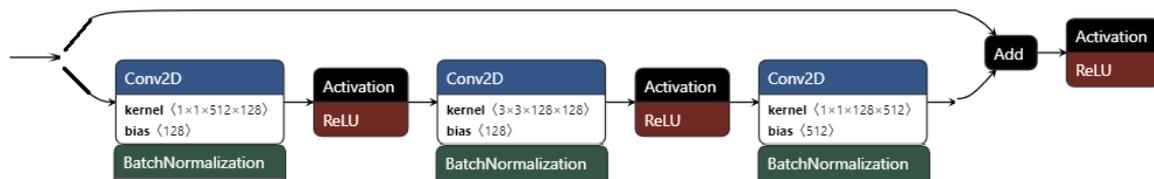


Figura 3.9 – Representação esquemática de um bloco de conexão de atalho de uma ResNet50. Diferentemente da ResNet de 34 camadas, aqui a conexão de atalho salta 3 camadas ao invés de duas, que por sua vez estão organizadas num design de gargalo.

As três camadas saltadas pelos atalhos da ResNet50 consistem de uma camada convolucional comum entre duas convoluções com núcleos de dimensão 1x1. A primeira camada convolucional de núcleo 1x1 reduz as dimensões da entrada, que segue para a camada convolucional comum de núcleo 3x3 responsável pela extração de características, cuja saída prossegue para a camada convolucional de núcleo 1x1 que retorna a saída para o tamanho original de entrada [96]. Por isso o nome “gargalo”. Do mesmo modo que na ResNet de 34 camadas, aqui cada camada convolucional é acompanhada do procedimento da normalização dos lotes e passa por uma função de ativação do tipo ReLU.

Quanto à ResNet50 utilizada no trabalho, optou-se por importá-la da biblioteca Keras para Python [97] que, por conseguinte, norteia a descrição precisa da arquitetura ResNet50 utilizada no trabalho a seguir [98]. Cabe lembrar que cada camada convolucional é seguida pela normalização dos lotes e uma função de ativação ReLU. A rede consiste em:

- 1 camada convolucional com 64 núcleos de tamanho 7x7 e passo de 2 unidades;

¹² tradução livre de *bottleneck design*.

- Agrupamento por máximo de tamanho 2x2.
- 9 camadas convolucionais com três blocos de conexão de atalho com a seguinte configuração: uma camada convolucional com 64 núcleos de tamanho 1x1, seguida por uma de 64 núcleos de tamanho 3x3, seguida por uma de 256 núcleos de tamanho 1x1.
- 12 camadas convolucionais com quatro blocos de conexões de atalho com a mesma configuração dos blocos do item acima, exceto que cada camada possui o dobro de núcleos correspondente.
- 18 camadas convolucionais com seis blocos de conexões de atalho com a mesma configuração dos 12 blocos anteriores, exceto que novamente foi dobrado o número de núcleos por camada.
- 9 camadas convolucionais com três blocos de conexões onde novamente foi dobrado o número de núcleos por camada em relação às 18 anteriores.
- Agrupamento por média global;
- 1 camada densamente conectada de 1000 nós, seguida por uma função de ativação sigmoide na saída. A função é designada por

$$f(x) = \frac{1}{1 + e^{-x}},$$

de modo que o seu valor mínimo é 0 e o máximo, 1. As previsões, portanto, ficam dentro desta faixa.

Somando todos os itens, excetuando os passos de agrupamento, temos portanto as 50 camadas da ResNet50. Com esta construção, a ResNet50 é uma rede que possui cerca de 23 milhões parâmetros.

3.5.2 EfficientNet B2

A EfficientNet é uma classe de redes neurais propostas por Tan & Le [99] em 2019, variando de $B0$ à $B7$, trabalho sobre o qual nos basearemos em grande parte a escrita desta subseção. A sua motivação se deve a, assim como a ResNet, resolver os problemas provenientes da inserção de muitas camadas numa RNC, já abordados na subseção 3.5.1. No intuito de dar prosseguimento ao escalonamento de redes neurais, Tan & Le [99] propõem o que se chama por *método de escalonamento composto*¹³ Entre algumas alternativas comuns para melhorar a acurácia de uma RNC tem-se: o aumento das dimensões das camadas usando imagens de maior resolução, o aumento da profundidade e da largura das camadas.

¹³ tradução livre de *compound scaling method*.

O método de escalonamento composto (doravante MECO) é uma maneira de automatizar esse processo, utilizando alguns coeficientes fixos. Assumindo que uma camada convolucional de uma rede é uma função que realiza a operação $F(X)$ sobre dados de entrada X , o MECO se ocupa por expandir o comprimento, profundidade ou resolução do bloco da rede sem alterar a operação $F(X)$ realizada. Todos os blocos devem passar por esse processo de escalonamento uniforme que opera de modo que, para um coeficiente fixo ϕ , a profundidade p da RNC aumente de acordo $p = \alpha^\phi$; largura l de acordo $l = \beta^\phi$ e a resolução das imagens r de acordo $r = \gamma^\phi$. As únicas exigências sobre α , β e γ são que

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2; \quad (3.9)$$

isto é, o escalonamento é quadrático com a largura e resolução mas linear com a profundidade, e

$$\alpha, \beta, \gamma \geq 1. \quad (3.10)$$

Uma vez realizada a sondagem destes índices de modo experimental, é possível construir toda a gama de variações da EfficientNet, da B0 à B7. Cada uma destas terá em suas dimensões os mesmos índices α , β e γ , porém, com coeficientes ϕ diferentes.

Para a EfficientNet B0 ficou definido $\phi = 1$, e foi encontrado através de destes valores para α , β e γ que satisfazem 3.9 e 3.10. A partir daí, para construir as redes seguintes o valor de ϕ foi escalando. Assim, a EfficientNet B1 é um pouco mais profunda que a B0, que é um pouco menos do que a B2, e assim por diante até a B7, que é a mais profunda. Mas, devido ao MECO, enquanto a profundidade aumenta, também aumenta a largura e a resolução, seguindo as proporções. O principal problema em fazer este ajuste em redes neurais de modo geral é que não é simples encontrar quais são os melhores parâmetros sem alterar a operação $F(X)$. O trunfo da EfficientNet é encontrar uma maneira de relacionar estes parâmetros, ao custo de um maior poder computacional devido a possuir menos parâmetros que RNCs mais profundas de performance semelhante (ver resultados em [99]).

Entre a classe de EfficientNets, vamos nos restringir a EfficientNet B2 por conta de ter sido a utilizada neste trabalho. A escolha pela EfficientNet B2 deu-se devido a estar em um cenário intermediário em relação às EfficientNets - nem profunda demais quanto a B7, nem rasa demais quanto a B0. As EfficientNets funcionam num esquema de blocos, de modo que, para cada uma delas, tenhamos sempre a mesma sequência de blocos que seguem a estrutura da figura 3.10.

Esse esquema de bloco é chamado por Bloco Convolucional de Gargalo Invertido Mobile¹⁴ (MBConv). Caracteriza-se por iniciar com, observando a figura 3.10 da esquerda para a direita, uma convolução comum Conv2D seguida por uma convolução de profundidade¹⁵, representada na figura por *DepthwiseConv2D*. Esta convolução se caracteriza por

¹⁴ tradução livre de *Mobile Inverted Convolutional Bottleneck Block*.

¹⁵ tradução livre de *Depthwise Convolution*

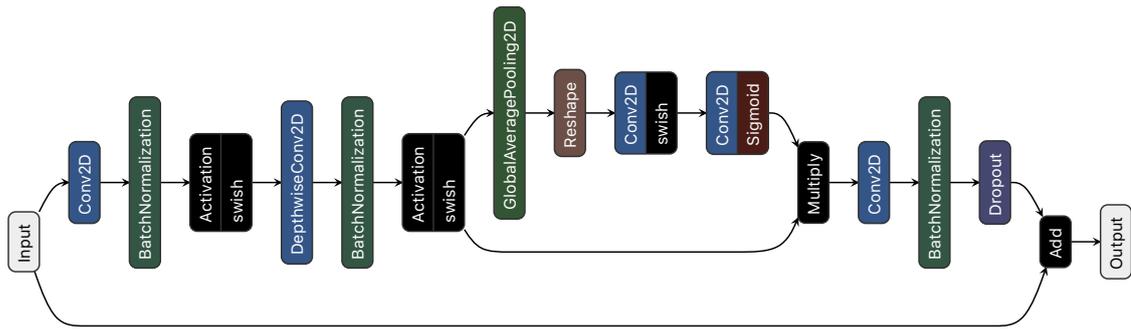


Figura 3.10 – Bloco Convolutivo de Gargalo Invertido da EfficientNet. A principal característica é de que ele inverte as conexões de atalho de modo que, ao invés de conectar saídas de blocos convolucionais 1x1, ele conecta os próprios blocos convolucionais centrais.

manter a profundidade do tensor de entrada na saída, através da realização de convolução em cada um dos canais de profundidade do tensor de entrada [100]. Depois, estes tensores são sobrepostos de modo que a profundidade se mantenha a mesma. Em seguida, temos o gargalo móvel invertido dentro do qual estão camadas convolucionais de núcleos de tamanho 1x1. Chama-se por "gargalo invertido" por as camadas de dentro da conexão de atalho possuírem núcleos pequenos, enquanto as de fora do gargalo tem núcleos maiores - o contrário do que acontece no bloco de conexão de atalho da ResNet50. O objetivo da adoção desse tipo de gargalo é o mesmo do gargalo adotado na ResNet50, a busca por melhorar a eficiência do fluxo do gradiente através das camadas. O esquema de gargalo invertido, no entanto, é mais eficiente do que o esquema de gargalo comum quanto ao uso de memória computacional [101]. Já o adjetivo "mobile" se deve a arquitetura onde ele foi proposto, a MobileNet e MobileNetV2.

Uma nova convolução ao fim seguido de um recurso no ramo de aprendizado de máquina chamado por *dropout* encerra este bloco. O *dropout* consiste de reiniciar alguns neurônios de certas camadas da rede aleatoriamente, que são reativados na próxima execução da rede. No entanto, o *dropout* pode fazer com que os neurônios daquela camada se vejam obrigados a buscar compensar a falta dos neurônios reiniciados, tornando-os mais especializados [102]. Este recurso pode deixar a classificação mais robusta.

Finalmente, agora como arquitetura geral para a EfficientNet B2, temos o esquema da figura 3.11. Optou-se por, nesta visualização, agrupar camadas de MBConv2D que se repetem ao longo de toda a rede, diferenciando-se umas das outras pelo número de canais de saída.

Esta arquitetura de blocos se mantém a mesma para todas as classes de EfficientNet, o que se modifica é o número de algumas camadas por blocos e, principalmente, o número de canais de saída [103]. Para a EfficientNet B2, o número de canais está expresso na figura, calculado através do MECO.

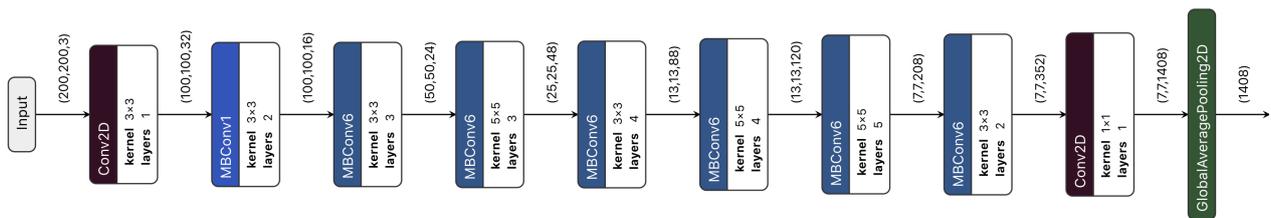


Figura 3.11 – Organização em blocos da EfficientNet B2. Há um bloco de entrada seguido de uma camada convolucional comum, após o qual há os blocos de Gargalo Móvel Invertido até a última camada.

CAPÍTULO 4

DETECÇÃO AUTOMATIZADA DE LENTES GRAVITACIONAIS COM REDES NEURAIIS ARTIFICIAIS

4.1 Histórico da Busca automatizada de Efeito Forte de Lente

Cabe agora realizar uma revisão da detecção de Lentes Gravitacionais utilizando redes neurais, que segue como proposta deste trabalho. Iremos repassar algumas maneiras de se observar o efeito forte de lente.

Uma das primeiras tentativas de se criar um algoritmo que faça a busca por arcos e anéis gravitacionais foi a proposta de Lenzen et. al [104] em 2004. O algoritmo consistia de um programa em linguagem C que usava a difusão anisotrópica para destacar os arcos do plano de fundo das imagens, extraídas do telescópio Hubble e do conjunto de telescópios VLT (*Very Large Telescope*). Starck et. al (2003) [105] usou transformadas *Wavelet* com o objetivo de reconstruir imagens ruídas, no entanto, a técnica permitia evidenciar estruturas alongadas como arco arcos gravitacionais. Outra iniciativa de 2007, por Cabanac et. al [106], utilizou dados do levantamento CFHTLS (*Canada-France-Hawaii Telescope Legacy Survey*) para realizar a busca de arcos gravitacionais num algoritmo de busca de objetos alongados para posterior inspeção visual, anéis de Einstein analisando o desvio pro vermelho fotométrico dos objetos no levantamento, e por fim, objetos de imagens múltiplas, englobando casos de lentes que não pudessem ser detectados pelo buscador de arcos. Os resultados do artigo já foram fruto da iniciativa de um novo projeto, focado na detecção automatizada de efeito forte de lente, o SL2S (*Strong Lensing Legacy Survey*).

Em 2006, Alard et al. desenvolveram o *Arcfinder*, um código computacional para encontrar arcos nas imagens da segunda edição do levantamento CFHTLS [107]. A estratégia desse código era utilizar um estimador local do alongamento de objetos na imagem do

levantamento que, caso se adequassem a uma função, eram indicadas como candidatos a arco gravitacional para posterior inspeção visual.

Um dos projetos de detecção de candidatos em que ocorre efeito forte de lente é a iniciativa do SLACS (*Sloan Lens ACS Survey*). O SLACS é um projeto de levantamento de dados que utiliza espectroscopia para encontrar efeito forte de lente no massivo catálogo do SDSS (*Sloan Digital Sky Survey*) [108]. O projeto identifica objetos próximos a diferentes desvios pro vermelho e, uma vez detectado um bom candidato, os dados de sua localização são enviados para a *Advanced Camera for Surveys* (ACS), que está a bordo do telescópio espacial Hubble. Assim, a câmera é apontada para o alvo e, caso hajam as distorções típicas do efeito forte de lente já abordadas na seção 2.2 do capítulo 2, o objeto é registrado como caso positivo de lenteamento. Na primeira publicação de seus resultados, o SLACS encontrou 19 casos de lenteamento em escala galáctica e 9 casos prováveis [108]. Ao longo dos anos, muitos outros candidatos de lentes foram encontrados pelo SLACS usando a câmera ACS [109, 110, 111, 112, 113]. O método de descoberta se manteve mesmo para os lançamentos mais recentes [113]. Um dos achados mais notórios foi a descoberta de um anel de Einstein duplo [114].

Uma maneira bastante direta de se identificar arcos é através da própria inspeção visual, que mostrou-se efetiva para alguns casos. Nord et al. [115], e a colaboração DES (*Dark Energy Survey*), num trabalho de 2016 realizaram uma inspeção visual sobre os dados do primeiro ano do DES, e detectaram 53 candidatos a lente ao buscar arcos e anéis. Parte deles foi levado à observação e outros à análise espectroscópica, a fim de obter a confirmação. A motivação do uso da inspeção visual segundo o grupo, ao invés do uso de um método computacional, deu-se por conta de a área de visualização ser bastante pequena. Além disso, candidatos que forem evidentes o bastante para se encontrar pela inspeção visual já seriam facilmente encontrados por algoritmos. 6 candidatos foram detectados neste trabalho por este método e 9 em outra iniciativa mais recente do mesmo grupo, de 2020 [116]. Apesar da qualidade dos achados, não se pode negar o poder do recurso computacional. Também nos dados do primeiro ano do DES, Diehl et. al (2017) [117] encontraram 374 candidatos a lentes gravitacionais na iniciativa *DES Bright Arcs Survey*. Ainda que em alguns dos procedimentos adotados a inspeção visual tenha sido essencial, a maior parte do trabalho se deu utilizando recursos computacionais.

No ramo do aprendizado de máquina, provavelmente uma das primeiras publicações do seu uso na busca de arcos foi o trabalho de Estrada et. al (2007), nos dados do SDSS. O grupo selecionou 825 aglomerados do SDSS e realizaram uma inspeção visual em busca de arcos, em conjunto ao uso de um algoritmo automatizado usando uma rede neural. Depois de selecionar possíveis candidatos analisando suas dimensões, os parâmetros medidos desses candidatos entram na rede neural que irá determinar se este pode ser considerado um bom candidato [118]. Apesar de as imagens propriamente ditas não consistirem dos dados de entrada, esta iniciativa foi pioneira na automatização do

processo de detecção. Abordagens mais recentes estão no trabalho de Joseph et al. (2014) [119]. A técnica usada foi o PCA (*Principal Component Analysis*), uma técnica de aprendizagem de máquina que consiste de calcular uma projeção nos dados ao longo de um vetor que minimiza a variância. Foi utilizada uma base de dados pública do domínio *Bologna Lens Factory*¹, cujas imagens simuladas visam reproduzir possíveis observações do satélite Euclid (conforme [120]). A mesma técnica foi aplicada em 2016 por Paraficz et al. [121] mas, desta vez nos dados do CFHTLS, na busca por arcos em escala galáctica. Foram encontrados possíveis 109 casos de lente. Em ambas as iniciativas, estiveram presentes técnicas para remover a galáxia de fundo na imagem no intuito de facilitar a detecção dos arcos e, buscar como teste final aplicar o algoritmo em dados reais a fim de verificar a precisão. Bom et al (2017) [122], propôs um buscador de arcos baseado no método de filamentação por mediatrizes usando Redes Neurais. A metodologia empregada foi realizar o treino numa base de dados simulada pelo gerador AddArcs [123] e, depois utilizar na classificação dados de imagens reais contendo arcos e não-arcos do telescópio espacial Hubble, extraídas pelo software SExtractor.

O uso das redes neurais convolucionais neste contexto de detecção é consideravelmente recente, de 2017, nas abordagens de Petrillo et al. [124] e Jacobs et al. [125]. Petrillo et al. utilizaram uma RNC para encontrar efeito forte de lente nos dados do KiDS (*Kilo-Degree Survey*), priorizando lenteamento por LRGs (*Luminous Red Galaxies*). Entre 21789 LRGs do catálogo, foi prevista pela RNC a ocorrência de 761 candidatos a objetos sofrendo lenteamento. Dois destes candidatos são ocorrências já conhecidas de lenteamento gravitacional, entre os três que já se conhecia na região.

Já Jacobs et al. realizaram uma classificação na base de dados do CFHTLS, gerando um conjunto de treinamento próprio de lentes de duas maneiras. Uma delas foi através de uma modificação de um código de geração de simulações de imagens de efeito forte de lente conhecido por LensPop². A base de dados LensPop visa reproduzir aspectos das imagens que se farão presentes em levantamentos futuros à época de seu lançamento em 2015, como os extraídos pelo DES, Euclid e LSST (*Legacy Survey of Space and Time*) [126]. A outra estratégia consistiu de tomar dados reais do CFHTLS e adicionar artificialmente efeito forte de lente à algumas das imagens. Após treinados nestas bases de dados separadamente, os códigos foram aplicados a toda a dimensão do CFHTLS, encontrando mais de cem possíveis casos de efeito forte de lente.

Posteriormente, muitos trabalhos utilizando aprendizagem profunda com RNCs surgiram [127, 128], mirando em levantamentos de grande área [128, 129, 130] ou, inspirados por competições de criar arquiteturas eficientes na busca do efeito forte de lente [131]. Basta comparar a quantidade de objetos identificados através da busca visual com a quantidade de identificados por RNCs, para perceber o potencial. Embora haja a preparação

¹ <https://bolognalensfactory.wordpress.com>

² <https://github.com/tcollett/LensPop>

de uma amostra de treino, teste e de candidatos a serem avaliados não seja uma tarefa trivial, o número de candidatos que são descobertos pelas RNCs é promissor.

No que se refere especificamente à arquitetura ResNet, utilizada neste trabalho, uma abordagem recente por Li et. al. (2021) [132] já utilizou a arquitetura [94] para criar uma RNC que encontre arcos nos dados do KiDS, usando como dados de treinamento as próprias imagens do levantamento com a adição artificial de arcos. Uma variação da ResNet18 também foi usada por Canameras et al. (2021) [133] para encontrar lentes nos dados do levantamento HSC (*Hyper Suprime-Cam*³). Especificamente a ResNet50, foi utilizada no trabalho de Stein et al. (2021) para encontrar candidatos a lente nos dados do DESI (*Dark Energy Spectroscopic Instrument*⁴), no contexto de aprendizado não-supervisionado. No mesmo levantamento também foi usada a ResNet46 para encontrar candidatos a lente [134]. Seja gerando as próprias bases de dados a partir de códigos já disponíveis na rede [135, 136] ou baseando-se em imagens geradas de levantamentos [137, 138, 139, 140] para treinar seus algoritmos, a comunidade científica já descobriu o potencial da ResNet na classificação de imagens.

Quanto à EfficientNet, a de classe B0 foi recentemente utilizada por Rojas et al. [141] (2021) para encontrar candidatos a efeito forte de lente nos dados da primeira edição do DES. A EfficientNet B0 também foi utilizada por Savary et al. (2021) [142] para encontrar candidatos a lente nos dados do levantamento CFIS (*Canada-France Imaging Survey*⁵). De fato, as EfficientNets e outras RNCs já encontraram uso geral no campo de detecção do fenômeno de lenteamento gravitacional não apenas para classificação [143], mas também na medida de parâmetros cosmológicos [144]. Cabe pontuar que muitos destes trabalhos lançaram mão, assim como o presente, do uso da AUC como parâmetro de estimar a eficiência da RNC [145, 146, 143]. Inclusive, as duas edições da competição de detecção de dados de lentes, responsáveis geraram as bases de dados da qual trataremos a seguir e das quais fizemos uso, tiveram a AUC como uma de suas métricas de ranqueamento.

4.2 As bases de dados do Gravitational Lens Finding Challenge (GLFC)

O projeto *Bologna Lens Factory* gerou milhares de simulações de imagens de lentes gravitacionais fortes, incluindo as que foram utilizados ao longo deste trabalho. A ideia do projeto foi gerar bases de imagens de lentes gravitacionais simuladas de diferentes tipos, e torná-las disponíveis para a comunidade (ver [147]). Além disso, desafios como este são capazes de engajar grande parte da comunidade científica ao redor da astronomia, sejam os cientistas familiarizados ou não com o tema (ver e.g., [148]). As imagens simuladas

³ <https://hsc.mtk.nao.ac.jp/ssp/>

⁴ <https://www.desi.lbl.gov>

⁵ <https://www.cfht.hawaii.edu/Science/CFIS/>

seguiram a prescrição de dois *surveys*, o KIDS e o EUCLID no caso do primeiro e segundo *data challenge*, respectivamente.

O KiDS (*Kilo-Degree Survey*⁶) é um levantamento de dados obtidos a partir do céu do hemisfério sul, utilizando as bandas em ótico do VLT (Very Large Telescope⁷). O levantamento foi realizado sobre uma área de 1500 graus quadrados em quatro filtros (U, G, R, I), ao longo de um período entre 3 e quatro anos que, ao ser combinado com dados do levantamento VIKING (*VISTA Kilo-Degree Infrared Galaxy Survey* [149]) realizado sobre o telescópio VISTA (*Visible and Infrared Survey Telescope for Astronomy*⁸) na faixa do infravermelho, fornecem uma base de dados de 9 bandas [6]. O tempo de exposição foi escolhido de modo a captar galáxias na faixa de desvio para o vermelho de 0,8. Depois de se completar todo o levantamento sobre os 1500 graus, toda a área foi novamente imageada apenas na banda *g* utilizando os mesmos parâmetros das observações anteriores, mas realizado ao longo de 2 anos. O intuito dessa repetição é aumentar a acurácia de medições do movimento próprio dos corpos. Apesar de ser designado principalmente para o estudo do fenômeno de lenteamento fraco e medir desvios pro vermelho fotométricos, os dados do levantamento já encontrou aplicações em outros tópicos de pesquisa, como a matéria escura, evolução de galáxias e aglomerados e a estrutura da Via Láctea [6].

A missão Euclid é a missão espacial que pretende lançar o telescópio espacial Euclid em 2022, cujo objetivo é contribuir na compreensão da matéria e energia escuras [150]. Acerca do telescópio, pode-se dizer que a parte do equipamento que realizará as observações é constituída por dois instrumentos, o instrumento VIS (*Visible Instrument*) e o NISP (*Near-Infrared Spectrometer and Photometer*) [151]. O VIS ficará responsável pelo imageamento visual, precisamente na faixa de comprimentos de onda de 550-900 nm, com sensibilidade de 25 mag para fontes extensas, tamanho de pixel de 0.1 arcseg e um campo de visão de 0.787×0.709 graus [152]. Já o instrumento NISP é um espectrômetro e fotômetro infravermelho, cujo canal fotométrico estará equipado com três filtros de banda Y, J e H, respectivamente nas faixas de 900-1192 nm, 1192-1372 nm e 1372-2000nm [153]. Espera-se que cubra uma área do céu de 0.763 x 0.722 graus, e que em todas as bandas haja uma sensibilidade de 24 mag para fontes pontuais, e tamanho de pixel de 0.3 arcseg.

Essas especificações técnicas do telescópio e das bandas de imagem foi o que se tentou reproduzir com as simulações que geraram a base de dados da primeira edição do GLFC. As imagens base foram feitas utilizando uma simulação de N-corpos, e nelas foram anexados e modelados os halos de matéria escura dos objetos e o efeito de lenteamento. Os dados foram projetados dentro de um cone de luz de 1.6 graus quadrados, estendendo-se até o desvio para o vermelho $z = 6$; enquanto o efeito de lenteamento de lentes extensas foi incorporado mapeando as cáusticas no intervalo de $z = 1$ até $z = 3$ com subdivisões de 0.1 nesse intervalo, e de $z = 3$ até $z = 6$ com subdivisões de 0.25 [154]. Os halos foram

⁶ <https://kids.strw.leidenuniv.nl>

⁷ <https://www.eso.org/public/brazil/teles-instr/paranal-observatory/vlt/>

⁸ <https://www.eso.org/public/teles-instr/paranal-observatory/surveytelescopes/vista/>

utilizados como as lentes e, os objetos que servem de fonte foram retirados do *Hubble Ultra Deep Field* (HUDF)⁹. O código que gera as simulações, de acordo Metcalf et. al [154], é capaz de combinar galáxias no plano de fundo, imagem lenteada e a quantidade de ruído desejada. O resultado são imagens como a da figura 4.1. Em algumas poucas o efeito forte de lente é claramente discernível, já em outras é mais difícil identificá-lo.

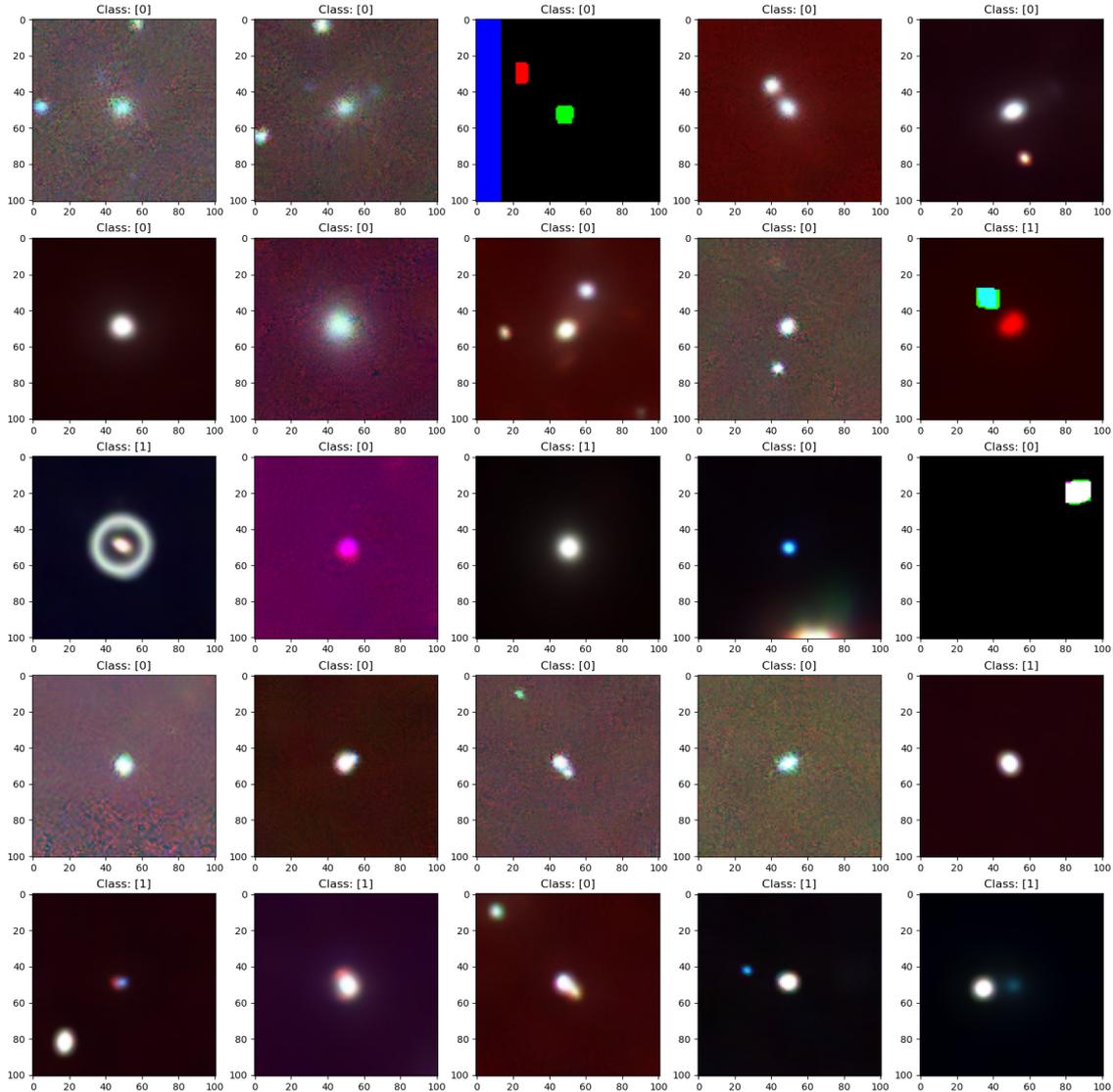


Figura 4.1 – 25 exemplos de imagens combinadas das bandas R, G, e U da primeira edição do GLFC. Fica evidente o ruído e a poluição visual que se espera de imagens obtidas a partir de um telescópio com a poluição atmosférica.

Na primeira edição do GLFC, foram fornecidos dois conjuntos gerados na direção de replicar os dados de acordo diferentes tipos de observações. Uma delas imita as imagens geradas a partir da visualização espacial que se espera obter do instrumento VIS do telescópio espacial Euclid, no que se configura como imagens *space-based*. São 20000 imagens com uma tabela verdade e 100000 imagens sem identificação, em uma banda

⁹ Já citado no capítulo 1, ver [155]

única. Já outro conjunto de imagens se propõe a imitar as imagens obtidas a partir de um telescópio da superfície da Terra, simulando as imagens do KiDS em quatro bandas: U, G, R e I. As imagens simuladas com essa ótica se configuram como imagens *ground-based*, e são constituídas por 20000 imagens de treino com quatro bandas cada, acompanhadas de uma tabela verdade, e outras 100.000 imagens sem identificação, também com quatro bandas. Para este trabalho, no primeiro momento nos ocupamos de utilizar as 20000 imagens de treino *ground-based*, nas bandas U, G e R, da primeira edição do GLFC.

No segundo momento deste trabalho, já utilizamos as imagens da segunda edição do GLFC. Foi disponibilizado pelo desafio dois arquivos, um de treino com imagens em quatro bandas, H, J, Y e VIS e um de teste com as imagens em mesmas bandas. Ambos os arquivos possuem 100.000 imagens e, um dos mais notórios diferenciais em relação à primeira edição da competição é que as imagens possuem dimensões diferentes. De modo geral, as imagens foram inspiradas em observações que se espera serem obtidas pela missão espacial Euclid, consistindo portando de imagens *space-based*, se importarmos a classificação da edição anterior. As bandas H, J, e Y consistem de recortes alguns comprimentos de onda já mencionados nesta seção em dimensões 66x66 pixels, enquanto a banda que simula as imagens do instrumento que fará observações no comprimento de onda visível possui dimensões de 200x200, e de maior resolução. Outro diferencial em relação ao desafio anterior é que os dados de treino não vieram acompanhados de uma tabela verdade, mas de um arquivo de dados astrofísicos de cada objeto simulado. Os organizadores definiram um critério a ser adotado e, com base nisso elaboramos uma tabela verdade própria seguindo esse critério, a fim de realizar a classificação. Os detalhes estarão mais explícitos na seção 5.3.1.

Essas duas bases de dados deverão ser capazes de fornecer arcabouço suficiente para um estudo preliminar da detecção de lentes gravitacionais usando redes neurais, do mesmo modo que já foram utilizadas extensamente em outros trabalhos [136, 146, 135, 138, 139, 143].

CAPÍTULO 5

RESULTADOS

Neste capítulo apresentamos o processo investigativo executado durante a pesquisa, acerca do problema de realizar classificação de um conjunto de imagens de lentes gravitacionais e, em particular, o regime de poucos dados. Por isso, está organizado de modo que cada seção seja um novo passo dado na investigação do problema, de acordo as lacunas encontradas no passo anterior ou, de modo a dar mais substância ao resultado obtido anteriormente.

Todos os códigos utilizados para reproduzir a performance dos modelos em cada caso encontram-se presentes no domínio público <https://github.com/KayqueTeles/DISSERTACAO.git>, cujos componentes serão referenciados ao longo de cada seção.

5.1 Problematização

Conforme exposto no capítulo 4, já houveram duas competições recentes de inteligência artificial para a classificação de lentes gravitacionais, os *GLFC*. A proposta era de criar o melhor classificador de uma base de dados de imagens simuladas de telescópios *ground-based* (no primeiro GLFC) e uma banda *space-based*. Os dados incluíram a adição de ruído e outros efeitos observacionais pela própria natureza dos equipamentos de observação, com base numa tabela verdade ¹. Por isso, consistem de bom material para início do estudo. O fato de esses dados já terem sido classificados por redes neurais, já estarem prontos para uso e tabelados contornam o problema de classificar imagens reais em que, seria necessário realizar cortes nas imagens que isolassem bem os objetos e, organizá-los de acordo a sua natureza. Um classificador que seja eficiente em detectar imagens reais de lenteamento gravitacional deve ter facilidade para detectar o mesmo fenômeno em imagens simuladas. Esta é a base de dados sobre a qual foram realizados os primeiros testes neste trabalho,

¹ A base de dados e a tabela verdade das duas edições pode ser obtida em http://metcalfe1.difa.unibo.it/blf-portal/gg_challenge.html

até o momento em que obtivemos os resultados desejados.

A problematização inicial, portanto, consiste de utilizar uma arquitetura de rede neural para realizar classificação sobre essa base de dados. A escolhida foi a vencedora da edição do *ILSVRC* de 2015, a arquitetura ResNet50 [94], já abordada na subseção 3.5.1. Além de toda a validação que o seu uso em larga escala para diversos projetos supracitados poderia fornecer, a sua implementação é bastante simples na linguagem de programação escolhida para os nossos estudos - o *python*. A rede pode ser importada com apenas uma linha de código de uma biblioteca de acesso aberto - o *tensorflow-keras*[156].

5.2 Desempenho da ResNet50 nos dados da primeira edição do GLFC

Para utilizar a base de dados *ground-based* da primeira edição do *GLFC*, que consiste de 20000 objetos, fez-se necessário um pré-processamento dos dados. Na entrada do CBPF da primeira edição *GLFC*, verificou-se que utilizar diferentes estratégias para realizar a inserção dos dados na rede retornava diferentes pontuações de AUC [154]. Tanto no caso *ground-based* quanto no caso *space-based*, a melhor alternativa de pré-processamento entre as possibilidades testadas consistiu de executar um ajuste de contraste e um filtro de Wiener[157] em cada uma das três bandas utilizadas (U, G e R).

Conforme o sugerido para um problema de classificação (ver capítulo 3, faz-se necessário estabelecer uma divisão dessa base de dados em conjuntos de treino, teste e validação. A divisão utilizada na primeira execução foi de reservar 10% dos dados para validação, 10% para teste e os 80% restantes para treino. Recuperando as métricas de avaliação dos resultados já abordadas na seção 3.4, nas primeiras execuções da rede buscou-se analisar o custo e a acurácia do treino ao longo das épocas de execução e, seu desempenho em um teste cego e um teste aberto.

Escolhemos realizar o treino com 50 épocas, com uma taxa de aprendizado de 0,01, *batch size* de 64 itens e de modo a otimizar uma função custo de entropia binária cruzada². Dado que acurácia e o custo da rede são formadas por valores que são calculados e registrados a cada época, ao final de toda a fase de treino é possível visualizar um registro das flutuações destes índices. Para a acurácia, este registro se encontra na figura 5.1.

Enquanto a linha azul se refere a acurácia da classificação sobre o próprio conjunto de treino da rede, a linha laranja se refere à acurácia sobre o conjunto de validação. É possível verificar que, no caso da acurácia, no início das épocas enquanto a acurácia de treino começa a partir de um certo valor, a acurácia de validação começa do zero. Ao longo das épocas ocorre um rápido crescimento da acurácia de validação, o que indica que há um rápido aprendizado da rede já no início. O esperado é que ao longo das épocas haja

² O código utilizado para reproduzir os dados desta seção é o arquivo "5-2.CH1.Resnet.py" presente no domínio público <https://github.com/KayqueTeles/DISSERTACAO.git>

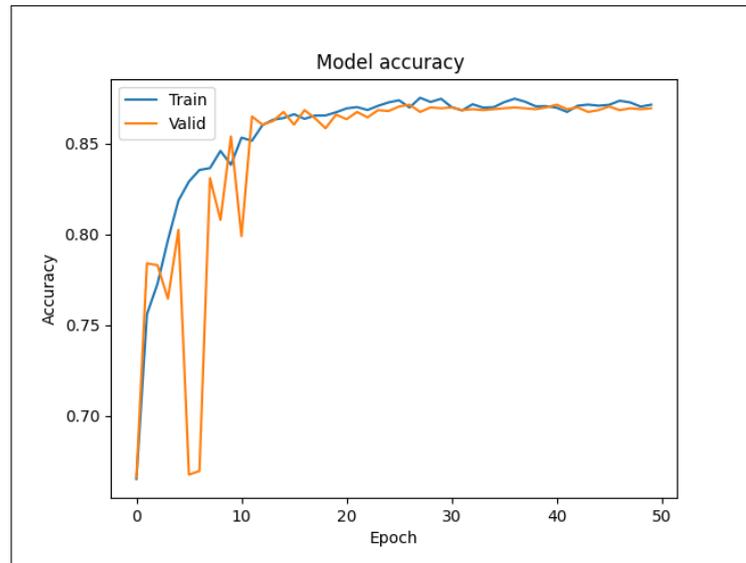


Figura 5.1 – Gráfico de Acurácia da rede ResNet50 em toda a base de dados.

uma convergência cada vez maior das duas linhas, até que idealmente elas se encontrem. Neste caso, isto não acontece evidentemente, mas observando os intervalos do gráfico de cerca da época 10 até a época 50, é possível ver que os valores dos dois oscila entre 0.8 e 0.9. Isto é, estão relativamente próximos. Além disso, cabe lembrar que esta é uma primeira tentativa de classificação dos dados da rede, que ainda será bastante refinada.

As primeiras oscilações na acurácia de validação dizem respeito a fortes penalidades aplicadas pela própria rede em pesos muito destoantes, através da função custo. A fim de otimizar os pesos, a rede penaliza bastante conexões de neurônios que não contribuíram bem para a classificação. O resultado é uma forte oscilação da acurácia de validação, até o momento em que as penalidades surtem efeito e os pesos começam a se aproximar de um valor otimizado para a classificação. A consequência disto é a convergência para a acurácia de treino.

Já no caso da função custo, conforme já abordado no capítulo 3, o esperado é que seja uma tendência de minimização. Neste contexto, o gráfico gerado está presente na figura 5.2.

Verifica-se na figura 5.2 a reprodução de um comportamento bastante comum para o gráfico da função custo em problemas de classificação, que se trata do início quase assintótico na primeira época para, em seguida, descer até atingir uma convergência para zero. Da mesma maneira que no gráfico da acurácia, a linha azul é referente ao custo calculado sobre os dados de treino, enquanto a amarela é sobre os dados de validação. No gráfico reproduzido, se verifica uma forte tendência de sobreajuste a partir da época ~ 20 , sugerindo que devemos usar a rede treinada em torno dessa época apenas.

Por fim, a última métrica adotada a fim de avaliar a classificação é a curva ROC. Consiste nada mais do que a avaliação do modelo em um teste cego, em que é registrado a cada limiar entre as possibilidades de classificação a taxa de falsos positivos com a de

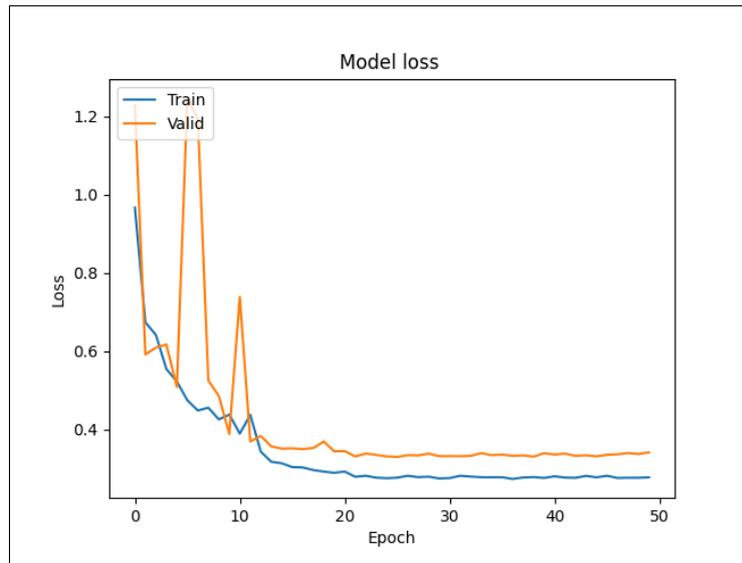


Figura 5.2 – Gráfico da Função Custo da rede ResNet50 em toda a base de dados.

positivos verdadeiros (ver subseção 3.4). Diferentemente das métricas anteriores, a ROC é calculada após o treino e assim, utilizando o modelo final da rede, temos para a curva o gráfico da figura 5.3.

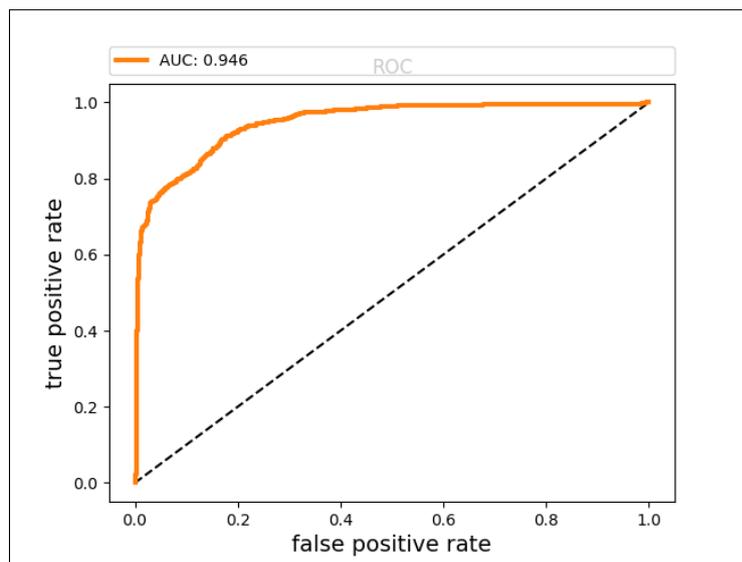


Figura 5.3 – Gráfico da curva ROC da rede ResNet50 em toda a base de dados.

Evidentemente, há uma reprodução do comportamento característico para curvas ROC já abordado: a subida bastante abrupta da taxa de falsos positivos inicialmente, seguida da convergência para o máximo de falsos positivos posterior conforme o limiar se torna alto demais. A linha tracejada indica a linha que teria que ser descrita pela curva para que a AUC fosse de 0.5. Na classificação que resultou na figura 5.3, o resultado para a AUC foi de 0.946, valor bastante próximo de 1.

Esse resultado da ROC nos indica que a ResNet50 já se mostrou ser bastante eficiente na classificação da base de dados da primeira edição do GLFC. Verifica-se que o valor

de AUC se aproxima do que foi obtido pelas redes que melhor performaram na primeira edição da competição, de modo que ficaria em quarto lugar na tabela de classificação (ver [154]). O seu sucesso já era conhecido, mas reproduzir esse dado confere a certeza da replicabilidade deste. Enquanto as acurácia e custo se mantiveram em convergência e sob controle, respectivamente, a AUC se aproximou bastante do ideal valor de 1.0. Isto, à princípio, resolve o problema de encontrar uma rede neural que execute a classificação nessa base de dados.

5.2.1 A investigação de um limite de poucos dados

Os resultados da seção anterior mostram a performance em todo o conjunto de dados. No entanto, conforme já abordado no capítulo 4, um problema comum na implementação de uma busca automatizada de arcos gravitacionais é a raridade do evento. Seria muito conveniente ter uma base de dados reais tão grande quanto o tamanho dessa base de dados simuladas, no entanto, não existe um número tão grande de registros desse fenômeno até o presente momento. Por isso, se torna relevante o desenvolvimento de um buscador de arcos gravitacionais, que pudesse ser treinado com um conjunto restrito de dados reais. Dessa forma o método não dependeria de um modelo de lente pré-estabelecido ou de alguma hipótese adicional sobre os dados. Quanto menor for este número, a rede neural receberá menos informações acerca da distinção entre imagens de galáxias comuns e imagens onde está ocorrendo o fenômeno de lenteamento. Assim, uma vez confrontada com imagens do fenômeno não vistas previamente, a rede neural possuirá uma dificuldade maior em discernir o fenômeno, uma vez que a sua capacidade de generalização poderá ficar prejudicada devido a baixa variedade.

No intuito de prosseguir com essa motivação, começamos a reduzir a quantidade de dados de entrada na rede, fixamos o tamanho do conjunto de validação e, acrescentamos o restante ao conjunto de teste. Uma vez treinando com menos dados, espera-se que a rede perca um pouco da sua capacidade de predição. A adição dos dados restantes ao conjunto de teste não impacta o desempenho da rede neural, poderíamos manter a mesma quantidade de dados de teste usadas na seção anterior. Possuir um conjunto de teste maior apenas adiciona mais pontos às curvas de predição no teste cego e, portanto, nos permite ter um pouco mais de precisão no desempenho da rede com novos dados. Seguindo essa estratégia, escolhemos alguns intervalos de valores para ir diminuindo a eficiência do treino até que a classificação se tornasse a pior possível - próximo do regime aleatório. A avaliação deste desempenho da classificação será feita pelos métodos de avaliação de resultados já discutidos previamente, mas em especial observando a AUC.

Uma preocupação importante a partir desse ponto é a randomização dos dados de entrada. Conforme fomos diminuindo o tamanho do conjunto de treino, a influência individual de cada imagem da base de dados, na eficiência da rede como um todo, aumenta. Portanto, podem haver grandes variações na capacidade de predição da rede se houver

qualquer viés na escolha desses dados. Um pequeno conjunto de dados onde o fenômeno a ser classificado esteja claramente discernível pode performar melhor do que uma base de dados maior, com dados mais confusos. Uma maneira de se contornar esse problema, à princípio, é sempre randomizar a base de dados na entrada, de modo a garantir que não haja privilégio na escolha das imagens. Essa randomização deve vir, inclusive, antes da separação da base de dados em conjuntos de treino, validação e teste, para garantir que também não ocorra o mesmo problema nestas outras duas etapas.

Portanto, o procedimento para a classificação diminuindo o tamanho do conjunto de treino foi o seguinte: a cada execução,

- embaralhe as imagens e a tabela verdade correspondente;
- separe uma certa quantidade de dados para validação;
- tome o conjunto de treino de acordo o tamanho da amostra cujo desempenho visamos analisar;
- use o restante dos dados como conjunto de teste;
- construa um novo modelo e treine-o utilizando o conjunto de treino e validação;
- realize as predições com o conjunto de teste (ainda não visto pelo modelo).

Por motivos de simplificação do trabalho realizado, nos focaremos neste primeiro momento em observar o valor da AUC do modelo final da rede. O resultado deste estudo está presente no gráfico da figura 5.4, em que o eixo horizontal corresponde ao tamanho do conjunto de treino, enquanto o vertical corresponde à AUC. A linha tracejada que corta o gráfico corresponde à AUC de 0,5 em cada altura, a fim de observar a proximidade do valor de AUC do correspondente valor de treino à esse valor. Conforme já mencionado na subseção 3.4.5, o valor de 0,5 para AUC corresponde a um regime de completa aleatoriedade na classificação, portanto quanto mais próximo desse valor, pior é o poder de predição da rede no teste cego.

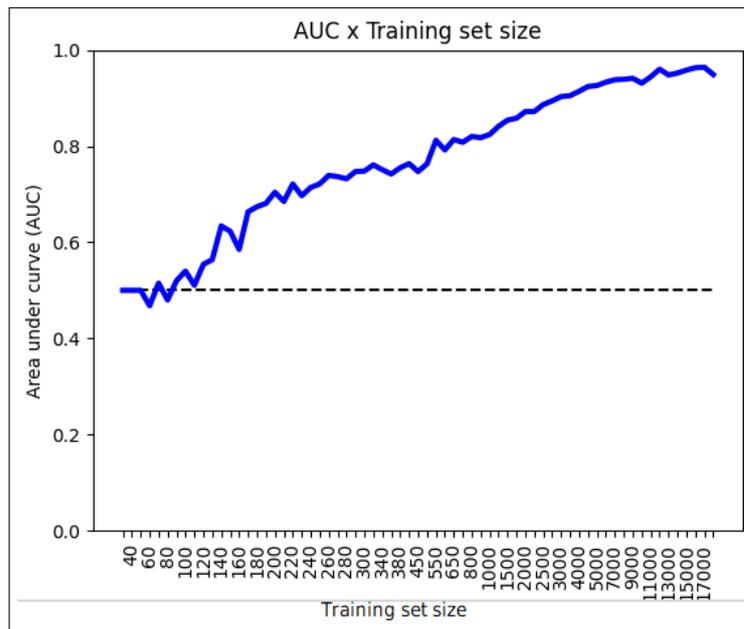


Figura 5.4 – Distribuição de AUCs de acordo o tamanho do conjunto de treinamento.

Como mostra a figura 5.4, na análise que foi realizada o valor máximo de AUC obtido no teste cego foi atingido na marca de 16000 objetos de treinamento, com AUC de aproximadamente 0.963. No contexto desta execução, foram utilizados 2000 objetos para validação e 2000 para teste, correspondentes a 10% cada da base de dados total, o que contribui para justificar a escolha dessa proporção de dados no contexto da seção anterior³.

Ao começarmos a avançar para o lado esquerdo do gráfico, percebemos que o desempenho da classificação se mantém bastante alto conforme diminuimos o tamanho do conjunto de treino, até uma porcentagem bem pequena do tamanho total da base de dados. Somente por volta dos 500 à 600 objetos em que a AUC começa a cair para abaixo de 0.8. Cabe apontar também que a continuidade do gráfico se deve apenas à ligação de cada par de pontos vizinhos, e não à ocorrência de treinos para toda a quantidade possível de objetos dentro desse intervalo.

De acordo com o critério sugerido por Hosmer (2013) [85], um regime “ótimo” para classificação seja o limiar de 0,8 para AUC, temos portanto que a faixa de 500 à 600 objetos de treino é uma faixa de quantidade de dados onde ainda é possível obter uma boa classificação em testes cegos. Deste modo, no problema de classificação de lentes e não lentes, seriam necessários portanto 600 objetos entre lentes e não lentes para ser possível afirmar que o modelo é eficiente em classificar os dados considerados. Esta quantidade, apesar de ser pequena em comparação ao tamanho da base de dados total utilizada neste problema, ainda seria um número bem grande de registros de fenômenos

³ O código utilizado para realizar esta execução é o mesmo do item anterior, isto é, o arquivo “5-2_CH1_Resnet.py”, no domínio público “<https://github.com/KayqueTeles/DISSERTACAO.git>”. A única alteração necessária é alterar o parâmetro que define o tamanho do conjunto de treino.

de lenteamento gravitacional forte para uma mostra homogênea de sistemas confirmados espectroscopicamente.

5.2.2 Múltiplos conjuntos de treino/teste

Embora não abordado na seção 5.2.1, conforme diminui-se o tamanho do conjunto de imagens de treinamento, pode-se verificar que sucessivos treinos com a mesma quantidade de dados pode gerar resultados com desvios significativos. Isto ocorre por conta do passo de randomização dos dados antes de sua separação, e também pelo componente estocástico do treinamento das redes, por exemplo, o uso de camadas de *dropout* e a inicialização dos pesos. Os conjuntos de treino que possuem uma melhor variedade e quantidade de objetos que poderão ser mais eficazes em realizar as previsões e, portanto, resultarão em valores de AUC maiores. Esse conjunto ideal não deve ser composto apenas por objetos facilmente identificados como lentes gravitacionais mas também objetos difíceis, de modo que haja a possibilidade de aprender pelo equilíbrio. Qualquer um dos casos diferentes pode acabar dificultando o treino - o que também justifica a randomização.

Pode-se concluir portanto, que quando diminui-se o tamanho do conjunto de treino, existe uma tendência a diminuir a variedade, o que pode prejudicar o desempenho da rede. Dado um conjunto fixo de simulações e reduzindo suficientemente, surge de modo natural a possibilidade de criar mais de um conjunto de treino, com imagens totalmente distintas, sem grandes prejuízos no conjunto de validação e de teste. Seria possível, por exemplo, dividir as 20000 imagens do GLFC de modo a usar 2000 para validação, 2000 para teste e, os 16000 restantes poderiam ser divididos em 10 conjuntos de treino de 1600 dados, permitindo assim, a realização de 10 treinos utilizando os mesmos conjuntos de treino e validação.

Um dos primeiros benefícios dessa estratégia é a análise do quão divergentes podem ser os resultados de acordo os dados que entram na rede, isto é, verificar o quão sensível está a classificação à escolha dos dados de entrada. Isto leva à natural consequência da possibilidade de realização de uma estatística sobre os resultados da rede, como o cálculo da média e/ou desvio padrão. Particularmente, podemos realizar essa estatística sobre a AUC do teste cego, uma métrica de bastante peso na avaliação final do modelo. Desta forma, torna-se possível calcular uma AUC média com barras de erro, calculadas a partir do desvio padrão. Outra utilidade para a medida inclui a possibilidade de verificar a ocorrência de *overfitting*, um problema comum na aprendizagem de máquina em que a rede neural consegue reconhecer bem os dados do conjunto de treino mas, tem dificuldades para generalizar os resultados para dados nunca antes vistos.

Como já mencionado, podemos adotar esta técnica no momento em que há uma quantidade suficientemente pequena de dados de treino entrando na rede neural, em um conjunto grande de simulações. Enquanto isto não ocorre, no entanto, ainda é possível definir uma estratégia para encontrar uma estatística do nosso modelo, inspirada num recurso

comumente empregado na aprendizagem profunda de máquina conhecido por *validação cruzada*. Uma das versões mais conhecidas da técnica de validação cruzada é o que se conhece por validação cruzada de k -*folds*, onde k é o número de subdivisões da base de dados. Nesta técnica, divide-se toda a base de dados que seria utilizada para treino e validação em k *folds*, ou k subconjuntos de mesmo volume de dados. Cada permutação destes conjuntos comporá a base de dados usada no treino do modelo, enquanto o restante é usado para validação.

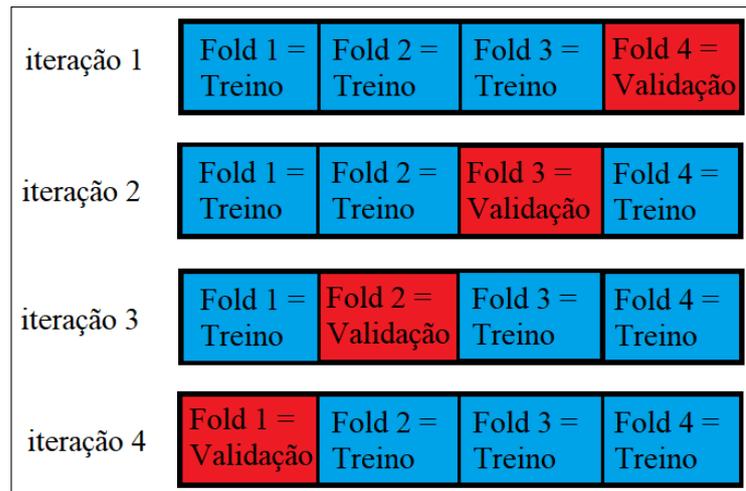


Figura 5.5 – Esquema representativo do uso de diferentes conjuntos de treino e de validação durante o treinamento de uma rede neural utilizando validação cruzada de k *folds* com 4 *folds*.

Conforme mostra a figura 5.5, na primeira iteração um dos *folds* é reservado para validação, enquanto os demais *folds* são agrupados para compôr o conjunto de treino. Já na segunda, outro *fold* é designado como conjunto de validação, enquanto os demais são reagrupados para compôr o conjunto de treino. Este processo é repetido k vezes até que todos os *folds* já tenham sido utilizados para validação em alguma das iterações. Conseqüentemente, o treinamento da rede foi executado k vezes, com diferentes conjuntos de treino de acordo as permutações possíveis no agrupamento de *folds*. A cada início de nova iteração é preciso reiniciar os pesos da rede, no intuito de obter diferentes resultados com cada agrupamento de *folds*.

É importante mencionar que, no presente trabalho, utilizamos este método apenas para obter diferentes conjuntos de treino. O conjunto de teste sempre foi separado antes da etapa de treinamento da rede e portanto, da subdivisão dos dados em *folds*. O motivo reside no interesse de ter um conjunto de teste de tamanho sempre fixo, e não dependente do tamanho dos *folds*. Neste trabalho, adotamos 10 como número fixo de *folds* para todas as ocasiões e, adotamos a estratégia da subdivisão dos dados de treino em *folds* nos contextos em que haviam mais de 1500 objetos no conjunto de treinamento da rede. Neste regime, a divisão dos dados é feita no esquema de subdivisão de dados em *folds*. Para abaixo de 1500 objetos no conjunto de treino, é possível obter 10 conjuntos de treino

totalmente disjuntos dentro do dataset de 20.000 dados. Por isso, julgamos adequado demarcar graficamente de alguma maneira a mudança de regime de múltiplos treinos, indicado na figura 5.6 por diferentes cores. Os pontos e barras demarcados em azul correspondem ao método de múltiplos treinos, enquanto os marcados por vermelho ao método inspirado na validação cruzada⁴.



Figura 5.6 – Distribuição de AUCs de acordo o tamanho do conjunto de treinamento e respectivas barras de erro.

Exceto pelas barras de erro, podemos ver que o gráfico da figura 5.6 é semelhante ao da figura 5.4, o que é de esperar visto que não adicionamos melhorias à classificação. No entanto, os pontos do gráfico 5.6 agora correspondem à mediana das AUCs obtidas pela predição da rede em cada uma das execuções, sobre o mesmo conjunto de teste. Apesar de muito pequenas, existem barras de erro também nos pontos em vermelho, porém, as margens estão tão próximas do valor original que podem à princípio parecerem ausentes. Poderia ser argumentado que isto se deve por conta de, nesse diferente regime de escolha dos conjuntos para os treinos consecutivos, os conjuntos de treino serem bastante parecidos e por isso, não serem muito divergentes um dos outros durante cada execução. Isto minimizaria bastante a margem de erro porque, usando conjuntos de treino bastante parecidos, não seriam esperadas grandes divergências no desempenho. No entanto, pode-se observar da figura que, nos pontos onde se mantém o regime de conjuntos de treino disjuntos próximos aos pontos onde ocorre a mudança de regime, as margens de erro já são bastante pequenas e já se confundem com os próprios pontos. Seria esperado, portanto,

⁴ Para gerar em código o gráfico acima, é preciso coletar os valores da AUC de múltiplas execuções, em dois diferentes regimes de escolhas dos folds. Um código em que essa implementação está feita é o arquivo “5-2-2.CH1_Resnet_Multi.py” presente em “<https://github.com/KayqueTeles/DISSERTACAO.git>”.

que mesmo que os conjuntos de treino com tamanhos superiores ao limiar de transição de regime fossem disjuntos, ainda teríamos pequenas margens de erro.

Por outro lado, conforme se prossegue para o lado esquerdo do gráfico é possível notar que as barras de erro vão aumentando quase progressivamente, até atingirem escalas que abrangem toda a margem de dados ou um dos lados. Portanto, na hora de observar esse limiar de otimização, podemos ver que na faixa dos 500 a 600 objetos as barras de erro ainda estão bem pequenas, apesar de haver um ponto dissidente no ponto dos 450 objetos - pode ser um ponto em que, usando este *batch size*, a rede não performa muito bem. Podemos assumir assim, que a escolha prévia de um regime razoável de classificação de 500 a 600 objetos se mantém, visto que nessas faixas as barras de erro ainda são pequenas.

Verifica-se portanto, o benefício do uso dos múltiplos treinos na classificação dos dados. A partir das divergências na classificação dados os diferentes conjuntos de treino, somos capazes de definir margens de erro para as execuções. Fica bastante clara a distribuição de regiões onde os resultados são bastante confiáveis dos locais onde estes não o são. Lançando mão deste recurso, podemos introduzir outras técnicas, agora, que ajudem efetivamente no treinamento do modelo e avaliar a sua contribuição para o problema em questão.

5.2.3 Avaliação da influência do pré-carregamento de pesos

Depois de se verificar a existência de um limite em que podemos, por convenção, classificar o desempenho da detecção como satisfatório, podemos pensar em alternativas que forcem ainda mais esse limiar. Até aqui já fizemos a estatística com múltiplos treinos, mas cabe a adoção de recursos que possamos inserir dentro da própria rede neural a fim de melhorar o desempenho da classificação. Nesta seção, avaliaremos o impacto do recurso chamado por “pré-carregamento de pesos” em nosso limite de poucos dados.

O treinamento de uma rede neural, como já vimos no capítulo 3, consiste, resumidamente, na otimização de seus parâmetros. Antes de começar a fase de treino, na construção do modelo aos seus parâmetros são atribuídos valores aleatórios que, de acordo as informações retiradas dos dados de entrada, vão sendo otimizados época à época de modo a se adequarem aos dados que estão entrando na rede. É possível que essa otimização dos parâmetros leve bastante tempo ou sequer ocorra, como ocorre no problema dos mínimos locais. O pré-carregamento de pesos consiste, apenas, de fornecer antes do treino da rede, valores otimizados em outra base de dados aos parâmetros iniciais da rede. Funciona como a importação do aprendizado da rede com outra base de dados, servindo de pontapé inicial para desenvolver aprendizado em uma nova base de dados. Mesmo que o objeto da classificação seja diferente ou ausente dos objetos presentes na base de dados treinada anteriormente, pode haver um impacto positivo na detecção da nova base⁵.

⁵ De fato, existe uma técnica de aprendizado de máquina focada especificamente na transferência de aprendizagem de uma rede neural para outra, chamada por *transfer learning*. Para mais detalhes, ver

Precisamente, no caso do problema em questão, aproveitando as facilidades de importação presentes nas bibliotecas do python, é possível importar os pesos pré-carregados do treino da rede sobre uma base de dados bastante conhecida: o ImageNet. Retreinamos a rede em todos os intervalos de dados usados previamente, usando o recurso de múltiplos treinos, a fim de verificar o quanto o pré-carregamento dos pesos impacta na classificação. A comparação feita está presente na figura 5.7, em que os pontos em vermelho indicam as medianas das AUCs sobre 10 conjuntos na ocasião anterior, sem pré-carregamento de pesos e, em azul, usando o recurso de pré-carregamento dos pesos. O eixo vertical consiste das AUCs com suas respectivas margens de erro, enquanto o horizontal constitui o valor do tamanho do conjunto de treino utilizado ou, para os casos em que isto é possível, o tamanho das conjuntos(*folders*) utilizadas em cada treino⁶.

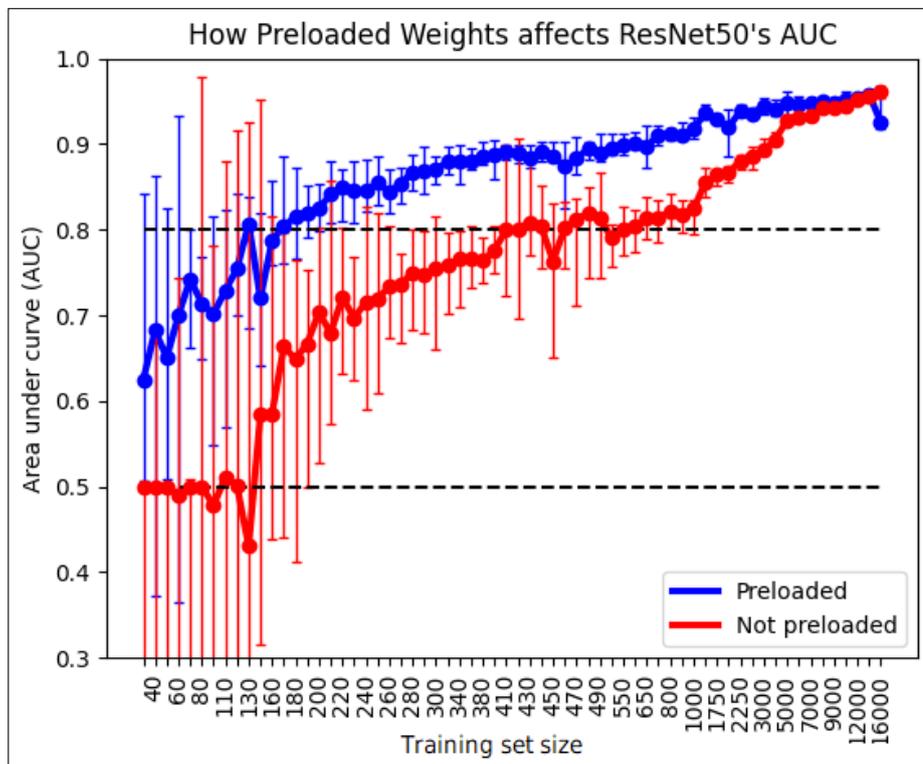


Figura 5.7 – Comparação entre as AUCs com e sem pré-carregamento de pesos.

Enquanto no canto direito do gráfico em que se atinge a totalidade dos dados de treino há uma evidente proximidade e até sobreposição dos valores de AUC, conforme diminui-se o tamanho da base de dados usada para treino surge uma evidente disparidade. Essa distância entre os valores de AUC só cresce até o mínimo de objetos de treino utilizados que, enquanto para o caso dos dados sem pré-carregamento de pesos atinge o limiar de 0.5 na faixa dos 150 dados de treino, para os treinos com o pré-carregamento esse limiar

[158]. No entanto, diferentemente do que se faz no *transfer learning*, não realizamos o treino na base de dados cujos pesos iremos importar.

⁶ Pode-se implementar esse pré-carregamento de pesos preenchendo a opção implícita “weights” com “imagenet”, na função “ResNet”, que importa o modelo diretamente da biblioteca Tensorflow[156]

sequer é atingido na mediana, porém já é consistente dentro dos erros por volta de 60. Além disso, as barras de erro para o caso sem pré-carregamento de pesos mantêm-se comportadas até por volta dos 500 dados de treino, no caso dos pesos pré-carregados estas margens se mantêm próximas ao valor da mediana até a faixa de cerca de 70 dados de treino.

Estas informações mostram que, dentro dos testes considerados nesta base de dados, o recurso do pré-carregamento dos pesos mostra-se indispensável para a melhora dos resultados atingidos, sobretudo nos casos em que o número de dados de treinamento é reduzido. Além disso, por conta de reduzir consideravelmente o tamanho das barras de erro, garante que as redes treinadas sejam mais estáveis. Tomando apenas por convenção, já discutida em seções anteriores, que o valor de 0.8 para a AUC é um limite de confiabilidade para a predição num teste cego, vimos que sem o pré-carregamento dos pesos esse valor definia o uso de cerca de 450 dados para treino como mínimo necessário. Usando o pré-carregamento dos pesos podemos levar esse limite um pouco mais além, exigindo um mínimo de cerca de 200 dados para treino que tornem a classificação aceitável, considerando as barras de erro.

Nesse limite seria possível obter 80 amostras de dados reais, em uma amostra uniforme em compatíveis com *surveys* recentes, para serem usadas no treinamento da rede, misturadas com 80 dados de galáxias comuns. No entanto, continuamos a nossa investigação buscando verificar se ainda é possível reduzir esse limite ainda mais.

5.2.4 Avaliação da influência da aumentação de dados

Outro recurso que pode impactar significativamente a classificação é o recurso da aumentação de dados. Isto consiste de aumentar a quantidade de dados de entrada na rede, criando cópias com modificações sutis dos dados já existentes. Entre essas modificações estão a rotação da imagem, transposição, ampliação, filtragem, entre outros. Essas cópias entram na rede neural junto às imagens originais como se houvessem mais dados de entrada do que o previamente estabelecido. Por continuarem sendo o mesmo dado com uma simples modificação visual, a tabela verdade para os novos dados continuam sendo a mesma. Isso é bastante útil principalmente na ausência de grandes quantidades de dados, pois alimenta a rede com novos dados que, serão tão eficazes quanto os antigos para o treino. Por consequência, é de se esperar um aumento na capacidade de generalização da rede em testes cegos, uma vez que, tendo sido alimentada com mais variações de um fenômeno, poderá se tornar mais fácil identificar novas ocorrências do fenômeno em diferentes padrões ou posições.

Nesse processo de investigar o desempenho da rede com poucos conjuntos de dados, o recurso da aumentação de dados pode ser um aliado forte no aumento da precisão da rede em suas predições. Fizemos o teste com o intervalo de dados sobre o qual já trabalhamos até agora. No entanto, apenas para fim de clarificar a exposição, optou-se por plotar

graficamente apenas no pequeno intervalo de dados. Os dados da aumentação escolhidos foram gerados a partir de:

- Rotações aleatórias das imagens em ângulos de múltiplos de 90° ;
- Espelhamento horizontal;
- Espelhamento vertical.

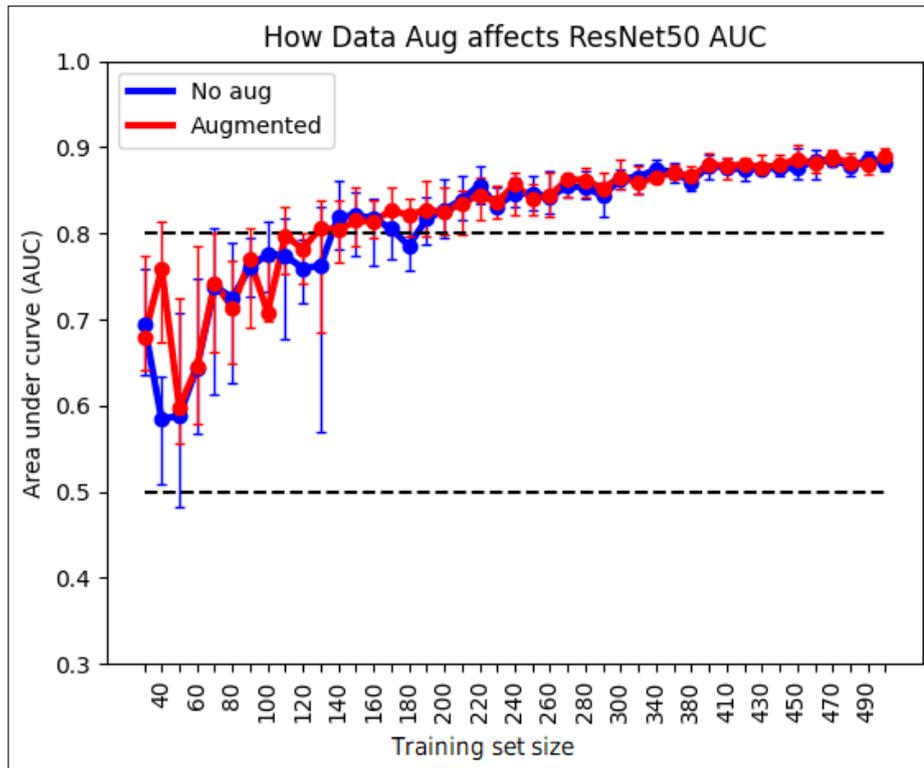


Figura 5.8 – Comparação entre as AUCs com e sem aumentação de dados.

Da figura 5.8, temos a justificativa para o por quê limitar a análise a poucos dados⁷. É possível visualizar que mesmo nos maiores valores do gráfico, os pontos que demarcam as AUCs com aumentação e sem aumentação se fazem bastante próximos até que, para um número de cerca de 200 ou mais, já chegam a se sobrepôr.

5.2.5 Uso do ensemble de redes nos dados da primeira edição do GLFC

Ao invés de substituir o método atual por outra arquitetura, podemos aproveitar o desempenho da arquitetura que temos e tentar encontrar outra que desempenhe tão bem quanto. Descobrimos uma arquitetura nova que desempenhe ainda melhor que a anterior, poderíamos efetuar apenas a substituição simples dessa nova arquitetura pela antiga mas, existe a possibilidade de ir além. Nesta seção avaliamos a possibilidade de combinar o

⁷ Para realizar as modificações especificadas, a própria biblioteca [156] possui a função interna “Image-DataGenerator” que faz a aumentação de dados, dadas as opções de modificação nas imagens.

resultado de duas arquiteturas diferentes, com intuito de superar a performance das duas, individualmente através de um *ensemble* de redes.

Neste trabalho resolvemos prosseguir o estudo com o ensemble de duas redes neurais, a ResNet, já utilizada previamente, e a EfficientNetB2. Fizemos uma breve abordagem destas na seção 3.5. A motivação para escolha da EfficientNetB2 se deu por conta de ser uma rede bastante atual, no estado do conhecimento da época, e pela agilidade na sua execução. Por possuir menos parâmetros do que a rede ResNet50 (quase 8 milhões, em comparação aos respectivos mais de 23 milhões), o tempo para treiná-la é ainda menor do que o tempo que leva o treino da própria ResNet50. Por isso, julgamos que seria interessante verificar como a EfficientNetB2 desempenha nessa base de dados estudada. Na figura 5.9, apresentamos os resultados de ambas as redes. Realizamos o treino da EfficientNetB2 utilizando os mesmos recursos para incremento das predições usados na ResNet50, como a aumento de dados e o pré-carregamento de pesos da base de dados *ImageNet*. O que temos na figura 5.9 são as medianas das AUCs devido aos múltiplos treinos da nova rede, nos mesmos intervalos de dados que já utilizamos para a ResNet50. Após o treino, fizemos o modelo executar um teste cego de classificação na base de dados utilizada a cada etapa do multitreino e depois retiramos as medianas.

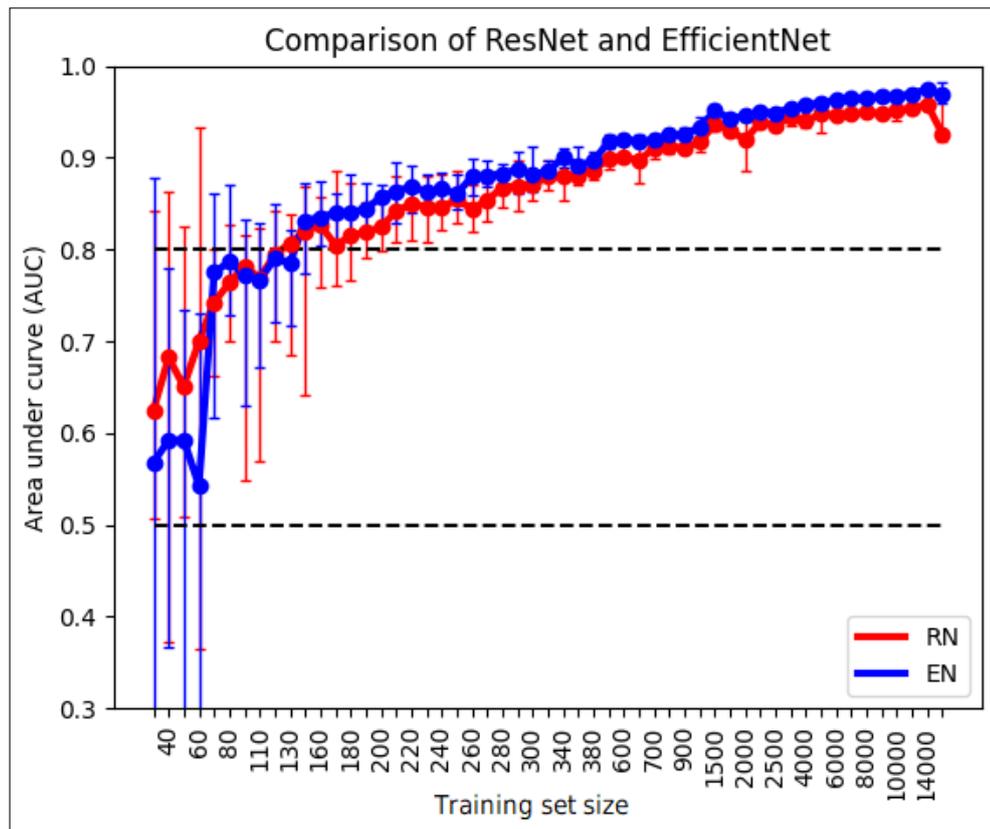


Figura 5.9 – Comparação de desempenhos das redes ResNet50 e EfficientNetB2.

Como pode-se verificar pela figura 5.9, a EfficientNetB2 (em azul) na média, performa para quase todos os pontos melhor do que a rede ResNet50, no entanto, as discrepâncias

são consistentes dentro da margem de erro. Comprova-se, desta maneira, a usabilidade da rede EfficientNetB2 na estratégia de classificar essa base de dados, conjuntamente à arquitetura que utilizamos antes.

Inicialmente, a primeira estratégia para se realizar um ensemble das redes foi a mais imediata que poderia se imaginar: uma média aritmética das predições. As predições de um modelo de rede neural sobre um conjunto de dados constituem de pontuações para aqueles dados, que podem ter diferentes valores de acordo a quantidade de classes para se classificar e, o tipo de função de ativação na saída da rede. No caso das redes que utilizamos, as saídas da rede foram as funções pré-definidas chamadas por *softmax*, que retornam para cada classe um número de 0 à 1, normalizado em 1 para todas as classes. Por conta de termos um problema de classificação binária, estão envolvidas apenas duas probabilidades: a chance de estar ocorrendo um lenteamento gravitacional ou não. Assim, a saída da rede neural retorna dois números com as probabilidades preditas de estar havendo lenteamento ou não, cuja soma deve ser 1. Cada arquitetura prevê este conjunto de probabilidades para cada imagem do conjunto de teste. Portanto, calcular a média aritmética destas predições seria calcular a média entre as probabilidades de estar havendo lenteamento e a de não estar, resultando em dois novos valores de probabilidades para aquele dado. Repetindo este mesmo processo para todos os dados de teste, o conjunto das predições constitui uma espécie de predição conjunta das duas arquiteturas.

Para verificar se o uso do ensemble produz um impacto positivo no desempenho das redes, além do treino que já realizamos nas imagens da base de dados utilizando a rede ResNet50, faz-se necessário repetir o treino com todos os intervalos mencionados com a rede EfficientNetB2 para, em cada caso, calcular a média dos valores das predições. Assim como fizemos ao longo dos passos anteriores deste trabalho, também lançamos mão do recurso do múltiplos treinos, da aumento de dados e do pré-carregamento dos pesos. A rede EfficientNetB2 também é uma rede pré-carregada pela biblioteca Keras[97] e assim, oferece uma função que carrega a rede com a opção de pré-carregar os pesos da base de dados *ImageNet*, conforme fizemos. Apresentamos os resultados na figura 5.10. Na legenda, “RN” corresponde à mediana das AUCs dos múltiplos treinos prevista pela rede ResNet50 (em vermelho), “EN” á respectiva mediana prevista pela rede EfficientNetB2 (em azul) e, “(RN+EN)/2” corresponde ao ensemble entre elas (em verde)⁸.

⁸ O código utilizado para treinar as redes e, por conseguinte, implementar este modelo está presente na pasta “CODE-ENSEMBLE”, dentro do repositório de GitHub “<https://github.com/KayqueTeles/DISSERTACAO.git>”. O arquivo principal a ser executado, dentro da árvore de arquivos, é o denominado “main_ensembles_final.py”.

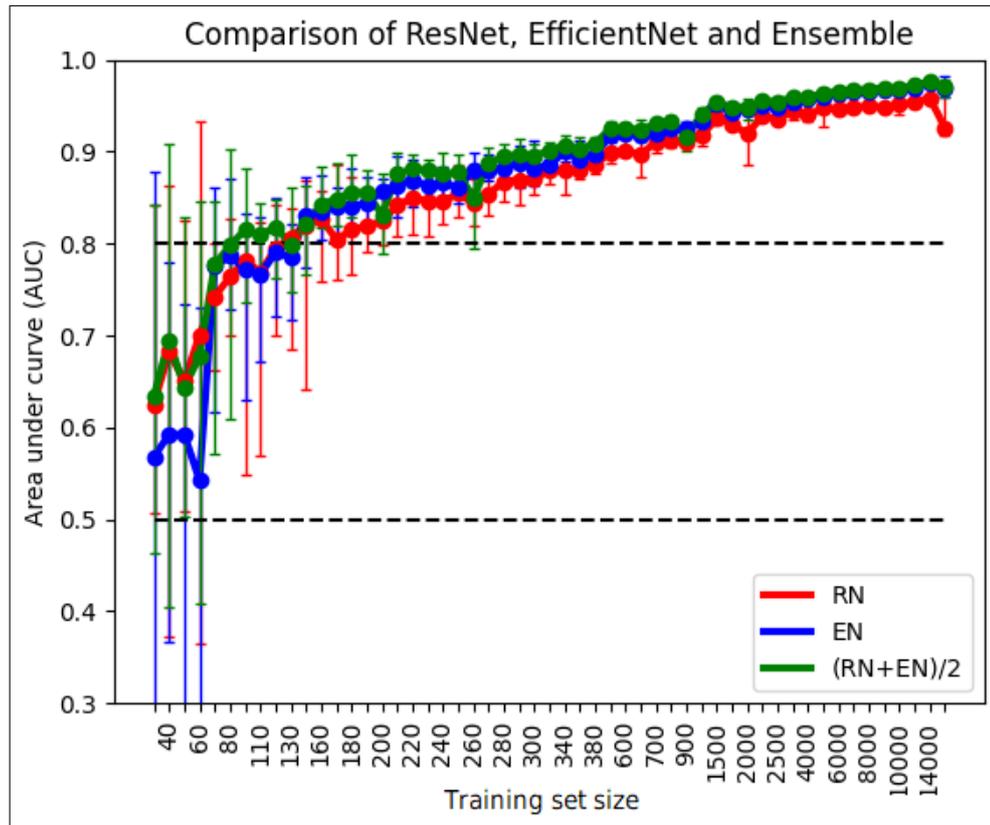


Figura 5.10 – Comparação de desempenhos da rede ResNet50, EfficientNetB2 e do ensemble entre elas.

Pode-se observar pela figura 5.10 que a rede EfficientNetB2 teve um desempenho superior, na média, à rede ResNet50, mas consistente dentro da incerteza. O mesmo ocorre com o ensemble, a média parece superar as redes individuais. Essa diferença é pouco perceptível nos casos em que o conjunto de treino individual é o maior possível, mas vai se acentuando conforme vamos reduzindo o tamanho do conjunto de treino, principalmente em relação à rede ResNet50. A conclusão dessa análise, no entanto, seria que é possível reduzir esse limiar de dados para cerca de 80 dados, apesar das barras de erro estarem bem grandes. Esse é o mínimo de dados que seria necessário para reproduzir um resultado de 0,8 através do ensemble de redes.

Existe toda uma gama de funções internas de pacotes do *Python* com alternativas para o desenvolvimento destes ensembles, usando métodos bastante utilizados na literatura. Fizemos uma análise comparativa do desempenho de alguns destes métodos de se realizar *ensemble* de redes, a fim de verificar qual método poderia ser mais eficiente na classificação da nossa base de dados. Observando o pacote *DeepStack*[159], carregamos a função para ensemble nele já pré-definida *DirichletEnsemble*, e da biblioteca *scikit-learn* importamos a função *LogisticRegression* para observar o seu desempenho⁹. Realizamos o treino para todas as faixas de treino utilizadas, obtendo o gráfico expresso na figura 5.11.

⁹ Para mais informações sobre as duas técnicas de se realizar ensemble, ver [159].

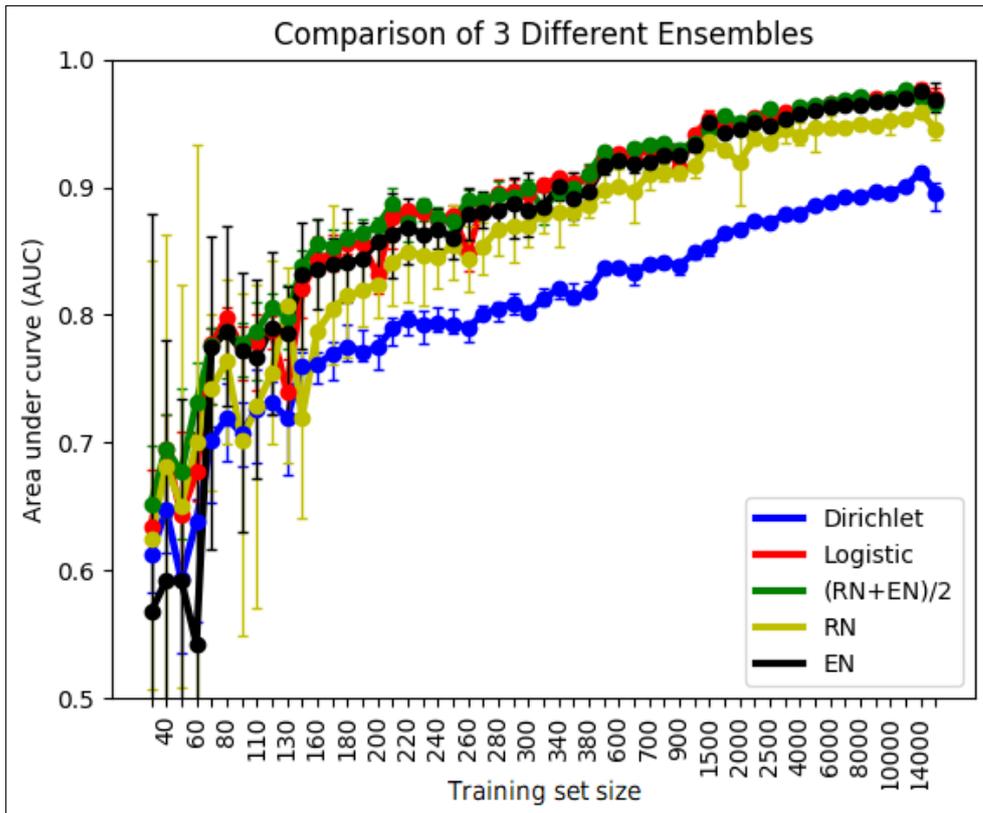


Figura 5.11 – Comparação de desempenhos de duas técnicas de ensemble pré-carregadas e do ensemble por média aritmética com o das redes individuais.

Resolvemos deixar na figura 5.11 o desempenho das redes individuais a fim de verificar o comparativo das estratégias de ensemble em relação aos resultados de cada uma. Fica evidente no gráfico que a curva azul possui a maior parte de seus pontos com valor bem menor que as demais. Esta curva corresponde ao desempenho do ensemble calculado a partir da função *DirichletEnsemble* do pacote *deepstack*, que executa o cálculo do ensemble das redes através da técnica chamada por ensemble de Dirichlet-Markov. Apesar de ser uma técnica estatística complexa, se mostrou pior do que o desempenho das redes individuais e portanto, não serve para refinar o resultado conforme queremos.

Presente na figura 5.11 ainda está o resultado obtido via função *LogisticRegression*, do pacote *scikit-learn*, que executa uma regressão logística dos valores imputados. A regressão sobre os dados das predições apresentou uma boa performance como meio de ensemble mas, manteve-se similar ao desempenho da EfficientNetB2. No topo do gráfico ainda está, na maior parte dos pontos, o ensemble calculado pela simples média aritmética das predições, ainda que consistente em 1σ com a EfficientNetB2. Poderiam ainda ser testadas outras funções pré-carregadas para o ensemble, mas esses resultados nos indicaram que a média se mostrou já uma estratégia de ensemble razoável e de simples implementação. Poderiam ser adicionadas mais redes, mas cabe pensar no custo computacional envolvido e no tempo necessário para treinar todas as redes ao longo de um grande número de épocas.

5.3 Classificação com dados da Segunda Edição do GLFC

Depois de explorar diversas alternativas para melhorar a detecção automatizada de lentes gravitacionais numa mesma base de dados, analisamos uma base de dados nova. Pois, apesar de todo o sucesso atingido, a base de dados utilizada ainda é uma base de dados simulados. Apesar de terem sido inspirados em imagens de dados reais obtidos por telescópios, constituem, como qualquer outra, de uma base de dados com os seus padrões de geração. Nem sempre transferir a capacidade de predição de uma rede em certa base de dados para outra base de dados é tarefa simples e, nesse intuito, daremos prosseguimento a nossa análise com outro conjunto de dados.

Uma outra base de dados simulados de lente gravitacional disponível na rede é a fornecida na segunda edição do GLFC, de 2020. Desta vez a base de dados de imagens de treino consiste de 100.000 (cem mil) objetos simulados em que pode haver ou não efeito forte de lente, de modo a tentarem reproduzir imagens do levantamento de galáxias do consórcio Euclid. Por sua vez, as imagens foram geradas de modo a tentar reproduzir as observações retiradas nas bandas VIS (espectro visível) e NISP J, Y e H (obtidos por fotometria no infravermelho). Portanto, cada objeto da base de dados possui uma imagem em 4 bandas, sendo as imagens na banda VIS de dimensão 200x200 pixels enquanto nas bandas NISP J, Y e H, de dimensões 66x66 pixels. A princípio, o primeiro desafio do uso dessa base de dados é o fato de haverem 4 bandas e, uma delas possuir imagens de dimensões diferentes das demais. Isso dificulta a entrada simultânea desses dados numa rede neural, mas foram realizados primeiros testes dessa base de dados, apenas normalizando as imagens onde foi obtido um *underfitting*.

Cabe aqui mencionar o motivo pelo qual, tendo conhecimento dessa segunda base de dados de lentes gravitacionais, não a utilizamos desde o início ao longo de todos os testes realizados. De fato, ocorre que a sequência deste trabalho também é, de certo modo, uma sequência histórica do desenvolvimento do trabalho. Até o momento em que se estava desenvolvendo o ensemble de redes, não havia sido lançada ainda essa segunda base de dados, uma vez que o intervalo entre as duas edições da competição foi de um ano. Por isso, o lançamento da segunda edição com essa nova base de dados foi uma novidade e, o envolvimento de ambos orientador e orientando na própria competição em si nos fez dedicar tempo a descobrir como desvendar essa nova base de dados.

Felizmente, os ganhadores desta segunda edição da competição foi o time de inteligência artificial do CBPF CAST (*CBPF Arc Search Team*). Portanto, foram adotados os mesmos passos realizados pelo grupo para classificar a base de dados, e a mesma rede [160]. Primeiramente, mencionaremos o tratamento das imagens que se fez necessário para que a rede neural conseguisse começar a executar a classificação. Diferentemente dos dados da primeira edição do GLFC, que requiseram apenas um pré-processamento simples, estes dados representaram um verdadeiro desafio para se descobrir qual a melhor maneira de pré-processá-los. No entanto, podemos refazer o caminho das estratégias que

se mostraram mais eficazes.

5.3.1 Pré-processamento dos dados da segunda edição do GLFC

Observamos que 9 imagens que não continham todas as 4 bandas. Optamos por retirar completamente estes objetos da base de dados, uma vez representando um número bastante pequeno em comparação ao tamanho total da base de dados.

O primeiro passo do pré-processamento da base de dados da segunda edição do GLFC foi a construção da tabela verdade para os objetos fornecidos. Diferentemente da primeira edição do GLFC, o catálogo dos objetos não possuía uma resposta pronta a respeito do seu caráter de ocorrer ou não lenteamento mas, ao invés, consiste de um arquivo de parâmetros das imagens simuladas. Entre os aspectos dos objetos descritos neste catálogo estão as (os):

- coordenadas x e y de suas curvas críticas;
- desvio para o vermelho da fonte e da lente;
- magnitude da fonte não lenteada presente no catálogo Euclid VIS;
- número de curvas críticas;
- distância da fonte ao centro de uma cáustica em radianos;
- área de curvas críticas largas em radianos quadrados;
- área da cáustica em radianos quadrados;
- número de pixels na imagem da fonte lenteada acima de 1σ ;
- fluxo total destes pixels em unidades de σ ;
- número de pixels na lente principal acima de 1σ ;
- fluxo total destes pixels em unidades de σ ;
- número de grupos separados de pixels da fonte;
- magnificação efetiva da fonte contabilizada em todas as bandas;
- contraste de brilho superficial médio entre a lente e a fonte em escala de pixels acima de um certo limiar;
- diferença de cor entre a lente e a fonte;
- massa do halo da lente principal;
- massa estelar da lente principal;

- magnitude da lente principal na banda de referência;
- número de fontes adicionadas;

Um critério sugerido pelos avaliadores no momento da classificação é tentar classificar como caso positivo de efeito forte de lente os objetos que possuem: número de grupos de pixels separados da fonte maior que zero; magnificação efetiva da fonte em todas as bandas maior que 1.6 e; número de pixels da fonte maior que 20. Seguindo este critério como definidor dos casos em que ocorre lenteamento, dentro do catálogo há um número de 49213 objetos que podem ser considerados como lentes.

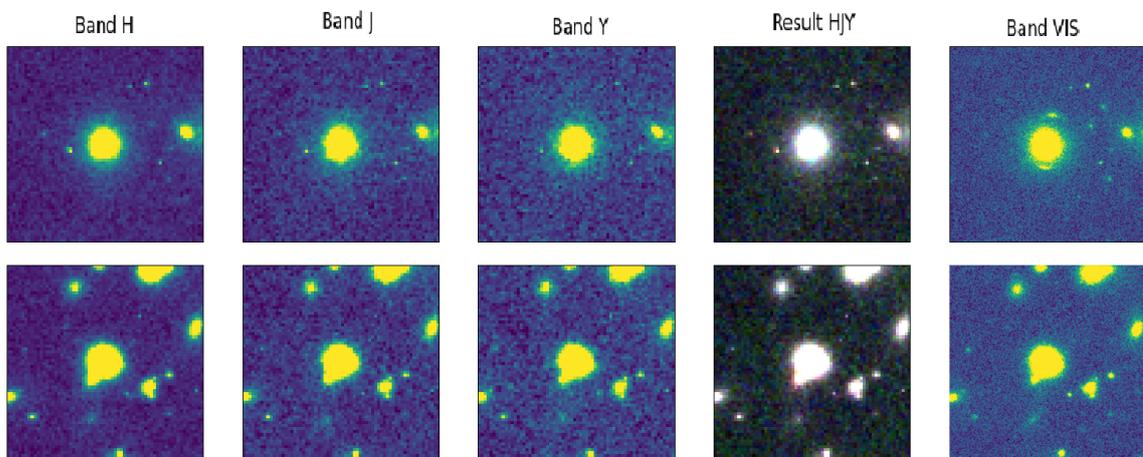


Figura 5.12 – Dois exemplos de imagens em suas bandas H, J, Y, VIS e o resultado da combinação das três bandas HJY.

Uma vez definida a tabela verdade dos objetos que entrarão na rede, cabe agora fazer uma análise das imagens propriamente ditas. Na figura 5.12 temos alguns exemplos de imagens em cada banda, já modificadas pelo pré-processamento que será discutido ao longo dessa seção. O objeto das imagens da primeira linha indica um caso positivo de lenteamento gravitacional, adotando os critérios propostos sobre a tabela verdade, enquanto o objeto da linha de baixo, não. No sentido de torná-las mais visíveis ao olho humano, durante o processo de gerar a visualização das imagens, foram realizados outros tipos de processamentos. Portanto, as cores das imagens não refletem exatamente as cores originais dos objetos diretamente extraídos do tensor de dados. A primeira função utilizada nesse processamento é um ajuste de contraste. Funciona através de um mapeamento dos valores de pixels das imagens em uma escala de cinza, eliminando os valores abaixo e acima de 1% do mínimo e máximo respectivamente, para executar uma normalização e por fim retornar esse valores de pixel à matriz de dados. O efeito final desse ajuste é aumentar o contraste dos pixels da imagem. É durante esse processo que são geradas as figuras nas bandas H, J, Y e VIS da figura 5.12, ganhando cores. Depois disso, foi usada

a subrotina `toimage`¹⁰ para converter a matriz de dados numa única imagem resultante da sobreposição dos canais. O processo é análogo para a conversão de imagens nos canais R, G, e B em RGB, resultando na imagem da coluna "Result HJY".

É possível verificar que as imagens ainda possuem bastante ruído, não sendo possível verificar com precisão, nestes dois exemplos, se ocorre o efeito forte de lente ou é apenas uma sobreposição de galáxias próximas. No entanto, neste exemplo, nas imagens da primeira linha está presente o efeito forte de lente enquanto, nas imagens da segunda, não. Sabendo que o primeiro exemplo apresenta um caso positivo de lente, é possível identificar as manchas foscas ao redor da mancha branca central como imagens múltiplas de uma fonte longínqua.

No pré-processamento dos dados, um passo bastante comum é realizar a normalização dos dados. E a primeira abordagem para realizar essa normalização é colocar os dados entre zero e 1. Desta maneira, diminui-se bastante as escalas de valores para que um valor muito discrepante não provoque um grande desequilíbrio nos pesos da rede. Isto pode ser feito identificando o valor máximo de pixel das imagens e, em seguida, dividindo todos os valores dos pixels por esse máximo. Desta maneira, os pixels mais próximos do máximo atingirão um valor próximo de 1 enquanto aos menos próximos, cabe um valor próximo de zero. Antes de qualquer coisa, cabe observar o histograma da distribuição de pixels das imagens 5.13.

Nesta figura, enquanto o gráfico abaixo mostra a distribuição de pixels para as imagens da banda VIS, a primeira mostra a distribuição das imagens nas bandas H, J, e Y, representadas nas cores azul, marrom e verde respectivamente, sobrepostas. A distribuição da banda VIS não foi incluída às demais por conta de seus valores estarem uma ordem de grandeza acima, dado que as imagens possuem maior resolução, potencialmente dificultando a visualização das distribuições de pixels nas demais bandas. Como pode-se verificar, existe um considerável contingente de imagens que possuem pixels de valores bastante destoantes de todo o resto. O fato de haver um número grande de valores de sinais opostos também é apresentada uma dificuldade para a normalização direta. Estando a distribuição de valores centrada em zero, a maior parte dos dados ao ser dividida diretamente pelo máximo resultará em um número ainda mais próximo de zero, o que irá afinar a curva. O efeito disto na visualização da imagem é um escurecimento, tornando ainda mais difícil a visualização a olho nu e, por conseguinte, distinção virtual da rede neural.

Uma saída natural decorrente desse problema é buscar, de algum modo, limitar esses valores de mínimo e máximo de modo a se manterem próximos do restante da amostra. Isso pode ser feito facilmente em python utilizando a função `clip` do pacote *Numpy*[163], que limita os valores de um conjunto de dados à duas bordas do intervalo transformando

¹⁰ Por sua vez, esta função é uma subrotina para a conversão de uma tabela de dados em uma imagem visível da biblioteca *Pillow*[161]. Essa função está presente na versão 1.0.0 da biblioteca *Scipy*[162], mas foi removida em versões mais recentes.

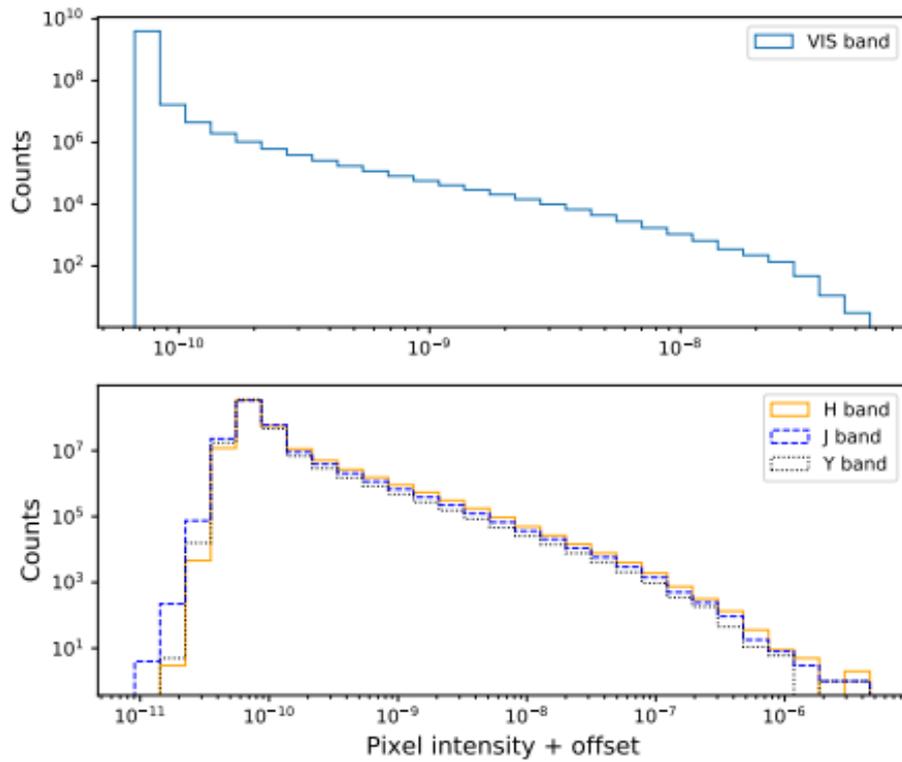


Figura 5.13 – Histograma de pixels das imagens nas bandas H, J, Y (superior) e VIS (inferior), antes do pré-processamento.

todos os valores além dessas bordas (seja acima da superior ou abaixo da inferior) para valores da borda. Desta maneira, os máximos e mínimos globais são limitados em certos pontos. Resta saber que valores de máximo e mínimo seriam interessantes de se retirar das amostras. Uma saída comumente adotada em estatística é calcular um dos percentis dos valores de dados, como o 98^o por exemplo, e tomá-lo como o máximo da distribuição. Desta forma, estamos excluindo uma minoria dos valores de pixel acima de um máximo comum ao resto e, com o auxílio da função para clipar dados, podemos transformar esses valores excluídos no valor deste percentil calculado.

Quanto ao mínimo, um problema que pudemos identificar foi a presença de valores negativos dentro da base de dados, o que fazia com que o ponto médio da distribuição da maioria dos valores se situasse num ponto próximo ao zero. De modo a evitar escolher um mínimo entre esses valores e acabar tendo um mínimo de valor negativo, o modo de contornar esse problema foi tomando o negativo do percentil de índice 99,9 da distribuição de pixels e, em seguida, tomamos o negativo desse número. Esse constituiu o valor mínimo de nossos dados. Finalmente, encontrando esse máximo e mínimo o processo de normalização dos dados foi executado da seguinte forma: para cada imagem i de um objeto no canal C , a nova imagem normalizada será tal que

$$C_i = \frac{C_i - L_{inferior}}{L_{superior} - L_{inferior}} \quad (5.1)$$

onde $L_{inferior}$ constitui o valor correspondente ao percentil do mínimo utilizado e

$L_{superior}$ o valor correspondente ao percentil do máximo. Repetido esse procedimento para cada canal de imagens, temos por fim as imagens normalizadas, prontas para entrar na rede neural, cujo histograma está presente na figura 5.14.

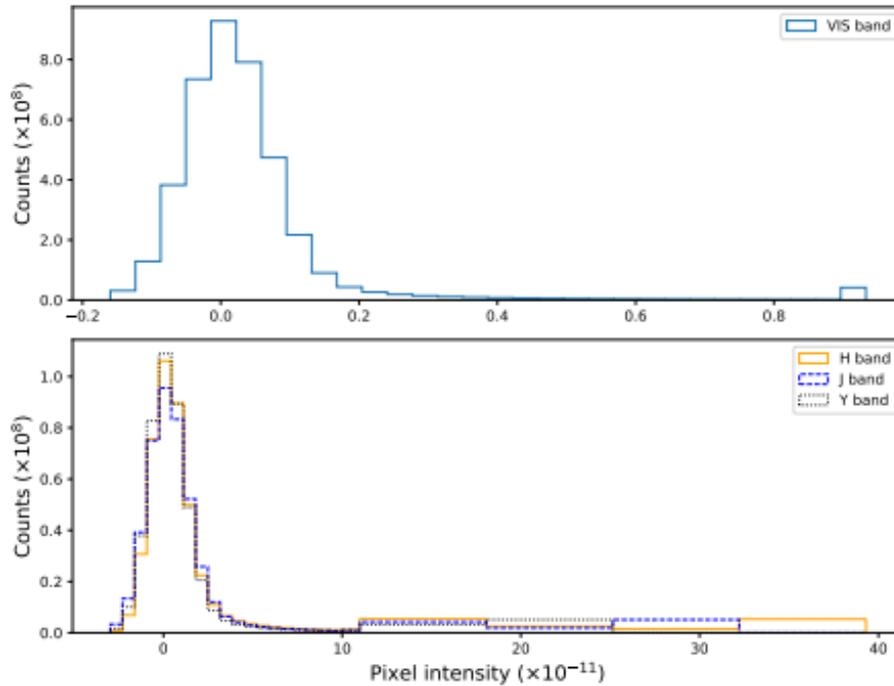


Figura 5.14 – Histograma final de pixels das imagens nas bandas H, J, Y (superior) e VIS (inferior), após pré-processamento.

Graças aos percentis serem calculados independentemente para cada canal de imagens, agora nessa figura os pixels de cada uma das bandas se distribuem em picos levemente diferentes - o que não constitui de exigência neste processo. O maior benefício dessa normalização, no entanto, está no fato de agora termos os dados distribuídos no intervalo de zero a um positivo. Isto poderá facilitar a entrada dos dados na rede neural, uma vez que sinais diferentes na entrada podem mudar os sinais dos pesos durante o processo de retropropagação¹¹.

5.3.2 Desempenho do Ensemble de redes nos dados da Segunda Edição do GLFC

Uma vez que os dados estão processados e preparados, caberia agora inserí-los num classificador. O algoritmo desenvolvido para classificar a base de dados da primeira edição consiste de um ensemble das redes ResNet50 e EfficientNetB2. Cada uma das duas, pelo modo como estão pré-definidas por biblioteca, aceitam no máximo um tensor de dados de imagem com três canais. Sendo o caso da base de dados do II GLFC um tensor com quatro

¹¹ O código do pré-processamento “CH2-Pre-Processing.py” pode ser encontrado no repositório do domínio Github <https://github.com/KayqueTeles/DISSERTACAO.git>

canais (H, J, Y e VIS), e tendo em mente um deles de maiores dimensões em relação ao anterior (VIS), fez-se necessário pensar sobre como esses dados poderiam entrar na rede de ensemble. Inicialmente, a ideia principal seria selecionar canais específicos ignorando os demais, dado que cada canal é referente ao mesmo objeto em diferentes filtros (e dimensões da imagem, no caso da banda VIS). Por isso, como primeira abordagem com essa nova base de dados experimentamos a classificação utilizando apenas os dados das camadas H, J e Y, que possuem dimensões semelhantes, como entradas das redes ResNet50 e EfficientNetB2.

Quanto ao tamanho da amostra, optamos por reproduzir os mesmos número de objetos para o treino 16000 (dezesesseis mil) utilizados na classificação da base de dados da primeira edição, aumentando o conjunto de validação para 3000 objetos e o de teste como 3000 também. Valor que sequer chega a metade de toda a base de dados, de 10^5 (cem mil) objetos, o que nos permite fazer algo inédito em relação a base de dados anterior: gerar mais de um conjunto disjunto de treino de 16000 objetos. No entanto, um dos pontos principais nesse trabalho seria buscar verificar se é possível definir um limite inferior fixo para a quantidade de dados necessária neste problema. Pela base de dados da primeira edição do GLFC, 80 seria um número mínimo necessário de dados que se fariam disponíveis, tendo em mente uma classificação com AUC de pelo menos 0,8. Foi mantido o número de épocas da execução (50), tamanho do lote de dados que entra na rede à cada etapa do treino (*batch size* de 64), escolha de otimizadores (RAdam) e a estrutura. Dado o tamanho da base de dados, realizamos a execução em 5 *fold*s de treino inspirados em validação cruzada com 80.000 dados. Mas, no intuito de evitar a poluição visual no trabalho, optamos por mostrar os gráficos referentes apenas à primeira de cada uma das duas redes. Para a acurácia, temos os gráficos da figura 5.15.

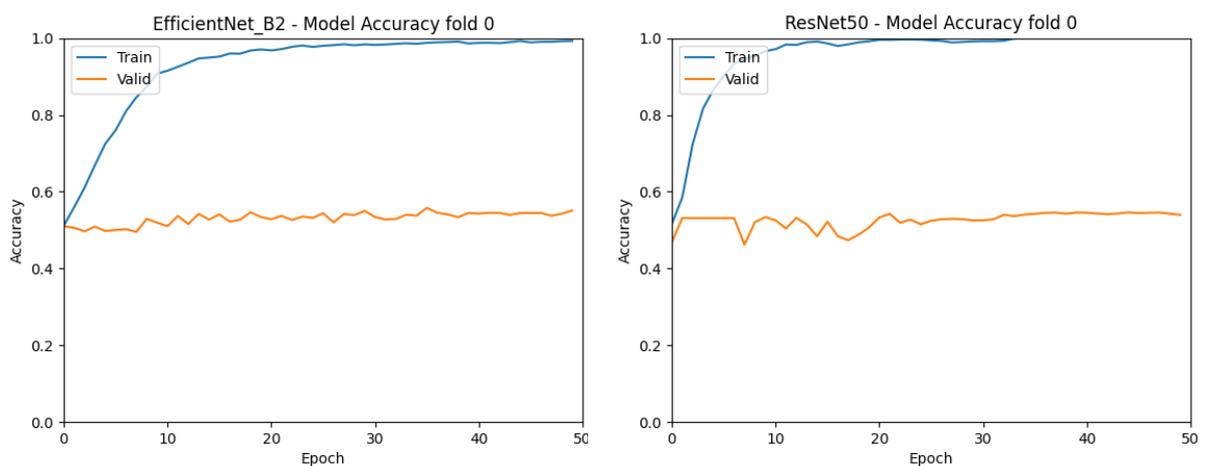


Figura 5.15 – Acurácia do Treino da primeira pasta das redes EfficientNet B2 (à esquerda) e ResNet50 (à direita) no Ensemble de Redes sobre a segunda base de dados do GLFC.

Nos gráficos de ambas as redes ocorre o seguinte fenômeno: enquanto a acurácia de

treino em azul converge para 1, a acurácia de validação em laranja se mantém próxima ao aleatório, flutuando por volta de 0,5. Fica evidente nos gráficos, portanto, uma distância entre as acurácias, um espaçamento que apenas cresce conforme a acurácia de treino converge para 1,0, configurando um caso de sobreajuste. Enquanto a linha azul que refere-se ao desempenho do modelo nos dados de treino demonstra uma ótima acurácia, a linha laranja diz respeito a mesma acurácia no teste cego sobre os dados de validação: um regime de classificação indistinguível da escolha aleatória. Esse comportamento também pode ser verificado na função custo do treino das duas redes, presente na figura 5.16.

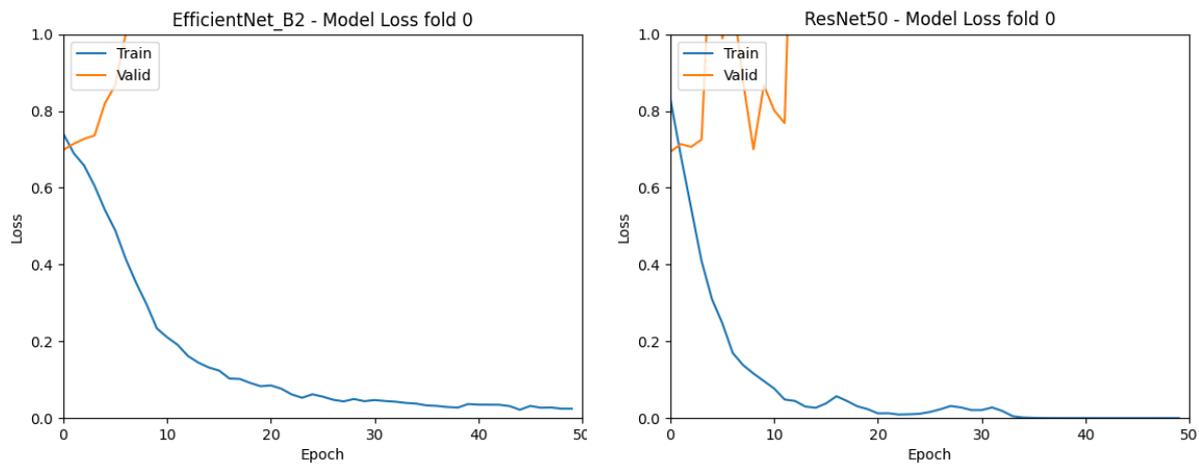


Figura 5.16 – Pontos da Função Custo do Treino da primeira pasta das redes EfficientNet B2 (à esquerda) e ResNet50 (à direita) no Ensemble de Redes sobre a segunda base de dados do GLFC.

Neste gráfico da função custo o sobreajuste fica ainda mais evidente. Enquanto o custo de treino cai convergindo para valores próximos de 0,0, o custo de validação cresce rapidamente, atingindo patamares muito maiores que 1,0, omitidos nesse gráfico para não dificultar a visualização. Diferentemente dos gráficos de função custo para classificações com sucesso, as linhas de custo de treino e validação não se encontram. Por fim, como último recurso de nossa avaliação, cabe verificar qual foi o desempenho em termos da curva ROC de ambas as redes, presentes na figura 5.17.

Aqui verifica-se o efeito final do sobreajuste na classificação de dados. A linha pontilhada se refere a uma AUC de 0,5, um regime de classificação próximo a escolha aleatória. Mas na figura, em ambos os casos a ROC fica apenas um pouco acima desse valor. O modelo não conseguiu realizar o treino recebendo em sua entrada as imagens das camadas H, J, e Y. Resta ainda verificar como se dá esse desempenho nas imagens da banda VIS, uma vez que possuem resolução maior. Para isto, como as redes do modo como estão arquitetadas precisam de tensores de três canais na entrada, preparamos um tensor com três canais, sendo cada um deles o próprio canal VIS. Por fim, executamos as redes nas mesmas condições, usando a mesma quantidade de objetos e, para as duas redes, obtivemos as curvas ROC da figura 5.18.

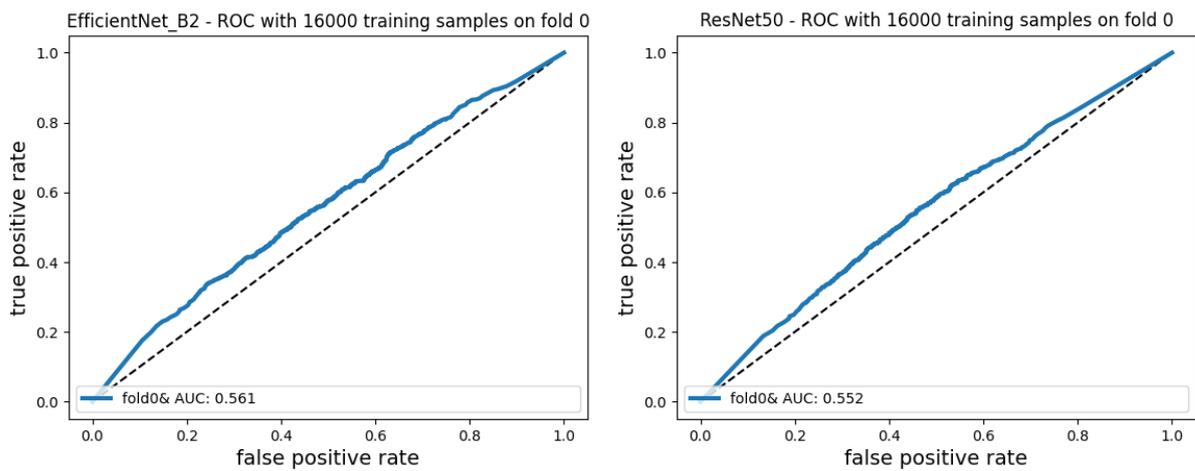


Figura 5.17 – Curva ROC e AUC de cada modelo ao final da primeira pasta das redes EfficientNet B2 (à esquerda) e ResNet50 (à direita) no Ensemble de Redes sobre a segunda base de dados do GLFC.

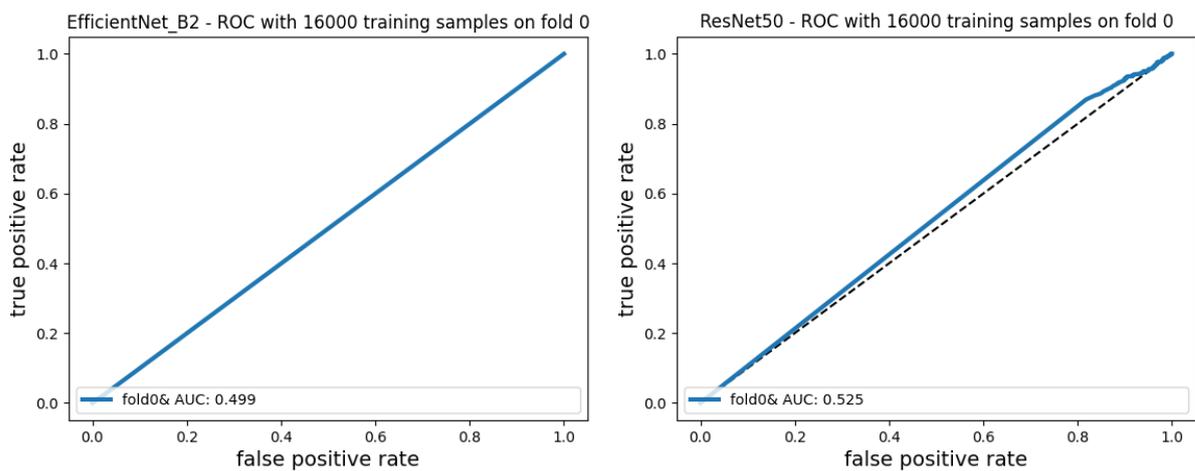


Figura 5.18 – Curva ROC e AUC de cada modelo ao final da primeira pasta das redes EfficientNet B2 (à esquerda) e ResNet50 (à direita) no Ensemble de Redes sobre a segunda base de dados do GLFC, usando a banda VIS.

Novamente, temos as duas redes apresentando uma área sob a curva próxima do valor do regime aleatório de 0,5. Apesar de, para o caso de usarmos a banda VIS não ser evidenciado nesse trabalho os gráficos de acurácia e custo, não se verifica que ocorre sobreajuste. Pelo contrário, neste caso as acurácia e custo de treino seguem com valor praticamente constante, indicando que nenhum aprendizado sequer foi extraído dos dados. Não é possível afirmar, portanto, que esse modelo é capaz de realizar previsões confiáveis num teste cego acerca da ocorrência ou não do fenômeno de lente. Cabe lembrar que foram utilizados 16000 objetos para treino na execução que retornou os resultados das figuras 5.15, 5.16 e 5.17. Existem técnicas computacionais a fim de diminuir ou retirar o sobreajuste, como a regularização da função custo¹², mudar o otimizador, dentre outras,

¹² Para uma breve introdução sobre reguladores de função custo usando *Tensorflow*[156], ver

mas não obtivemos sucesso. Este resultado também se mostrou presente ao realizar o treino com os dados sem o pré-processamento e com o pré-processamento. A melhor alternativa é escolher uma nova arquitetura para a rede neural, no intuito de que ela possa se adaptar melhor à base de dados. Desta forma, descartamos a possibilidade de utilizar o ensemble das redes ResNet50 e EfficientNet B2 na expectativa de encontrar um bom desempenho na classificação com apenas 80 objetos de treino, diferentemente da primeira base de dados.

5.3.3 Treino usando a concatenação de duas EfficientNet B2

Uma vez que a nossa base de dados está devidamente normalizada, pronta e tendo passado nos primeiros testes, o passo seguinte na classificação é o de buscar encontrar redes neurais que sejam eficientes para lidar com o caso que temos em mãos. A nossa base de dados, à título de recapitulação, consiste de cem mil imagens em quatro bandas, sendo três bandas de imagens com dimensões 66x66 pixels e uma extra com imagens em dimensões 200x200 pixels. Normalmente seria interessante redimensionar as imagens 200x200 para o mesmo tamanho das imagens das demais bandas, no entanto, ao fazê-lo estaríamos abrindo mão da maior resolução da imagem na banda VIS, que poderia contribuir na classificação dos objetos. Por outro lado, se fosse feito o redimensionamento das imagens 66x66 para o tamanho 200x200, isso poderia distorcer as imagens e gerar artefatos. O desafio era, portanto, tentar encontrar uma maneira de imputar na rede as imagens em todas as bandas que nos foram fornecidas. Nesse intuito, o modo como se obteve a decisão final acerca do melhor procedimento que poderia ser utilizado seguiu de modo experimental. Primeiro testamos a rede numa arquitetura já conhecida e, caímos num problema de sobreajuste. O passo seguinte consiste de encontrar uma outra rede neural que seja capaz de resolver o nosso problema.

Reconhecendo o pré-processamento dos dados feito pelo grupo CAST vencedor da segunda edição do GLFC, uma primeira aposta possível para a rede neural também poderia ser a rede neural utilizada pelo grupo: a EfficientNet B2. O fato de ser uma rede com poucos parâmetros e, por conseguinte, cujo treino pode ser bem mais rápido do que outras a tornou bastante interessante para a competição. Dadas as pretensões de realizar várias execuções neste trabalho, investigando sempre o problema de classificar uma base de dados com o mínimo possível de objetos.

No entanto, o modo de realizar a entrada dos dados que utilizamos ao tentar o *ensemble* de redes que é o diferencial. Quanto ao problema do tensor de imagens possuir quatro dimensões, uma delas maior que as demais, a solução adotada pelo grupo consistiu de utilizar duas EfficientNet B2. Em uma delas entra parte dos dados, na outra entra apenas os dados na banda VIS e, por fim, a última camada dessas duas redes estará concatenada

<https://www.ritchieng.com/machine-learning/deep-learning/tensorflow/regularization/>. Acesso em 24/02/2023.

de modo que a previsão final receba contribuições de cada uma das entradas. Isto é, as duas EfficientNet B2 serão treinadas ao mesmo tempo, com suas diferentes entradas, motivo pelo qual iremos nos referir a essas redes como Efficient Net B2 concatenadas. Precisamente, em uma das EfficientNet B2 a entrada consiste em um tensor com 3 canais de imagens 66x66, sendo um deles o canal de imagens na banda Y e, os dois canais seguintes, matrizes nulas de dimensão 66x66. Já na outra EfficientNet B2, a entrada consiste de um tensor de 3 canais de imagens 200x200, o primeiro deles com as imagens na banda VIS e os dois seguintes também consistindo de matrizes nulas de dimensão 200x200. Essa escolha se deu por conta de estas serem as bandas que se provaram mais influentes na classificação da base de dados [160].

Para entrar na prática, cabe por fim colocar este modelo à prova. Quanto a divisão dos dados, foi utilizada toda a base de dados de 100.000 objetos, fazendo uma divisão de 80.000 para treino, 10.000 para validação e 10.000 para o último teste cego. Utilizamos o otimizador RAdam (*Rectified Adam* [164]), que consiste de uma versão do otimizador Adam¹³ em que é introduzido um termo para retificar a variância da taxa de aprendizado, resolvendo alguns problemas de convergência [164]. Foram mantidos o tamanho do lote e o valor da taxa de aprendizado. Nestas condições, o resultado obtido em termos de curva ROC foi a performance de um teste cego de acordo a figura 5.19.

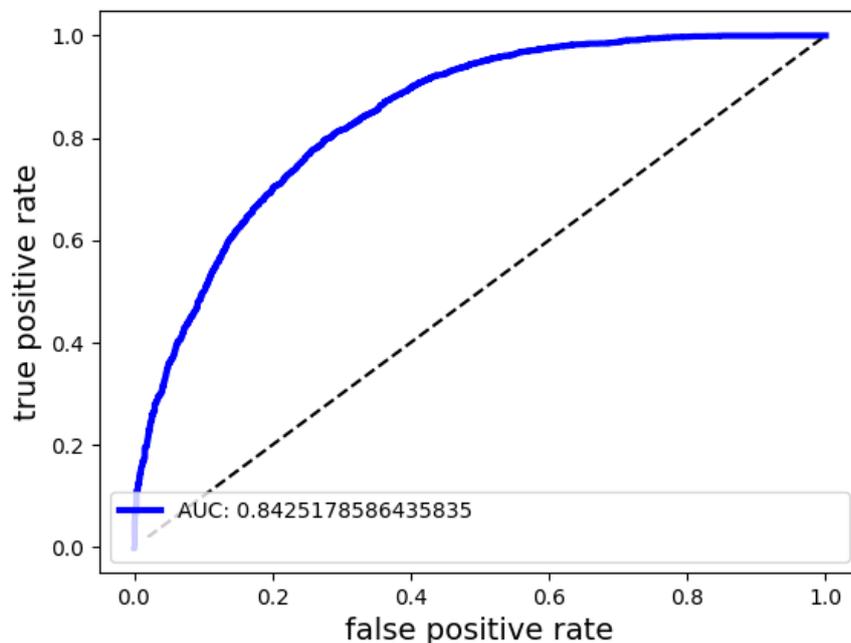


Figura 5.19 – Desempenho das redes EfficientNet B2 concatenadas nos dados da segunda edição do GLFC.

A curva em azul representa a ROC para as previsões dos casos tabelados como lente, para a qual a área sob a curva fica por volta de 0,84, sendo esta a pontuação da classificação. Tal pontuação já demonstra que a arquitetura utilizada adquiriu certa capacidade

¹³ Um método estocástico de otimização de descida de gradiente proposto em [165]

de discernimento entre os conjuntos de dados, diferentemente dos testes anteriores, o que valida o uso da arquitetura proposta.

A partir deste resultado, utilizamos esta arquitetura para definir múltiplos treinos a fim de investigar o limite de poucos dados para uma classificação ótima nessa base de dados. Foram mantidos o tamanho dos conjuntos de validação e de teste como 10.000, dada a grande quantidade de dados disponível¹⁴. Lançando mão do recurso dos múltiplos treinos em *folders*, fizemos as execuções reduzindo cada vez mais o tamanho dos conjuntos de treinamento e registramos as médias dos *folders*, resultando na figura 5.20.

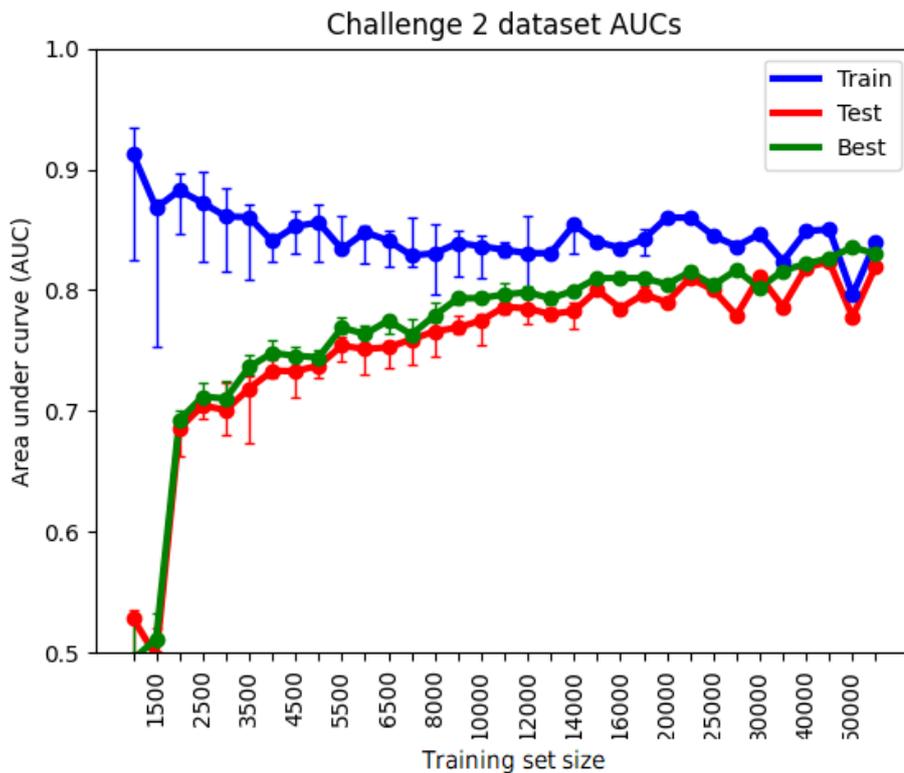


Figura 5.20 – Desempenho das redes EfficientNet B2 concatenadas nos dados da segunda edição do GLFC, para diversos tamanhos de conjuntos de treino.

Nesta figura, tem-se no eixo horizontal o tamanho dos conjuntos de treino enquanto o vertical refere-se aos valores de AUC. A linha em azul legendada como "Train" refere-se aos valores médios de AUC no caso de um teste do modelo em sua última época sob os próprios dados de treino. A linha em vermelho "Teste" indica a performance do modelo em sua última época num teste cego. E por fim, a linha em verde indica a performance do modelo em sua melhor época, de acordo a função custo de validação calculada. Tomamos como melhor época aquela em que o desempenho da rede no conjunto de validação foi o mais alto.

Estes três casos foram colocados no gráfico da figura 5.20 a fim de observar como podem influenciar as possibilidades de tomarmos um limite de poucos dados. Observando

¹⁴ O código utilizado para gerar os treinos desta base de dados, para dados já pré-processados, está presente na pasta "CH2" do repositório <https://github.com/KayqueTeles/DISSERTACAO.git>

a curva do melhor modelo, seria possível afirmar que uma quantidade mínima de dados de treino necessários para se atingir um limite de poucos dados seria 9.000 ou 10.000 dados. Já pela curva de teste, esse valor precisaria ser um pouco mais alto, apresentando um pico em 15.000 acompanhado de baixas até um pico um pouco maior em 20.000 dados. Portanto, enquanto de acordo o raciocínio da base de dados da primeira edição teríamos 15.000 como limite inferior aqui, é possível diminuir um pouco esse limite se fizermos a seleção do modelo que melhor performou ao longo das épocas para 10.000 ou até 9.000 objetos. Este será, portanto, o parâmetro a ser utilizado como último resultado destes testes, no que se refere ao limite de poucos dados.

A queda abrupta nos valores iniciais expõe mais uma vez o regime de sobreajuste: enquanto o aprendizado com dados de treino atingem uma alta taxa de acerto, as previsões de testes cegos tornam-se indistinguíveis do regime aleatório na classificação binária.

5.3.4 Desempenho das Efficient Net B2 concatenadas na base de dados do GLFC variando o critério da tabela-verdade

Um teste que pode se mostrar promissor para trabalhos futuros, além de no sentido de explorar o potencial de classificação puro da Efficient Net B2 concatenada nesta base de dados é explorar como melhora a classificação de acordo o critério de seleção dos casos de lente. Ao introduzir esta nova base de dados, exploramos de modo geral quais são as variáveis disponíveis na base de dados e quais delas se mostraram relevantes para definir a tabela verdade. De acordo a seção 5.3.1, seriam considerados casos positivos de lenteamento os objetos que demonstrassem: número de grupos de pixels acima de 1σ do *background* maior que zero; magnificação efetiva da fonte em todas as bandas maior que 1.6 e; número de pixels da fonte maior que 20. Por esse critério, entre os 100.000, 49.213 objetos foram tabelados como lentes.

No entanto, este é um critério mínimo de objetos que serão adotados como lentes, isto é, existem muitos casos de objetos que são difíceis de se discernir como lentes, entre alguns em que o fenômeno é bem evidente. Alterando esses critérios, pode ser possível filtrar com maior precisão os resultados da classificação. Resolvemos fazer as seguintes flutuações para este estudo: magnificação efetiva da fonte em todas as bandas no intervalo 1,1-1,6-2,0 e número de pixels da fonte de 10-50-100. Durante as execuções utilizamos 20.000 imagens de treino, 10.000 para validação e 10.000 para testes, e os demais parâmetros que já se provaram úteis nesta base de dados. Na tabela 1, as AUCs foram escolhidas como parâmetro para verificar as eficácias dos treinos em cada caso, destacando o desempenho do modelo final com os dados de treino e num teste cego, e um teste cego usando o melhor modelo. A linha destacada em amarelo corresponde à performance adotando o critério de lente proposto no desafio.

Na tabela, “Mag.” refere-se à magnificação mínima, “Nº Pixels” se refere ao número

Tabela 1 – Comparação entre AUCs variando o critério de Lente Gravitacional

Mag.	Nº Pixels	Nº Lentes	AUC		
			Final - Treino	Final - Cego	Melhor - Cego
1,1	10	67102 (67,1 %)	0,83	0,78	0,78
1,1	50	36961 (37 %)	0,93	0,89	0,89
1,1	100	22647 (22,6 %)	0,94	0,91	0,94
1,6	10	57887 (57,9 %)	0,80	0,75	0,77
1,6	20	49213 (49,2 %)	0,86	0,79	0,80
1,6	50	33065 (33 %)	0,92	0,87	0,88
1,6	100	20834 (20,8 %)	0,95	0,91	0,93
2,0	10	45684 (45,7 %)	0,80	0,76	0,78
2,0	50	28040 (28 %)	0,93	0,88	0,88
2,0	100	18528 (18,5 %)	0,95	0,92	0,92

mínimo de pixels da fonte. “Nº Lentes” refere-se ao número de objetos da base de dados que atende a este critério e, portanto, constituem de casos positivos de ocorrência de lenteamento gravitacional. As colunas sob o termo “AUC” referem-se aos valores de AUC encontrados ao final do treino para cada um dos seguintes casos: “Final - Treino” referentes ao modelo final utilizando dados de treino, “Final - Cego” referentes ao modelo final num teste cego e, “Melhor - Cego” referindo-se ao melhor modelo escolhido com base na melhor época. No caso do critério já adotado previamente, de magnificação acima de 1,6 e número de pixels maior que 20, os seguintes valores de AUC foram obtidos: 0,84 para o final com dados de treino, 0,78 para o final no teste cego e 0,80 para o melhor modelo num teste cego.

Deste teste é possível verificar que as situações que apresentam maiores valor de AUC correspondem aos valores de magnificação maior que 1,1 e número de pixels de 50 e 100, maior que 1,6 e número de pixels de 50 e 100, e de magnificação maior que 2,0 com número de pixels maior que 50 e 100. Em especial, os casos de número de pixels maior que 100 foram os que tiveram melhor desempenho, alcançando um desempenho do melhor modelo num teste cego de 0,93. Este, portanto, pode ser dito como o fator determinante para uma maior AUC no modelo. Este valor está bem acima do valor utilizado na análise do número de samples, de 0,80. Esta análise sugere, portanto, que utilizar esse critério de lente para o número de objetos é capaz de melhorar consideravelmente a performance da classificação do modelo, em relação ao critério de lente descrito na proposição da segunda edição do GLFC.

5.4 Conclusões e Discussão

Considerando tudo o que foi exposto no capítulo, podemos realizar algumas considerações.

- No início da seção 5.2, vimos que a ResNet50 por si só, sem grandes adequações, é capaz de reproduzir os resultados competitivos da primeira edição do GLFC. A AUC encontrada foi de 0,965 enquanto o valor mais alto em Metcalf et al. é de 0,98 [154]. Este resultado foi obtido usando 16000 dados de treino, 2000 de validação e 2000 de teste, numa execução única.
- Ao começar a realizar múltiplos treinos em dez pastas na subseção 5.2.2, foi possível adicionar ao gráfico barras de erro e tomar o valor pela média. Começamos também a verificar o desempenho da classificação diminuindo o tamanho da base de dados de treino. Vimos que aqui, o regime ótimo de classificação de acordo Hosmer (2013) [85] seria atingido por volta dos 450 objetos, na figura 5.6. Significa que em média, 450 amostras de lentes e não-lentes seriam necessários para atingir uma qualidade de classificação próxima a 0,8. Idealmente, um número próximo a metade destes deveriam ser casos positivos de lenteamento, isto é, 225. Este valor já é menor do que o tamanho da base de dados do *Master Lens Database*, podendo resultar na execução até de mais de um *fold*.
- Observamos o efeito da aumentação de dados e do pré-carregamento de pesos, nas respectivas subseções 5.2.4 e 5.2.3. Enquanto a aumentação de dados pouco contribuiu, inicializar os pesos evitando o regime aleatório chegou a reduzir o número de recortes de lentes e não lentes para 160. Significaria ter 80 casos de lentes conhecidos, o que reduz ainda mais esse limiar necessário.
- Já na subseção 5.2.5, este limite foi reduzido para 80 objetos na base de dados, através da estratégia do ensemble de redes pela média simples. Isto aponta um indício de que o ensemble de redes, portanto, é uma ferramenta promissora. Este é semelhante ao obtido em uma das pulições relacionadas a dissertação, Castro & Teles et al. (2021) [143], entretanto, no referido trabalho, algumas escolhas metodológicas fizeram as barras de erro menores. De qualquer modo, o valores concordam dentro do erro. Significaria precisar de 40 casos de efeito de lente a fim de fazer uma base de dados que pode ser treinada. Tomando os 675 casos do *Master Lens Database*, seria possível fazer 10 pastas com dados de lentes (resultando em 400 lentes utilizadas) e ainda sobrariam 275 para compor um conjunto de validação.
- Cabe enunciar, no entanto, que esse resultado é válido apenas para a rede neural de ensemble da ResNet50 e EfficientNet B2, nos dados da primeira edição do GLFC. Nos dados da segunda edição, como vimos na subseção 5.3.2, esse mesmo ensemble de redes não consegue recuperar performance similar, se tornando indiferente da classificação aleatória. Para esta base de dados foi necessária uma outra rede neural, como a concatenação de duas Efficient et B2, que tratam a situação de imagens com resoluções diferentes.

- Na subseção 5.3.3, vimos o desempenho da EfficientNetB2 na base de dados da segunda edição. Seria bastante satisfatório conseguir reproduzir o limite mínimo de 80 imagens de treino necessárias, no entanto, a arquitetura utilizada não foi capaz de tanto na nossa condição de uso. Neste trabalho, atingimos o limiar com cerca de 10.000 dados, valor bem alto em relação a quantidade de registros de eventos de lenteamento já catalogados. Não se pode esquecer, no entanto, que a base de dados da segunda edição é muito mais complexa.
- Por fim, na subseção 5.3.4 tivemos o uso de um recurso que poderia ser usado para refinar a os resultados do uso das duas EfficientNetB2 concatenadas nos dados da segunda edição: a restrição do critério de lente. Verificamos que restringir principalmente o número de pixels da fonte para acima de 100 teve efeitos bastante positivos no aumento da AUC.

Estas foram as conclusões que puderam ser retiradas do trabalho, e que poderão nortear esforços futuros.

CAPÍTULO 6

CONSIDERAÇÕES FINAIS

Neste trabalho verificamos algumas estratégias que podem ser usadas para melhorar a classificação de lentes gravitacionais nas bases de dados do *Gravitational Lens Finding Challenge*. Alcançamos o limiar de 80 objetos na base de dados da primeira edição do GLFC, como número que deveria contemplar amostras contendo lentes gravitacionais e não, enquanto para os dados da segunda edição a marca atingida foi de 10.000 objetos. Cabe observar o efeito da escolha da arquitetura neste resultado final, e a importância da otimização das variáveis utilizadas na classificação. À princípio, não se imaginava que um esquema de rede neural que performou tão bem nas bases de dados da primeira edição, como o ensemble de uma ResNet50 e EfficientNet B2, precisaria ser revisto na base de dados da segunda edição, desde que se executasse o treino sem uso da primeira base [160].

Por fim, a não obtenção do mesmo limiar de dados obtidos com a base de dados da primeira edição do GLFC nos apontou que ainda havia trabalho a ser feito. Provavelmente há alternativas metodológicas que poderão melhorar esse limiar de 10.000 para estas bases de dados. De fato, para bases de dados restritas a um pequeno número de exemplos, a capacidade de generalização da rede é essencial e portanto não encontramos uma solução única, devendo cada caso ser discutido.

No presente, um gargalo para o desenvolvimento deste tipo de técnica costuma ser tempo de máquina. No caso da ResNet50, por exemplo, a execução inicial exige o aguardo de um treino durante 50 épocas. Separar os dados em dez pastas exigiria aguardar este tempo de 50 épocas dez vezes, e treinar com diferentes tamanhos a base de dados de treino exigiria aguardar 50 épocas dez vezes para cada tamanho de conjunto de treino. Cada novo recurso incorporado à rede aumentava ainda mais o tempo de execução necessário, e no final ainda havia uma rede a mais pra ser treinada em cada um destes passos, quando optamos pelo ensemble. Já no caso da concatenação de EfficientNet B2, apesar da EfficientNet possuir menos parâmetros, as camadas convolucionais da rede eram largas devido a maior resolução das imagens, o que também atrasou o tempo de treino. O uso de

GPUs é capaz de fazer este tipo de análise em curto intervalo de tempo. No entanto, nem sempre GPUs e clusters de GPUs não se encontram largamente disponíveis na comunidade acadêmica.

Como sugestão de trabalhos futuros, tem-se:

- Como consequência direta da descoberta da melhora da performance das EfficientNet B2 concatenadas na base de dados da segunda edição do GLFC, por conta da mudança do critério de lente, conforme a subseção 5.3.4, executar novas execuções reduzindo o tamanho do conjunto de treinamento. Do mesmo modo que foi feito nas seções 5.3.3 e 5.2.1, a fim de investigar qual seria o limiar de poucos dados alcançados com o melhor (ou cada) critério.
- Verificar como o uso de outras classes de EfficientNet concatenadas poderia alterar o resultado da classificação na base de dados da segunda edição do GLFC.
- Verificar como as EfficientNet B2 concatenadas se sairiam na base de dados da primeira edição do GLFC.
- Treinar tanto as EfficientNet B2 concatenadas quanto o ensemble da ResNet50 e EfficientNet B2 nas respectivas bases de dados em que melhor performam e, realizar um teste cego com ambas utilizando dados reais (do *Master Lens Database*, por exemplo).
- E como consequência do acima, utilizar uma base de dados reais como acervo de treino destes modelos de RNC supracitados, a fim de avaliar o seu desempenho de classificação. Diferentes levantamentos poderiam oferecer diferentes desafios nessa classificação de imagens. Não seria de se surpreender que as diferenças de filtros e instrumentos utilizados nos levantamentos fosse o suficiente para viabilizar uma arquitetura de RNC em um deles e, inviabilizar em outro.

REFERÊNCIAS

- [1] José Fernando Rocha. Origens e evolução das idéias da física. SciELO-EDUFBA, 2002.
- [2] S George Djorgovski, AA Mahabal, AJ Drake, MJ Graham, and Ciro Donalek. Sky surveys. Planets, Stars, and Stellar Systems, 2:223–281, 2013.
- [3] Stephen M Kent. Sloan digital sky survey. Astrophysics and Space Science, 217(1):27–30, 1994.
- [4] Kevork Abazajian, Jennifer K Adelman-McCarthy, Marcel A Agüeros, Sahar S Allam, Scott F Anderson, James Annis, Neta A Bahcall, Ivan K Baldry, Steven Bastian, Andreas Berlind, et al. The first data release of the sloan digital sky survey. The Astronomical Journal, 126(4):2081, 2003.
- [5] Katherine Accetta, Conny Aerts, Victor Silva Aguirre, Romina Ahumada, Nikhil Ajgaonkar, N Filiz Ak, Shadab Alam, Carlos Allende Prieto, Andres Almeida, Friedrich Anders, et al. The seventeenth data release of the sloan digital sky surveys: Complete release of manga, mastar and apogee-2 data. arXiv preprint arXiv:2112.02026, 2021.
- [6] Jelte TA de Jong, Gijs A Verdoes Kleijn, Konrad H Kuijken, and Edwin A Valentijn. The kilo-degree survey. Experimental Astronomy, 35(1):25–44, 2013.
- [7] K Kuijken, C Heymans, A Dvornik, H Hildebrandt, JTA de Jong, AH Wright, T Erben, M Bilicki, B Giblin, H-Y Shan, et al. The fourth data release of the kilo-degree survey: ugrI imaging and nine-band optical-ir photometry over 1000 square degrees. Astronomy & Astrophysics, 625:A2, 2019.
- [8] Dark Energy Survey Collaboration et al. The dark energy survey. arXiv preprint astro-ph/0510346, 2005.
- [9] Santiago Javier Avila Perez, Bob Nichol, Will Percival, Daniel B Thomas, DES Collaboration, et al. The dark energy survey: Data release 1. Astrophysical Journal, Supplement Series, 239(2):18, 2018.

- [10] TMC Abbott, M Adamow, M Aguena, S Allam, A Amon, J Annis, S Avila, D Bacon, Manda Banerji, K Bechtol, et al. The dark energy survey data release 2. The Astrophysical Journal Supplement Series, 255(2):20, 2021.
- [11] J Anthony Tyson. Large synoptic survey telescope: overview. In Survey and Other Telescope Technologies and Discoveries, volume 4836, pages 10–20. International Society for Optics and Photonics, 2002.
- [12] René J Laureijs, Ludovic Duvet, Isabel Escudero Sanz, Philippe Gondoin, David H Lumb, Tim Oosterbroek, and Gonzalo Saavedra Criado. The euclid mission. In Space Telescopes and Instrumentation 2010: Optical, Infrared, and Millimeter Wave, volume 7731, page 77311H. International Society for Optics and Photonics, 2010.
- [13] Silvia Mollerach and Esteban Roulet. Gravitational lensing and microlensing. World Scientific, 2002.
- [14] Joachim Wambsganss. Gravitational lensing in astronomy. Living Reviews in Relativity, 1(1):1–74, 1998.
- [15] Philip J Marshall, Tommaso Treu, Jason Melbourne, Raphaël Gavazzi, Kevin Bundy, S Mark Ammons, Adam S Bolton, Scott Burles, James E Larkin, David Le Mignant, et al. Superresolving distant galaxies with gravitational telescopes: Keck laser guide star adaptive optics and hubble space telescope imaging of the lens system sdss j0737+ 3216. The Astrophysical Journal, 671(2):1196, 2007.
- [16] Johan Richard, Tucker Jones, Richard Ellis, Daniel P Stark, Rachael Livermore, and Mark Swinbank. The emission line properties of gravitationally lensed 1.5 z_i 5 galaxies. Monthly Notices of the Royal Astronomical Society, 413(1):643–658, 2011.
- [17] H Ebeling, M Stockmann, J Richard, J Zabl, G Brammer, S Toft, and A Man. Thirty-fold: extreme gravitational lensing of a quiescent galaxy at $z= 1.6$. The Astrophysical Journal Letters, 852(1):L7, 2017.
- [18] Mohammad Akhshik, Katherine E Whitaker, Gabriel Brammer, Guillaume Mahler, Keren Sharon, Joel Leja, Matthew B Bayliss, Rachel Bezanson, Michael D Gladders, Allison Man, et al. Requiem-2d methodology: spatially resolved stellar populations of massive lensed quiescent galaxies from hubble space telescope 2d grism spectroscopy. The Astrophysical Journal, 900(2):184, 2020.
- [19] Allison WS Man, Johannes Zabl, Gabriel B Brammer, Johan Richard, Sune Toft, Mikkel Stockmann, Anna R Gallazzi, Stefano Zibetti, and Harald Ebeling. An exquisitely deep view of quenching galaxies through the gravitational lens: Stellar

- population, morphology, and ionized gas. *The Astrophysical Journal*, 919(1):20, 2021.
- [20] L. et al Moustakas. The master lens database and the orphan lenses project. 2022 (in prep).
- [21] Shotaro Kikuchihara, Masami Ouchi, Yoshiaki Ono, Ken Mawatari, Jacopo Chevallard, Yuichi Harikane, Takashi Kojima, Masamune Oguri, Gustavo Bruzual, and Stéphane Charlot. Early low-mass galaxies and star-cluster candidates at $z \sim 6-9$ identified by the gravitational-lensing technique and deep optical/near-infrared imaging. *The Astrophysical Journal*, 893(1):60, 2020.
- [22] Jean-Baptiste Jolly, Kirsten Knudsen, Nicolas Laporte, Johan Richard, Seiji Fujimoto, Kotaro Kohno, Yiping Ao, Franz E Bauer, Eiichi Egami, Daniel Espada, et al. Alma lensing cluster survey: A spectral stacking analysis of [c ii] in lensed $z \sim 6$ galaxies. *Astronomy & Astrophysics*, 652:A128, 2021.
- [23] N Laporte, A Zitrin, RS Ellis, S Fujimoto, G Brammer, J Richard, M Oguri, GB Caminha, K Kohno, Y Yoshimura, et al. Alma lensing cluster survey: a strongly lensed multiply imaged dusty system at $z \sim 6$. *Monthly Notices of the Royal Astronomical Society*, 505(4):4838–4846, 2021.
- [24] Frederick B Davies, Feige Wang, Anna-Christina Eilers, and Joseph F Hennawi. Constraining the gravitational lensing of $z \sim 6$ quasars from their proximity zones. *The Astrophysical Journal Letters*, 904(2):L32, 2020.
- [25] Robert M Quimby, Masamune Oguri, Anupreeta More, Surhud More, Takashi J Moriya, Marcus C Werner, Masayuki Tanaka, Gaston Folatelli, Melina C Bersten, Keiichi Maeda, et al. Detection of the gravitational lens magnifying a type ia supernova. *Science*, 344(6182):396–399, 2014.
- [26] Patrick L Kelly, Steven A Rodney, Tommaso Treu, Ryan J Foley, Gabriel Brammer, Kasper B Schmidt, Adi Zitrin, Alessandro Sonnenfeld, Louis-Gregory Strolger, Or Graur, et al. Multiple images of a highly magnified supernova formed by an early-type cluster galaxy lens. *Science*, 347(6226):1123–1126, 2015.
- [27] Isaac Spitzer. *Measuring time delays in strongly lensed quasars using wide-field survey data*. PhD thesis, San Diego State University, 2016.
- [28] AO Jaunsen and J Hjorth. Detection of a spiral lens galaxy and optical variability in the gravitational lens system b1600+ 434. *arXiv preprint astro-ph/9611159*, 1996.
- [29] Viatcheslav Mukhanov. *Physical foundations of cosmology*. Cambridge university press, 2005.

- [30] Sjur Refsdal. On the possibility of determining hubble's parameter and the masses of galaxies from the gravitational lens effect. Monthly Notices of the Royal Astronomical Society, 128(4):307–310, 1964.
- [31] Charles R Keeton and Leonidas A Moustakas. A new channel for detecting dark matter substructure in galaxies: gravitational lens time delays. The Astrophysical Journal, 699(2):1720, 2009.
- [32] AM Nierenberg, T Treu, SA Wright, CD Fassnacht, and MW Auger. Detection of substructure with adaptive optics integral field spectroscopy of the gravitational lens b1422+ 231. Monthly Notices of the Royal Astronomical Society, 442(3):2434–2445, 2014.
- [33] Simona Vegetti and Léon VE Koopmans. Bayesian strong gravitational-lens modeling on adaptive grids: objective detection of mass substructure in galaxies. Monthly Notices of the Royal Astronomical Society, 392(3):945–963, 2009.
- [34] GG Nyambuya and W Simango. On the gravitational bending of light—was sir arthur stanley eddington right? International Journal of Astronomy and Astrophysics, 2014, 2014.
- [35] Martin Makler. Lentes gravitacionais: Um rico recurso didático na ocasião da celebração dos 100 anos do eclipse de sobral. Conexões-Ciência e Tecnologia, 13(2):21–36, 2019.
- [36] Jaume Giné. On the origin of the deflection of light. Chaos, Solitons & Fractals, 35(1):1–6, 2008.
- [37] Frank Watson Dyson, Arthur Stanley Eddington, and Charles Davidson. Ix. a determination of the deflection of light by the sun's gravitational field, from observations made at the total eclipse of may 29, 1919. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 220(571-581):291–333, 1920.
- [38] David Valls-Gabaud. The conceptual origins of gravitational lensing. In AIP Conference Proceedings, volume 861, pages 1163–1163. American Institute of Physics, 2006.
- [39] Albert Einstein. Lens-like action of a star by the deviation of light in the gravitational field. Science, 84(2188):506–507, 1936.
- [40] Fritz Zwicky. Nebulae as gravitational lenses. Physical Review, 51(4):290, 1937.
- [41] Dennis Walsh, Robert F Carswell, and Ray J Weymann. 0957+ 561 a, b: twin quasistellar objects or gravitational lens? Nature, 279(5712):381–384, 1979.

- [42] Jaan Pelt, Rainer Kayser, Sjur Refsdal, and Thomas Schramm. The light curve and the time delay of qso 0957+ 561. arXiv preprint astro-ph/9501036, 1995.
- [43] Shadab Alam, Franco D Albareti, Carlos Allende Prieto, Friedrich Anders, Scott F Anderson, Timothy Anderton, Brett H Andrews, Eric Armengaud, Éric Aubourg, Stephen Bailey, et al. The eleventh and twelfth data releases of the sloan digital sky survey: final data from sdss-iii. The Astrophysical Journal Supplement Series, 219(1):12, 2015.
- [44] IWA Browne, PN Wilkinson, NJF Jackson, ST Myers, CD Fassnacht, LVE Koopmans, DR Marlow, M Norbury, D Rusin, CM Sykes, et al. The cosmic lens all-sky survey-ii. gravitational lens candidate selection and follow-up. Monthly Notices of the Royal Astronomical Society, 341(1):13–32, 2003.
- [45] Masamune Oguri, Naohisa Inada, Bartosz Pindor, Michael A Strauss, Gordon T Richards, Joseph F Hennawi, Edwin L Turner, Robert H Lupton, Donald P Schneider, Masataka Fukugita, et al. The sloan digital sky survey quasar lens search. i. candidate selection algorithm. The Astronomical Journal, 132(3):999, 2006.
- [46] Massimo Meneghetti. Introduction to Gravitational Lensing: With Python Examples, volume 956. Springer Nature, 2021.
- [47] Matts Roos. Introduction to cosmology. John Wiley & Sons, 2015.
- [48] Ian Morison. Introduction to astronomy and cosmology. John Wiley & Sons, 2013.
- [49] Cristina Furlanetto. Arcos gravitacionais em aglomerados de galáxias: detecção, caracterização e modelamento. 2012.
- [50] Francesco De Paolis, Mosè Giordano, Gabriele Ingrosso, Luigi Manni, Achille Nucita, and Francesco Strafella. The scales of gravitational lensing. Universe, 2(1):6, 2016.
- [51] Vanessa Pacheco De Freitas. Arcos gravitacionais na escala galáctica: Modelagem analítica e seções de choque. 2018.
- [52] Nilo Sylvio Costa Serpa. COSMOLOGIA DE LEMAÎTRE-TOLMAN: LENTEAMENTO GRAVITACIONAL FRACO E OUTRAS APLICAÇÕES. PhD thesis, Universidade Federal do Rio de Janeiro Rio de Janeiro, 2009.
- [53] Patrick Kelly. An individual star at redshift 1.5 extremely magnified by a galaxy-cluster lens. In American Astronomical Society Meeting Abstracts# 231, volume 231, pages 415–05, 2018.

- [54] Nicha Leethochawalit, Tucker A Jones, Richard S Ellis, Daniel P Stark, Johan Richard, Adi Zitrin, and Matthew Auger. A keck adaptive optics survey of a representative sample of gravitationally lensed star-forming galaxies: high spatial resolution studies of kinematics and metallicity gradients. The Astrophysical Journal, 820(2):84, 2016.
- [55] G Soucail, OB Fort, Y Mellier, and JP Picat. A blue ring-like structure in the center of the a 370 cluster of galaxies. Astronomy and astrophysics, 172:L14–L16, 1987.
- [56] J Lehar, RG McMahon, M Irwin, SR Conner, and BF Burke. A four-image gravitational lens in mg0751+ 2716. In Bulletin of the American Astronomical Society, volume 25, page 1343, 1993.
- [57] Frédéric Courbin and Dante Minniti. Gravitational lensing: an astrophysical tool, volume 608. Springer Science & Business Media, 2002.
- [58] Vanessa P de Freitas, Martin Makler, and Habib S Dúmet-Montoya. Strong lensing cross-sections for isothermal models. i. finite source effects in the circular case. Monthly Notices of the Royal Astronomical Society, 481(2):2189–2204, 2018.
- [59] Ana Carolina Lorena, João Gama, and Katti Faceli. Inteligência Artificial: Uma abordagem de aprendizado de máquina. Grupo Gen-LTC, 2000.
- [60] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4):115–133, 1943.
- [61] Rafael Izbicki and Tiago Mendonça dos Santos. Aprendizado de máquina: uma abordagem estatística. Rafael Izbicki, 2020.
- [62] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat’s striate cortex. The Journal of physiology, 148(3):574–591, 1959.
- [63] Matteo Carandini. What simple and complex cells compute. The Journal of physiology, 577(Pt 2):463, 2006.
- [64] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. The Journal of physiology, 160(1):106–154, 1962.
- [65] Kuniyiko Fukushima. Neocognitron—a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. NHK , (15):p106–115, 1981.
- [66] Kuniyiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In Competition and cooperation in neural nets, pages 267–285. Springer, 1982.

- [67] Kunihiko Fukushima, Sei Miyake, and Takayuki Ito. Neocognitron: A neural network model for a mechanism of visual pattern recognition. IEEE transactions on systems, man, and cybernetics, (5):826–834, 1983.
- [68] Yann LeCun, Lawrence D Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, Urs A Muller, Eduard Sackinger, Patrice Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. Neural networks: the statistical mechanics perspective, 261(276):2, 1995.
- [69] The mnist database of handwritten digits. [Disponível em: <https://http://yann.lecun.com/exdb/mnist/>]. Acessado em 10-Jan-2022.
- [70] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning, pages 448–456. PMLR, 2015.
- [71] Aurélien Géron. Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow. Alta Books, 2019.
- [72] Raúl Rojas. Neural networks: a systematic introduction. Springer Science & Business Media, 2013.
- [73] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.
- [74] Keiron O’Shea and Ryan Nash. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458, 2015.
- [75] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Transactions on Neural Networks and Learning Systems, 2021.
- [76] Lars Lien Ankile, Morgan Feet Heggland, and Kjartan Krange. Deep convolutional neural networks: A survey of the foundations, selected improvements, and some current applications. arXiv preprint arXiv:2011.12960, 2020.
- [77] David Stutz. Learning shape completion from bounding boxes with cad shape priors, 2017.
- [78] Marina Chane. Classificação de imagens médicas: entendendo a rede neural convolucional, 2021. [Disponível em: <https://www.imaios.com/br/Empresa/blog/Classificacao-de-imagens-medicas-entendendo-a-rede-neural-convolucional-CNN>]. Acessado em 12-Jan-2022.

- [79] Shipra Saxena. Binary cross entropy/log loss for binary classification, 2021. [Disponível em: <https://www.analyticsvidhya.com/blog/2021/03/binary-cross-entropy-log-loss-for-binary-classification/>]. Acessado em 12-Jan-2022.
- [80] T Fawcett. Introduction to roc analysis, 27, 861–874. *Science Direct*, 2006.
- [81] Zhe Hui Hoo, Jane Candlish, and Dawn Teare. What is an roc curve?, 2017.
- [82] Shengping Yang and Gilbert Berdine. The receiver operating characteristic (roc) curve. *The Southwest Respiratory and Critical Care Chronicles*, 5(19):34–36, 2017.
- [83] Tom Fawcett. Roc graphs: Notes and practical considerations for researchers. *Machine learning*, 31(1):1–38, 2004.
- [84] Tatiana Cristina Figueira Polo and Hélio Amante Miot. Use of roc curves in clinical and experimental studies, 2020.
- [85] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.
- [86] Tatiana Cristina Figueira Polo and Hélio Amante Miot. Aplicações da curva roc em estudos clínicos e experimentais, 2020.
- [87] Jayawant N Mandrekar. Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9):1315–1316, 2010.
- [88] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- [89] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [90] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [91] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [92] M Mattheakis and P Protopapas. Recurrent neural networks: Exploding vanishing gradients & reservoir computing. In *Advanced Topics in Data Science*. Harvard Press, 2019.

- [93] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2):157–166, 1994.
- [94] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5353–5360, 2015.
- [95] M Siddhartha. Understanding resnet and analyzing various models on the cifar-10 dataset, 2021. [Disponível em: <https://www.analyticsvidhya.com/blog/2021/06/understanding-resnet-and-analyzing-various-models-on-the-cifar-10-dataset/>. Acessado em 15-Jan-2022.
- [96] Daquan Zhou, Qibin Hou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Rethinking bottleneck structure for efficient mobile network design. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, pages 680–697. Springer, 2020.
- [97] Francois et. al Chollet. Keras, 2015.
- [98] Aakash Kaushik. Understanding resnet50 architecture, 2021.
- [99] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning, pages 6105–6114. PMLR, 2019.
- [100] Atul Pandey. Depth-wise convolution and depth-wise separable convolution, 2018.
- [101] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4510–4520, 2018.
- [102] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1):1929–1958, 2014.
- [103] Vardan Agarwal. Complete architectural details of all efficientnet models, 2020.
- [104] Frank Lenzen, Sabine Schindler, and Otmar Scherzer. Automatic detection of arcs and arclets formed by gravitational lensing. Astronomy & Astrophysics, 416(1):391–401, 2004.

- [105] Jean-Luc Starck, David L Donoho, and Emmanuel J Candès. Astronomical image representation by the curvelet transform. *Astronomy & Astrophysics*, 398(2):785–800, 2003.
- [106] RA Cabanac, C Alard, M Dantel-Fort, B Fort, R Gavazzi, P Gomez, JP Kneib, O Le Fevre, Y Mellier, R Pello, et al. The cftls strong lensing legacy survey-i. survey overview and t0002 release sample. *Astronomy & Astrophysics*, 461(3):813–821, 2007.
- [107] C Alard. Automated detection of gravitational arcs. [arXiv preprint astro-ph/0606757](https://arxiv.org/abs/astro-ph/0606757), 2006.
- [108] Adam S Bolton, Scott Burles, Léon VE Koopmans, Tommaso Treu, and Leonidas A Moustakas. The sloan lens acs survey. i. a large spectroscopically selected sample of massive early-type lens galaxies. *The Astrophysical Journal*, 638(2):703, 2006.
- [109] Adam S Bolton, Scott Burles, Léon VE Koopmans, Tommaso Treu, Raphaël Gavazzi, Leonidas A Moustakas, Randall Wayth, and David J Schlegel. The sloan lens acs survey. v. the full acs strong-lens sample. *The Astrophysical Journal*, 682(2):964, 2008.
- [110] Tommaso Treu, Léon V Koopmans, Adam S Bolton, Scott Burles, and Leonidas A Moustakas. The sloan lens acs survey. ii. stellar populations and internal structure of early-type lens galaxies. *The Astrophysical Journal*, 640(2):662, 2006.
- [111] MW Auger, T Treu, AS Bolton, R Gavazzi, LVE Koopmans, PJ Marshall, K Bundy, and LA Moustakas. The sloan lens acs survey. ix. colors, lensing, and stellar masses of early-type galaxies. *The Astrophysical Journal*, 705(2):1099, 2009.
- [112] Yiping Shu, Adam S Bolton, Joel R Brownstein, Antonio D Montero-Dorta, Léon VE Koopmans, Tommaso Treu, Raphaël Gavazzi, Matthew W Auger, Oliver Czoske, Philip J Marshall, et al. The sloan lens acs survey. xii. extending strong lensing to lower masses. *The Astrophysical Journal*, 803(2):71, 2015.
- [113] Yiping Shu, Joel R Brownstein, Adam S Bolton, Léon VE Koopmans, Tommaso Treu, Antonio D Montero-Dorta, Matthew W Auger, Oliver Czoske, Raphaël Gavazzi, Philip J Marshall, et al. The sloan lens acs survey. xiii. discovery of 40 new galaxy-scale strong lenses. *The Astrophysical Journal*, 851(1):48, 2017.
- [114] Raphael Gavazzi, Tommaso Treu, Léon VE Koopmans, Adam S Bolton, Leonidas A Moustakas, Scott Burles, and Philip J Marshall. The sloan lens acs survey. vi. discovery and analysis of a double einstein ring. *The Astrophysical Journal*, 677(2):1046, 2008.

- [115] B Nord, E Buckley-Geer, H Lin, HT Diehl, J Helsby, N Kuropatkin, Adam Amara, T Collett, S Allam, GB Caminha, et al. Observation and confirmation of six strong-lensing systems in the dark energy survey science verification data. The Astrophysical Journal, 827(1):51, 2016.
- [116] Brian Nord, E Buckley-Geer, H Lin, N Kuropatkin, T Collett, DL Tucker, HT Diehl, A Agnello, A Amara, TMC Abbott, et al. Observation and confirmation of nine strong-lensing systems in dark energy survey year 1 data. Monthly Notices of the Royal Astronomical Society, 494(1):1308–1322, 2020.
- [117] H Thomas Diehl, EJ Buckley-Geer, KA Lindgren, B Nord, H Gaitsch, S Gaitsch, H Lin, S Allam, TE Collett, C Furlanetto, et al. The des bright arcs survey: hundreds of candidate strongly lensed galaxy systems from the dark energy survey science verification and year 1 observations. The Astrophysical Journal Supplement Series, 232(1):15, 2017.
- [118] J Estrada, J Annis, HT Diehl, PB Hall, T Las, H Lin, M Makler, KW Merritt, V Scarpine, S Allam, et al. A systematic search for high surface brightness giant arcs in a sloan digital sky survey cluster sample. The Astrophysical Journal, 660(2):1176, 2007.
- [119] R Joseph, F Courbin, RB Metcalf, C Giocoli, P Hartley, N Jackson, F Bellagamba, J-P Kneib, L Koopmans, G Lemson, et al. A pca-based automated finder for galaxy-scale strong lenses. Astronomy & Astrophysics, 566:A63, 2014.
- [120] Rene Laureijs, J Amiaux, S Arduini, J-L Augueres, J Brinchmann, R Cole, M Cropper, C Dabin, L Duvet, A Ealet, et al. Euclid definition study report. arXiv preprint arXiv:1110.3193, 2011.
- [121] D Paraficz, F Courbin, A Tramacere, R Joseph, RB Metcalf, J-P Kneib, P Dubath, D Droz, F Filleul, D Ringeisen, et al. The pca lens-finder: application to cfhtls. Astronomy & Astrophysics, 592:A75, 2016.
- [122] CR Bom, M Makler, MP Albuquerque, and CH Brandt. A neural network gravitational arc finder based on the mediatrix filamentation method. Astronomy & Astrophysics, 597:A135, 2017.
- [123] C Brandt. Simulacao e Busca automatizada de arcos gravitacionais, 2012. PhD thesis, Dissertação (Mestrado).
- [124] CE Petrillo, CRESCENZO Tortora, S Chatterjee, G Vernardos, LVE Koopmans, G Verdoes Kleijn, NICOLA ROSARIO Napolitano, G Covone, P Schneider, ANIELLO Grado, et al. Finding strong gravitational lenses in the kilo degree survey

- with convolutional neural networks. Monthly Notices of the Royal Astronomical Society, 472(1):1129–1150, 2017.
- [125] Colin Jacobs, Karl Glazebrook, Thomas Collett, Anupreeta More, and Christopher McCarthy. Finding strong lenses in cftls using convolutional neural networks. Monthly Notices of the Royal Astronomical Society, 471(1):167–181, 2017.
- [126] Thomas E Collett. The population of galaxy–galaxy strong lenses in forthcoming optical imaging surveys. The Astrophysical Journal, 811(1):20, 2015.
- [127] James Pearson, Clara Pennock, and Tom Robinson. Auto-detection of strong gravitational lenses using convolutional neural networks. Emergent Scientist, 2:1, 2018.
- [128] Ana Diaz Rivero and Cora Dvorkin. Direct detection of dark matter substructure in strong lens images with convolutional neural networks. Physical Review D, 101(2):023515, 2020.
- [129] Milad Pourrahmani, Hooshang Nayyeri, and Asantha Cooray. Lensflow: A convolutional neural network in search of strong gravitational lenses. The Astrophysical Journal, 856(1):68, 2018.
- [130] Yashar D Hezaveh, Laurence Perreault Levasseur, and Philip J Marshall. Fast automated analysis of strong gravitational lenses with convolutional neural networks. Nature, 548(7669):555–557, 2017.
- [131] C Schaefer, M Geiger, T Kuntzer, and J-P Kneib. Deep convolutional neural networks as strong gravitational lens detectors. Astronomy & Astrophysics, 611:A2, 2018.
- [132] R Li, NR Napolitano, C Spiniello, C Tortora, K Kuijken, LVE Koopmans, P Schneider, F Getman, L Xie, L Long, et al. High-quality strong lens candidates in the final kilo-degree survey footprint. The Astrophysical Journal, 923(1):16, 2021.
- [133] R Canameras, S Schuldt, Y Shu, SH Suyu, S Taubenberger, T Meinhardt, L Leal-Taixé, DC-Y Chao, KT Inoue, AT Jaelani, et al. Holismokes. vi. new galaxy-scale strong lens candidates from the hsc-ssp imaging survey. arXiv preprint arXiv:2107.07829, 2021.
- [134] Christopher Storfer, Andrew Pilon, Varun Ravi, and Matthew Domingo. Finding strong gravitational lenses with residual neural networks. 2020.
- [135] Sandeep Madireddy, Nan Li, Nesar Ramachandra, Prasanna Balaprakash, and Salman Habib. Modular deep learning analysis of galaxy-scale strong lensing images. ArXiv, 2019.

- [136] Sandeep Madireddy, Nan Li, Nesar Ramachandra, James Butler, Prasanna Balaprakash, Salman Habib, and Katrin Heitmann. A modular deep learning pipeline for galaxy-scale strong gravitational lens detection and modeling. arXiv preprint arXiv:1911.03867, 2019.
- [137] Rui Li, Yiping Shu, Jianlin Su, Haicheng Feng, Guobao Zhang, Jiancheng Wang, and Hongtao Liu. Using deep residual networks to search for galaxy-ly α emitter lens candidates based on spectroscopic selection. Monthly Notices of the Royal Astronomical Society, 482(1):313–320, 2019.
- [138] Xiaosheng Huang, Christopher Storfer, V Ravi, A Pilon, M Domingo, DJ Schlegel, S Bailey, A Dey, RR Gupta, D Herrera, et al. Finding strong gravitational lenses in the desi decam legacy survey. The Astrophysical Journal, 894(1):78, 2020.
- [139] R Li, NR Napolitano, C Tortora, C Spiniello, LVE Koopmans, Z Huang, N Roy, G Vernardos, S Chatterjee, B Giblin, et al. New high-quality strong lens candidates with deep learning in the kilo-degree survey. The Astrophysical Journal, 899(1):30, 2020.
- [140] Abia Khan Akhazhanov. Discovering strong gravitational lensing with deep learning. University of California, Los Angeles, 2018.
- [141] K Rojas, E Savary, B Clément, M Maus, F Courbin, C Lemon, JHH Chan, G Vernardos, R Joseph, R Cañameras, et al. Strong lens systems search in the dark energy survey using convolutional neural networks. arXiv preprint arXiv:2109.00014, 2021.
- [142] E Savary, K Rojas, M Maus, B Clément, F Courbin, R Gavazzi, JHH Chan, C Lemon, G Vernardos, R Cañameras, et al. A search for galaxy-scale strong gravitational lenses in the ultraviolet near infrared optical northern survey (unions). arXiv preprint arXiv:2110.11972, 2021.
- [143] Icaro Castro, Kayque Teles, Clécio Bom, and Tatiana Escovedo. Desenvolvendo um ensemble de redes profundas para identificação de lentes gravitacionais: Aplicação em regime de poucos dados. NOTAS TÉCNICAS, 11(1), 2021.
- [144] Ji Won Park, Sebastian Wagner-Carena, Simon Birrer, Philip J Marshall, Joshua Yao-Yu Lin, Aaron Roodman, LSST Dark Energy Science Collaboration, et al. Large-scale gravitational lens modeling with bayesian neural networks for accurate and precise inference of the hubble constant. The Astrophysical Journal, 910(1):39, 2021.
- [145] Daniel Magro, Kristian Zarb Adami, Andrea DeMarco, Simone Riggi, and Eva Sciacca. A comparative study of convolutional neural networks for the detection of

- strong gravitational lensing. Monthly Notices of the Royal Astronomical Society, 2021.
- [146] Manuel Speranza Torres Veras and Cristina Furlanetto. Detecção de lentes gravitacionais utilizando aprendizado de máquina. In Universidade Presente. UFRGS, 2019.
- [147] Bologna lens factory: a database of simulated gravitational lenses. Bologna Lens Factory, 2013. Disponível em: <http://metcalf1.difa.unibo.it/blf-portal/index.html>. Acessado em 29-Ago-2021.
- [148] Renée Hložek. Data challenges as a tool for time-domain astronomy. Publications of the Astronomical Society of the Pacific, 131(1005):118001, 2019.
- [149] Alastair Edge, William Sutherland, Konrad Kuijken, Simon Driver, Richard McMahon, Steve Eales, and Jim P Emerson. The vista kilo-degree infrared galaxy (viking) survey: bridging the gap between low and high redshift. The Messenger, 154:32–34, 2013.
- [150] Euclid and the origin of the accelerating universe. Euclid Consortium, 2017. Disponível em: <https://www.euclid-ec.org>. Acessado em 29-Ago-2021.
- [151] Spacecraft. Euclid at European Space Agency, 2019. Disponível em: <https://sci.esa.int/web/euclid/-/spacecraft>. Acessado em 29-Ago-2021.
- [152] Euclid vis instrument. Euclid at European Space Agency, 2019. Disponível em: <https://sci.esa.int/web/euclid/-/euclid-vis-instrument>. Acessado em 29-Ago-2021.
- [153] Euclid nisp instrument, 2019. [Disponível em: <https://sci.esa.int/web/euclid/-/euclid-nisp-instrument>. Acessado em 29-Ago-2021.
- [154] 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.
- [155] Hubble’s deepest view ever of the universe unveils earliest galaxies, 2004. [Disponível em: <https://hubblesite.org/contents/news-releases/2004/news-2004-07.html?Year=2004&page=3&filterUUID=8a87f02e-e18b-4126-8133-2576f4fdc5e2>. Acessado em 03-Set-2021.
- [156] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing

- Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from [tensorflow.org](https://www.tensorflow.org).
- [157] Norbert Wiener, Norbert Wiener, Cyberneticist Mathematician, Norbert Wiener, Norbert Wiener, and Cybernéticien Mathématicien. Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications, volume 113. MIT press Cambridge, MA, 1949.
- [158] Emilio Soria Olivas, Jos David Mart Guerrero, Marcelino Martinez-Sober, Jose Rafael Magdalena-Benedito, L Serrano, et al. Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques. IGI global, 2009.
- [159] Julio Borges. Deepstack: Ensembles for deep learning.
- [160] CR Bom, BMO Fraga, LO Dias, P Schubert, M Blanco Valentin, C Furlanetto, M Makler, K Teles, M Portes de Albuquerque, and R Benton Metcalf. Developing a victorious strategy to the second strong gravitational lensing data challenge. arXiv preprint arXiv:2203.09536, 2022.
- [161] P Umesh. Image processing in python. CSI Communications, 23, 2012.
- [162] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17:261–272, 2020.
- [163] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Pícus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser,

- Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [164] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.
- [165] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.