



Universidade Federal do Rio de Janeiro
Centro de Ciências Matemáticas e da Natureza
Observatório do Valongo

Identificação de lentes gravitacionais utilizando técnicas de aprendizagem profunda

Viviane de Mattos Matioli

Orientador: Dr Fabio Andre Machado Porto
Laboratório Nacional de Computação Científica

Coorientador: Rafael Silva Pereira
Laboratório Nacional de Computação Científica

Rio de Janeiro, RJ - Brasil

Agosto de 2020

Viviane de Mattos Matioli

Identificação de lentes gravitacionais utilizando técnicas de aprendizagem profunda

Trabalho de Conclusão de Curso submetido à Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Bacharel em Astronomia.

Universidade Federal do Rio de Janeiro
Centro de Ciências Matemáticas e da Natureza
Observatório do Valongo

Orientador(es): Dr Fabio Andre Machado Porto e Rafael Silva Pereira

Rio de Janeiro, RJ - Brasil
Agosto de 2020

CIP - Catalogação na Publicação

M433i Matioli, Viviane
Identificação de lentes gravitacionais utilizando técnicas de aprendizagem profunda / Viviane Matioli. -- Rio de Janeiro, 2020.
58 f.

Orientadora: Fabio Porto.

Coorientadora: Rafael Pereira.

Trabalho de conclusão de curso (graduação) - Universidade Federal do Rio de Janeiro, Observatório do Valongo, Bacharel em Astronomia, 2020.

1. Lentes gravitacionais. 2. Deep Learning. I. Porto, Fabio, orient. II. Pereira, Rafael, coorient. III. Título.



UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
CCMN - OBSERVATÓRIO DO VALONGO
DEPARTAMENTO DE ASTRONOMIA



PROJETO FINAL

RELATÓRIO DA COMISSÃO JULGADORA

ALUNA: Viviane de Mattos Matioli (DRE 115062733)

TÍTULO DO TRABALHO: “IDENTIFICAÇÃO DE LENTES GRAVITACIONAIS UTILIZANDO TÉCNICAS DE APRENDIZAGEM PROFUNDA”

DATA DA DEFESA: 24 de agosto de 2020 às 14:00 h

MEMBROS DA COMISSÃO JULGADORA:

Prof. Paulo Afrânio Antônio Lopes – Presidente - (OV/UFRJ)

Dr. Fabio André Machado Porto – Orientador - (LNCC)

Dr. Ricardo Ogando – (ON/MCTI)

CANDIDATA:

Viviane de Mattos Matioli

Rio de Janeiro, 24 de agosto de 2020.

Prof. Wagner L. F. Marcolino
Coord. de Grad. do Curso de Astronomia

Resumo

Estima-se que grandes levantamentos planejados para o futuro próximo devam observar uma quantidade de lentes gravitacionais fortes algumas ordens de magnitude maior que o volume conhecido atualmente, em particular na escala de galáxias. Tradicionalmente, o processo de identificação destes tipos de lentes envolve inspeção visual de imagens, o que não será viável neste cenário devido a grande quantidade de dados totais produzidos, dentre os quais lentes fortes são fenômenos relativamente raros. Assim será necessário o desenvolvimento de métodos automatizados mais eficientes para esta tarefa. Neste trabalho buscamos desenvolver um método para auxiliar no processo de identificação destas lentes, utilizando técnicas de aprendizagem profunda, mais especificamente redes neurais siamesas. O método é treinado e avaliado tanto em imagens reais quanto simuladas. Resultados obtidos em alguns cenários indicam a possibilidade de sua utilização como uma etapa inicial para filtragem de candidatos a lentes, eliminando metade dos candidatos falsos, enquanto classifica a maior parte das lentes verdadeiras de maneira correta.

Palavras-chave: Lentes gravitacionais fortes. Redes neurais artificiais. Aprendizagem profunda.

Abstract

It is estimated that large surveys planned for the near future should observe an amount of strong gravitational lenses some orders of magnitude greater than the volume currently known, particularly galaxy-scale lenses. Traditionally, the identification process of these types of lenses involves visual inspection of images, which would not be feasible in this scenario due to the large amount of data produced, among which strong lenses are relatively rare phenomena. Thus, the development of more efficient, automated methods for this task will be necessary. In this work we seek to develop a method to assist in the identification process of these lenses, using deep learning techniques, more specifically Siamese neural networks. The method is trained and evaluated in both real and simulated images. Results obtained in some scenarios indicate the possibility of its use as an initial step for filtering lens candidates, eliminating half of the false candidates, while correctly classifying most lens examples.

Keywords: Strong gravitational lensing. Artificial Neural Networks. Deep learning.

Lista de figuras

Figura 1 – Configuração de uma lente fina. D_L , D_S e D_{LS} são distâncias de diâmetro angular entre observador e lente (defletor), observador e fonte e entre lente e fonte, respectivamente.	15
Figura 2 – Algumas configurações possíveis para um sistema de lente elíptica e fonte compacta. (a) Ilustração do plano da fonte indicando suas diferentes posições em diferentes cores. Linhas representam cóusticas. (b) Imagens correspondentes formadas no plano da lente. Linhas representam curvas críticas.	17
Figura 3 – Relação entre principais elementos de um método de aprendizado de máquina	20
Figura 4 – Modelo de um perceptron.	22
Figura 5 – Dois exemplos de redes neurais <i>feedforward</i> tradicionais.	23
Figura 6 – Exemplos de funções de ativação comumente utilizadas em redes neurais artificiais. ReLU e <i>Leaky</i> ReLU, muito utilizadas em camadas intermediárias, e as funções sigmoide e tangente hiperbólica, também utilizadas em camadas de saída em problemas de classificação binária.	23
Figura 7 – Matriz de confusão para classificação binária. A diagonal principal corresponde ao número de exemplos classificados corretamente (verdadeiros positivos <i>VP</i> e verdadeiros negativos <i>VN</i>), enquanto a diagonal secundária contém os exemplos classificados de maneira incorreta (falsos negativos <i>FN</i> e falsos positivos <i>FP</i>)	26
Figura 8 – Visualização de um exemplo de operação de convolução em uma entrada contendo um canal. Uma janela contendo o <i>kernel</i> é deslocada sobre toda a imagem de entrada, as duas matrizes são multiplicadas elemento a elemento e todos os termos somados. Uma função de ativação é aplicada sobre os resultados, produzindo a o <i>feature map</i> de saída da camada.	28
Figura 9 – Exemplo de operação <i>max pooling</i> com <i>kernel</i> 2x2 e <i>stride</i> 2. Um tensor de entrada de dimensões 4x4x6 contendo 6 <i>feature maps</i> é reduzido espacialmente, sem ter sua profundidade modificada.	28
Figura 10 – Exemplo de CNN utilizada para identificar imagens de dígitos manuscritos. A rede recebe as imagens como entrada, e sua saída consiste nas probabilidades da imagem pertencer a cada uma das dez classes, correspondentes aos dígitos de 0 a 9.	29
Figura 11 – Exemplo de aumento de complexidade em características identificadas pela rede em um problema de reconhecimento de faces	30

Figura 12 – Exemplo de arquitetura de uma rede neural siamesa. Duas imagens \mathbf{x}_i e \mathbf{x}_j são recebidas como entrada e processadas pelas sub-redes idênticas, resultando nos respectivos vetores de <i>feature</i> \mathbf{h}_i e \mathbf{h}_j . A distância entre eles é calculada em seguida.	30
Figura 13 – Funcionamento do algoritmo KNN em um hiperespaço de duas dimensões	31
Figura 14 – Imagens reais	34
Figura 15 – Imagens simuladas	34
Figura 16 – Modelo da rede siamesa utilizada na segunda etapa. Para a primeira etapa, utilizamos a mesma arquitetura, porém com apenas três canais para cada imagem.	35
Figura 17 – Fluxograma - Etapa 1	37
Figura 18 – Fluxograma - Etapa 2. Fluxograma indicando diferentes conjuntos utilizados para teste na segunda etapa e suas fontes: Simulação do desafio de <i>BolognaLens Factory</i> , lentes galáxia-quasar com pré-processamento (CASTLES), lentes galáxia-quasar sem pré-processamento (HLA), lentes galáxia-galáxia de diferentes levantamentos (<i>Master Lens Database</i>), além de não lentes reais (HLA)	38
Figura 19 – Na sequência da esquerda para direita, a primeira e terceira colunas apresentam imagens originais anteriores à limpeza, das fontes CASTLES e <i>Master Lens Database</i> , respectivamente. Os resultados após o pré-processamento são vistos na segunda e quarta colunas.	39
Figura 20 – Matriz de confusão - Exemplo 1 - Lentes galáxia-quasar	40
Figura 21 – Matriz de confusão - Exemplo 2 - Lentes galáxia-quasar	40
Figura 22 – Distribuição de probabilidades - Exemplo 1 - Lentes galáxia-quasar	41
Figura 23 – Distribuição de probabilidades - Exemplo 2 - Lentes galáxia-quasar	41
Figura 24 – Matriz de confusão - Exemplo 1 - Lentes galáxia-galáxia	42
Figura 25 – Matriz de confusão - Exemplo 2 - Lentes galáxia-galáxia	43
Figura 26 – Distribuição de probabilidades - Exemplo 1 - Lentes galáxia-galáxia	43
Figura 27 – Distribuição de probabilidades - Exemplo 2 - Lentes galáxia-galáxia	44
Figura 28 – Matriz de confusão - Etapa 2 - Lentes galáxia-galáxia simuladas	45
Figura 29 – Distribuição de probabilidade -Etapa 2 - Lentes galáxia-galáxia simuladas	45
Figura 30 – Matriz de confusão - Etapa 2 - Lentes reais galáxia-quasar	46
Figura 31 – Distribuição de probabilidade - Etapa 2 - Lentes reais galáxia-quasar	47
Figura 32 – Matriz de confusão - Etapa 2 - Lentes reais galáxia-quasar sem pré-processamento CASTLES	48
Figura 33 – Distribuição de probabilidades - Etapa 2 - Lentes reais galáxia-quasar sem pré-processamento CASTLES	48

Figura 34 – Lentes pertencentes ao levantamento CASTLES com pré-processamento realizado pelos autores originais (esquerda) ao lado de suas imagens antes deste pré-processamento (direita)	49
Figura 35 – Matriz de confusão - Etapa 2 - Lentes reais galáxia-galáxia	50
Figura 36 – Distribuição de probabilidade - Etapa 3 - Lentes reais galáxia-galáxia	50
Figura 37 – Componentes do PCA de imagens de treinamento e teste	52
Figura 38 – Componentes do PCA incluindo imagens de lentes do levantamento CASTLES com pré-processamento	52

Lista de tabelas

Tabela 1 – Métricas - Etapa 1 - lentes galáxia-quasar	42
Tabela 2 – Métricas - Etapa 1 - Lentes galáxia-galáxia	44
Tabela 3 – Métricas - Etapa 2 - Lentes galáxia-galáxia simuladas	46
Tabela 4 – Métricas - Etapa 2 - Lentes reais galáxia-quasar	47
Tabela 5 – Métricas - Etapa 2 - Lentes reais galáxia-quasar sem pré-processamento CASTLES	48
Tabela 6 – Métricas - Etapa 3 - Lentes reais galáxia-galáxia	51

Sumário

1	Introdução	12
1.1	Organização do trabalho	13
2	Fundamentação teórica	14
2.1	Lentes gravitacionais	14
2.1.1	Identificação de lentes fortes	18
2.2	Inteligência artificial	19
2.2.1	Aprendizado de máquina	19
2.2.1.1	Classificação	20
2.2.2	Redes Neurais Artificiais	21
2.2.2.1	Perceptron	21
2.2.2.2	Redes neurais densas	22
2.2.2.3	Treinamento	24
2.2.2.4	Avaliação	25
2.2.3	Reconhecimento de imagens	26
2.2.3.1	Camadas convolucionais e <i>pooling</i>	26
2.2.3.2	Redes neurais convolucionais	29
2.2.3.3	Redes neurais siamesas	30
3	Metodologia	33
3.1	Dados	33
3.2	Desenvolvimento	34
3.2.1	Pré-processamento	36
3.2.2	Primeira etapa - Treinando com imagens reais	36
3.2.3	Segunda etapa - Treinando com imagens simuladas	37
4	Resultados	39
4.1	Pré-processamento	39
4.2	Primeira etapa - Treinando com imagens reais	40
4.2.1	Lentes galáxia-quasar	40
4.2.2	Lentes galáxia-galáxia	42
4.3	Segunda etapa - Treinando com imagens simuladas	44
4.3.1	Avaliação em imagens simuladas	45
4.3.2	Avaliação em imagens reais	46
4.3.2.1	Lentes galáxia-quasar	46
4.3.2.2	Lentes galáxia-galáxia	50
4.3.2.3	Visualizando o problema através da análise de componentes principais (PCA)	51
5	Conclusão	53

Referências 54

1 Introdução

Lentes gravitacionais são fenômenos observados quando a luz de uma fonte distante tem sua trajetória desviada devido a curvatura do espaço-tempo causada pela presença de outro corpo massivo entre a fonte e o observador. Lentes fortes são casos em que os dois objetos estão suficientemente alinhados de forma a levar à formação de múltiplas imagens associadas à mesma fonte. Nestes casos normalmente temos uma galáxia ou quasar como fonte, enquanto o objeto defletor se trata de outra galáxia ou aglomerado.

Sistemas de lente forte são ferramentas extremamente úteis, com diversas aplicações astrofísicas e cosmológicas. Podem ser utilizadas para estimar a constante de *Hubble* e outros parâmetros cosmológicos (Refsdal, 1964; Suyu, 2013; Treu, 2016), determinar a massa de objetos agindo como defletores e analisar a distribuição de matéria escura em galáxias e aglomerados (Kochanek, 1991; Tessore, 2016; Vegetti, 2009; Dye, 2005), além de possibilitar a observação de fontes em altos *redshifts* que não seriam detectadas sem seu efeito de magnificação (Bellagamba, 2016; Shu, 2016; Marshall, 2007; Deane, 2013).

Apesar de sua importância, atualmente são conhecidos menos de 1000 exemplares de sistemas de lentes fortes e seu processo de identificação tradicional envolve inspeção visual de um grande volume de imagens selecionadas de acordo com parâmetros como luminosidade ou cor. Porém o volume crescente de dados astronômicos sendo produzidos torna evidente a necessidade de métodos automatizados mais eficientes para identificação de lentes em grandes levantamentos. Estima-se que levantamentos recentes como *Dark Energy Survey* (DES) e *Kilo-Degree Survey* (KiDS) contenham $\sim 10^3$ exemplos de lentes fortes na escala de galáxias, enquanto projetos planejados para o futuro, como o *Legacy Survey of Space and Time* (LSST) e *Euclid* devem observar cerca de 10^5 lentes deste tipo (Collett, 2015). É esperado também que o projeto *Square Kilometer Array* (SKA) observe uma quantidade de lentes nesta mesma ordem de magnitude em frequências de rádio (McKean, 2015).

Neste cenário, diversos métodos têm sido desenvolvidos atualmente para automatizar a busca por lentes em levantamentos utilizando técnicas de aprendizado de máquina, como (Petrillo, 2017; Bom, 2017; Jacobs, 2017; Hartley, 2017; Ostrovski, 2017). Após treinados, métodos deste tipo são capazes de classificar um grande volume de imagens rapidamente de maneira eficiente, o que os torna mais adequados para esta tarefa.

Nosso objetivo neste trabalho, seguindo esta linha de pesquisa, é desenvolver um método automatizado que possa auxiliar na tarefa de identificação de lentes gravitacionais utilizando técnicas de aprendizagem profunda. Para isto utilizamos mais especificamente redes neurais siamesas, desenvolvidas para classificar imagens de entrada nas classes lente

e não-lente.

O trabalho foi desenvolvido em duas etapas. Na primeira etapa treinamos e avaliamos o método com imagens reais. Como esperado, o modelo treinado desta forma não apresenta resultados satisfatórios, devido à pouca quantidade e heterogeneidade de dados disponíveis de lentes conhecidas atualmente. Na segunda etapa utilizamos um conjunto de imagens simuladas para treinamento, obtendo resultados significativamente melhores ao aplicar este modelo tanto em simulações quanto em imagens reais.

1.1 Organização do trabalho

O capítulo 2 é dedicado a apresentar os fundamentos básicos da teoria de lentes gravitacionais e aprendizado de máquina, em particular redes neurais artificiais. No capítulo 3 detalhamos a metodologia adotada no desenvolvimento do trabalho, descrevendo suas etapas. Os resultados obtidos em cada etapa são apresentados no capítulo 4. Concluimos o trabalho com as considerações finais no capítulo 5.

2 Fundamentação teórica

Neste capítulo são discutidos alguns dos principais conceitos importantes para a compreensão do problema tratado no trabalho. Primeiramente apresentamos os fundamentos da teoria de lentes gravitacionais (2.1), em particular lentes fortes, foco principal do trabalho, e sua identificação (2.1.1). Posteriormente abordamos a área de inteligência artificial e aprendizado de máquina (2.2.1), redes neurais (2.2.2) e sua utilização para reconhecimento de imagens (2.2.3).

2.1 Lentes gravitacionais

Com a formulação da teoria da relatividade geral de Einstein em 1915, o conceito de gravidade passou a ser entendido de forma distinta, e assim também o comportamento da luz na presença de campos gravitacionais. A partir dela tornou-se possível descrever de forma quantitativa como a presença de um objeto massivo causa o desvio de luz se propagando em sua vizinhança, devido à curvatura no espaço-tempo causada por sua concentração de massa.

De acordo com a relatividade geral um raio de luz passando próximo a um corpo esférico de massa M a uma distância mínima ξ é desviado por um ângulo

$$\hat{\alpha} = \frac{4GM}{c^2\xi} = \frac{2R_s}{\xi} \quad (2.1)$$

onde G representa a constante gravitacional, c a velocidade da luz, e R_s o raio de Schwarzschild $R_s = \frac{2GM}{c^2}$.

Um dos primeiros experimentos desenvolvidos para testar a teoria da relatividade geral envolvia justamente este efeito de desvio da luz por um objeto massivo, neste caso, o Sol. O experimento foi realizado em 1919 durante um eclipse solar, quando seria possível observar estrelas projetadas próximas a superfície solar no céu, medindo suas posições. Assim verificou-se que a diferença nas posições observadas correspondia ao desvio previsto $\hat{\alpha}$, duas vezes maior que a previsão realizada utilizando física newtoniana, o que contribuiu para fortalecer a teoria de Einstein.

A figura 1 ilustra este efeito, apresentando uma configuração geométrica típica de uma lente gravitacional utilizando a aproximação de lente fina, em que as trajetórias da luz entre fonte, defletor e observador são aproximadas por retas. Esta aproximação é adequada quando a distribuição de massa do objeto defletor é fina comparada à extensão de todo o sistema da lente, como para casos em que os defletores são galáxias ou aglomerados, não

sendo apropriada para lentes extensas como estruturas em grande escala no Universo.

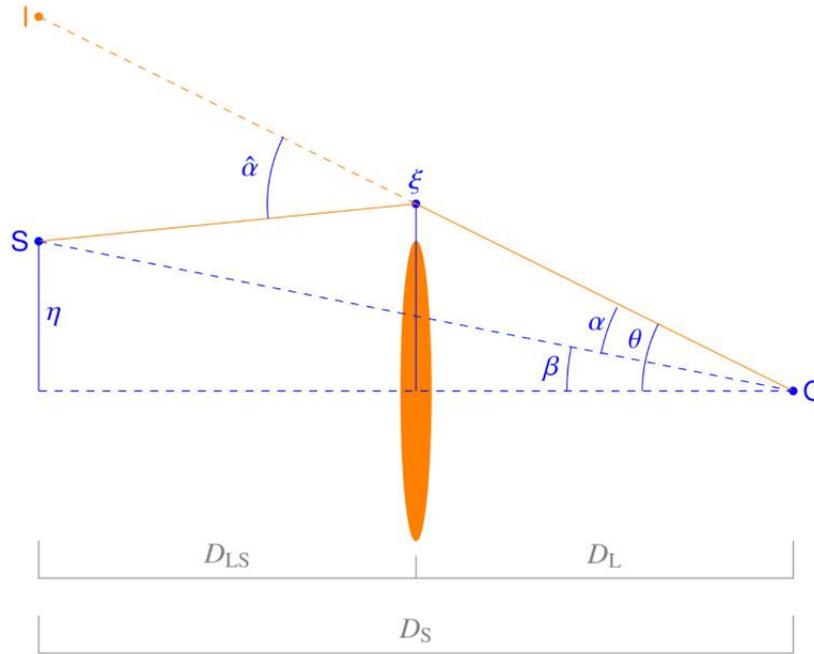


Figura 1 – Configuração de uma lente fina. D_L , D_S e D_{LS} são distâncias de diâmetro angular entre observador e lente (defletor), observador e fonte e entre lente e fonte, respectivamente.

Fonte: (Bartelmann, 2016)

A chamada equação da lente relaciona os ângulos presentes na imagem, correspondentes à posição verdadeira da fonte (β), sua posição observada no céu (θ) e o ângulo de deflexão $\hat{\alpha}$

$$\beta = \theta - \frac{D_{ls}}{D_s} \hat{\alpha} = \theta - \alpha \quad (2.2)$$

onde α é o chamado ângulo de deflexão reduzido.

Uma outra quantidade importante relacionada a problemas de lentes gravitacionais é o potencial de lenteamento ou deflexão $\psi(\theta) = \frac{2}{c^2} \frac{D_{LS}}{D_L D_S} \int \Phi(D_L \theta, z) dz$, onde Φ é o potencial gravitacional Newtoniano. A maior parte das propriedades de uma lente estão relacionadas a ele, e o ângulo de deflexão reduzido pode ser expresso como $\alpha = \nabla_{\theta} \psi$.

Lentes gravitacionais fortes correspondem ao caso em que a equação da lente possui múltiplas soluções θ , onde são formadas assim múltiplas imagens da fonte. Enquanto em lentes gravitacionais fracas é observada apenas uma imagem no plano da lente correspondente à fonte em β . Em ambos os casos as imagens formadas, associadas a uma fonte extensa, podem também sofrer distorções em sua forma e alteração no tamanho. Para uma fonte pequena comparada a escala de variação do ângulo de desvio, estes efeitos podem

ser descritos pela matriz jacobiana

$$A_{ij} = \frac{\partial \beta_i}{\partial \theta_j} = \delta_{ij} - \frac{\partial^2 \psi}{\partial \theta_i \partial \theta_j} = \begin{pmatrix} 1 - k - \gamma_1 & -\gamma_2 \\ -\gamma_2 & 1 - k + \gamma_1 \end{pmatrix} \quad (2.3)$$

onde k é a chamada convergência, relacionada ao potencial de lenteamento pela equação de Poisson ($\nabla^2 \psi = 2k$), e $\gamma_1 = \frac{1}{2}(\psi_{11} - \psi_{22})$ e $\gamma_2 = \psi_{12}$ são as componentes do chamado cisalhamento (*shear*) $\gamma^2 = \gamma_1^2 + \gamma_2^2$. (Utilizando a notação abreviada $\psi_{ij} \equiv \frac{\partial^2}{\partial \theta_i \partial \theta_j}$)

A matriz A (eq. 2.3) mapeia pequenas distâncias $\delta\theta$ em uma imagem em pequenas distâncias $\delta\beta$ de volta à fonte. Podemos verificar que na ausência de um objeto defletor e seu potencial de lenteamento, A é apenas uma matriz identidade, e a imagem observada é idêntica à fonte. Já na presença de uma lente as propriedades deste mapa são definidas pela curvatura do potencial ψ , expressado pela matriz de suas derivadas de segunda ordem, conhecida como matriz Hessiana.

O inverso de A é chamado tensor de magnificação, e seu determinante é o fator de magnificação

$$\mu = \det M = \frac{1}{\det A} = \frac{1}{(1 - k)^2 - \gamma^2} \quad (2.4)$$

A convergência k é responsável pela magnificação isotrópica da imagem formada, ou seja, o aumento de seu tamanho em relação a fonte, sem interferir na forma. Já o cisalhamento γ introduz a anisotropia ao mapeamento, modificando a forma da imagem. Uma fonte circular, por exemplo, sob os efeitos de k e γ pode possuir uma imagem elíptica no plano da lente, com semieixos proporcionais aos autovalores de M .

Os pontos no plano da lente onde $\det A = 0$, formam as chamadas curvas críticas. As curvas correspondentes no plano da fonte, obtidas através da equação da lente, são chamadas cóusticas. Estes dois tipos de curvas são úteis para análise qualitativa e entendimento do mapeamento envolvido entre os dois planos.

As cóusticas dividem o plano da fonte em regiões de diferente multiplicidade de imagens. Quando a posição de uma fonte varia, cruzando uma cóustica, duas imagens são formadas ou destruídas (ao cruzar em direção à fonte ou ao exterior, respectivamente). Este efeito pode ser observado na figura 2. Já as curvas críticas correspondentes dividem o plano da lente em regiões de paridade positiva (onde $\mu > 0$) e negativa (onde $\mu < 0$, imagens simétricas à fonte).

Outro efeito importante relacionado a lentes fortes é o atraso temporal entre as diferentes imagens formadas, ou *time delay*. Os intervalos de tempo necessário para que a luz de cada imagem alcance o observador podem ser diferentes, dependendo de sua

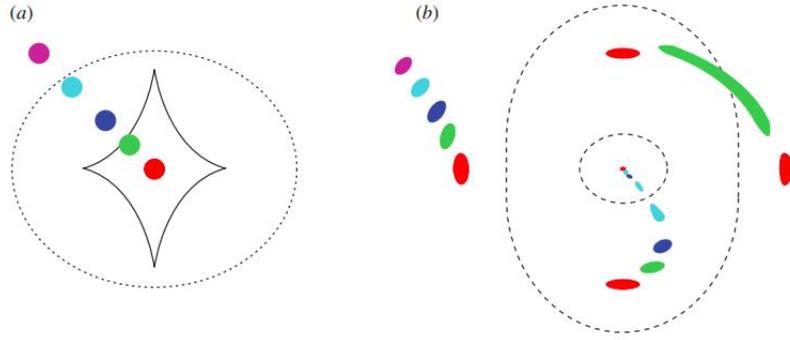


Figura 2 – Algumas configurações possíveis para um sistema de lente elíptica e fonte compacta. (a) Ilustração do plano da fonte indicando suas diferentes posições em diferentes cores. Linhas representam cústicas. (b) Imagens correspondentes formadas no plano da lente. Linhas representam curvas críticas.

Fonte: (Ellis, 2010)

trajetória e do potencial gravitacional da lente. O *time delay*, entre duas imagens localizadas nas posições θ_i e θ_j é expresso por

$$\Delta t_{ij} = \frac{D_{\Delta t}}{c} \Delta \tau_{ij} = \frac{D_{\Delta t}}{c} \left[\frac{(\theta_i - \beta)^2}{2} - \psi(\theta_i) - \frac{(\theta_j - \beta)^2}{2} + \psi(\theta_j) \right] \quad (2.5)$$

onde $D_{\Delta t} = (1 + z_l) \frac{D_l D_s}{D_{ls}}$ é a chamada distância de *time delay* e $\tau(\theta) = \frac{(\theta - \beta)^2}{2} - \psi(\theta)$ é o chamado potencial de Fermat.

Analisando a configuração e morfologia de um sistema de lente forte, é possível modelar a distribuição de massa da lente, determinando $\psi(\theta)$ e a posição da fonte β . Para sistemas com fontes variáveis, como AGNs (*active galactic nuclei*), é possível também medir este *delay* Δt_{ij} entre diferentes imagens e a partir da equação 2.5, inferir a distância $D_{\Delta t}$. Assim lentes fortes são ferramentas úteis para estimar a constante de Hubble, inversamente proporcional a $D_{\Delta t}$, e outros parâmetros cosmológicos (Treu, 2016; Suyu, 2013; Witt, 2000).

Como visto, o efeito causado por lentes é determinado pela variação espacial do potencial gravitacional associado ao objeto defletor e assim não dependem da natureza e estado da matéria que o compõe, podendo esta ser bariônica ou escura, estando ou não em equilíbrio. Por isto outra aplicação de lentes fortes é a determinação da massa total de galáxias e aglomerados e sua distribuição de matéria escura (Tessore, 2016; Vegetti, 2009; Dye, 2005). Além destas aplicações, estes tipos de lentes podem ainda funcionar como telescópios naturais, tornando possível observar objetos em alto *redshift* que não poderiam ser detectados sem a lente (Shu, 2016; Marshall, 2007).

Nesta seção foram abordados alguns conceitos básicos da teoria por trás de lentes gravitacionais, para uma abordagem muito mais profunda e abrangente, podem ser úteis

livros, palestras e *reviews* como (Schneider, 2006; Petters, 2012; Narayan, 1996; Bartelmann, 2010; Wambsganss, 1998)

2.1.1 Identificação de lentes fortes

O primeiro sistema de lente gravitacional forte foi descoberto em 1979, quando foram observados dois quasares de mesmo *redshift* e espectros idênticos, concluindo que se tratavam do mesmo objeto (Walsh, 1979). Desde então, a identificação de lentes, em sua maior parte, tem sido feita de maneira não automatizada. O método mais comumente utilizado é a inspeção visual de imagens de objetos selecionados a partir de levantamentos de acordo com parâmetros como luminosidade, *redshift* e classificação espectral, quando disponíveis. Esse tipo de abordagem foi adotado, por exemplo, em (Jackson, 2008; Faure et al., 2008; Sygnet et al., 2010; Pawase et al., 2014).

Um método similar, porém utilizando a estratégia de *crowdsourcing* para inspeção visual das imagens através de uma plataforma online foi desenvolvido pelo projeto SPACE WARPS (Marshall, 2016; More, 2016; Geach, 2015). Apesar de apresentar resultados interessantes, este tipo de método ainda não é adequado para utilização em quantidades muito maiores de dados futuros, além de ainda apresentar a dificuldade de avaliação de eficiência da performance humana neste tipo de tarefa.

Também foram desenvolvidos métodos para busca por lentes na escala de galáxias utilizando dados espectroscópicos, como os projetos SLACS (*Sloan lens ACS survey*) e BELLS (*BOSS Emission-line lens survey*) (Bolton et al., 2006; Brownstein, 2011; Smith, 2015; Willis, 2006). Estes métodos procuram por linhas correspondentes a estrelas em alto *redshift*, pertencentes à galáxia fonte, superpostas ao espectro de galáxias massivas de menor *redshift*, que agem como defletoras. Novamente, apesar de bons resultados obtidos, como espectroscopia é uma técnica de custo muito maior comparada a fotometria, demandando mais tempo, ainda haverá, pelo futuro próximo, um volume muito maior de dados fotométricos disponíveis para identificação de lentes.

Com grandes telescópios e levantamentos planejados para o futuro, como o *Legacy Survey of Space and Time* (LSST), *Euclid* e *Square Kilometer Array* (SKA) espera-se um aumento de algumas ordens de grandeza no número de lentes fortes observáveis, especialmente na escala de galáxias (Collett, 2015; McKean, 2015), sendo necessário o desenvolvimento de métodos automatizados mais eficientes para sua identificação entre todos os dados produzidos.

Assim, diversos métodos automatizados como (Seidel, 2007; Gavazzi, 2014; Joseph, 2014; Sonnenfeld et al., 2018) já foram desenvolvidos, normalmente para a identificação de arcos e anéis. Mais recentemente, tem-se desenvolvido também métodos adotando técnicas como aprendizado de máquina supervisionado, que após treinados, podem ser utilizados

para analisar grande volume de dados de maneira rápida e eficiente. Métodos deste tipo foram desenvolvidos, por exemplo, em (Bom, 2017; Petrillo, 2017; Jacobs, 2017; Ostrovski, 2017; Hartley, 2017) e no desafio para identificação de lentes fortes (Metcalf, 2019).

2.2 Inteligência artificial

O termo inteligência artificial (IA) foi utilizado pela primeira vez no ano de 1956, pelo cientista da computação John McCarthy, durante a primeira conferência dedicada ao tema. A definição do termo de acordo com seu criador era "a ciência e engenharia de produzir máquinas inteligentes". (McCarthy,)

Desde então, a área passou por períodos prósperos, chamados verões da IA, recebendo grandes investimentos, intercalados a períodos conhecidos como invernos da IA, onde interesse e desenvolvimento na área foram drasticamente reduzidos, devido a frustração de expectativas desproporcionais criadas a seu respeito.

Nas últimas duas décadas observamos o crescimento acelerado da produção de dados e sua diversificação, adentrando a era do *Big Data*, além de avanços possibilitando maior capacidade de armazenamento e poder de processamento, com a popularização de GPUs. Estes fatores contribuíram para a criação de um ambiente propício ao renascimento de interesse e desenvolvimento na área de inteligência artificial, que hoje é amplamente disseminada, com aplicações em diversas áreas da indústria e ciência.

Assim foram desenvolvidas também definições mais atualizadas para a área, em geral relacionadas a "sistemas que exibem comportamento inteligente, ao analisar seu ambiente e realizar ações - com algum grau de autonomia - para atingir objetivos específicos". (HLEG, 2019).

A subárea de IA relacionada ao desenvolvimento de técnicas que habilitam estes sistemas a aprender as melhores decisões a serem tomadas, sem uma programação explícita realizada por humanos de forma determinística, é chamada aprendizado de máquina (*machine learning*).

2.2.1 Aprendizado de máquina

Aprendizado de máquina (AM) pode ser definido como um conjunto de métodos capazes de detectar padrões em dados automaticamente e utilizá-los para realizar previsões sobre dados futuros. (Murphy, 2012)

De forma geral, um método de aprendizado de máquina pode ser caracterizado por três elementos principais: uma ou mais **tarefas**, um **modelo** e os **atributos** (*features*) selecionados a partir dos dados analisados (Flach, 2012). A relação entre estes elementos é ilustrada na figura 3. Um **modelo** é ajustado a partir da análise de atributos de **dados de**

treinamento, por um algoritmo de aprendizado, e pode então ser utilizado para realizar determinada **tarefa** sobre novos dados.

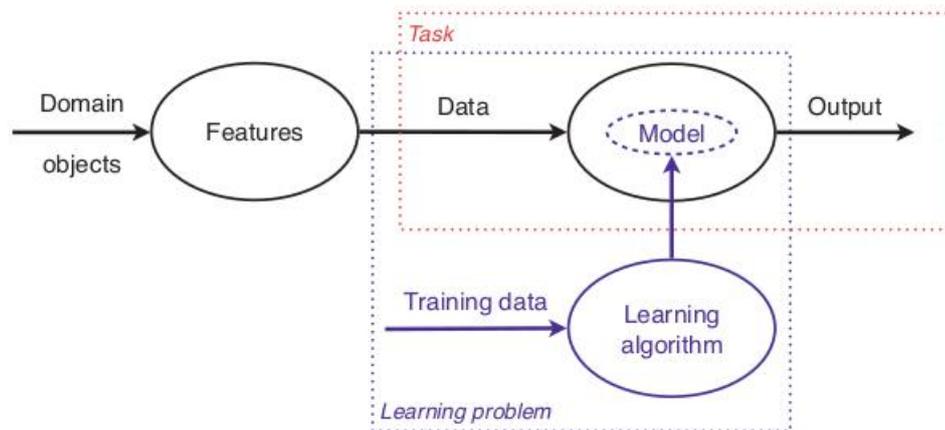


Figura 3 – Relação entre principais elementos de um método de aprendizado de máquina

Fonte: (Flach, 2012)

Métodos de aprendizado de máquina são normalmente divididos ainda em grupos quanto ao tipo de aprendizado: aprendizado supervisionado ou preditivo, não supervisionado ou descritivo, semi-supervisionado e aprendizado por reforço. (Murphy, 2012)

Os métodos mais amplamente utilizados atualmente pertencem ao grupo de aprendizado supervisionado, no qual consiste também o foco deste trabalho. Neste tipo de método, o conjunto de treinamento utilizado possui a forma $D = \{(\mathbf{x}^{(j)}, \mathbf{y}^{(j)})\}_{j=1}^N$, onde N é o número de exemplos de treinamento. Assumindo uma função desconhecida f que mapeia $\mathbf{y} = f(\mathbf{x})$, o objetivo do algoritmo de aprendizado pode ser descrito então como estimar uma aproximação \hat{f} a partir do conjunto de treinamento D . Esta aproximação $\hat{\mathbf{y}} = \hat{f}(\mathbf{x})$ representa um **modelo**, que então é utilizado para realizar previsões sobre novos dados de entrada \mathbf{x}

Estes exemplos de entrada $\mathbf{x}^{(j)}$ podem consistir em vetores contendo valores reais, ou estruturas mais complexas como imagens, textos, séries temporais como áudio e vídeo.

A forma da saída $\mathbf{y}^{(j)}$ também pode variar, de acordo com o objetivo do algoritmo e a tarefa que o modelo deve realizar. Os dois tipos de tarefas mais comuns são a **regressão**, na qual $\mathbf{y}^{(j)}$ consiste em um valor contínuo, e **classificação**, quando $\mathbf{y}^{(j)}$ é uma variável categórica correspondente a uma classe. (Murphy, 2012)

2.2.1.1 Classificação

De forma geral, em um problema de classificação, aprende-se uma distribuição $p_c = p(y | \mathbf{x}, \mathbf{D}, \mathbf{M})$ que indica a probabilidade do exemplo de entrada \mathbf{x} pertencer a cada

classe, dados o conjunto de treinamento \mathbf{D} e o modelo M . A saída é então dada por

$$\hat{y} = \hat{f}(\mathbf{x}) = \operatorname{argmax} [p(y = c | \mathbf{x}, \mathbf{D}, M)] \quad (2.6)$$

ou seja, a classe c de maior probabilidade.

Um caso especial de problema de classificação é a chamada classificação binária, em que a variável inferida \mathbf{y} pode assumir apenas dois valores, correspondentes a duas classes. Normalmente utiliza-se $y \in \{0, 1\}$, onde $y = 1$ representa a classe de interesse, ou positiva.

2.2.2 Redes Neurais Artificiais

As chamadas redes neurais artificiais (ANNs) estão entre os métodos de aprendizado de máquina supervisionado de maior popularidade atualmente. Nesta seção serão apresentados brevemente seus principais elementos, propriedades, treinamento e métodos de avaliação.

2.2.2.1 Perceptron

O algoritmo precursor das redes neurais artificiais modernas foi desenvolvido ainda em 1958, pelo psicólogo americano Frank Rosenblatt, inspirado por trabalhos anteriores e pelo modelo de interação entre neurônios no cérebro humano (Rosenblatt, 1958).

Conhecido como *Perceptron*, este algoritmo ilustrado na figura 4, consiste em um classificador linear binário. Ou seja, ele recebe como entrada um vetor \mathbf{x} contendo valores reais (atributos), e a partir de uma combinação linear de seus elementos produz uma saída binária única, correspondente à classe prevista \hat{y} .

Para realizar esta classificação, o *perceptron* utiliza um vetor de pesos reais \mathbf{w} , cujo elemento w_i está associado ao atributo x_i de entrada, indicando sua importância para o resultado, além de um parâmetro interno b chamado *bias*. Assim a decisão final é dada por

$$\hat{y} = \sigma(\mathbf{x} \cdot \mathbf{w} + b) \quad (2.7)$$

onde σ é a chamada função de ativação, que no caso do *perceptron* tradicional corresponde a uma função degrau. Os coeficientes \mathbf{w} e b são ajustados automaticamente pelo algoritmo durante seu treinamento. Este conjunto de parâmetros aprendidos corresponde ao modelo do classificador.

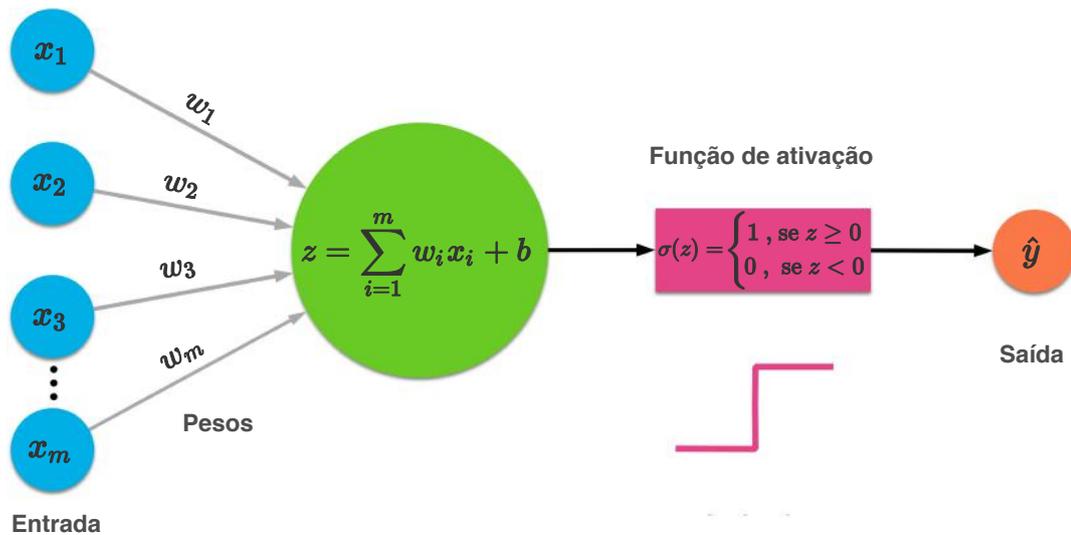


Figura 4 – Modelo de um perceptron.

Fonte: Autora.

2.2.2.2 Redes neurais densas

As redes neurais artificiais utilizadas atualmente são constituídas por várias unidades de neurônios ou nós similares ao *perceptron*, formando diferentes camadas, como ilustrado na figura 5. O tipo de camada ilustrada, em que cada neurônio está conectado a todas as unidades da camada seguinte são chamadas densas ou *fully connected*, assim as redes formadas exclusivamente por este tipo de camada são chamadas redes densas.

A primeira camada da rede é chamada de camada de entrada, e recebe os vetores de dados iniciais do problema. A rede pode possuir uma ou mais camadas intermediárias, chamadas de camadas escondidas, com diferentes números de nós. Uma rede com várias camadas intermediárias é chamada de profunda (*deep neural network*).

Já a última camada possui um nó associado a cada elemento do vetor de saída da rede. Um classificador binário, por exemplo, possui apenas um neurônio em sua camada de saída, que calcula a probabilidade do exemplo de entrada pertencer à classe positiva. Já em classificadores multiclasse a última camada possui número de neurônios igual ao número de classes, e sua saída consiste em um vetor de probabilidades do exemplo pertencer a cada classe.

Unidades de diferentes camadas podem possuir diferentes tipos de função de ativação, responsáveis por introduzir o caráter não linear à rede. Algumas das funções mais utilizadas são apresentadas na figura 6. A função ReLU (*Rectified Linear Units*) é amplamente utilizada em camadas intermediárias, pois é uma função simples, facilitando o processamento, mas capaz de produzir bons resultados. Já funções do tipo sigmoide podem ser utilizadas em camadas de saída para problemas de classificação.

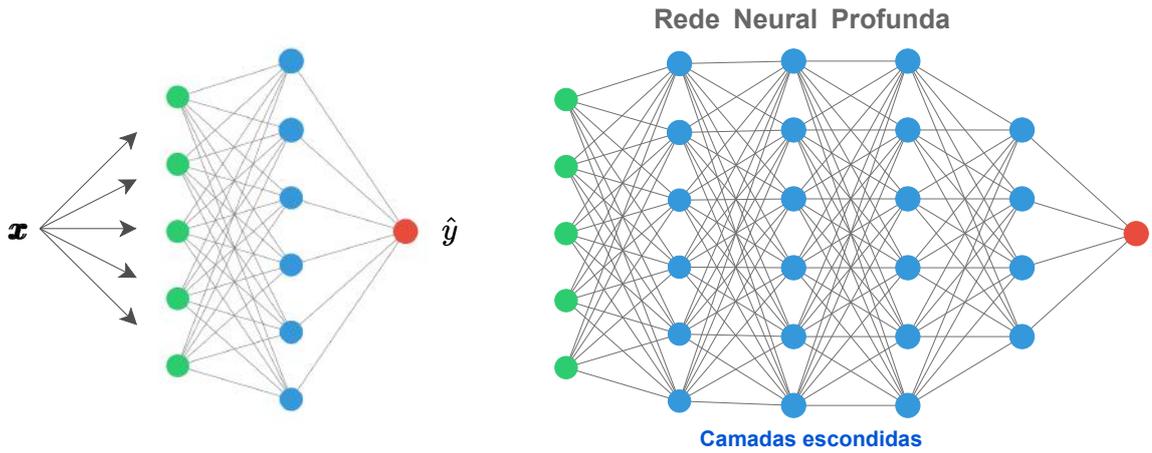


Figura 5 – Dois exemplos de redes neurais *feedforward* tradicionais.

Fonte: Autora.

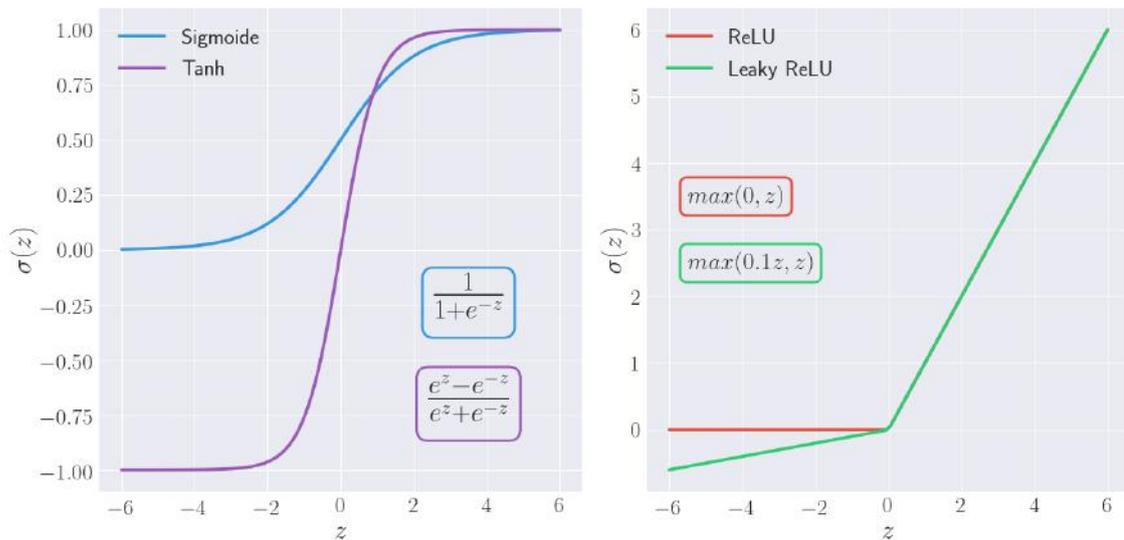


Figura 6 – Exemplos de funções de ativação comumente utilizadas em redes neurais artificiais. ReLU e *Leaky* ReLU, muito utilizadas em camadas intermediárias, e as funções sigmoide e tangente hiperbólica, também utilizadas em camadas de saída em problemas de classificação binária.

Fonte: Autora.

Desta forma, em uma rede neural densa, cada unidade possui uma função de ativação (unidades de uma mesma camada possuem mesma função de ativação) e parâmetro *bias* próprios, e cada conexão entre duas unidades i e j está associada a um peso w_{ij} . Assim um determinado neurônio j recebe como entrada as saídas de todas as unidades da camada anterior, chamadas ativações, e calcula sua saída a partir da regra vista no caso do *perceptron* $a_j = \sigma_j(\sum_{i=0}^n w_{ij}a_i + b_j)$. Esta ativação é em seguida passada como entrada para todas as unidades da camada seguinte. Este processo de cálculo da saída da rede a partir de um exemplo de entrada, através da propagação de ativações por todas as

camadas intermediárias é conhecido como *forward propagation*.

2.2.2.3 Treinamento

Para que uma rede neural artificial possa ser utilizada para realizar a tarefa pretendida sobre novos dados, esta precisa primeiramente ser treinada sobre o conjunto de treinamento D com o objetivo de determinar os parâmetros adequados para o problema.

Dois conceitos importantes envolvidos no processo de treinamento são a função de custo e método do gradiente descendente. Uma função de custo é utilizada para quantificar a discrepância entre os valores previstos pela rede $\hat{\mathbf{y}}^{(j)}$ e os valores reais $\mathbf{y}^{(j)}$. Uma função de custo muito utilizada para problemas de classificação binária é a *binary cross entropy*:

$$C = -\frac{1}{N} \sum_{j=1}^N y^{(j)} \cdot \log(\hat{y}^{(j)}) + (1 - y^{(j)}) \cdot \log(1 - \hat{y}^{(j)}) \quad (2.8)$$

Durante o treinamento o objetivo é minimizar esta função, ou seja, reduzir o erro nas previsões da rede. Para isto são utilizados métodos de otimização iterativos que visam encontrar o conjunto de parâmetros (pesos e *bias*) correspondentes a um local de mínimo da função de custo. Estes tipos de otimizadores normalmente se baseiam no gradiente da função de custo (*gradient descent*), possuindo algumas variações. Tomando como exemplo o método do gradiente descendente estocástico, todos os parâmetros da rede são atualizados após a passagem de cada exemplo de treinamento. Esta atualização é feita no sentido oposto ao gradiente da função. Para cada peso w_i da rede, temos:

$$w_i \leftarrow w_i + \Delta w_i \quad (2.9)$$

onde

$$\Delta w_i = -\alpha \frac{\partial C}{\partial w_i} \quad (2.10)$$

e α é o hiperparâmetro chamado *learning rate*, que determina a magnitude do passo de atualização dos pesos.

Como a saída da rede, e consequentemente a função de custo C , são funções dos parâmetros de todas as camadas anteriores, o processo de cálculo do gradiente da função de custo e atualização destes parâmetros faz uso extensivo da regra da cadeia, e é realizado por um algoritmo chamado *backpropagation*, explicado de forma mais abrangente em (Nielsen, 2015; Mitchell, 1997; LeCun et al., 2012).

Vistos os principais componentes do processo de treinamento, podemos resumi-lo de forma simplificada: Os pesos e *bias* da rede são inicializados de forma aleatória. Cada exemplo de treinamento passa pela rede resultando em uma saída $\hat{\mathbf{y}}$, sobre a qual é

calculada a função de custo (etapa conhecida como *forward step*). Em seguida os pesos e bias da rede são atualizados no sentido oposto ao gradiente da função de custo, com o objetivo de minimizá-la (*backward step*).

Este processo é realizado iterativamente para todos os exemplos de treinamento diversas vezes. De forma geral o treinamento é encerrado quando o cálculo da função de custo em diferentes iterações converge para um determinado valor mínimo. Como geralmente também é necessário um conjunto de treinamento volumoso para que a rede possa melhor aproximar a distribuição real, este pode ser um processo demorado.

O modelo final é determinado pelo conjunto de parâmetros ajustados automaticamente no treinamento e os chamados hiperparâmetros, relacionados à arquitetura da rede, como o número de camadas, número de neurônios em cada camada, e suas funções de ativação. A otimização desta arquitetura normalmente é realizada através da utilização de um conjunto de avaliação. Observando a performance da rede sobre este conjunto, ajustamos seus hiperparâmetros de forma a otimizar seus resultados.

Após o treinamento, são realizados testes, nos quais a rede otimizada é utilizada para classificar exemplos ainda não vistos, partes do chamado conjunto de teste. Seu desempenho em tais testes é avaliado de acordo com determinadas métricas, descritas brevemente na próxima seção.

2.2.2.4 Avaliação

Uma das principais ferramentas utilizadas para avaliação em tarefas de classificação é a chamada matriz de confusão, que facilita a visualização da performance da rede, apresentando as frequências das diferentes classificações. A configuração de uma matriz de confusão para um modelo de classificação binária é ilustrada na figura 7.

Métricas de avaliação úteis podem ser calculadas a partir dos valores apresentados na matriz de confusão. Algumas das métricas mais utilizadas são:

- Acurácia, que representa a proporção de classificações corretas dentre todas as realizadas pela rede ao processar os exemplos de teste

$$Acurácia = \frac{VP + VN}{VP + FP + VN + FN} \quad (2.11)$$

- Precisão, indicando qual proporção de classificações positivas realizadas pela rede estão corretas

$$Precisão = \frac{VP}{VP + FP} \quad (2.12)$$

		Classe prevista	
		P	N
Classe Real	P	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	N	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Figura 7 – Matriz de confusão para classificação binária. A diagonal principal corresponde ao número de exemplos classificados corretamente (verdadeiros positivos *VP* e verdadeiros negativos *VN*), enquanto a diagonal secundária contém os exemplos classificados de maneira incorreta (falsos negativos *FN* e falsos positivos *FP*)

Fonte: Autora.

- *Recall*, proporção de exemplos pertencentes a classe real positiva que são classificados corretamente

$$Recall = \frac{VP}{VP + FN} \quad (2.13)$$

- *F1-Score*, correspondente a média harmônica entre precisão e *recall*

$$F1 = 2 \times \frac{precisão \times recall}{precisão + recall} \quad (2.14)$$

2.2.3 Reconhecimento de imagens

Seres humanos possuem uma habilidade natural de interpretar dados não estruturados como textos, áudios e imagens, característica historicamente não compartilhada com máquinas. Porém, ao longo das últimas décadas, com o desenvolvimento de redes neurais profundas, a área de visão computacional tem observado grandes avanços, apresentando técnicas de reconhecimento de imagens com performance comparável a humanos. Redes neurais profundas para reconhecimento de imagens normalmente possuem diferentes tipos de camadas especializadas, além das camadas densas vistas nas seções anteriores.

2.2.3.1 Camadas convolucionais e *pooling*

Os dois principais tipos de camadas voltadas para processamento de imagens em redes neurais, e os únicos abordados e utilizados neste trabalho são as chamadas camadas

convolucionais e *max pooling*.

De acordo com seu nome, as camadas convolucionais realizam a operação de convolução entre as imagens e diferentes filtros, também chamados de *feature detectors* ou *kernels*, para reconhecimento de padrões.

No caso unidimensional contínuo, a operação de convolução entre duas funções f e g é definida como

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (2.15)$$

Porém ao trabalhar com imagens digitais, representadas por matrizes contendo valores reais correspondentes a cada pixel, necessitamos da versão discreta bidimensional:

$$(I * K)_{ij} = \sum_m \sum_n K_{m,n} \cdot I_{i-m,j-n} \quad (2.16)$$

Onde I e K são as matrizes que representam a imagem e o *kernel*, respectivamente. Generalizando ainda para casos em que as imagens possuem diferentes canais c , temos

$$(I * K)_{ij} = \sum_c \sum_m \sum_n K_{m,n,c} \cdot I_{i-m,j-n,c} \quad (2.17)$$

Os *kernels* normalmente possuem dimensões menores que a matriz da imagem de entrada, e a operação de convolução entre os dois pode ser visualizada como ilustrado na figura 8. Cada elemento ou pixel da imagem resultante da operação corresponde a uma combinação linear dos elementos de uma área da imagem original, onde os pesos são os elementos do *kernel*, sendo ajustados durante o treinamento. Para uma imagem com diferentes canais, esta operação é realizada para cada canal separadamente, e os resultados somados.

Após a operação de convolução uma função de ativação, normalmente ReLU, é aplicada a cada elemento $(I * K)_{ij}$, resultando no que chamamos de *feature map*, a saída da camada convolucional.

Este tipo de camada possui parâmetros como os chamados *stride* e *padding*, relacionados ao tamanho do passo no deslocamento do *kernel* sobre a imagem, e ao tamanho do *feature map* resultante da convolução, respectivamente.

As camadas *pooling* são comumente utilizadas após camadas convolucionais e realizam a redução espacial dos *feature maps*. Com isso elas reduzem o número de parâmetros na rede e o poder de processamento necessário, além de extrair características dominantes invariantes a pequenas translações e rotações.

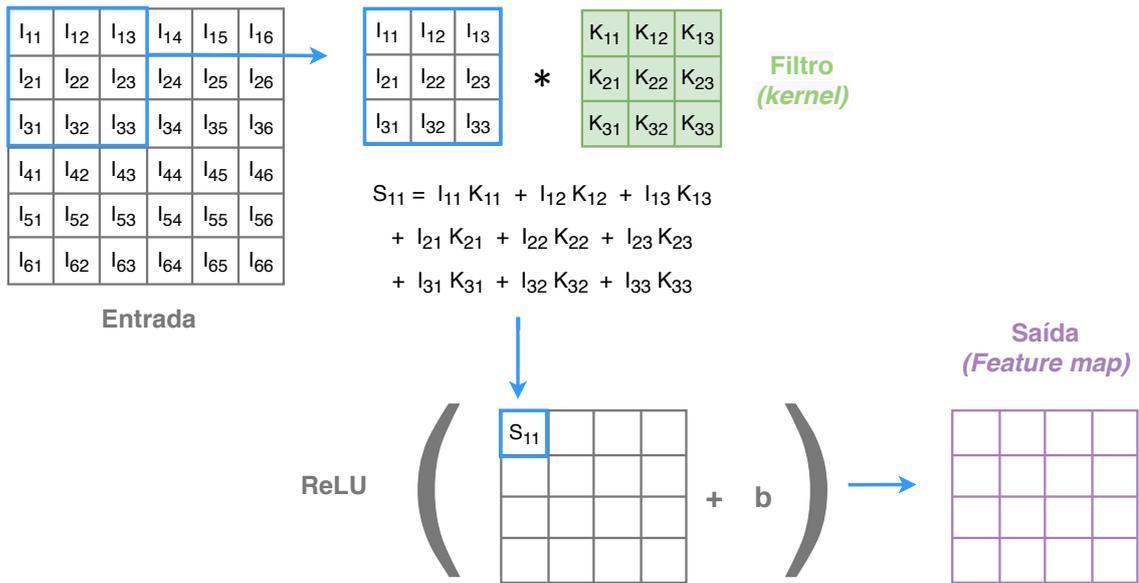


Figura 8 – Visualização de um exemplo de operação de convolução em uma entrada contendo um canal. Uma janela contendo o *kernel* é deslocada sobre toda a imagem de entrada, as duas matrizes são multiplicadas elemento a elemento e todos os termos somados. Uma função de ativação é aplicada sobre os resultados, produzindo a o *feature map* de saída da camada.

Fonte: Autora.

A camada *pooling* mais comumente utilizada é a chamada *max pooling*, ilustrada na figura 9. A operação é realizada com o deslocamento de um *kernel* sobre os *feature maps* de entrada, de forma similar ao que ocorre nas camadas de convolução, porém extraindo apenas os valores mais altos em seu interior. Este processo é realizado para cada canal de forma independente.

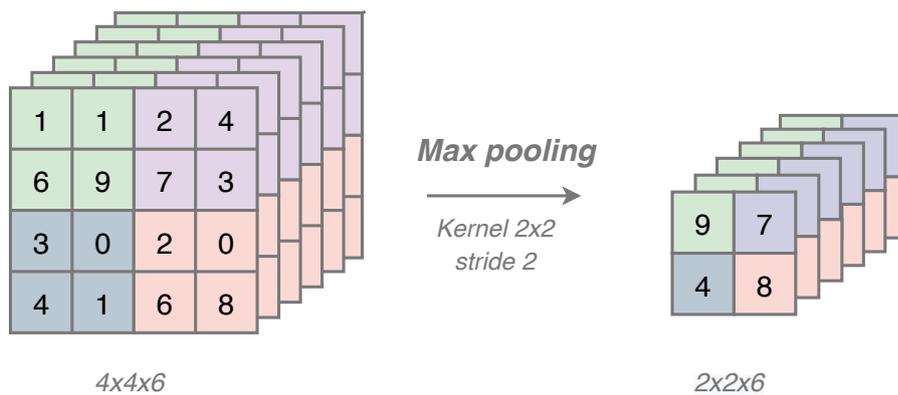


Figura 9 – Exemplo de operação *max pooling* com *kernel* 2x2 e *stride* 2. Um tensor de entrada de dimensões 4x4x6 contendo 6 *feature maps* é reduzido espacialmente, sem ter sua profundidade modificada.

Fonte: Autora.

A utilização destes dois tipos de camadas adicionais antes das camadas densas em uma rede visa principalmente facilitar o processamento de um grande volume de

imagens, reduzindo-as e identificando padrões e características essenciais para o processo de classificação.

2.2.3.2 Redes neurais convolucionais

Um tipo de rede neural amplamente utilizada para reconhecimento de imagens são as redes neurais convolucionais (CNNs). As CNNs são normalmente formadas por diferentes combinações dos tipos de camadas básicas abordados: convolucionais, *pooling* e densas.

Normalmente as imagens passam por camadas convolucionais e *pooling* na fase conhecida como extração de características (*feature extraction*) e são em seguida "achata-das" para o formato de um vetor coluna e passadas como entrada para camadas densas, na fase de classificação. Um exemplo de rede neural convolucional como este é ilustrado na figura 10, onde vemos os três tipos de camadas e alguns de seus parâmetros.

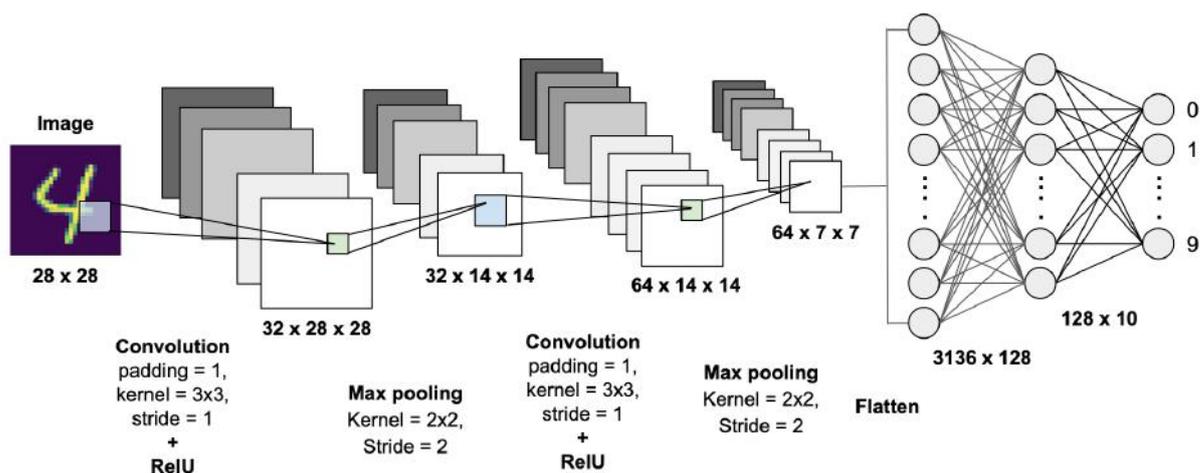


Figura 10 – Exemplo de CNN utilizada para identificar imagens de dígitos manuscritos. A rede recebe as imagens como entrada, e sua saída consiste nas probabilidades da imagem pertencer a cada uma das dez classes, correspondentes aos dígitos de 0 a 9.

Fonte: ([TowardsDataScience, 2019](#))

As primeiras camadas de uma CNN capturam características de mais baixo nível, estruturas como arestas e bordas em diversas orientações, enquanto camadas posteriores são capazes de identificar combinações cada vez mais complexas destas estruturas, como partes de objetos presentes nas imagens, até o objeto como um todo, como o exemplo apresentado na figura 11.

Alguns exemplos de CNNs populares são: a clássica LeNet-5, uma das primeiras CNNs a ser desenvolvida ([LeCun et al., 1998](#)), AlexNet, uma versão já bem mais profunda ([Krizhevsky, 2012](#)), VGG ([Simonyan, 2014](#)), ResNet ([He et al., 2016](#)), e Xception ([Chollet, 2017](#)).

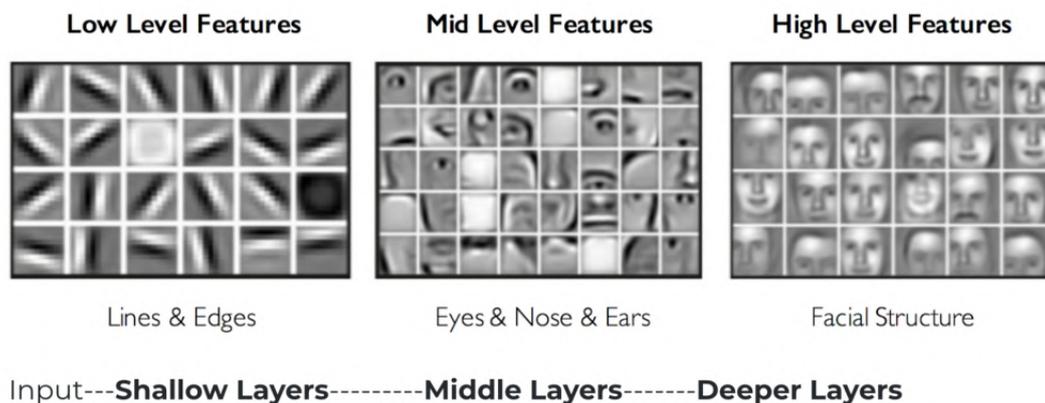


Figura 11 – Exemplo de aumento de complexidade em características identificadas pela rede em um problema de reconhecimento de faces

Fonte:(MIT..., 2018)

2.2.3.3 Redes neurais siamesas

Um outro tipo de rede neural artificial também utilizada para reconhecimento de imagens são as redes neurais siamesas (SNNs). Este tipo de rede pode ser formada pelos mesmos tipos de camadas vistos anteriormente, porém é estruturada de forma distinta, contendo duas sub-redes idênticas.

Uma SNN como a ilustrada na figura 12 recebe duas imagens de entrada, que são processadas pelas duas sub-redes para extração de *features* e reconhecimento de padrões. As saídas destas sub-redes são os chamados vetores de *feature* associados as respectivas imagens, ou seja, vetores contendo características relevantes identificadas pela rede.

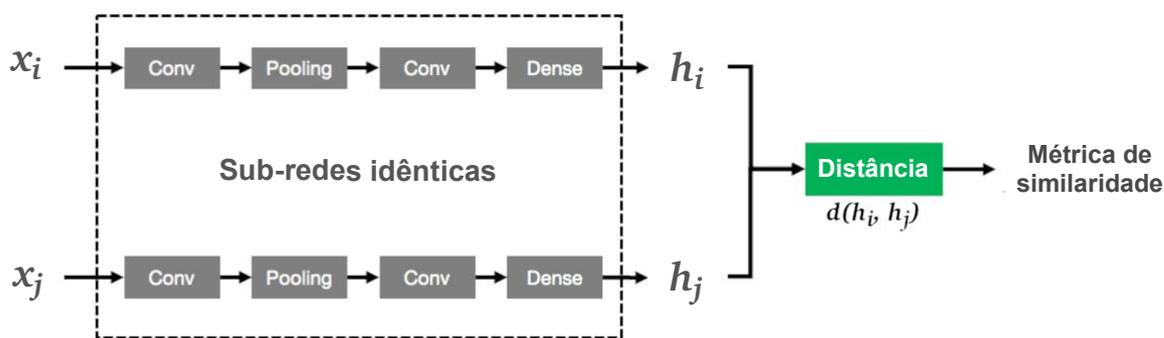


Figura 12 – Exemplo de arquitetura de uma rede neural siamesa. Duas imagens x_i e x_j são recebidas como entrada e processadas pelas sub-redes idênticas, resultando nos respectivos vetores de *feature* h_i e h_j . A distância entre eles é calculada em seguida.

Fonte: Adaptado de (Packt, 2019)

Em seguida estes vetores são comparados em uma camada especial unindo as duas sub-redes, responsável por calcular uma métrica de distância entre eles. A última camada,

que consiste apenas em um neurônio, geralmente com função de ativação sigmoide, recebe como entrada esta métrica e calcula a saída da rede, que corresponde à probabilidade de que as duas imagens de entrada pertençam a mesma classe.

Assim a ideia principal por trás da rede siamesa é encontrar uma função que projete exemplos de entrada em um hiperespaço onde vetores de *feature* de mesma classe estão mais próximos que vetores de classes distintas. Este processo é realizado automaticamente durante o treinamento com a minimização da função de custo. (Chopra, 2005)

Após o treinamento da rede, uma maneira eficiente de classificar novas imagens é através da utilização do algoritmo KNN (*K-Nearest Neighbors*) no hiperespaço resultante. As novas imagens são processadas pela rede siamesa e seus vetores de *feature* associados são passados como entrada para o KNN, que calcula a distância entre o novo vetor e os k vizinhos mais próximos correspondentes a vetores de *feature* de imagens utilizadas para treinamento. A classificação é então realizada considerando a distância e classe dos k vizinhos mais próximos (*distance weighted KNN*). Este processo é ilustrado na imagem 13.

De forma mais precisa, dado um vetor de *feature* h associado a uma imagem a ser classificada em uma das classes c , a saída do algoritmo é dada por

$$\hat{f}(h) = \underset{c}{\operatorname{argmax}} \left[\sum_{i=1}^k \frac{1}{d(h, n_i)} \delta(c, y_{n_i}) \right] \quad (2.18)$$

onde n_i corresponde a um vizinho, $d(h, n_i)$ a distância entre eles, e $\delta(c, y_{n_i}) = 1$ se $y_{n_i} = c$ e $\delta(c, y_{n_i}) = 0$ caso contrário. Ou seja, a saída é a classe que maximiza o valor do somatório.

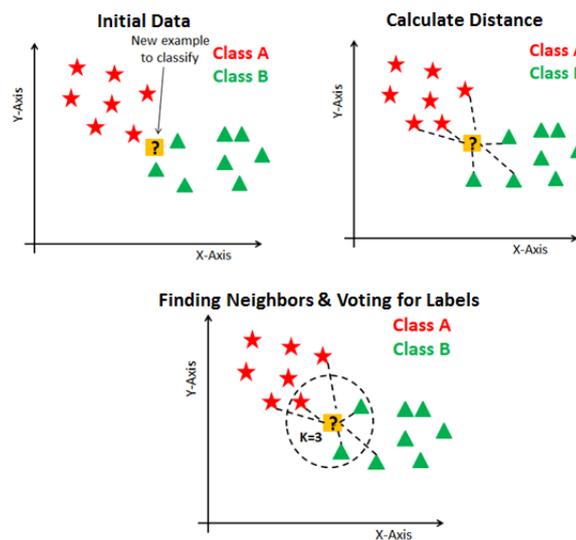


Figura 13 – Funcionamento do algoritmo KNN em um hiperespaço de duas dimensões

Fonte: (Datacamp, 2018)

Esta forma diferenciada de funcionamento das SNNs apresenta algumas vantagens sobre a utilização de CNNs tradicionais. Ao contrário destas, as redes siamesas não requerem grande volume de exemplos de treinamento para cada classe, e após treinadas podem ser utilizadas para classificação de imagens em novas classes, não vistas durante o treinamento, sem a necessidade de retreinar a rede, como ocorreria no caso das CNNs. Para isto basta adicionar ao menos uma imagem da nova classe aos exemplos utilizados pelo método para realizar a classificação (*One shot learning*).

Devido a estas vantagens, as redes siamesas são muito utilizadas em aplicações como reconhecimento de faces e detecção de fraudes em assinaturas. (Koch, 2015)

3 Metodologia

De acordo com a proposta do trabalho descrita anteriormente, buscamos um método inicial rápido e automatizado que possa auxiliar na identificação de lentes gravitacionais. Para isto as ferramentas escolhidas foram redes siamesas, juntamente com o algoritmo KNN, vistos em 2.2.3.3. O objetivo deste capítulo é explicar como estas ferramentas foram empregadas, a organização do desenvolvimento do trabalho em diferentes etapas e os dados utilizados em cada uma.

3.1 Dados

Para o desenvolvimento do trabalho foram utilizadas imagens reais de lentes gravitacionais, imagens reais de galáxias como exemplos de não-lentes, e imagens simuladas tanto de lentes quanto não-lentes. As figuras 14 e 15 apresentam exemplos de imagens destes conjuntos.

Imagens reais de lentes do tipo galáxia-quasar observadas no levantamento CASTLES, realizado com o Telescópio Espacial Hubble, foram retiradas diretamente de seu website (C.S. Kochanek, ; Falco, 2001) e já possuem um pré-processamento realizado pelos autores. Selecionando apenas as lentes que possuem imagens em todos os três filtros do levantamento (H, I e V), obtemos 55 sistemas.

Utilizamos também imagens reais de lentes do tipo galáxia-galáxia resultantes de diferentes levantamentos e reunidas no portal *Master Lens Database* (MASTER...), de onde as extraímos. Foram selecionados 67 objetos da classe A, ou seja, com a maior probabilidade de serem realmente lentes. Utilizamos apenas uma imagem para cada sistema, obtidas com telescópios e filtros diferentes.

Como exemplos de não-lentes reais utilizamos imagens *cutouts* de tamanho 10" de galáxias obtidas a partir do portal *Hubble Legacy Archive* (HLA) (Brad Whitmore,). Foram selecionadas galáxias que possuíssem imagens obtidas com o mesmo instrumento utilizado no levantamento CASTLES (WFPC2) e nos mesmos dois de seus três filtros, V (F555W) e I (F814W).

Por fim, como o número de imagens de lentes reais não é suficiente para nosso objetivo, utilizamos também imagens simuladas de lentes do tipo galáxia-galáxia e de não-lentes, parte do primeiro desafio de identificação de lentes fortes realizado no portal *Bologna Lens Factory* (Metcalf, 2018; Metcalf, 2019). Seu conjunto de imagens disponibilizado para treinamento contém lentes puramente simuladas, obtidas através do código GLAMER (Metcalf, 2014; Petkova, 2014) e lentes construídas a partir de imagens reais de galáxias

do levantamento *Kilo-Degree Survey* (KiDS) com a adição de novas fontes. Utilizamos aproximadamente 17000 objetos desta simulação, cada um destes possuindo imagens em quatro filtros, simulando as bandas u , g , r e i do levantamento KiDS.

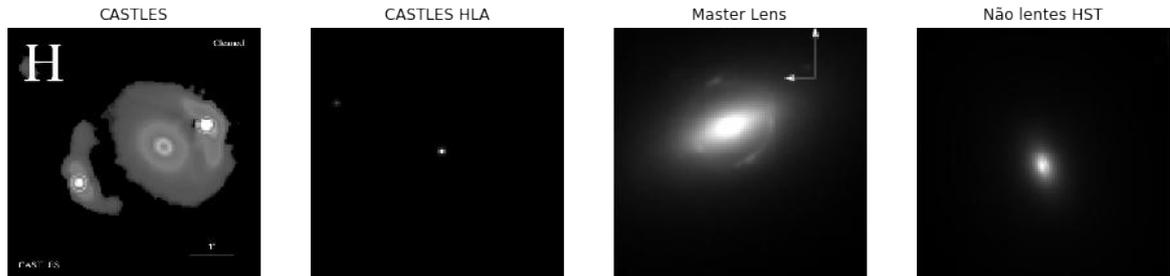


Figura 14 – Imagens reais

Fonte: Autora



Figura 15 – Imagens simuladas

Fonte: Autora.

3.2 Desenvolvimento

O modelo desenvolvido consiste em um classificador binário, que após treinado, classifica imagens de entrada nas classes lente e não-lente. Para desenvolvê-lo utilizamos a arquitetura de rede siamesa apresentada na figura 16, com função de ativação ReLU em todas as camadas intermediárias, função sigmoide na camada de saída, e função de perda *binary cross entropy*. A rede foi utilizada em conjunto com o algoritmo KNN.

O método foi treinado e avaliado utilizando imagens contendo lentes e imagens de galáxias como exemplos de não-lentes. Para reduzir o tempo de treinamento, utilizamos um conjunto de dez imagens âncoras para cada classe, selecionadas a partir do conjunto maior de treinamento. Como visto anteriormente, a cada iteração a rede siamesa recebe um par de imagens a serem comparadas, utilizando esta técnica nos certificamos de que uma das componentes deste par seja sempre uma imagem âncora, à qual é comparada a outra imagem de treinamento.

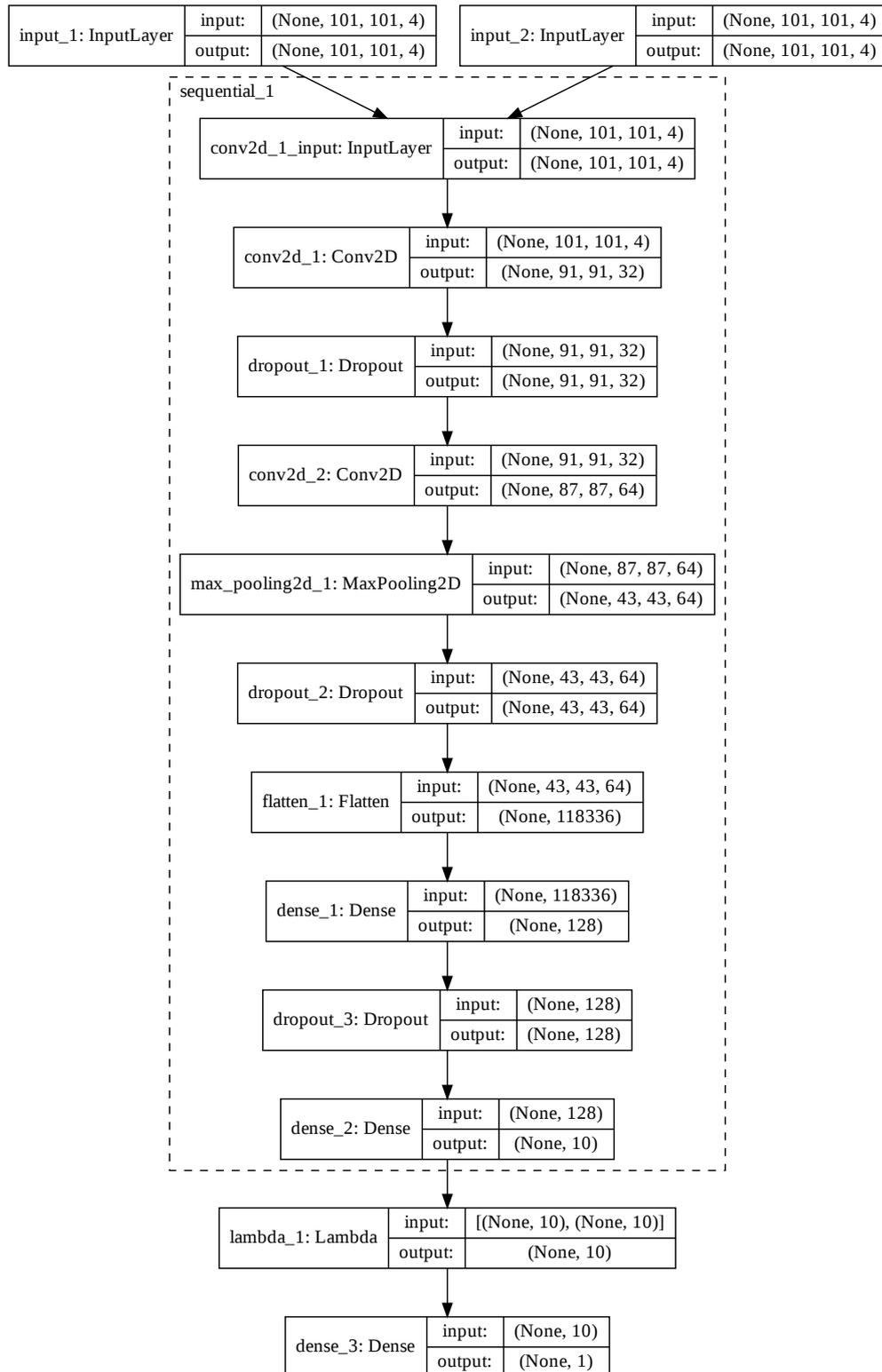


Figura 16 – Modelo da rede siamesa utilizada na segunda etapa. Para a primeira etapa, utilizamos a mesma arquitetura, porém com apenas três canais para cada imagem.

Fonte: Autora.

Após o treinamento, a classificação de novas imagens é realizada de forma análoga. O algoritmo KNN é utilizado sobre a projeção das imagens no hiperespaço de *features* obtido pela rede, classificando novas imagens de acordo com sua distância às âncoras.

O desenvolvimento do trabalho foi dividido em duas etapas, nas quais o método é treinado e avaliado utilizando diferentes conjuntos de imagens, descritas nas próximas subseções.

3.2.1 Pré-processamento

Anteriormente às duas etapas principais, foi necessário realizar um pré-processamento das imagens utilizadas para treinamento e avaliação da rede.

Algumas imagens de lentes reais possuem textos e símbolos, normalmente na cor branca, como pode ser visto na imagem 14 apresentada na seção sobre os dados. Para retirá-los foram utilizadas máscaras preenchendo *pixels* com valores acima de um *threshold* de acordo com *pixels* em sua vizinhança na imagem original.

Todas as imagens foram também reduzidas às mesmas dimensões, e organizadas de forma que observações de um objeto em diferentes filtros correspondem a diferentes canais. Por fim, todas foram também normalizadas para que o valor de seus *pixels* se encontre dentro do intervalo de 0 a 1, facilitando o processamento pela rede.

3.2.2 Primeira etapa - Treinando com imagens reais

Em uma primeira etapa nosso objetivo é utilizar esta abordagem de aprendizado de máquina para identificação de lentes gravitacionais no contexto atual de dados e imagens reais conhecidos e disponíveis hoje para a comunidade.

Assim utilizamos apenas imagens reais de lentes fortes já identificadas para treinar e avaliar o método. Realizamos esta etapa para sistemas de lentes do tipo galáxia-quasar e galáxia-galáxia separadamente, como ilustrado no fluxograma da figura 17. Em ambos os casos 70% das imagens do conjunto é utilizada para treinamento e 30% para teste.

A avaliação da performance do modelo é realizada através das métricas abordadas na seção 2.2.2.4. Refletindo a pouca quantidade e heterogeneidade dos dados e imagens de lentes reais disponíveis atualmente, não esperamos obter resultados satisfatórios nesta etapa.

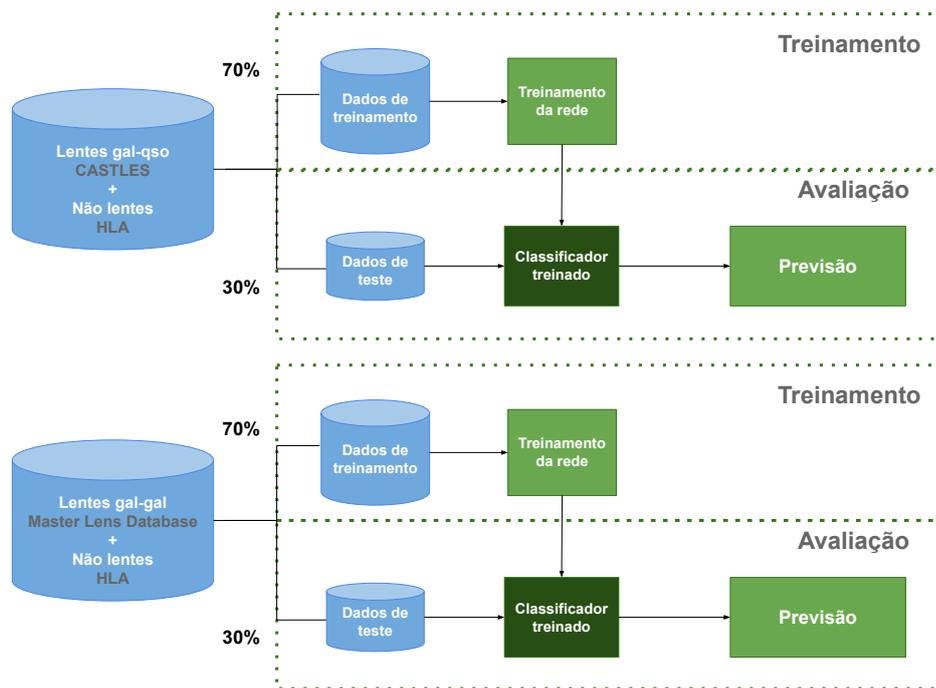


Figura 17 – Fluxograma - Etapa 1

Fonte: Autora.

3.2.3 Segunda etapa - Treinando com imagens simuladas

Devido à quantidade e condições desfavoráveis dos dados reais mencionadas anteriormente, utilizamos então, nesta segunda etapa, imagens simuladas no treinamento da rede siamesa, como realizado em artigos recentes utilizando técnicas de aprendizado de máquina, como os citados na seção 2.1.1. A simulação utilizada é disponibilizada já finalizada pelos realizadores do desafio, não sendo otimizada para nosso trabalho, assim nesta etapa verificamos a qualidade de resultados que podemos obter sem o custo necessário para realizar uma simulação própria.

Por isso alguns fatores que podem ter impacto sobre os resultados devem ser levados em consideração, como a diferença na qualidade e níveis de ruído entre as imagens simuladas e reais utilizadas nos testes, além dos filtros, disponíveis em diferentes frequências e quantidades. Devido a estas diferenças, a distribuição aprendida pela rede durante o treinamento com imagens simuladas não corresponde exatamente à distribuição real, mas pode ser considerada como uma aproximação, levando em conta estes fatores.

Após o treinamento, avaliamos o desempenho da rede em algumas situações diferentes, com imagens simuladas e reais, com e sem pré-processamento, e de dois tipos de lentes, como mostrado no fluxograma da figura 18. Utilizamos, mais especificamente, os seguintes conjuntos para teste:

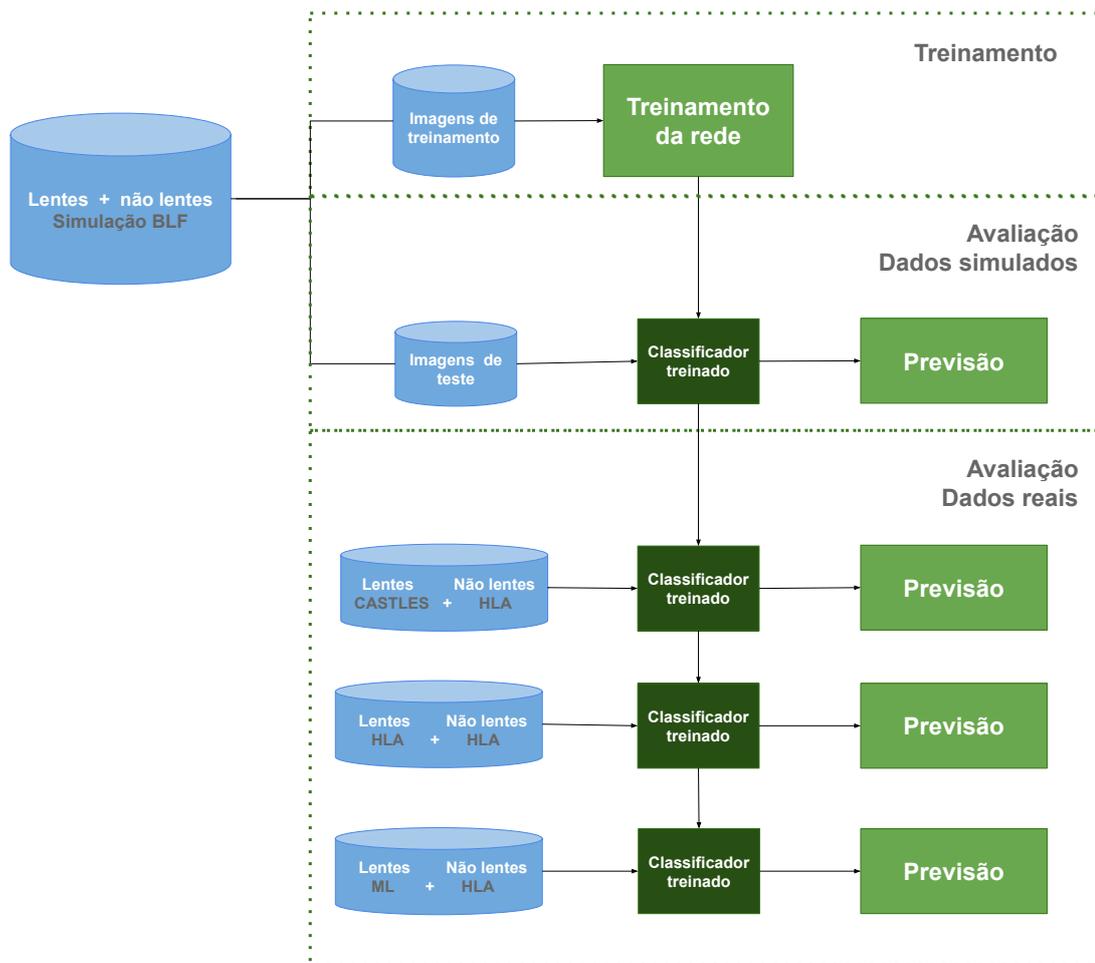


Figura 18 – Fluxograma - Etapa 2. Fluxograma indicando diferentes conjuntos utilizados para teste na segunda etapa e suas fontes: Simulação do desafio de *BolognaLens Factory*, lentes galáxia-quasar com pré-processamento (CASTLES), lentes galáxia-quasar sem pré-processamento (HLA), lentes galáxia-galáxia de diferentes levantamentos (*Master Lens Database*), além de não lentes reais (HLA)

Fonte: Autora.

- Imagens simuladas de lentes galáxia-galáxia e de não-lentes, parte da mesma simulação utilizada para treinamento:
- Imagens reais de lentes de tipo galáxia-quasar do levantamento CASTLES, com pré-processamento específico realizado pelos autores originais, e imagens reais exemplos de não-lentes obtidas a partir do *Hubble Legacy Archive*;
- Imagens reais de lentes de tipo galáxia-quasar do levantamento CASTLES sem o pré-processamento realizado pelos autores, e mesmos exemplos reais de não-lentes.
- Imagens reais de lentes galáxia-galáxia pertencentes a diferentes levantamentos, e mesmos exemplos reais de não-lentes;

4 Resultados

Para desenvolvimento do projeto foi utilizada a linguagem de programação Python em sua versão 3.7.5 e suas bibliotecas numpy 1.17.3, sklearn 0.21.3, opencv 4.2.0.32, keras 2.3.1, astropy 3.2.3, pandas 0.25.3, matplotlib 3.1.1 e plotly 4.8.1. A obtenção e pré-processamento das imagens foram realizados em um ambiente com processador Intel(R) Core(TM) i7-4790, de frequência base 3.60GHz, e 15GB de RAM. Já as etapas envolvendo construção, treinamento e avaliação das redes foram realizadas com a ferramenta *google colab*, utilizando a opção de aceleração por GPU.

4.1 Pré-processamento

Alguns exemplos de imagens antes e pós pré-processamento descrito na seção 3.2.1 são apresentados na figura 19.

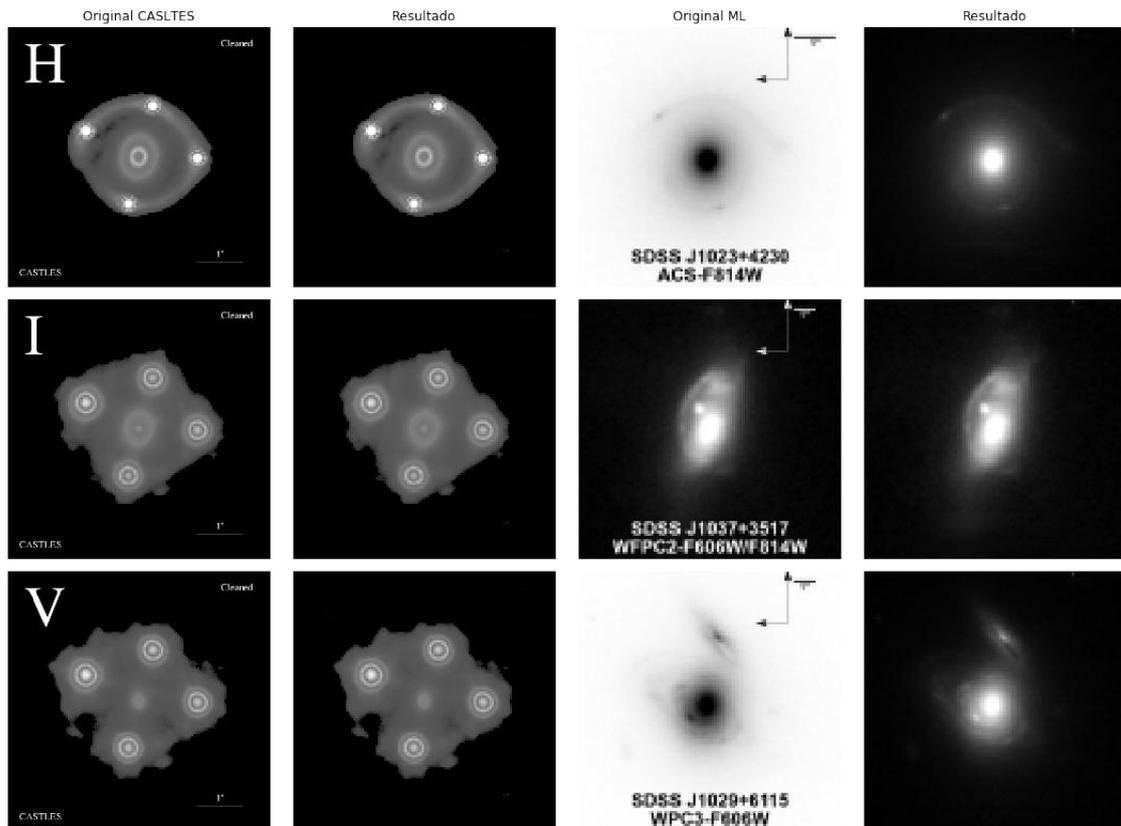


Figura 19 – Na sequência da esquerda para direita, a primeira e terceira colunas apresentam imagens originais anteriores à limpeza, das fontes CASTLES e *Master Lens Database*, respectivamente. Os resultados após o pré-processamento são vistos na segunda e quarta colunas.

Fonte: Autora.

4.2 Primeira etapa - Treinando com imagens reais

Nesta primeira etapa o desempenho da rede apresenta grande variação de acordo com a divisão de nosso conjunto de dados entre imagens de treinamento e de teste, e escolha de âncoras para as duas classes. Para ilustrar este efeito, apresentamos exemplos de resultados obtidos pelo mesmo processo, apenas variando estes fatores.

4.2.1 Lentes galáxia-quasar

Treinando e avaliando a rede apenas com imagens reais de lentes do tipo galáxia-quasar e não-lentes, obtemos em um primeiro exemplo a matriz de confusão da figura 20 e o histograma da figura 22, que indica a probabilidade com que a rede classifica lentes como lentes, e não-lentes como não-lentes. Neste exemplo vemos que o modelo classifica todas as imagens como lentes, com probabilidade maior que 80%.

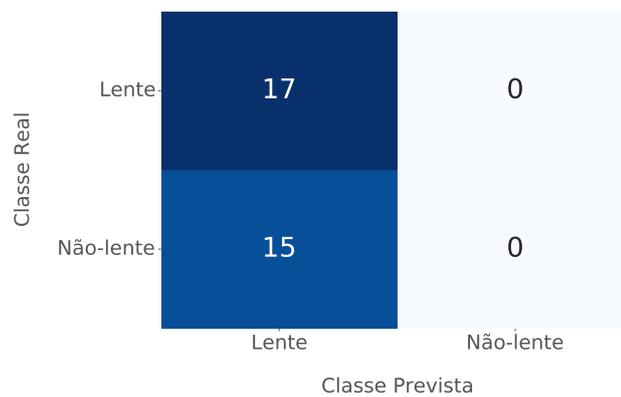


Figura 20 – Matriz de confusão - Exemplo 1 - Lentes galáxia-quasar

Fonte: Autora.

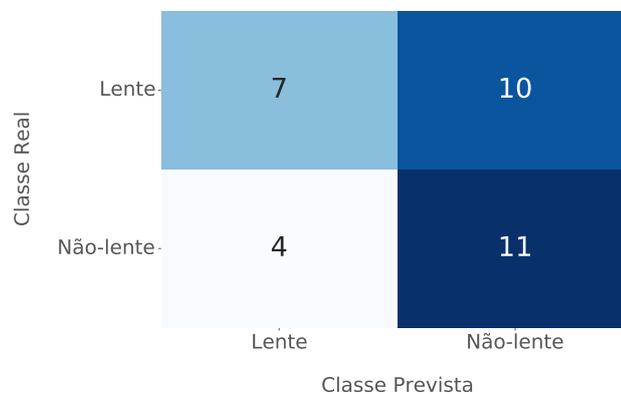


Figura 21 – Matriz de confusão - Exemplo 2 - Lentes galáxia-quasar

Fonte: Autora.

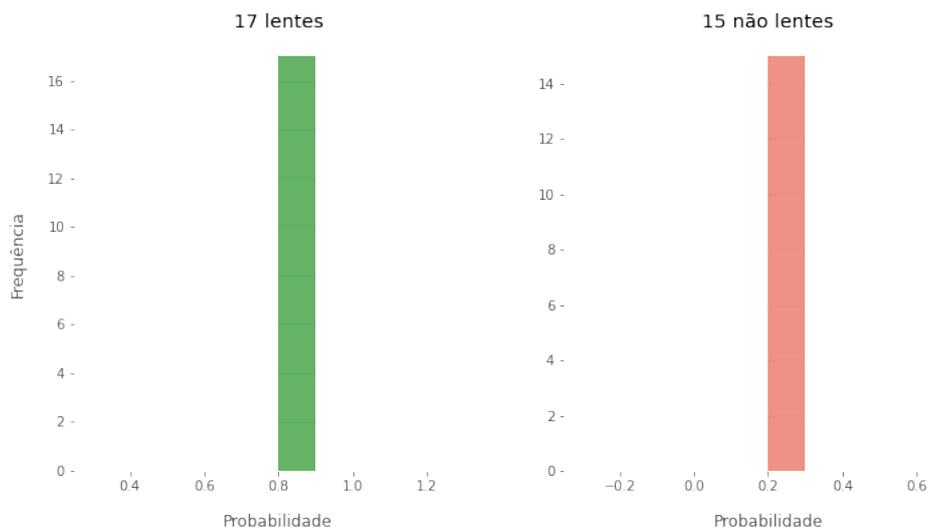


Figura 22 – Distribuição de probabilidades - Exemplo 1 - Lentes galáxia-quasar

Fonte: Autora.

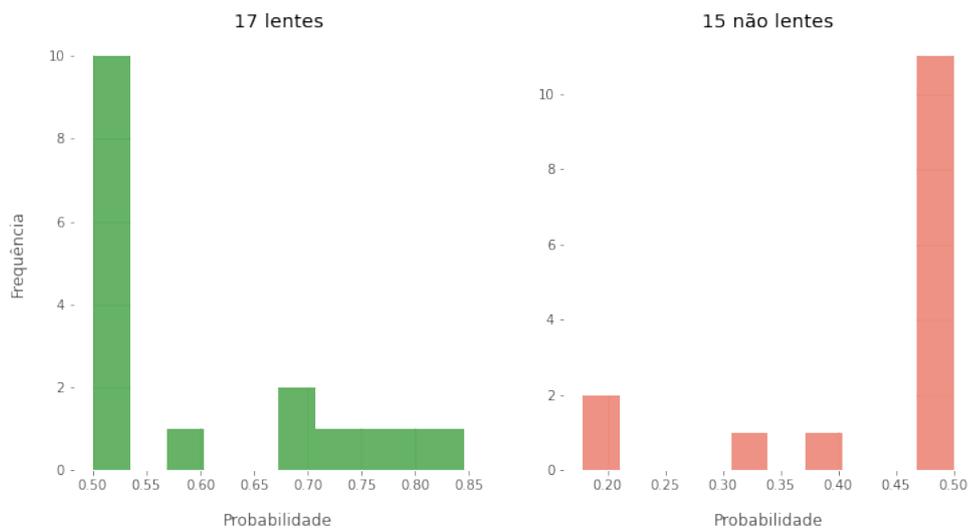


Figura 23 – Distribuição de probabilidades - Exemplo 2 - Lentes galáxia-quasar

Fonte: Autora.

Realizando o mesmo processo, porém com uma divisão diferente do conjunto de dados em treinamento e teste, e utilizando diferentes âncoras, podemos obter resultados completamente diferentes, como a matriz de confusão 21 e o histograma de probabilidades 23. Apesar de possuir melhor desempenho na classificação de não-lentes, apenas 41% das lentes reais são identificadas de forma correta. A maior parte das classificações são realizadas com probabilidade próxima de 50%, o que indica a dificuldade da rede em associar as novas imagens à uma das classes de treinamento. A tabela 1 compara as

métricas de avaliação obtidas nestes dois exemplos.

Tabela 1 – Métricas - Etapa 1 - lentes galáxia-quasar

Métrica	Exemplo 1	Exemplo 2
Acurácia	0.53	0.56
Precisão	0.53	0.64
Recall	1.0	0.41
F1	0.69	0.5

Fonte – Autora.

4.2.2 Lentes galáxia-galáxia

Em seguida treinamos e avaliamos a rede utilizando, desta vez, as imagens de lentes galáxia-galáxia, mantendo as mesmas imagens de não-lentes. Em um primeiro exemplo obtemos a matriz de confusão da figura 24 e o histograma da figura 26. 76% das lentes são identificadas, em sua maioria com probabilidade acima de 80%.

Outro exemplo, com âncoras e divisões diferentes do conjunto de dados apresenta os resultados vistos na matriz de confusão da figura 25 e histograma 27. Neste caso apenas 36% das lentes são identificadas, enquanto 40% das lentes são classificadas de forma incorreta com probabilidade maior que 90%, apesar de possuir melhor desempenho na classificação de não-lentes. As métricas do desempenho da rede nos dois exemplos são apresentadas na tabela 2.

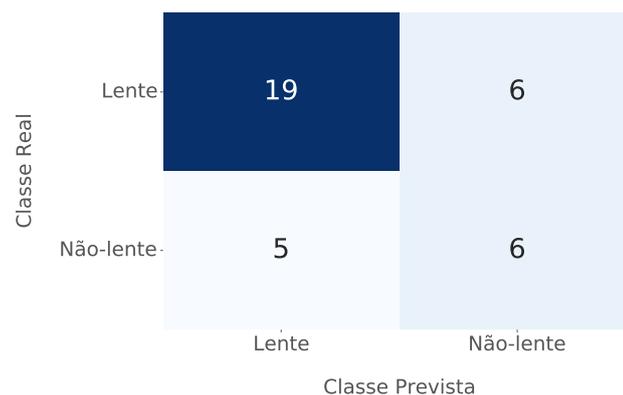


Figura 24 – Matriz de confusão - Exemplo 1 - Lentes galáxia-galáxia

Fonte: Autora.

Esta alta variação observada nos resultados indica, como já esperávamos, que nosso conjunto de imagens reais não é suficientemente completo e representativo das classes lente e não-lente. Outros fatores, já mencionados, que também podem ter influência sobre

Classe Real	Lente	9	16
	Não-lente	3	8
		Lente	Não-lente
		Classe Prevista	

Figura 25 – Matriz de confusão - Exemplo 2 - Lentes galáxia-galáxia

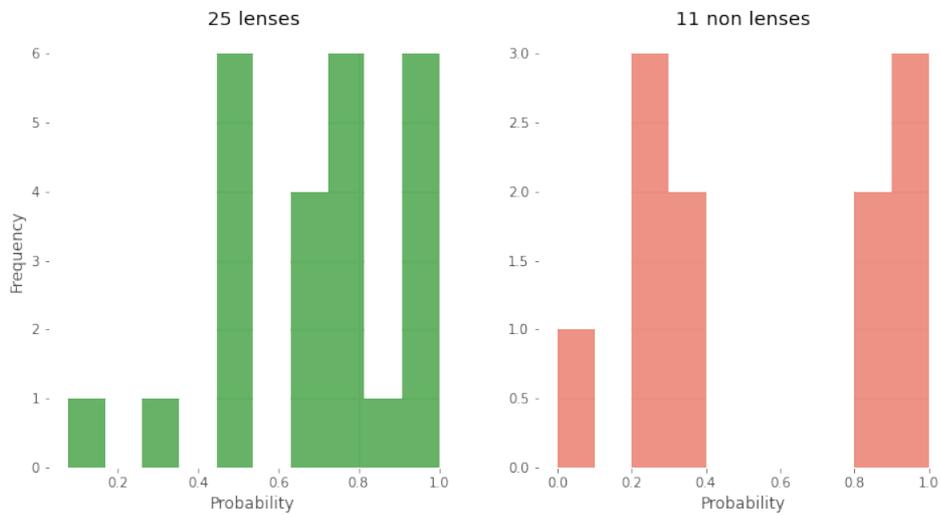


Figura 26 – Distribuição de probabilidades - Exemplo 1 - Lentes galáxia-galáxia

Fonte: Autora.

os resultados e devem ser considerados, estão relacionados a heterogeneidade dos dados disponíveis, incluindo a utilização de filtros diferentes, e as diferenças de pré-processamento.

Verificamos então que a proposta desta etapa não é viável no presente, sendo mais adequada para um cenário futuro, com um número maior de lentes já identificadas disponíveis, e assim um conjunto mais volumoso e representativo de imagens para ser utilizado no treinamento da rede.

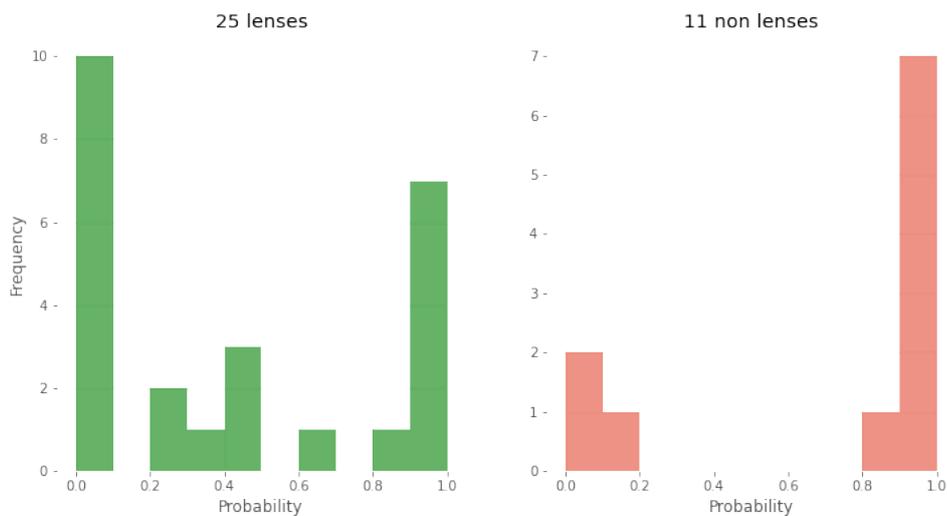


Figura 27 – Distribuição de probabilidades - Exemplo 2 - Lentes galáxia-galáxia

Fonte: Autora.

Tabela 2 – Métricas - Etapa 1 - Lentes galáxia-galáxia

Métrica	Exemplo 1	Exemplo 2
Acurácia	0.69	0.47
Precisão	0.79	0.75
Recall	0.76	0.36
F1	0.77	0.49

Fonte – Autora.

4.3 Segunda etapa - Treinando com imagens simuladas

Nesta segunda etapa, realizamos o treinamento da rede com imagens simuladas de lentes galáxia-galáxia e de não-lentes, avaliando seu desempenho em diferentes conjuntos.

Durante a avaliação dos resultados obtidos e sua apresentação nas próximas seções a métrica *recall* representa a proporção de lentes verdadeiras identificadas corretamente, enquanto acurácia, precisão e F1 dependem também do desempenho do método na classificação de não-lentes. Como este método foi desenvolvido com o propósito de funcionar como uma etapa inicial para filtragem de candidatos a lentes, estamos interessados em soluções de alto *recall*, indicando que candidatos verdadeiros não são descartados neste processo.

4.3.1 Avaliação em imagens simuladas

Avaliando o desempenho do modelo primeiramente na classificação de imagens também simuladas, parte do mesmo desafio, obtemos a matriz de confusão da figura 28, e o histograma de distribuição de probabilidade das classificações da figura 29.

De acordo com estes resultados, vemos que 87% das classificações feitas estão corretas (acurácia). A rede identifica 83% das lentes utilizadas para o teste (*recall*), mais de 60% destas com probabilidade maior que 80%. Os histogramas possuem configuração próxima a desejada, indicando que a maior parte dos exemplos são classificados corretamente com alta probabilidade, e o número de classificações com menor probabilidade decai rapidamente. As métricas são apresentadas na tabela 3.

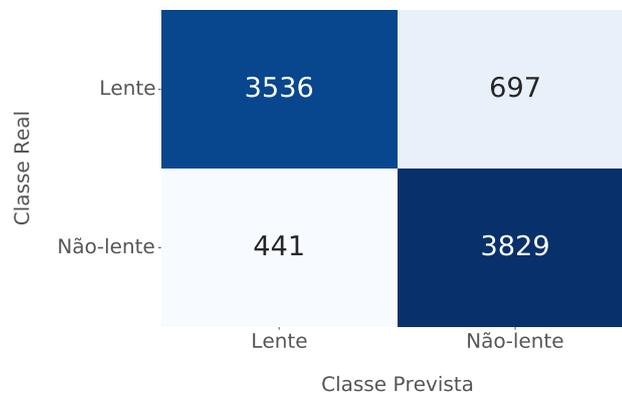


Figura 28 – Matriz de confusão - Etapa 2 - Lentes galáxia-galáxia simuladas

Fonte: Autora.

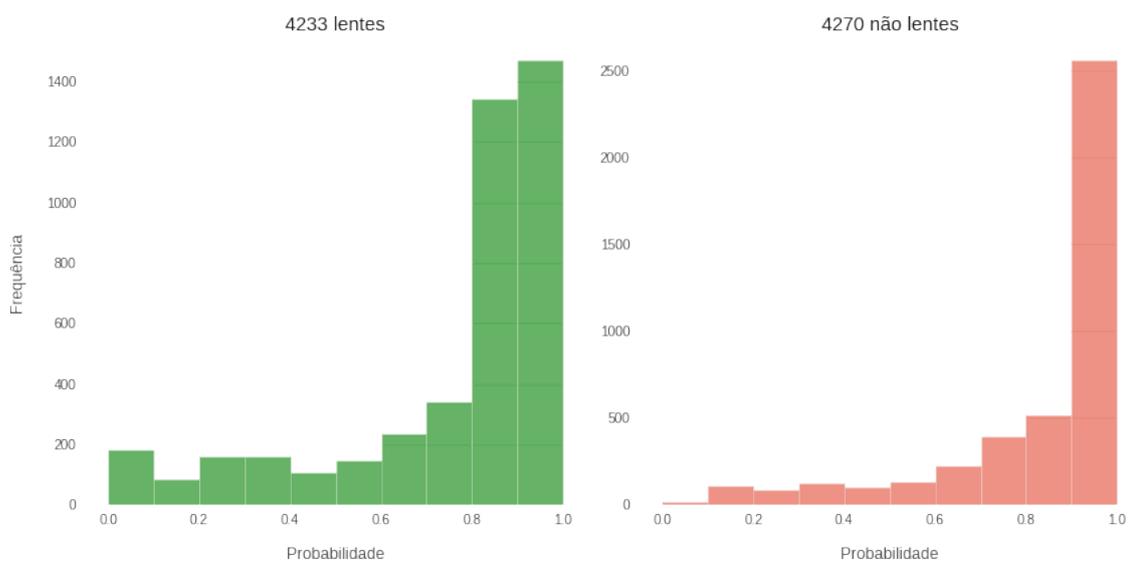


Figura 29 – Distribuição de probabilidade -Etapa 2 - Lentes galáxia-galáxia simuladas

Fonte: Autora.

Tabela 3 – Métricas - Etapa 2 -
Lentes galáxia-galáxia
simuladas

Acurácia	0.87
Precisão	0.89
Recall	0.83
F1	0.86

Fonte – Autora.

4.3.2 Avaliação em imagens reais

4.3.2.1 Lentes galáxia-quasar

Avaliando o desempenho da rede ao classificar imagens reais de lentes galáxia-quasar do levantamento CASTLES, obtemos a matriz de confusão da figura 30 e a distribuição de probabilidade de classificações da figura 31.

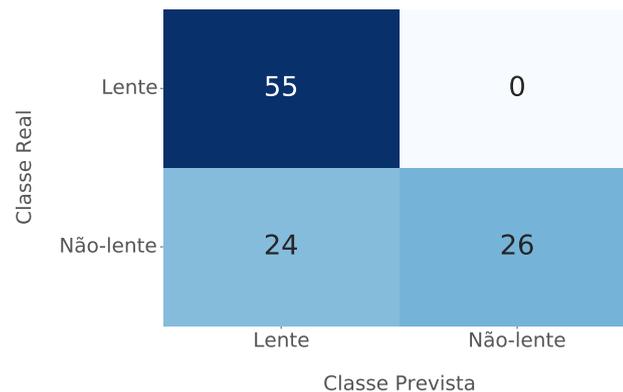


Figura 30 – Matriz de confusão - Etapa 2 - Lentes reais galáxia-quasar

Fonte: Autora.

Observamos que todas as lentes reais foram classificadas corretamente, com probabilidade maior que 90%, apesar de serem de tipo diferente daquelas utilizadas para treinamento. Já quanto a classificação das não-lentes reais, a rede não possui desempenho satisfatório, sendo quase metades destas classificadas incorretamente como lentes, enquanto aquelas classificadas corretamente possuem probabilidade menor que 70%. A tabela 4 apresenta as métricas de avaliação obtidas.

Avaliando, ainda, o desempenho da rede na classificação de lentes galáxia-quasar também parte do levantamento CASTLES, porém sem o pré-processamento realizado pelos autores do mesmo, obtemos a matriz de confusão da figura 32, e histograma de probabilidades da figura 33.

Verificamos, como previsto, que quase todas as lentes (93%) sem este pré-processamento

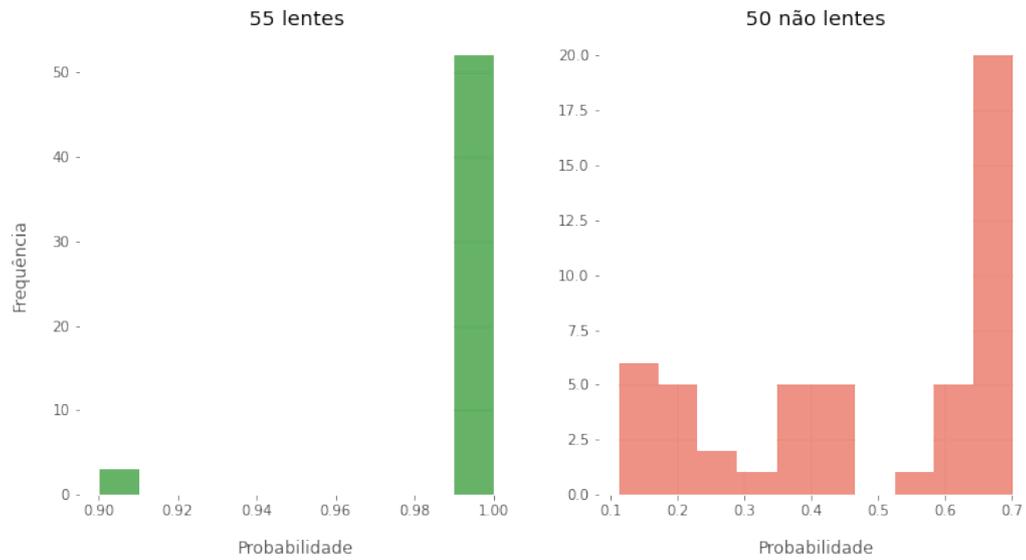


Figura 31 – Distribuição de probabilidade - Etapa 2 - Lentes reais galáxia-quasar

Fonte: Autora.

Tabela 4 – Métricas - Etapa 2 - Lentes reais galáxia-quasar

Acurácia	0.77
Precisão	0.70
Recall	1.0
F1	0.82

Fonte – Autora.

são classificadas de maneira incorreta, como não lentes, com alta probabilidade. As métricas são apresentadas na tabela 5 .

Este resultado era esperado, pois como pode ser visto na figura 34, em muitas imagens sem este pré-processamento, características importantes para identificação visual de um sistema como lente estão menos destacadas ou ausentes, como a presença da galáxia defletora. Não detectando estas características, a rede as classifica como não-lentes.

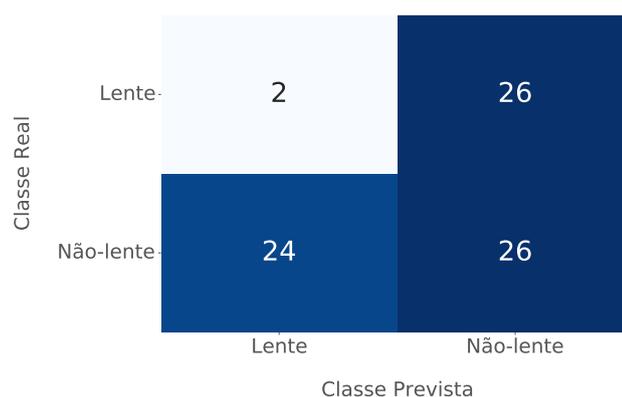


Figura 32 – Matriz de confusão - Etapa 2 - Lentes reais galáxia-quasar sem pré-processamento CASTLES

Fonte: Autora.

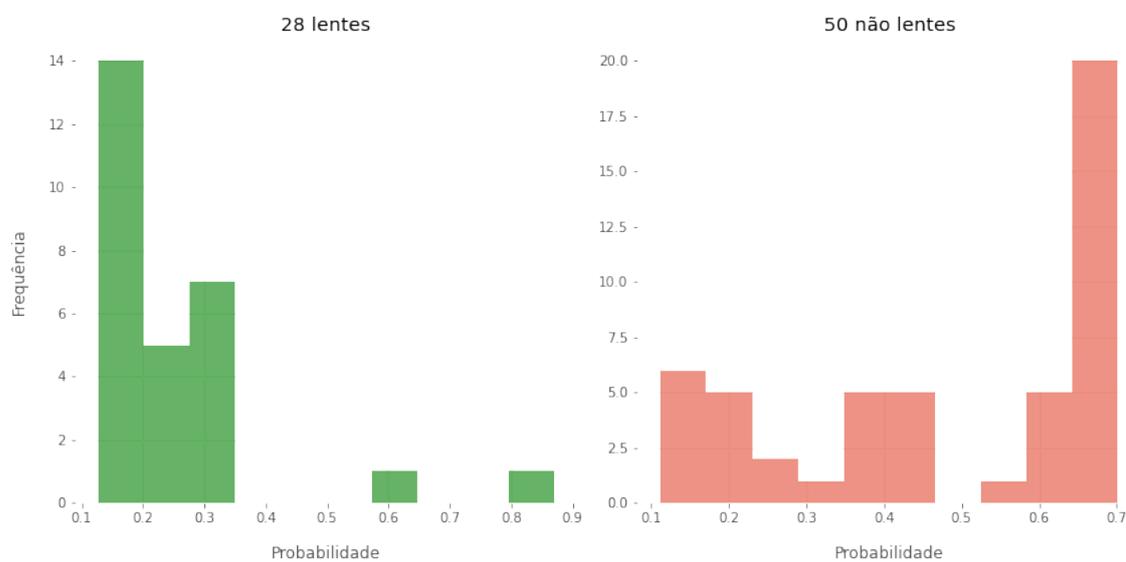


Figura 33 – Distribuição de probabilidades - Etapa 2 - Lentes reais galáxia-quasar sem pré-processamento CASTLES

Fonte: Autora.

Tabela 5 – Métricas - Etapa 2 - Lentes reais galáxia-quasar sem pré-processamento CASTLES

Acurácia	0.35
Precisão	0.08
Recall	0.07
F1	0.07

Fonte – Autora.

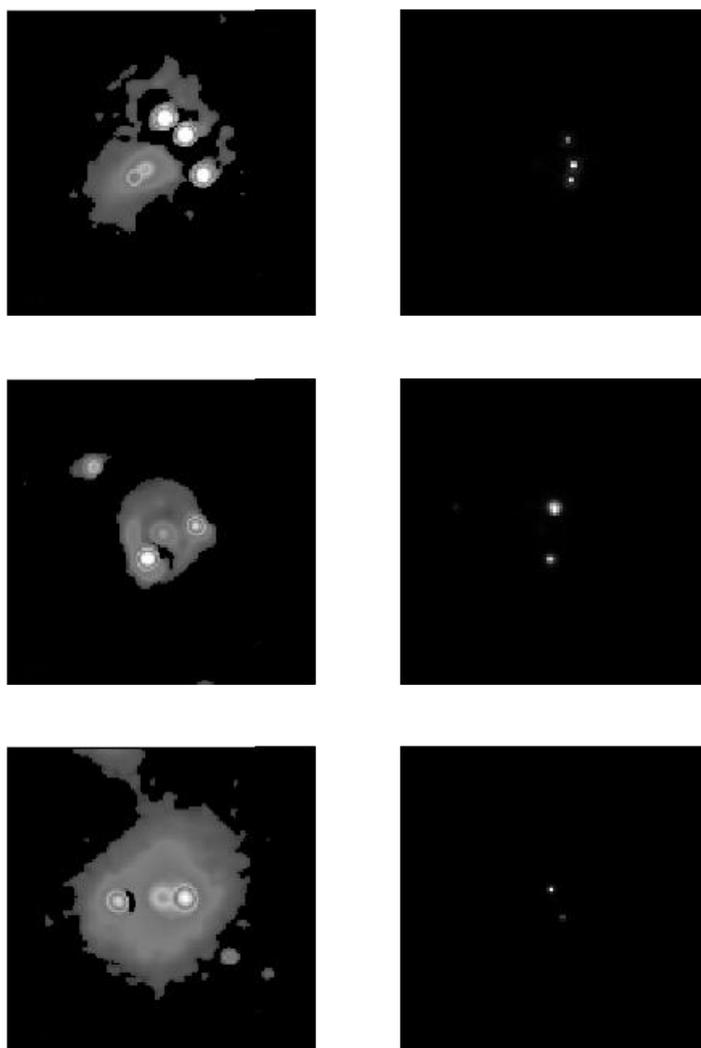


Figura 34 – Lentes pertencentes ao levantamento CASTLES com pré-processamento realizado pelos autores originais (esquerda) ao lado de suas imagens antes deste pré-processamento (direita)

Fonte: Autora.

4.3.2.2 Lentes galáxia-galáxia

Por fim, testando a rede também com imagens de lentes reais do tipo galáxia-galáxia, mesmo tipo daquelas utilizadas no treinamento, obtemos a matriz de confusão da figura 35 e distribuição de probabilidade 36.

Apesar de este ser um conjunto bastante heterogêneo, formado por imagens de lentes de diferentes levantamentos, vemos que 95% delas foram classificadas corretamente, em sua maioria com alta probabilidade, como visto no histograma de classificação das lentes. As métricas de avaliação são apresentadas na tabela 6.

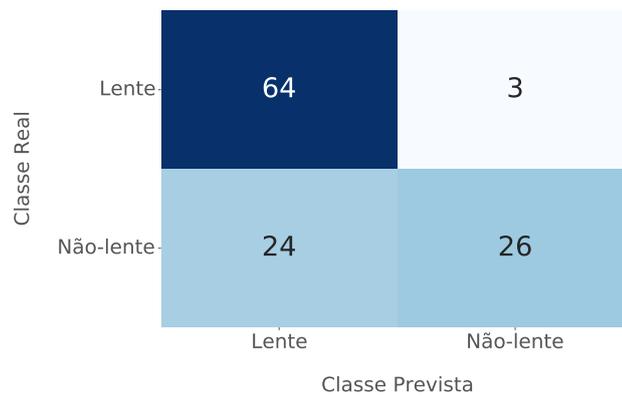


Figura 35 – Matriz de confusão - Etapa 2 - Lentes reais galáxia-galáxia

Fonte: Autora.

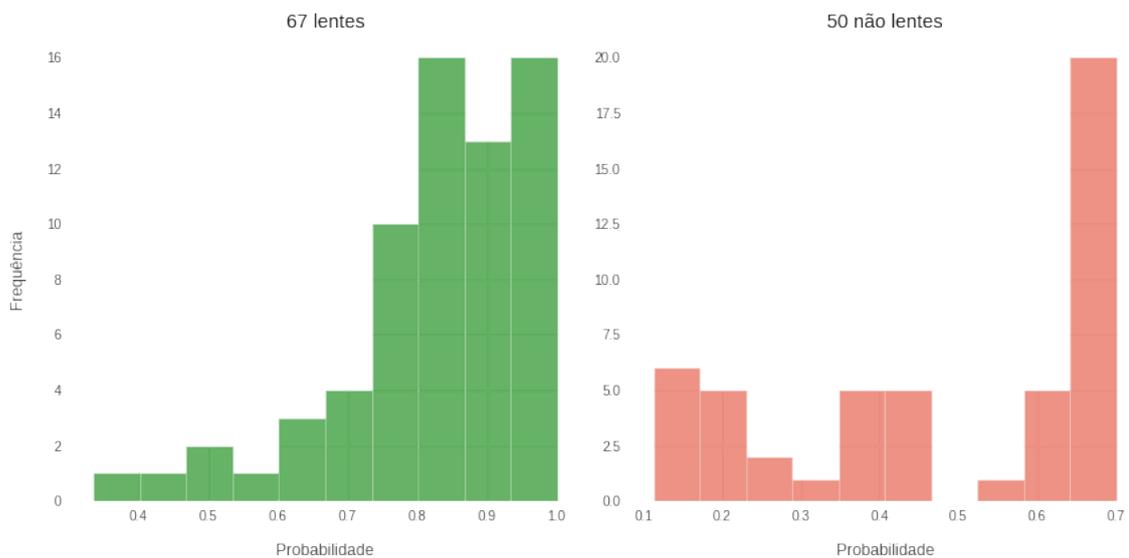


Figura 36 – Distribuição de probabilidade - Etapa 3 - Lentes reais galáxia-galáxia

Fonte: Autora.

Tabela 6 – Métricas - Etapa 3 -
Lentes reais galáxia-
galáxia

Acurácia	0.77
Precisão	0.73
Recall	0.95
F1	0.82

Fonte – Autora.

4.3.2.3 Visualizando o problema através da análise de componentes principais (PCA)

Como visto na seção 2.2.3.3, a rede siamesa projeta imagens de entrada em um hiperespaço de *features*, onde exemplos de mesma classe estão mais próximos entre si. No método desenvolvido com nossa arquitetura de rede siamesa, este hiperespaço possui 10 dimensões. Assim, com o objetivo de poder visualizar esta projeção, auxiliando na interpretação dos resultados obtidos, utilizamos o algoritmo de Análise de Componentes Principais (PCA) para reduzir a dimensionalidade dos vetores de *feature*. Em um espaço resultante de 3 dimensões, podemos gerar os gráficos das figuras 37 e 38, contendo exemplos de treinamento e teste. O funcionamento do PCA e como ele realiza redução de dimensionalidade é descrito em (Jolliffe, 2016).

Ao fazermos isto, a variância total explicada pelos 3 componentes do PCA é de 98%, o que indica que o espaço tridimensional obtido com a sua utilização é uma boa aproximação para a projeção original realizada pela rede, no espaço de 10 dimensões.

No gráfico 37 verificamos que tanto as não-lentes reais utilizadas nos testes como as lentes do levantamento CASTLES sem pré-processamento se encontram próximas à divisão entre as duas classes do treinamento, apresentando assim maior dificuldade de classificação pela rede, de acordo com os resultados observados.

Já no gráfico 38, verificamos que o conjunto de imagens de lentes galáxia-quasar do levantamento CASTLES está mais distante das imagens de treinamento, porém mais próximo da classe de lentes entre as imagens de treinamento, e assim todas as suas imagens são classificadas corretamente, como visto nos resultados apresentados.

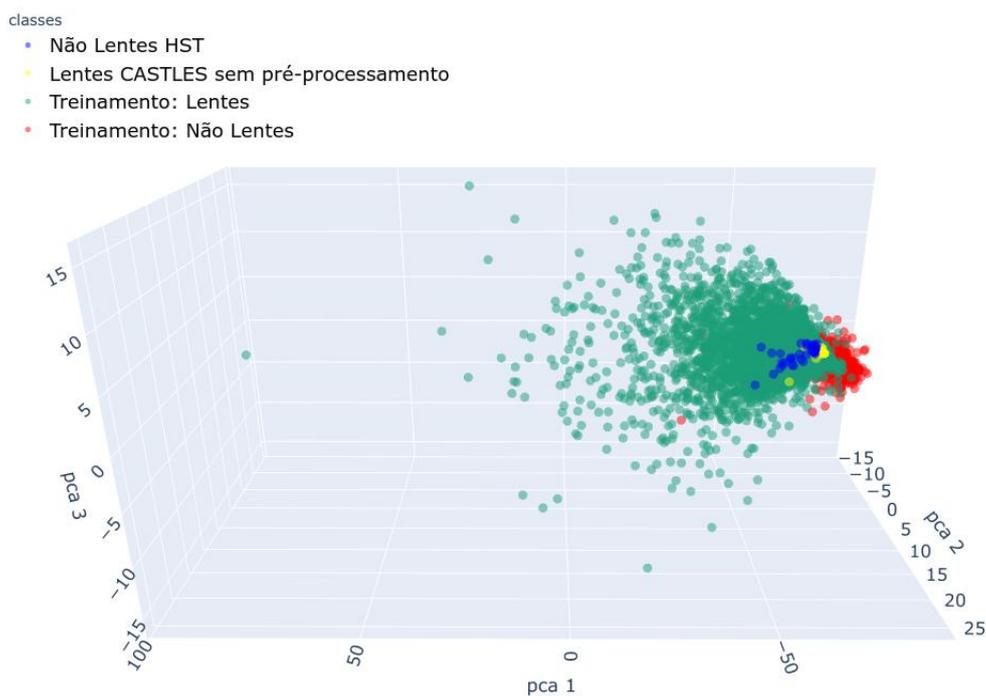


Figura 37 – Componentes do PCA de imagens de treinamento e teste

Fonte: Autora

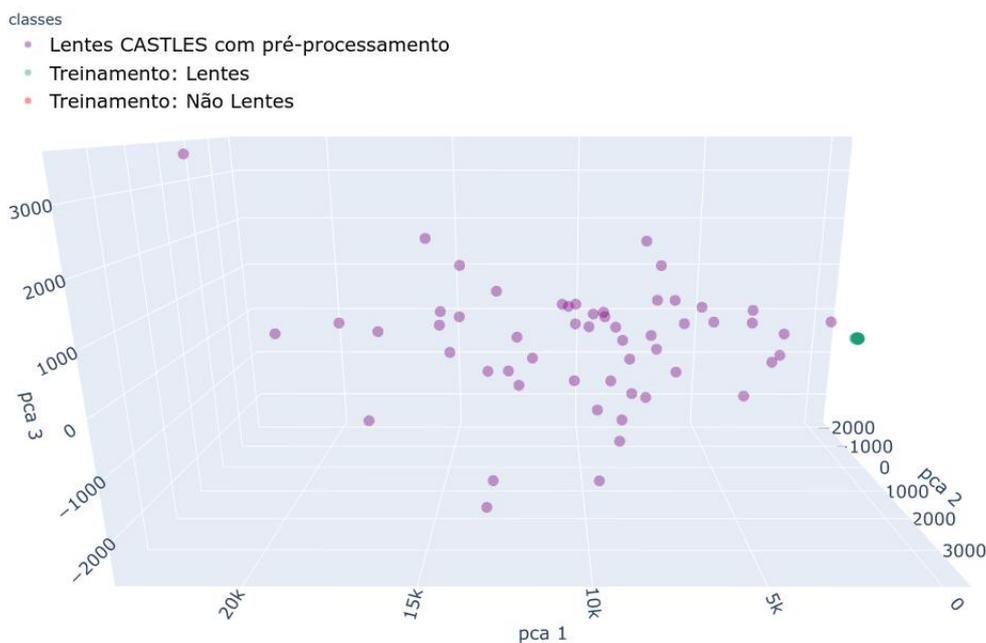


Figura 38 – Componentes do PCA incluindo imagens de lentes do levantamento CASTLES com pré-processamento

Fonte: Autora.

5 Conclusão

No decorrer deste trabalho procuramos desenvolver um método automatizado para auxiliar na identificação de lentes gravitacionais, selecionando possíveis candidatos. Para isto utilizamos uma rede neural siamesa que realiza a classificação de imagens nas classes lente e não-lente.

Em uma primeira etapa realizamos treinamento e avaliação da performance do método utilizando um conjunto de imagens reais. Como os dados de lentes conhecidas atualmente são pouco volumosos e heterogêneos, não foram obtidos resultados satisfatórios, como já era esperado.

Em uma segunda etapa, treinamos a rede com imagens simuladas, disponíveis em maior quantidade, contendo lentes galáxia-galáxia. O método apresentou bons resultados quando avaliado em imagens da mesma simulação, imagens reais de lentes galáxia-galáxia e imagens reais de lentes galáxia-quasar com pré-processamento próprio do levantamento, identificando nestes casos 83%, 95% e 100% das lentes, respectivamente.

Apesar do método não apresentar a mesma capacidade de generalização para a classificação de não-lentes reais, ele elimina metade destas, mantendo bom desempenho na classificação de lentes, que é a classe de interesse. Assim ele pode ser utilizado como uma etapa inicial sobre um conjunto de imagens candidatas a lentes, reduzindo-o pela metade, sem eliminar grande quantidade de candidatos correspondentes a lentes reais (até 5% nos testes).

O método apresenta ainda diversas possibilidades de desenvolvimento posterior, como uma análise mais profunda do efeito de pré-processamento em imagens reais, como o realizado pelo levantamento CASTLES, sobre sua classificação. Pode ser realizada uma investigação do efeito do tipo das galáxias utilizadas como exemplos de não-lentes nos testes e como cada tipo é classificado, além de realizar testes com maior volume de exemplos de não-lentes reais. Outro aspecto interessante a ser explorado é o desenvolvimento de uma simulação própria de lentes gravitacionais para treinamento da rede. Por fim também a sua integração a outro método para identificação de lentes.

Referências

- Bartelmann. Gravitational lensing. *Classical and Quantum Gravity*, IOP Publishing, v. 27, n. 23, p. 233001, 2010. Citado na página 18.
- Bartelmann. Weak gravitational lensing. *arXiv preprint arXiv:1612.06535*, 2016. Citado na página 15.
- Bellagamba. Zooming into the cosmic horseshoe: new insights on the lens profile and the source shape. *Monthly Notices of the Royal Astronomical Society*, Oxford University Press, p. stw2726, 2016. Citado na página 12.
- Bolton, A. S. et al. The sloan lens acs survey. i. a large spectroscopically selected sample of massive early-type lens galaxies. *The Astrophysical Journal*, IOP Publishing, v. 638, n. 2, p. 703, 2006. Citado na página 18.
- Bom. A neural network gravitational arc finder based on the mediatrix filamentation method. *Astronomy & Astrophysics*, EDP Sciences, v. 597, p. A135, 2017. Citado 2 vezes nas páginas 12 e 19.
- Brad Whitmore. *Welcome to the Hubble Legacy Archive*. Disponível em: <<https://hla.stsci.edu/>>. Acesso em: 25 mai. 2020. Citado na página 33.
- Brownstein. The boss emission-line lens survey (bells). i. a large spectroscopically selected sample of lens galaxies at redshift 0.5. *The Astrophysical Journal*, IOP Publishing, v. 744, n. 1, p. 41, 2011. Citado na página 18.
- Chollet, F. Xception: Deep learning with depthwise separable convolutions. p. 1251–1258, 2017. Citado na página 29.
- Chopra. Learning a similarity metric discriminatively, with application to face verification. v. 1, p. 539–546, 2005. Citado na página 31.
- Collett. The population of galaxy–galaxy strong lenses in forthcoming optical imaging surveys. *The Astrophysical Journal*, IOP Publishing, v. 811, n. 1, p. 20, 2015. Citado 2 vezes nas páginas 12 e 18.
- C.S. Kochanek. *CASTLES Survey*. Disponível em: <<https://www.cfa.harvard.edu/castles/>>. Acesso em: 25 mai. 2020. Citado na página 33.
- Datacamp. 2018. Disponível em: <<https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>>. Acesso em: 30 mai. 2020. Citado na página 31.
- Deane. The preferentially magnified active nucleus in iras f10214+ 4724–iii. vlbi observations of the radio core. *Monthly Notices of the Royal Astronomical Society*, The Royal Astronomical Society, v. 434, n. 4, p. 3322–3336, 2013. Citado na página 12.
- Dye. Decomposition of the visible and dark matter in the einstein ring 0047–2808 by semilinear inversion. *The Astrophysical Journal*, IOP Publishing, v. 623, n. 1, p. 31, 2005. Citado 2 vezes nas páginas 12 e 17.

- Ellis, R. S. Gravitational lensing: a unique probe of dark matter and dark energy. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, The Royal Society Publishing, v. 368, n. 1914, p. 967–987, 2010. Citado na página 17.
- Falco. The castles gravitational lensing tool. v. 237, p. 25, 2001. Citado na página 33.
- Faure, C. et al. First catalog of strong lens candidates in the cosmos field. *The Astrophysical Journal Supplement Series*, IOP Publishing, v. 176, n. 1, p. 19, 2008. Citado na página 18.
- Flach, P. *Machine learning: the art and science of algorithms that make sense of data*. [S.l.]: Cambridge University Press, 2012. Citado 2 vezes nas páginas 19 e 20.
- Gavazzi. Ringfinder: automated detection of galaxy-scale gravitational lenses in ground-based multi-filter imaging data. *The Astrophysical Journal*, IOP Publishing, v. 785, n. 2, p. 144, 2014. Citado na página 18.
- Geach. The red radio ring: a gravitationally lensed hyperluminous infrared radio galaxy at $z=2.553$ discovered through the citizen science project space warps. *Monthly Notices of the Royal Astronomical Society*, Oxford University Press, v. 452, n. 1, p. 502–510, 2015. Citado na página 18.
- Hartley. Support vector machine classification of strong gravitational lenses. *Monthly Notices of the Royal Astronomical Society*, Oxford University Press, v. 471, n. 3, p. 3378–3397, 2017. Citado 2 vezes nas páginas 12 e 19.
- He, K. et al. Deep residual learning for image recognition. p. 770–778, 2016. Citado na página 29.
- HLEG, E. *A definition of Artificial Intelligence: main capabilities and scientific disciplines*. [S.l.]: Retrieved from High-Level Expert Group on Artificial Intelligence (AI HLEG ... , 2019. Citado na página 19.
- Jackson, N. Gravitational lenses and lens candidates identified from the cosmos field. *Monthly Notices of the Royal Astronomical Society*, Blackwell Publishing Ltd Oxford, UK, v. 389, n. 3, p. 1311–1318, 2008. Citado na página 18.
- Jacobs. Finding strong lenses in cfhtls using convolutional neural networks. *Monthly Notices of the Royal Astronomical Society*, Oxford University Press, v. 471, n. 1, p. 167–181, 2017. Citado 2 vezes nas páginas 12 e 19.
- Jolliffe. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, The Royal Society Publishing, v. 374, n. 2065, p. 20150202, 2016. Citado na página 51.
- Joseph. A pca-based automated finder for galaxy-scale strong lenses. *Astronomy & Astrophysics*, EDP Sciences, v. 566, p. A63, 2014. Citado na página 18.
- Koch. Siamese neural networks for one-shot image recognition. v. 2, 2015. Citado na página 32.
- Kochanek. The implications of lenses for galaxy structure. *The Astrophysical Journal*, v. 373, p. 354–368, 1991. Citado na página 12.

- Krizhevsky. Imagenet classification with deep convolutional neural networks. 2012. Citado na página 29.
- LeCun, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Ieee, v. 86, n. 11, p. 2278–2324, 1998. Citado na página 29.
- LeCun, Y. A. et al. Efficient backprop. Springer, p. 9–48, 2012. Citado na página 24.
- Marshall. 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*, IOP Publishing, v. 671, n. 2, p. 1196, 2007. Citado 2 vezes nas páginas 12 e 17.
- Marshall. Space warps–i. crowdsourcing the discovery of gravitational lenses. *Monthly Notices of the Royal Astronomical Society*, Oxford University Press, v. 455, n. 2, p. 1171–1190, 2016. Citado na página 18.
- MASTER Lens Database. Disponível em: <<http://masterlens.astro.utah.edu>>. Acesso em: 25 mai. 2020. Citado na página 33.
- McCarthy. *What is AI? / Basic Questions*. [S.l.]. Disponível em: <<http://jmc.stanford.edu/artificial-intelligence/what-is-ai/index.html>>. Acesso em: 25 jun. 2020. Citado na página 19.
- McKean. Strong gravitational lensing with the ska. *arXiv preprint arXiv:1502.03362*, 2015. Citado 2 vezes nas páginas 12 e 18.
- Metcalf. Glamer–i. a code for gravitational lensing simulations with adaptive mesh refinement. *Monthly Notices of the Royal Astronomical Society*, Oxford University Press, v. 445, n. 2, p. 1942–1953, 2014. Citado na página 33.
- Metcalf. *Gravitational Lens Finding Challenge*. 2018. Disponível em: <http://metcalf1.difa.unibo.it/blf-portal/gg_challenge.html>. Acesso em: 25 mai. 2020. Citado na página 33.
- Metcalf. The strong gravitational lens finding challenge. *Astronomy & Astrophysics*, EDP Sciences, v. 625, p. A119, 2019. Citado 2 vezes nas páginas 19 e 33.
- MIT Deep Learning. 2018. Disponível em: <introtodeeplearning.com>. Acesso em: 30 mai. 2020. Citado na página 30.
- Mitchell, T. M. *Machine learning*. [S.l.: s.n.], 1997. Citado na página 24.
- More. Space warps–ii. new gravitational lens candidates from the cfhtls discovered through citizen science. *Monthly Notices of the Royal Astronomical Society*, Oxford University Press, v. 455, n. 2, p. 1191–1210, 2016. Citado na página 18.
- Murphy, K. P. *Machine learning: a probabilistic perspective*. [S.l.]: MIT press, 2012. Citado 2 vezes nas páginas 19 e 20.
- Narayan. Lectures on gravitational lensing. *arXiv preprint astro-ph/9606001*, 1996. Citado na página 18.
- Nielsen, M. A. *Neural networks and deep learning*. [S.l.]: Determination press San Francisco, CA, 2015. v. 2018. Citado na página 24.

- Ostrovski. Vdes j2325- 5229 az= 2.7 gravitationally lensed quasar discovered using morphology-independent supervised machine learning. *Monthly Notices of the Royal Astronomical Society*, Oxford University Press, v. 465, n. 4, p. 4325–4334, 2017. Citado 2 vezes nas páginas 12 e 19.
- Packt. 2019. Disponível em: <https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781789138900/7/ch07/v11sec86/creating-a-siamese-neural-network-in-keras>. Acesso em: 10 jun. 2020. Citado na página 30.
- Pawase, R. et al. A 7 deg² survey for galaxy-scale gravitational lenses with the hst imaging archive. *Monthly Notices of the Royal Astronomical Society*, Narnia, v. 439, n. 4, p. 3392–3404, 2014. Citado na página 18.
- Petkova. Glamer–ii. multiple-plane gravitational lensing. *Monthly Notices of the Royal Astronomical Society*, Oxford University Press, v. 445, n. 2, p. 1954–1966, 2014. Citado na página 33.
- Petrillo. Finding strong gravitational lenses in the kilo degree survey with convolutional neural networks. *Monthly Notices of the Royal Astronomical Society*, Oxford University Press, v. 472, n. 1, p. 1129–1150, 2017. Citado 2 vezes nas páginas 12 e 19.
- Petters. *Singularity theory and gravitational lensing*. [S.l.]: Springer Science & Business Media, 2012. v. 21. Citado na página 18.
- 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*, Narnia, v. 128, n. 4, p. 307–310, 1964. Citado na página 12.
- Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citado na página 21.
- Schneider. *Gravitational lensing: strong, weak and micro: Saas-Fee advanced course 33*. [S.l.]: Springer Science & Business Media, 2006. v. 33. Citado na página 18.
- Seidel. Arcfinder: an algorithm for the automatic detection of gravitational arcs. *Astronomy & Astrophysics*, EDP Sciences, v. 472, n. 1, p. 341–352, 2007. Citado na página 18.
- Shu. The boss emission-line lens survey. iv. smooth lens models for the bells gallery sample. *The Astrophysical Journal*, IOP Publishing, v. 833, n. 2, p. 264, 2016. Citado 2 vezes nas páginas 12 e 17.
- Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Citado na página 29.
- Smith. The sinfoni nearby elliptical lens locator survey: discovery of two new low-redshift strong lenses and implications for the initial mass function in giant early-type galaxies. *Monthly Notices of the Royal Astronomical Society*, The Royal Astronomical Society, v. 449, n. 4, p. 3441–3457, 2015. Citado na página 18.
- Sonnenfeld, A. et al. Survey of gravitationally-lensed objects in hsc imaging (sugohi). i. automatic search for galaxy-scale strong lenses. *Publications of the Astronomical Society of Japan*, Oxford University Press, v. 70, n. SP1, p. S29, 2018. Citado na página 18.

- Suyu. Two accurate time-delay distances from strong lensing: Implications for cosmology. *The Astrophysical Journal*, IOP Publishing, v. 766, n. 2, p. 70, 2013. Citado 2 vezes nas páginas 12 e 17.
- Sygnnet, J.-F. et al. A search for edge-on galaxy lenses in the cfht legacy survey. *Astronomy & Astrophysics*, EDP Sciences, v. 517, p. A25, 2010. Citado na página 18.
- Tessore. Lensed: a code for the forward reconstruction of lenses and sources from strong lensing observations. *Monthly Notices of the Royal Astronomical Society*, The Royal Astronomical Society, v. 463, n. 3, p. 3115–3128, 2016. Citado 2 vezes nas páginas 12 e 17.
- TowardsDataScience. 2019. Disponível em: <<https://towardsdatascience.com/mnist-handwritten-digits-classification-using-a-convolutional-neural-network-cnn-af5fafbc35e9>>. Acesso em: 10 jun. 2020. Citado na página 29.
- Treu. Time delay cosmography. *The Astronomy and Astrophysics Review*, Springer, v. 24, n. 1, p. 11, 2016. Citado 2 vezes nas páginas 12 e 17.
- Vegetti. Bayesian strong gravitational-lens modelling on adaptive grids: objective detection of mass substructure in galaxies. *Monthly Notices of the Royal Astronomical Society*, Blackwell Publishing Ltd Oxford, UK, v. 392, n. 3, p. 945–963, 2009. Citado 2 vezes nas páginas 12 e 17.
- Walsh. 0957+ 561 a, b: twin quasistellar objects or gravitational lens? *Nature*, Springer, v. 279, n. 5712, p. 381–384, 1979. Citado na página 18.
- Wambsganss, J. Gravitational lensing in astronomy. *Living Reviews in Relativity*, Springer, v. 1, n. 1, p. 12, 1998. Citado na página 18.
- Willis. The ols-lens survey: the discovery of five new galaxy–galaxy strong lenses from the sdss. *Monthly Notices of the Royal Astronomical Society*, Blackwell Publishing Ltd Oxford, UK, v. 369, n. 3, p. 1521–1528, 2006. Citado na página 18.
- Witt. Analytic time delays and h_0 estimates for gravitational lenses. *The Astrophysical Journal*, IOP Publishing, v. 544, n. 1, p. 98, 2000. Citado na página 17.