dx.doi.org/10.7437/NT2236-7640/2021.02.002
Notas Técnicas, v. 11, n. 2, p. 1–18, 2021

Classificação Morfológica de Galáxias no S-PLUS por Combinação de Redes Convolucionais

Morphological Classification of Galaxies in S-PLUS using an Ensemble of Convolutional Networks

N. M. Cardoso[1],* G. B. Oliveira Schwarz[2],† L. O. Dias[3], C. R. Bom[3,4], L. Sodré Jr.[5], e C. Mendes de Oliveira[5]
[1]*Escola Politécnica, Universidade de São Paulo, Av. Prof. Luciano Gualberto, 380, Butantã, São Paulo – SP, CEP 05508-010, Brasil*
[2]*Universidade Presbiteriana Mackenzie, Rua da Consolação, 930, Consolação, São Paulo – SP, CEP 01302-907, Brasil*
[3]*Centro Brasileiro de Pesquisas Físicas, Rua Dr. Xavier Sigaud, 150, Urca, Rio de Janeiro – RJ, CEP 22290-180, Brasil*
[4] *Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Rodovia Mário Covas, lote J2, quadra J Distrito Industrial de Itaguaí, Itaguaí – RJ. CEP: 23810-000, Brasil e*
[5]*Departamento de Astronomia, Instituto de Astronomia, Geofísica e Ciências Atmosféricas da USP, Cidade Universitária, São Paulo – SP, CEP 05508-900, Brasil*

Abstract: The universe is composed of galaxies that have diverse shapes. Once the structure of a galaxy is determined, it is possible to obtain important information about its formation and evolution. Morphologically classifying galaxies means cataloging them according to their visual appearance and the classification is linked to the physical properties of the galaxy. A morphological classification made through visual inspection is subject to biases introduced by subjective observations made by human volunteers. For this reason, systematic, objective and easily reproducible classification of galaxies has been gaining importance since the astronomer Edwin Hubble created his famous classification method. In this work, we combine accurate visual classifications of the Galaxy Zoo project with *Deep Learning* methods. The goal is to find an efficient technique at human performance level classification, but in a systematic and automatic way, for classification of elliptical and spiral galaxies. For this, a neural network model was created through an Ensemble of four other convolutional models, allowing a greater accuracy in the classification than what would be obtained with any one individual. Details of the individual models and improvements made are also described. The present work is entirely based on the analysis of images (not parameter tables) from DR1 (www.datalab.noao.edu) of the Southern Photometric Local Universe Survey (S-PLUS). In terms of classification, we achieved, with the Ensemble, an accuracy of $\approx 99\%$ in the test sample (using pre-trained networks).

Keywords: Galaxy Morphology, Galaxy Classification, Computer Vision, Deep learning, Convolutional Neural Networks.

Resumo: O universo é composto de galáxias que apresentam variadas formas. Uma vez determinada a estrutura de uma galáxia, é possível obter informações importantes desde sua formação até sua evolução. A classificação morfológica é a catalogação de galáxias de acordo com a sua aparência visual e a classificação está ligada com as propriedades físicas da galáxia. Uma classificação morfológica feita através de inspeção visual está sujeita a um viés causado pela subjetividade da observação humana. Por isso, a classificação sistemática, objetiva e facilmente reproduzível de galáxias vem ganhando importância desde quando o astrônomo Edwin Hubble criou seu famoso método de classificação. Neste trabalho, nós combinamos classificações visuais acuradas do projeto Galaxy Zoo com métodos de *Deep Learning*. O objetivo é encontrar uma técnica eficiente que consiga simular a classificação visual humana, mas de forma sistematizada e automática, para classificação de galáxias elípticas e espirais. Para isto, um modelo de rede neural foi criado através de um Ensemble de outros quatro modelos convolucionais, possibilitando uma maior acurácia na classificação do que o que seria obtido com qualquer um individualmente. Detalhes dos modelos individuais e melhorias feitas nestes também são descritas. O presente trabalho é totalmente baseado na análise de imagens (não tabelas de parâmetros) do DR1 (www.datalab.noao.edu) do Southern Photometric Local Universe Survey (S-PLUS). Em termos de classificação, alcançamos, com o Ensemble, uma precisão de $\approx 99\%$ na amostra de teste (usando redes pre-treinadas).

Palavras chave: Morfologia de Galáxias, Classificação de Galáxias, Visão Computacional, Aprendizagem Profunda, Redes Neurais Convolucionais.

1. INTRODUÇÃO

Classificação morfológica é a categorização das galáxias conforme sua forma. Quando esta classificação é baseada na inspeção visual das imagens, elementos subjetivos são

*Electronic address: nauxmac@gmail.com
†Electronic address: gustavo.b.schwarz@gmail.com

agregados. Em 1926, o astrônomo Edwin Hubble, na tentativa de relacionar as formas das galáxias com sua origem e evolução, criou um método hoje conhecido como *Hubble Sequence* ou *Tunning Fork* [1, 2], que é uma tentativa de atribuir classes discretas às galáxias, de acordo com suas formas. Esta classificação, com algumas pequenas modificações e adições, ainda é usada até hoje. No *Tunning Fork*, as galáxias são classificadas como elípticas, espirais ou lenticulares, mas as formas predominantes de grandes galáxias na natureza são elípticas e espirais [2], pois acredita-se que a classe das lenticulares seja uma classe de transição. Lenticulares são muitas vezes classificadas como galáxias elípticas (mais comumente) ou espirais (menos comum). Com isto, foram criadas as classes "early-type", contendo as elípticas e lenticulares e "late-type", contendo as espirais e outras galáxias de tipo mais tardio ainda, chamadas de irregulares, que só foram incluídas no sistema de classificação muitos anos mais tarde.

O final do século 20 conheceu uma revolução na maneira de se estudar galáxias na Astronomia quando os primeiros mapeamentos de grandes áreas do céu começaram a ser feitos. O mapeamento que mais impactou a Astronomia nas últimas décadas foi o chamado SDSS[1].

Um programa que envolveu o SDSS e milhões de cidadãos comuns (chamado, em inglês, de projeto *citizen science*) foi o chamado GalaxyZoo[2], um projeto realizado em sua maioria por cidadãos sem vínculo acadêmico, que contribuíram com suas observações para a classificação de um grande número de galáxias do SDSS. A segunda liberação de dados do GalaxyZoo possui um catálogo com classificações morfológicas de trezentas mil galáxias, revisadas segundo o método de Hart et al. [3]. Uma subamostra destes dados, que coincide com o chamado *Stripe-82*[3], foi utilizada neste trabalho como *true table* na classificação de galáxias elípticas e espirais.

Com o avanço dos levantamentos (*surveys*) digitais e conseguente aumento da quantidade de dados coletados, se torna crucial o desenvolvimento de métodos rápidos e automatizados para a classificação morfológica de galáxias sem a perda da acurácia da tradicional classificação visual [4]. O uso de aprendizado de máquina e, mais recentemente *Deep Learning* tem mostrado resultados relevantes para problemas de classificação em diversos problemas nas áreas de visão computacional e astronomia, dentre outras.

O *Deep Learning* [5] é uma segmento específico dentro da área de aprendizado de máquina e, por conseguinte, da área de Inteligência Artificial. Consiste no desenvolvimento de redes neurais artificiais que são combinadas em um número significativamente maior do que as redes neurais tradicionais. Este tipo de técnica se transformou no estado-da-arte do reconhecimento de padrões em imagens devido a um tipo específico de rede neural conhecida como convolucional. As redes neurais convolucionais ou CNNs da sigla em inglês *Convolutional Neural Networks* [6], são inspiradas e propostas com certa analogia ao processamento das imagens realizadas no córtex visual de mamíferos. O processo começa quando um estímulo visual alcança a retina e equivale a um sinal que atravessa regiões específicas do cérebro. Essas regiões são responsáveis pelo reconhecimento de cada uma dessas características correspondentes [7]. Os neurônios biológicos das primeiras regiões respondem pela identificação de formatos geométricos primários, enquanto neurônios das camadas finais têm a função de detectar características mais complexas, formadas pelas formas simples anteriormente reconhecidas [7, 8]. Características com padrões muito específicos do objeto são estabelecidas depois que o procedimento se repete. De forma análoga, a CNN decompõe a tarefa de reconhecimento de um objeto em subtarefas. Para isso, durante a aprendizagem, a CNN divide a tarefa em subníveis de representação das características, posteriormente aprendendo a reconhecer novas amostras da mesma classe [6, 8]. Desta forma, as CNNs são capazes de predizer características complexas sem a necessidade de um pré processamento e são invariantes â escala e à rotação dos dados, o que torna essencial a classificação em imagens.

Este trabalho utilizou dados do S-PLUS para classificar imagens. O S-PLUS [9] é um levantamento de galáxias do Universo Local, liderado por brasileiros, feito com um telescópio de 0.8m e com uma câmera de grande campo, localizado no Chile. A parte do mapeamento que cobre a região do chamado *Stripe-82* é uma área de grande interesse dado que é coberta por diversos projetos, permitindo assim comparações e análises complementares. O S-PLUS cobriu a região com medidas de fluxo (magnitudes) em 12 bandas para três milhões de fontes (liberadas para a comunidade internacional no DR1, [9]).

Neste artigo, a Seção 2 versa sobre a preparação dos dados. A Seção 3 apresenta as seis redes com suas performances e seus respectivos hiperparâmetros. Quatro das redes com melhor performance são então escolhidas para a fase seguinte do trabalho, para construção de um meta-modelo, com os resultados apresentados na Seção 4. Mostramos que as características das galáxias sem prévia classificação, classificadas com nosso método, apresentam as cores esperadas, indicativo de uma classificação robusta. Finalmente, a Seção 5 apresenta a discussão dos resultados. Comparações são feitas entre os nossos resultados e os obtidos através de classificações feitas por outros autores.

2. CONJUNTO DE DADOS

Neste trabalho vamos apresentar uma técnica eficiente e automatizada para a classificação morfológica de galáxias usando *Deep Learning*. Para isso, primeiramente serão apresentados os dados utilizados e como os preparamos.

2.1. Aquisição dos dados

A imagem da galáxia, com sua respectiva classificação morfológica, é elemento crucial para se fazer o treinamento supervisionado de nossa Rede Neural Artificial. Para treinar a rede, de modo que esta aprenda a classificar galáxias através das imagens, é necessário fazê-la aprender as for-

[1] SDSS: Sloan Digital Sky Survey – `https://www.sdss.org`.
[2] `https://galaxyzoo.org`
[3] Este é um campo equatorial do céu de 336 graus^2, que cobre a região com ascensão reta das 20:00h às 4:00h e declinação de -1,26° a +1,26°

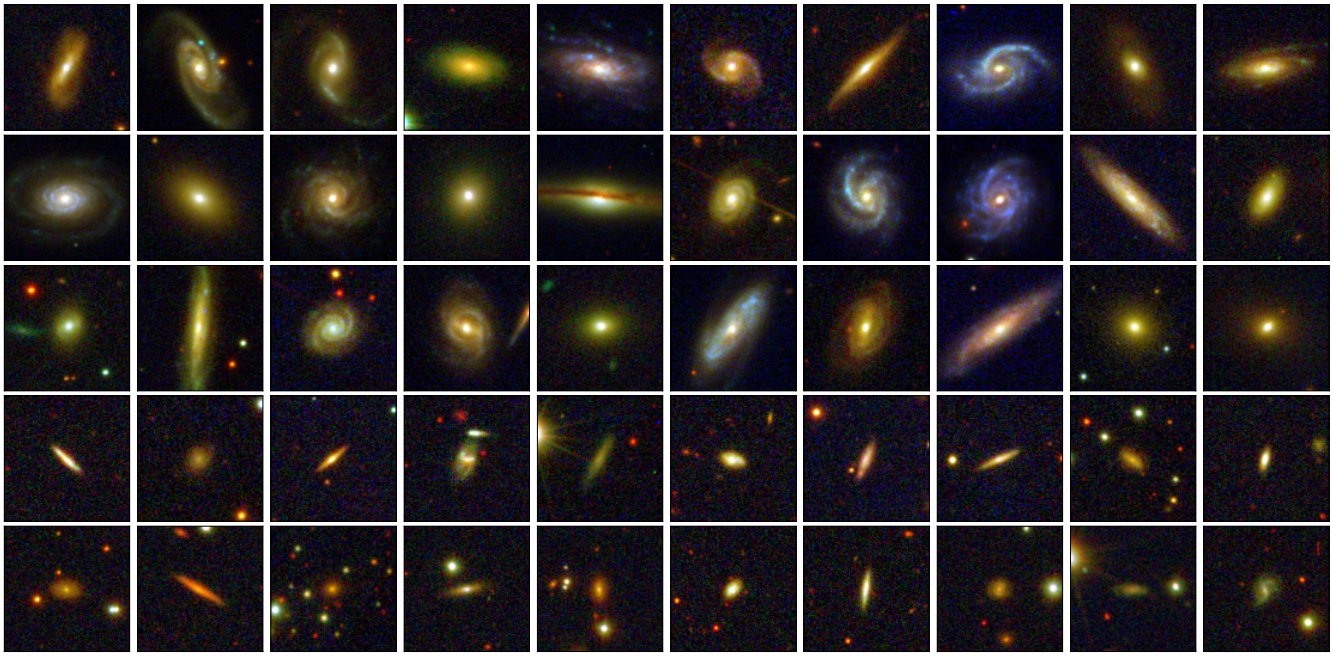


Figura 1: Exemplos de galáxias utilizadas nos conjuntos de treinamento, validação e teste, coloridas usando o método descrito na Seção 3.1. As primeiras três linhas mostram galáxias com $r_{auto} < 17$ e as últimas duas mostram galáxias no intevalo de magnitude $17 < r_{auto} < 17.5$. Note a diferença entre as imagens de diferentes magnitudes, na Seção 4 será mostrado que esta diferença tem impacto na performance do modelo.

mas e padrões das galáxias. No nosso caso, utilizamos uma grande amostra de imagens de galáxias já classificadas visualmente por humanos.

Os dados aqui utilizados são as imagens das galáxias do levantamento S-PLUS e as classificações morfológicas do GalaxyZoo, que separam as galáxias entre espiral e elíptica. A associação destes dois conjuntos de dados é feita pela correlação das coordenadas do objeto no espaço. Ademais, apenas galáxias no S-PLUS com magnitudes r_{auto} menores que 17.5 foram utilizadas. O valor de r_{auto}, que é dado em uma das colunas do catálogo do S-PLUS, representa aproximadamente o fluxo total de uma dada galáxia. Mais adiante, na Seção 4, faremos uma análise usando dois conjuntos de dados, um com galáxias com $r_{auto} < 17$ e outro com galáxias com $r_{auto} < 17.5$, para mostrar a importância do limite de magnitude (r_{auto}) nos resultados. O número de galáxias em cada um destes conjuntos é dado na tabela I. As imagens do S-PLUS foram obtidas através do banco de dados do projeto[4]. A Figura 1 mostra exemplos d imagens RGB do S-PLUS utilizadas neste trabalho.

2.2. Divisão do conjunto de dados

Todos os modelos usam os mesmos subconjuntos de dados para garantir que o desempenho de classificação do modelo não seja enviesado pela escolha dos lotes.

A distribuição de galáxias para cada subconjunto foi de 81% para treinamento, 9% para validação e 10% para teste. A proporção de galáxias elípticas e espirais entre os subconjuntos de treinamento, validação e teste é a mesma. Os conjuntos são constituídos por, aproximadamente, 68% de galáxias espirais e 32% de galáxias elípticas. A tabela I mostra a quantidade de galáxias em cada subconjunto.

Tabela I: Quantidade de galáxias em cada conjunto de dados.

r_{auto}	Treinamento	Validação	Teste	Total
< 17	2231	248	276	2757
< 17.5	3349	373	414	4136

2.3. Comparação dos conjuntos de dados

Para o desenvolvimento do trabalho, dividimos as amostras em subconjuntos distintos, com o objetivo final de fazer uma classificação de galáxias elípticas e espirais numa amostra ainda não classificada, denominada amostra *blind*. Nesta seção, mostramos que as amostras de galáxias utilizadas nos conjuntos de treinamento, validação, teste e *blind* têm distribuições equilibradas de medidas de brilho, determinadas através das magnitudes na banda r (que é a banda com maior sinal/ruído) e redshifts (desvios para o vermelho, que são proporcionais às distâncias dos objetos). Isto é importante para que a comparação dos diagramas cor-cor que serão mostrados na Figura 11, na Seção 4 faça sentido. A Figura 2 mostra o histograma de magnitude das galáxias para

[4] https://splus.cloud

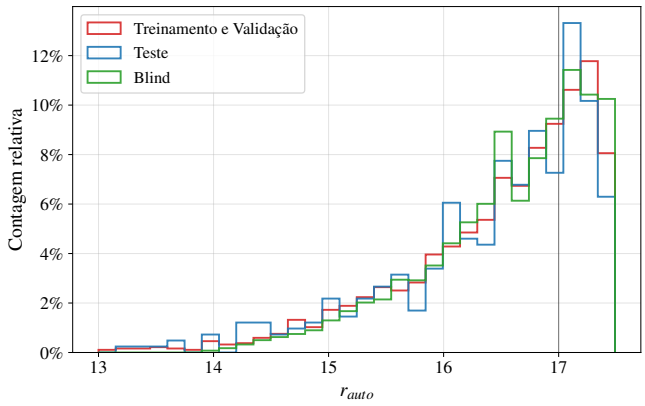


Figura 2: Histograma com a contagem relativa dos valores de r_{auto} para as amostras de treinamento e validação, teste e blind.

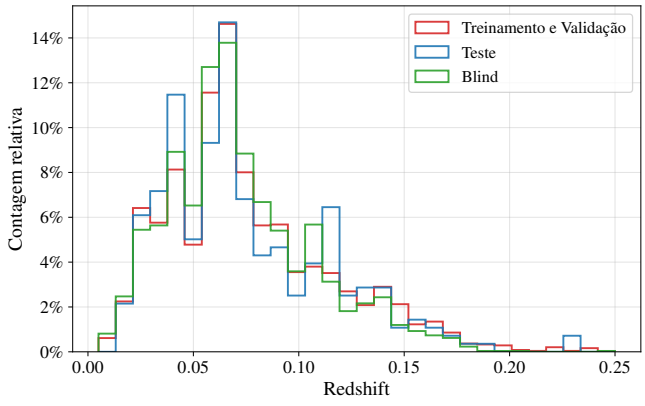


Figura 3: Histograma com a contagem relativa dos valores de redshift para as amostras de treinamento e validação, teste e blind para $r_{auto} < 17$.

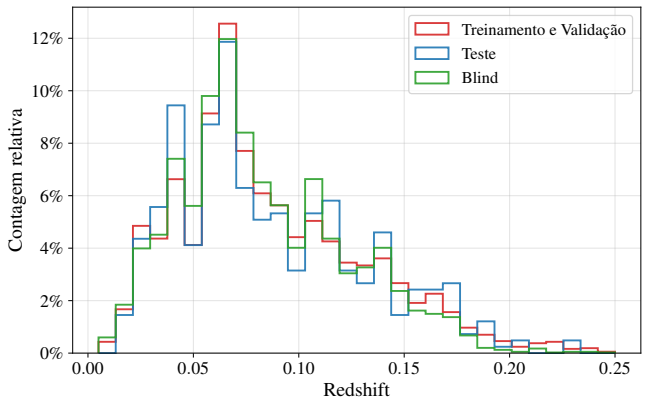


Figura 4: Histograma com a contagem relativa dos valores de redshift para as amostras de treinamento e validação, teste e blind para $r_{auto} < 17.5$.

valores de r_{auto} entre 13 e 17.5. Uma linha em 17 mostra que galáxias à direita estão presentes somente no conjunto $r_{auto} < 17.5$. As Figuras 3 e 4 mostram as distribuições de redshift para os conjuntos $r_{auto} < 17$ e $r_{auto} < 17.5$, respectivamente.

2.4. Conjunto de dados desbalanceado

Como visto na Seção 2.2, aproximadamente 68% das galáxias do conjunto de dados são espirais. Contudo, o desempenho dos algorítmos de *machine learning* são afetados negativamente pela quantidade desproporcional de objetos entres as classes. Algumas técnicas testadas para melhorar o desempenho da rede são listadas abaixo.

Subamostragem aleatória

A subamostragem aleatória, *random undersampling*, é a retirada aleatória de objetos do conjunto de trainamento pertencentes à classe com maior quantidade de elementos até que a proporção de objetos entre as classes fique equilibrada ($\approx 1:1$).

Sobreamostragem aleatória

Ao contrário da subamostragem, a sobreamostragem aleatória, *random oversampling*, é a replicação de elementos da classe em minoria até que a proproção de objetos entre as classes fique equilibrada.

Ponderamento das classes

Ao contrário das técnicas anteriores, o ponderamento das classes (*class weight*) não é uma técnica de reamostragem. Ela consiste na atribuição de pesos a cada classe, proporcionais a sua quantidade de elementos. O peso w_i da i-ésima classe tem o valor dado pela equação (1), baseada na heurística apresentada em [10].

$$w_i = \frac{Q}{N \times C_i} \tag{1}$$

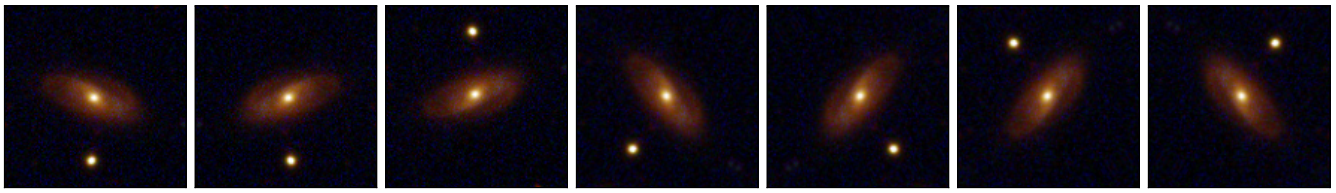
onde, Q é a quantidade total de objetos, N é o número de classes e C_i é o número de objetos da i-ésima classe.

3. MÉTODOS DE DEEP LEARNING

Nesta seção, explicamos sobre a preparação dos dados e como fizemos o aumento artificial dos dados para obter melhores resultados na avaliação dos modelos. Em seguida, descrevemos as redes convolucionais utilizadas: VGG, Inception Resnet, EfficientNet e DenseNet. Introduzimos o conceito de (*Ensemble*) e descrevemos as técnicas usadas anteriormente e que fundamentaram nossas escolhas. Em seguida, apresentamos as principais definições das redes e parâmetros utilizados neste trabalho e por fim detalhamos como foram feitas as modelagens e treinamentos dos classificadores e do nosso meta-modelo.

3.1. Preparação dos dados

O pré-processamento é a preparação das imagens para serem usadas pelo modelo, ou seja, é a transformação dos dados não processados em dados prontos para entrada na rede. Isso envolve representar as imagens por matrizes multidimensionais, onde cada elemento da matriz representa um pixel da imagem, e aplicar algumas transformações, especificadas a seguir.



(a) (b) (c) (d) (e) (f) (g)

Figura 5: Exemplo do aumento artificial de dados em uma imagem original, mostrada no painel (a). Os painéis (b), (c), (d), (e), (f) e (g) contêm os resultados da equação (3) substituindo M por diferentes combinações das transformações da equação (2). Em (b) $M = V$, em (c) $M = H$, em (d) $M = R(30°)$, em (e) $M = VR(30°)$, em (f) $M = HR(30°)$ e em (g) $M = HVR(30°)$.

3.1.1. Agrupamento das bandas para confecção das imagens RGB

Como as imagens do S-PLUS foram obtidas em 12 bandas fotométricas (listadas em [9]), para representá-las no sistema de cor RGB fizemos o seguinte mapeamento: em R colocamos as 4 bandas vermelhas r_SDSS, i_SDSS, J0861 e z, em G as bandas g_SDSS J0515 e J0660 e em B as cinco bandas mais azuis u_JAVA, J0373, J0395, J0410 e J0430 (as características destes filtros são dadas na Tabela 1 de [9]). Na combinação de bandas em cada canal, foi feita uma soma simples dos valores dos píxeis. Depois de reduzidas a três bandas, as imagens são usadas como entrada do programa Trilogy[11][5].

3.1.2. ImageNet

Como já mostrado em trabalhos anteriores, a inicialização dos pesos provenientes de uma rede pré-treinada usando a base de dados *ImageNet*[6] traz uma grande melhoria na precisão dos resultados da classificação. Essa base de dados possui milhões de imagens de objetos do cotidiano e já foi utilizada especificamente para a classificação de objetos astronômicos (veja por exemplo [12]), com excelentes resultados.

O uso deste dataset para pré-treinamento respeitou o pré-processamento utilizado originalmente pelos autores de cada rede, este procedimento este foi crucial para garantir um fit competitivo, isto é, no *benchmarking* original destas redes para os dados da *ImageNet*. Para a rede VGG16 (Seção 3.3), a ordem das bandas foi trocada de RGB para BGR, e cada banda foi centrada em zero em relação à *ImageNet*, sem escalonamento, ou seja, os píxeis de cada banda tiveram o valor da média da respectiva banda *ImageNet* subraído. Para a rede InceptionResNetV2 (Seção 3.4), os píxeis de entrada foram escalonados entre -1 e 1 em relação a amostra de treino. Para a rede EfficientNet (Seção 3.5), os píxeis de entrada foram escalonados entre 0 e 1 em relação à amostra de treino. E, para a rede DenseNet (Seção 3.6), os píxeis de entrada foram escalonados entre 0 e 1 e cada banda foi padronizada em relação à *ImageNet*, isto é, os píxeis de cada banda tiveram o valor da média subtraído e o resultado foi dividido pelo desvio padrão da distribuição da respectiva banda da *ImageNet*.

[5] https://www.stsci.edu/~dcoe/trilogy/Intro.html
[6] http://www.image-net.org/

3.2. Aumento artificial de dados

Aumento artificial de dados [13] é a aplicação de transformações afins nas imagens do conjunto de treinamento, por exemplo rotação, reflexão, translação e mudança de escala. As matrizes da equação (2) definem as transformações usadas.

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad V = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

onde $R(\theta)$ é a transformação rotação por um ângulo θ, H é a transformação reflexão horizontal e V é a transformação reflexão vertical.

Seja M a matriz das transformações combinadas, (x,y) a coordenada do píxel da imagem original e (x^*,y^*) a coordenada transformada do píxel, as transformações nas imagens são feitas remapeando as coordenadas dos píxeis originais aplicando uma combinação das matrizes da equação (2) em cada píxel da imagem original usando a equação (3), onde (t_x,t_y) é a coordenada do centro da imagem e as matrizes ao redor de M são as matrizes translação. Isso é feito para que a transformação M tenha o centro da imagem como ponto de simetria.

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} M \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3}$$

Além disso, ainda é aplicada uma interpolação bilinear como *anti-aliasing* [14, 15]. Durante o treinamento da rede, novas imagens de entrada são geradas a cada época a partir da transformação das imagens originais. A Figura 5 mostra a imagem original, no painel (a), e diversas transformações, nos demais painéis, aplicadas substituindo M da equação (3) por combinações (multiplicação matricial) das transformações da equação (2). Tais transformações não

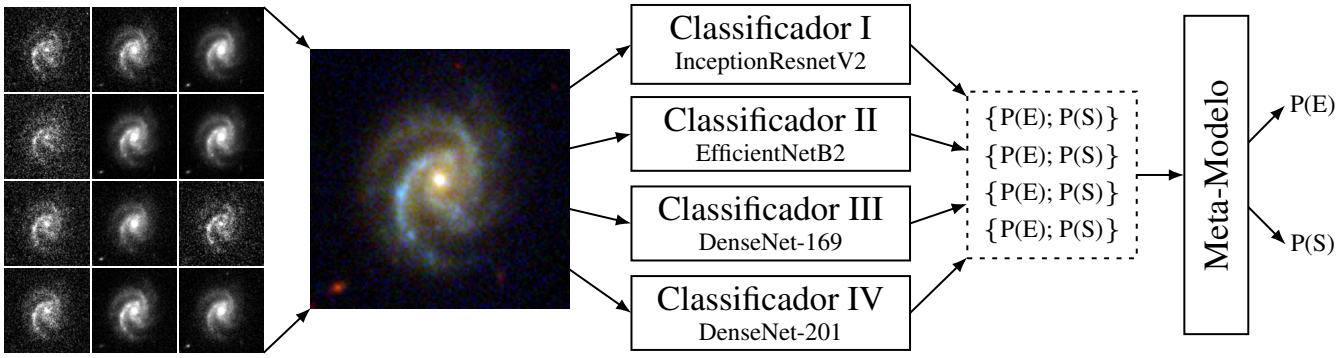


Figura 6: Diagrama que descreve a arquitetura da rede. Da esquerda para a direita, as 12 imagens de cada banda são agrupadas em uma única imagem RGB (Seção 3.1), que é a entrada dos classificadores indidviduais. Estes classificadores (Seção 3.9) têm a função de extrair características visuais da imagem e retornar a probabilidade de ser elíptica ou espiral. O meta-modelo (Seção 3.10) tem a função de combinar as predições dos classificadores em uma única predição final mais robusta.

mudam a interpretação da classe da imagem original, pois o espaço visual é invariante a elas. Logo, o objetivo de aplicar estas transformações nas imagens de entrada da rede é deixar que o algorítmo infira tal invariância, criando, assim, uma "noção" do espaço visual, o que resulta no aumento do potencial de generalização da rede [16, 17]. Frequentemente são relatados bons resultados com o uso desta técnica [18–20], principalmente quando existe grande similaridade entre as classes.

3.3. VGG

A arquitetura VGG [21] foi criada durante a competição de classificação de imagens *Large Scale Visual Recognition Challenge* [22]. Ela se destaca por estar entre as primeiras redes a adotar, com sucesso, o escalamento em profundidade (quantidade de camadas) para aumentar o desempenho na classificação de imagens usando redes convolucionais. Ela já foi usada em diversas tarefas de classificação, como a classificação de software malicioso [23], de plantas [24] e de tumores cerebrais [25].

3.4. InceptionResNetV2

A arquitetura InceptionResNetV2 [26] usa os blocos Inception, que são convoluções fatorizadas, introduzidos em [27], motivada pela construção de redes mais profundas com um menor custo computacional e menor overfitting, com a adição de conexões residuais [28] motivada pelo problema de dissipação do Gradiente (*vanishing gradients*). Isso permite treinar redes profundas com maior acurácia e mais rápido. Esta arquitetura já foi usada, por exemplo, para classificação de imagens de satélite [29], de ultrasonografia [30] e de células cancerígenas [31].

3.5. EfficientNet

A arquitetura EfficientNet [32] foi desenvolvida como uma resposta à questão de como escalar modelos de convolução. Foram considerados três diferentes aspectos: profundidade, largura e resolução da imagem de entrada. Em vez de dimensionar cada aspecto manualmente, o modelo implementa um escalonamento composto que equilibra os aspectos para obter melhor desempenho, com isso a rede consegue uma alta acurácia usando muito menos parâmtros e operações de ponto flutuante por segundo (*FLOPS*). Esta rede já foi usada na classificação de doenças em vegetais [33], eletrocardiogramas [18] e cristalização de proteínas [19].

3.6. DenseNet

A rede DenseNet [34] também usa conexões residuais que conectam cada camada a todas as outras camadas seguintes, o que reduz ainda mais o número de parâmetros na rede sem perda significativa da precisão. O uso desta rede incluem predição do mapa de contato de proteínas [35], classificação de músicas [36], câncer de mama [37] e esclerose múltipla [38].

3.7. Ensemble

Abaixo mostraremos como podemos combinar os resultados das redes descritas na última seção, usando um método de *Ensemble*, que combina os vários modelos treinados para resolver um mesmo problema. Essse ensemble foi feito inspirado em [39], que mostraram uma grande melhoria nos resultados quando várias redes são combinadas numa regressão logística. O uso de Ensemble tem se mostrado uma importante técnica no desenvolvimento de modelos de *machine learning* ainda nos anos 90, quando Hansen & Salamon [40] mostraram que as predições feitas pela combinação de um conjunto de classificadores são frequentemente mais precisas do que as feitas somente pelo melhor classificador.

Existem vários relatos de sucesso do uso desta técnica em *deep learning*, dentre eles a classificação de retina humana [41], melanoma [42] e de anomalias na Via Láctea [43]. Existem várias técnicas de agrupamento (*Ensemble*), como *Boosting* [44, 45], *Bagging* [46] e *Stacking* [47–49]. Este último foi o escolhido para ser usado neste trabalho. Ele se diferencia dos demais pela presença de um meta-modelo, que recebe as predições dos classificadores – treinados individualmente – e retorna uma predição final, como é mostrado na Figura 6. A Figura 6 mostra a arquitetura do *Ensemble* a partir do processo de inferência de uma imagem. Pelo diagrama, é possível notar que o modelo é composto de duas camadas de redes neurais artificiais, a primeira é composta pelos classificadores e a segunda é composta pelo meta-modelo. As Seções 3.9 e 3.10 detalham o desenvolvimento da primeira e segunda camadas respectivamente.

Uma outra etapa importante no desenvolvimento de redes neurais artificiais é o ajuste dos hiper-parâmetros, alguns deles mostrados na Seção 3.8.

3.8. Definições das redes e parâmetros utilizados

Nesta seção descrevemos as definições dos principais conceitos, no contexto de deep learning, que serão úteis para o entendimento dos métodos aqui utilizados. A função de ativação, função de custo, o otimizador, o learning rate, o número de épocas, além do número de camadas dos modelos, são importantes parâmetros responsáveis pela contrução do modelo definido a seguir.

Função de ativação

A função de ativação é responsável por adicionar não-linearidade à rede. Sem ela, a saída de uma camada seria apenas uma transformação linear dos dados de entrada e a rede não seria beneficiada pelo empilhamento de diversas camadas lineares, pois isso não aumentaria o espaço de hipóteses. Logo, a função de ativação viabiliza representações mais complexas da rede, uma vez que define a complexidade de um modelo e, consequentemente, sua capacidade de generalização [17]. Neste trabalho, a função $\mathrm{ReLU}(x) := \max(0, x)$ é usada nas camadas densas dos classificadores, a equação tangente hiperbólica é usada nas camadas densas do meta-modelo e a função Softmax [50] foi usada na última camada, tanto dos classificadores quanto do meta-modelo.

Função de Custo

A função de custo é utilizada com o objetivo de determinar o quão longe o modelo está do esperado, definindo a necessidade de atualização dos pesos da rede. Utilizamos a função Entropia Cruzada (*Cross-Entropy*)

$$J = \sum_{i=1}^{C} y_i \cdot \log(\hat{y}_i) \tag{4}$$

onde y_i representa a probabilidade da classe dada pelo conjunto de treinamento do objeto i e $\hat{y}_i$ representa a previsão da rede para este mesmo objeto.

Otimizador

O otimizador é um algorítmo iterativo com objetivo de minimizar a função de custo. Uma escolha típica é o método de gradiente descente e suas demais variações. Este tipo de algoritmo tem um parâmetro livre relacionado ao passo da iteração conhecido como taxa de aprendizado ou *learning rate*. Neste trabalho foram testados diversos algoritmos considerados como estado-da-arte dos otimizadores como Adam [51], NAdam [52], RAdam [53] e RMSprop [54].

Número de Épocas

O Número de épocas se referem a quantidade de vezes que o dataset de treino foi utilizado completamente no processo de otimização iterativa da função de custo. Um número de épocas adequado é necessário para que a função de custo seja minimizada.

Tamanho do Batch

O processo de otimização acontece em batches, cada iteração para minimizar a função custo é realizada com um número fixo de amostras, quando todas as amostras de treino são utilizadas se completa uma época.

Unidades de neurônios na última camada

A última camada da rede antes da camada de saída é responsável por condensar toda a informação extraída da rede para o processo de classificação final. Por esta razão, a quantidade de neurônios nessa camada pode ser particularmente sensível para a performance da rede. Neste trabalho utilizamos diferentes valores de neurônios para encontrar a quantidade que pode gerar a melhor performance.

Dropout

Dropout [55] é uma técnica de regularização muito utilizada em redes neurais por seu bom desempenho e baixo custo computacional. Aplicar esta regularização em uma camada consiste em eliminar aleatoriamente uma taxa dos neurônios de saída desta camada durante o treinamento, sendo geralmente escolhido um valor entre 0.2 e 0.5 para esta taxa [17].

3.9. Modelagem e Treinamento dos Classificadores

A estrutura de cada classificador é composta por um extrator de características ligado a um bloco de camadas densamente conectadas, também chamado de bloco de predição. O extrator de características é uma rede neural convolucional usada como camada de abstração das características visuais, detectando padrões como geometrias, contrastes, texturas e cores na imagem de entrada. Enquanto que o bloco de predição recebe estas característcas e retornam as probabilidades da imagem pertencer à classe elíptica e espiral.

Para o extrator de características, foram testadas arquiteturas conhecidas de redes convolucionais descritas anteriormente: InceptionResNet, EfficientNet, DenseNet e VGG. Cada uma destas arquiteturas foi pré-treinada usando a base de dados *ImageNet*.

Para as camadas densas, foram testadas várias configurações de escalamento, tanto em largura (quan-

Tabela II: Variação dos parâmetros dos classificadores com arquitetura InceptionResNetV2 usando *oversample*.

Parâmetro	Valor	ROC AUC	PR AUC
Otimizador	Adam	0.983551	0.984164
	RAdam	0.981333	0.982089
	NAdam	**0.985507**	**0.986090**
	RMSprop	0.976397	0.977327
Learning Rate	$5 \cdot 10^{-6}$	0.970411	0.971590
	$1 \cdot 10^{-5}$	0.978090	0.978931
	$5 \cdot 10^{-5}$	**0.986820**	**0.987339**
	$1 \cdot 10^{-4}$	0.986380	0.986919
	$5 \cdot 10^{-4}$	0.985809	0.986337
	$1 \cdot 10^{-3}$	0.981214	0.979992
	$5 \cdot 10^{-3}$	0.977913	0.976907
Batch Size	32	0.974966	0.972293
	64	0.981530	0.980156
	128	0.976305	0.974936
	192	**0.985100**	**0.985659**
	256	0.983728	0.984385
Unidades	128/2	0.977434	0.978307
	256/2	0.979423	0.980226
	512/2	**0.986420**	**0.986917**
	1024/2	0.981792	0.982526
	256/128/2	0.979862	0.980575
	1024/256/128/2	0.982252	0.982909

tidade de neurônios) quanto em profundidade (quantidade de camadas). As configurações testadas consistem de m camadas ocultas com n unidades de neurônios (m variando de 0 a 3 e $n = 2^t$, com t variando de 6 a 10) ligadas a uma camada final com 2 unidades de neurônios e função de ativação *Softmax*. Cada unidade desta última camada representa a probabilidade do objeto pertencer a cada classe [5], i.e., elíptica e espiral. Por isso o número de neurônios é igual ao número de classes e suas configurações não foram variadas.

Além disso, as camadas ocultas foram regularizadas com *dropout*, que garante uma melhora na capacidade de generalização da rede [56]. A avaliação dos modelos nos conjuntos de treinamento e de validação mostraram que a melhor taxa de *dropout* é 0.4, superando a avalição das redes sem regularizador e das redes com outras taxas de *dropout* no intervalo entre 0.1 e 0.6. Do mesmo modo, a função de ativação *ReLU* garantiu as melhores avaliações dos modelos quando comparadas com outras funções de ativação, como tanh e *ELU*.

Outras configurações da rede também foram testadas, tanto relacionadas aos dados, como a amostragem, quanto relacionadas ao treinamento, como o tamanho do *batch*, o algorítmo de otimização e a sua taxa de aprendizagem. Com a variação de um parâmetro por vez, é possível detectar os valores que contribuem para a melhor avaliação da rede, como visto na Tabela II. Nesta tabela, é mostrado que um modelo treinado usando o otimizador NAdam com uma taxa de aprendizagem de $5 \cdot 10^{-5}$ e 192 exemplares por *batch*, além de um bloco de predição contendo uma camada de 512 unidades de neurônios seguida de uma camada com 2 unidades, obteve a melhor avaliação no conjunto de validação em relação às outras configurações. Isso mostra como foram determinados os parâmetros do Modelo A da Tabela III, mostrada na Seção 4. Os parâmetros para os demais modelos foram obtidos analogamente.

3.10. Modelagem e Treinamento do Meta-Modelo

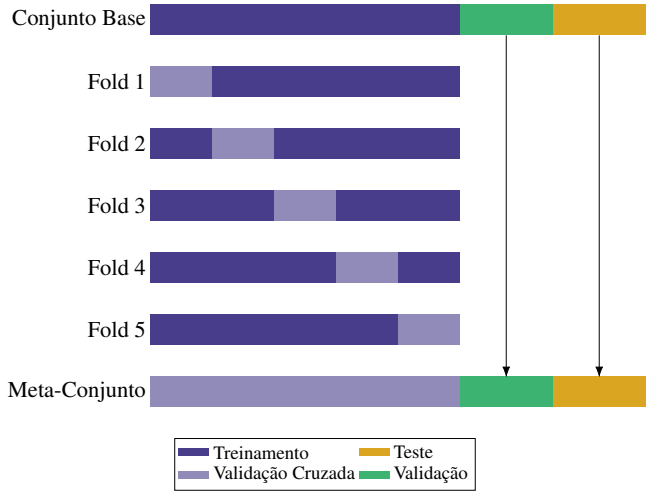


Figura 7: O diagrama representa a construção do conjunto de treinameto do meta-modelo (meta-conjunto) usado o método de validação cruzada *k-fold*.

O objetivo do meta-modelo é receber as predições de cada um dos classificadores e retornar uma predição final. Para isso, foi usada uma rede neural com estrutura composta por uma camada de entrada de 8 unidades de neurônios seguidas de duas camadas ocultas de 128 unidades e uma camada de saída com 2 unidades. Assim como nos classificadores, diversas configurações de hiperparâmetros foram testadas para o meta-modelo. Sendo que a configuração que gerou o melhor resultado na avaliação foi usando a função de ativação tangente hiperbólica nas camadas ocultas, *batch size* de 32 e o otimizador Adam com uma taxa da aprendizagem de $1 \cdot 10^{-5}$.

Como a entrada do meta-modelo são as predições dos classificadores, o conjunto de treinamento deve ser gerado a partir destes valores. Sendo assim, para criação do conjunto de treinamento do meta-modelo, foi usado um processo iterativo chamado validação cruzada k-fold (*k-fold cross-validation*). Neste método, o conjunto de treinamento é dividido em k amsotras de mesmo tamanho, sendo uma delas usada obter as predições enquanto as $k-1$ restantes são usadas como treino. O processo se repete por k vezes, variando a amostra usada para predição. No final da repetição deste processo para cada classificador, é obtido um conjunto de treinamento para o meta-modelo do mesmo tamanho do conjunto de treinamento original, com a vantagem que as predições foram feitas em uma amostra não usada no treinamento, como mostrado na Figura 7. Para o treinamento deste meta-modelo, foi usado um valor de $k = 5$.

Tabela III: Avaliação, no conjunto de teste, dos classificadores com os melhores hiperparâmetros.

Parâmetro	Modelo A	Modelo B	Modelo C	Modelo D	Modelo E	Modelo F
Arquitetura	InceptionResNetV2	EfficientNet-B2	DenseNet-169	DenseNet-201	EfficientNet-B7	VGG16
Learning Rate	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$1 \cdot 10^{-5}$
Batch Size	192	64	192	192	192	192
Amostragem	*Oversample*	*Class Weight*	*Oversample*	*Oversample*	*Oversample*	*Oversample*
Optimizador	NAdam	Adam	NAdam	NAdam	RAdam	NAdam
Unidades	512/2	256/128/2	1024/2	1024/2	512/2	1024/2
Acurácia (%)	92.75 ± 0.77	93.12 ± 0.94	93.12 ± 0.88	93.48 ± 0.84	91.30 ± 1.06	93.48 ± 1.38
F_1-Score (%)	96.56 ± 0.65	96.26 ± 1.04	97.44 ± 0.77	97.74 ± 0.91	94.79 ± 1.02	96.37 ± 0.97
ROC AUC (%)	97.63 ± 0.15	97.89 ± 0.21	97.57 ± 0.14	97.90 ± 0.27	96.84 ± 0.19	96.69 ± 1.14
PR AUC (%)	97.73 ± 0.14	97.87 ± 0.30	97.38 ± 0.20	97.96 ± 0.30	96.94 ± 0.17	95.85 ± 1.45

Para reduzir o impacto da variância das predições do conjunto de treinamento na predição final do meta-modelo, o valor usado como entrada do meta-modelo foi a mediana de 12 repetições da validação cruzada feitas para cada um dos classificadores. Além disso, os dados de entrada ainda receberam um pré-processamento, eles foram subtraídos da média e divididos pelo desvio padrão em relação às predições de cada classificador. Ao contrário dos classificadores, as camadas ocultas do meta-modelo não foram regularizadas com *dropout*.

4. RESULTADOS

Existem várias métricas que ajudam a definir a capacidade de classificação de um modelo de *deep learning*. Na Seção 4.1 serão apresentadas as métricas utilizadas, suas expressões e o que elas avaliam. Na Seção 4.2, será comparada a performance do modelo entre o conjunto de treinamento e de validação usando metricas quantitativas. Na Seção 4.3, os modelos serão avaliados no conjunto de teste usando métricas quantitativas. E, na Seção 4.4, será feita uma avaliação qualitativa dos modelos no conjunto blind.

4.1. Métricas de avaliação quantitativa dos modelos

As métricas aqui utilizadas baseiam-se no erro ou acerto na associação dos objetos às classes pelo modelo. Para relacionar a probabilidade calculada pelo modelo à classe, é definido um limiar de discriminação, que é a probabilidade mínima para que um exemplar pertença à uma classe. Para o cálculo das métricas, foi considerado um de limiar de 0.5.

A primeira métrica apresentada é a acurácia. Ela representa o número de objetos classificados corretamente em relação ao número total de objetos. Sua expressão é mostrada na equação (5).

$$Acurácia = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

onde TP, TN, FP, e FN são, respectivamente, a quantidade de Verdadeiro Positivo (*True Positive*), Verdadeiro Negativo (*True Negative*), Falso Positivo (*False Positive*) e Falso Negativo (*False Negative*).

Contudo, a acurácia nem sempre é uma métrica confiável para conjunto de dados desbalanceados, pois quanto maior a desproporção do número de objetos entre as classes, menor é o impacto das predições incorretas da classe em minoria no valor da acurácia, levando à uma avaliação superotimista do modelo. Para lidar com isso, usamos outras duas métricas: *Precision* e *Recall*, definidas nas equações (6) e (7), respectivamente.

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

As equações (6) e (7) mostram que *Precision* tem o valor máximo na ausência de Falso Positivo e *Recall* tem o valor máximo na ausência de Falso Negativo. Ao adotarmos a classe em minoria como positiva, temos uma avaliação que reflete a capacidade do modelo na tarefa mais difícil – a classificação correta de objetos na classe com menor representação no conjuto de treinamento. Para sumarizar estas duas métricas, usamos uma outra, chamada F_1-*score*, que é a média harmônica entre *Precision* e *Recall*, como definida na equação (8).

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

4.2. Performance dos modelos no treinamento

A Figura 8 mostra a avaliação do modelo em cada época de treinamento. Nos painéis superiores, criados a partir do conjunto de treinamento, notamos que, para a configuração escolhida, ambos os classificadores treinados executam a tarefa de minimizar a função de custo. Mas, para avaliar a capacidade de generalização da rede, ou seja, o potencial de reconhecimento de padrões em um caso real, é importante comparar os resultados com uma amostra diferente do trei-

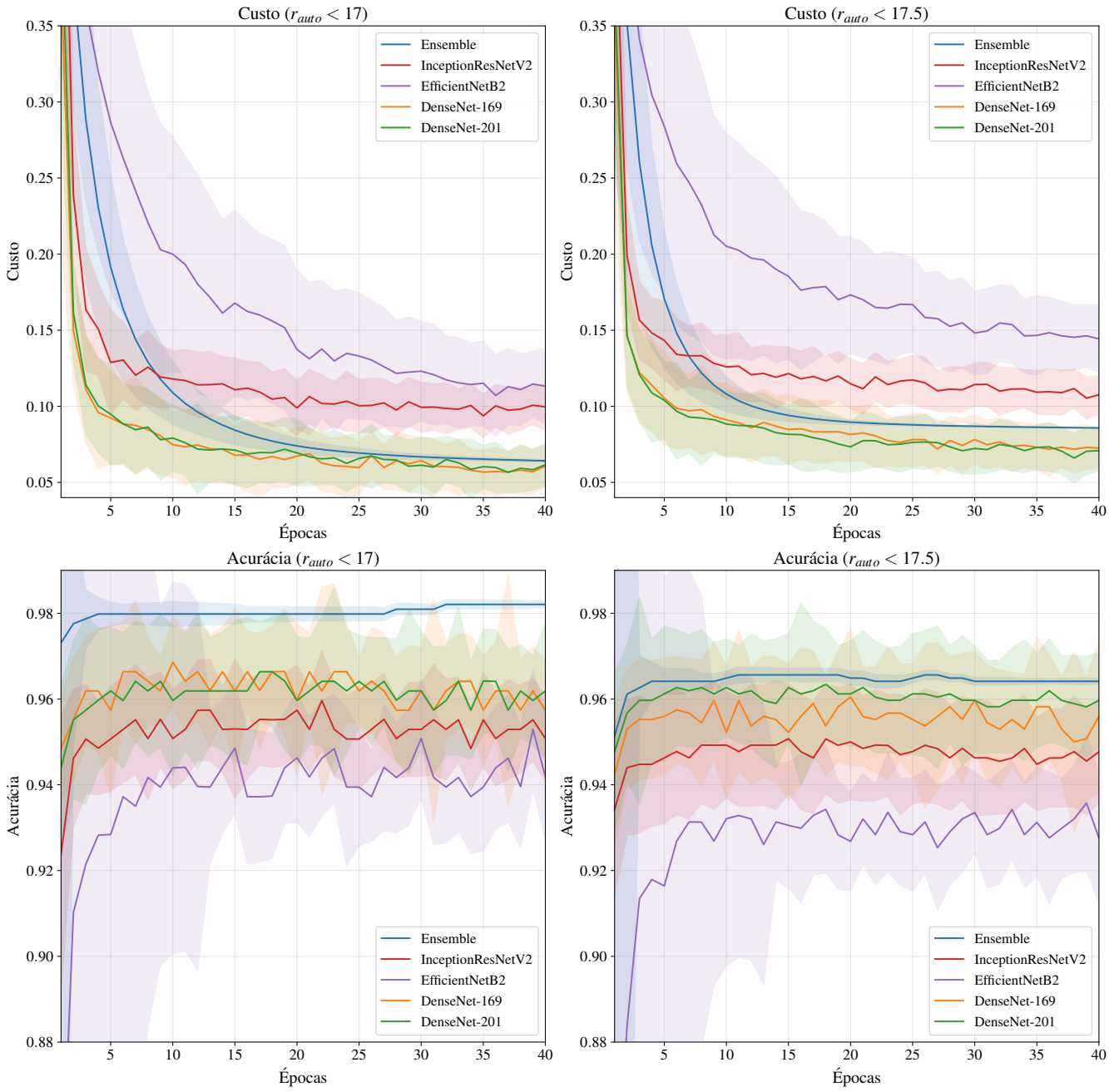


Figura 8: Os gráficos mostram a avaliação dos classificadores individuais e do Ensemble para modelos treinados com $r_{auto} < 17$ (esquerda) e $r_{auto} < 17.5$ (direita). É mostrada a função custo no conjunto de treinamento (painéis de cima) e a acurácia no conjunto de validação (painéis de baixo) para cada época de treinamento. As linhas contínuas representam a mediana de 60 medições, enquanto que as regiões sombreadas representam o desvio padrão.

namento. Os gráficos da acurácia no conjunto de validação, exibidos nos painéis inferiores, mostram que os classificadores conseguem bom desempenho em uma amostra não usada no treinamento.

Como visto na Seção 2.2, o modelo treinado no conjuto $r_{auto} < 17.5$ possui todos os objetos do conjunto $r_{auto} < 17$ mais objetos de magnitude entre 17 e 17.5. Sendo assim, a diferença de desempenho observada entre os modelos da direita e da esquerda mostram o impacto da inclusão destes objetos no treinamento. Este comportamento já era esperado, visto que são objetos difíceis de classificar até mesmo visualmente.

4.3. Avaliação do modelo no conjunto de teste

A Tabela III sumariza as melhores configurações obtidas para cada uma das arquiteturas testadas (conforme a Seção 3.9) e suas respectivas avaliações no conjunto de teste. Na parte inferior da tabela, foram incluídas as métricas definidas em 4.1, para cada classificador, com os melhores hiperparâmetros, como definidos em 3.8. Note que os resultados

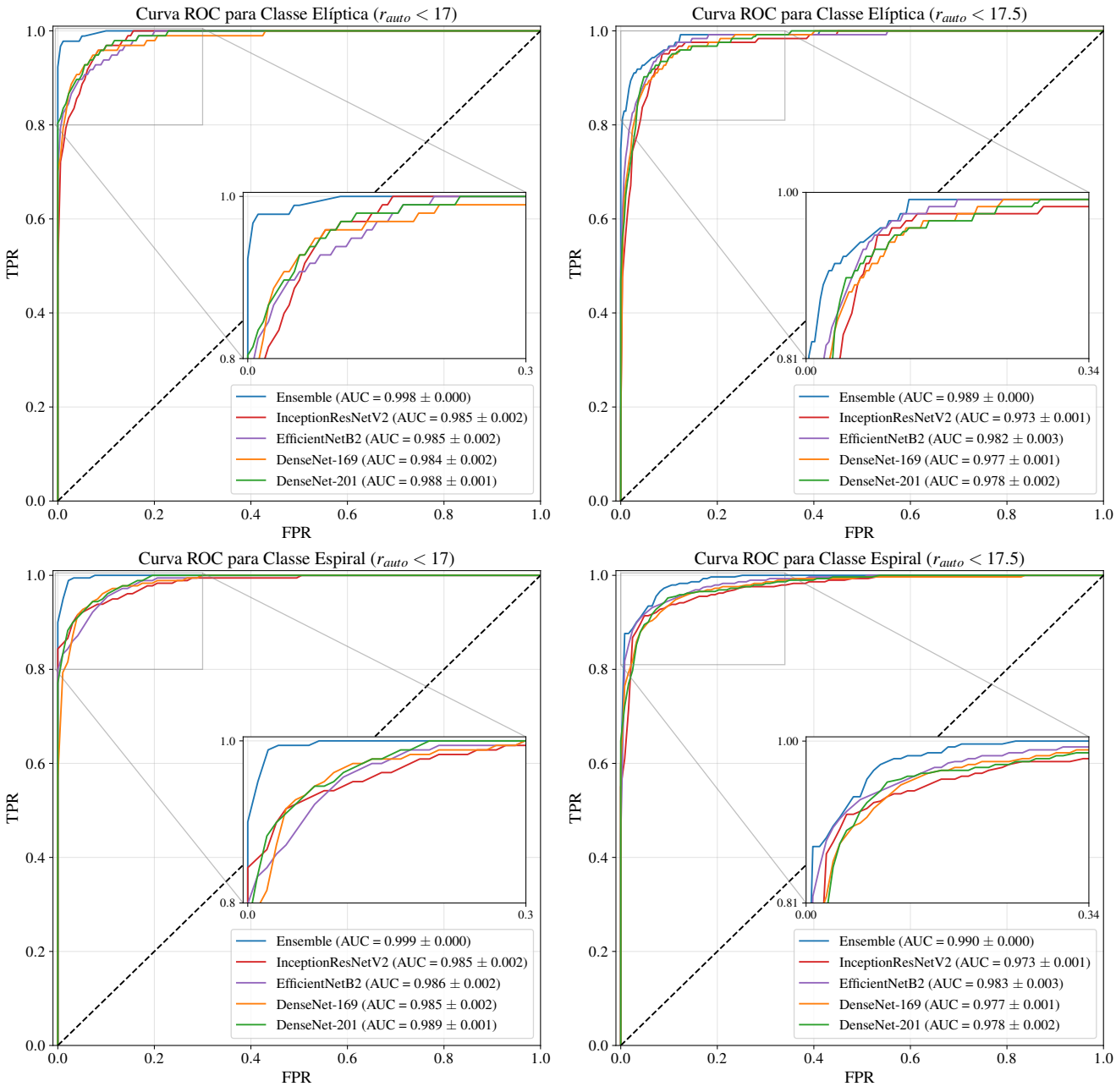


Figura 9: Curvas ROC dos classificadores individuais e do *Ensemble* para os modelos treinados com $r_{auto} < 17$ (esquerda) e $r_{auto} < 17.5$ (direita) separadas pelas classes Elíptica (nos painéis de cima) e Espiral (nos painéis de baixo). As linhas contínuas mostram a mediana de 60 curvas e o valor de AUC, na legenda, representa a mediana da área abaixo destas curvas e seu respectivo desvio padrão.

listados na tabela mostram a configuração dos classificadores que comporão o *Ensemble*, com exceção dos Modelos E (EfficientNetB7) e F (VGG16), que não serão incluídos no *Emsemble*, por não apresentarem resultados tão bons quanto os demais. A avaliação da rede VGG16 apresenta um alto desvio padrão para todas as métricas e a rede EfficientNetB7 não apresenta resultados compatíveis com o custo computacional.

A Tabela IV mostra os resultados para o *Ensemble*, obtido ao se combinar os modelos de A a D, como descrito na Seção 3.10. Outra análise, mais robusta do que é mostrado nessa tabela, é avaliar os modelos sob diferentes limiares de discriminação, ao invés de sob apenas um. Para isto, usamos a Curva Característica de Operação do Receptor *(Re-*

Tabela IV: Avaliação dos modelos no conjunto de teste.

Métrica (%)	$r_{auto} < 17$	$r_{auto} < 17.5$
Acurácia	98.52 ± 0.13	93.48 ± 0.12
F_1-Score	98.52 ± 0.13	92.42 ± 0.11
ROC AUC	99.81 ± 0.01	98.79 ± 0.01
PR AUC	99.82 ± 0.01	98.94 ± 0.01

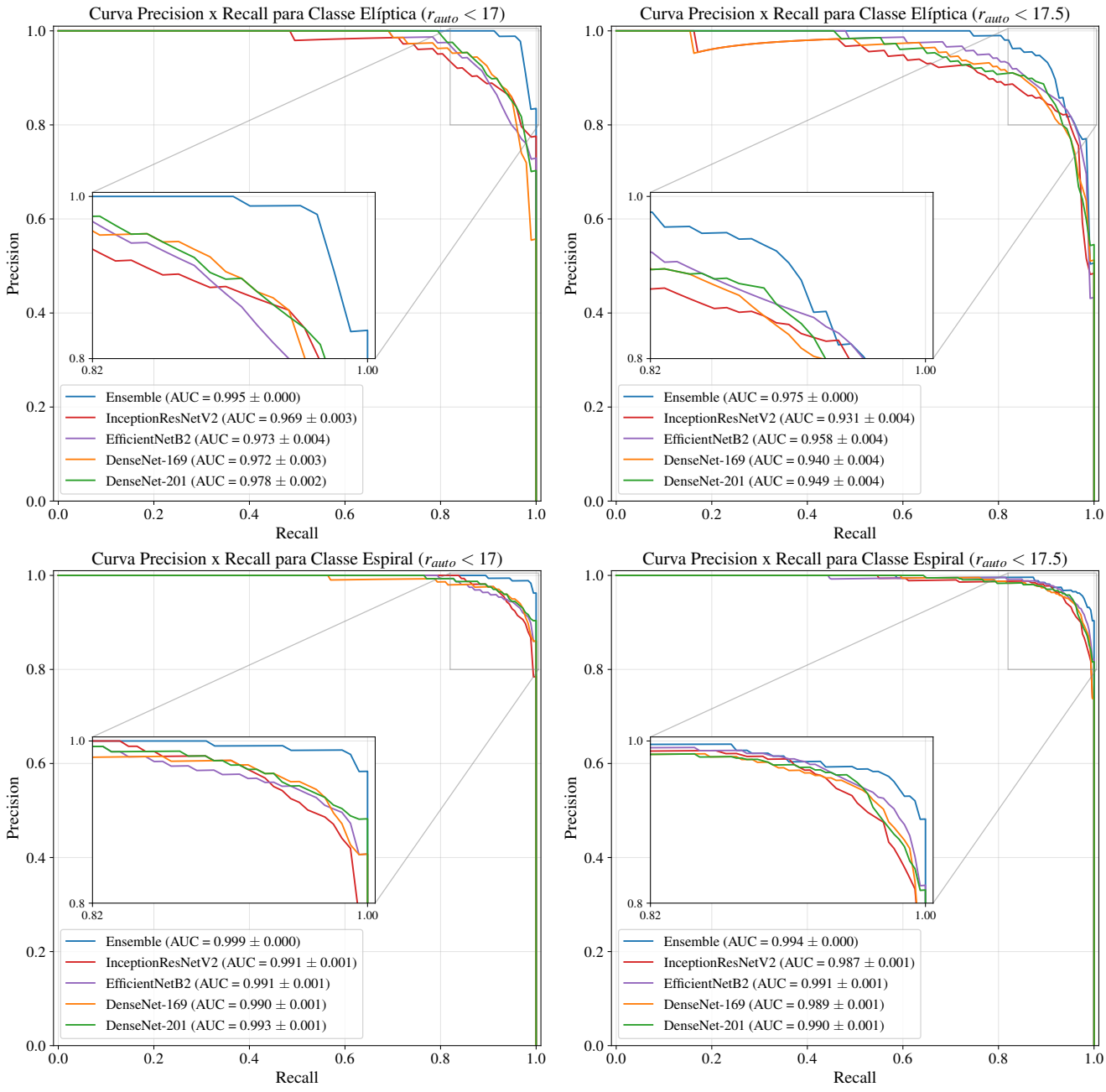


Figura 10: Curvas *Precision x Recall* dos classificadores individuais e do Ensemble para os modelos treinados com $r_{auto} < 17$ (esquerda) e $r_{auto} < 17.5$ (direita) separadas pelas classes Elíptica (nos painéis de cima) e Espiral (nos painéis de baixo). As linhas contínuas mostram a mediana de 60 curvas e o valor de AUC, na legenda, representa a mediana da área abaixo destas curvas e seu respectivo desvio padrão.

ceiver Operating Characteristic – ROC) [57, 58], que é um gráfico de FPR (False Positive Rate) versus TPR (True Positive Rate). Usamos também a *Curva Precision-Recall* (PR) que é um diagnóstico pontente para avaliar a classificação de modelos treinados a partir de conjuntos de dados desbanlanceados, como é o caso aqui.

A curva ROC pode ser visualizadas na Figura 9, onde resultados para os modelos individuais e para o Emsemble são comparados. Note que a curva ROC é tão melhor quanto maior for a área abaixo desta (*Area Under Curve – AUC*), pois essa área representa o grau de separabilidade de um modelo. Na Figura 10 mostramos a curva PR e o mesmo comentário vale aqui sobre o AUC. Fica claro ao analisar ambas as Figuras 9 e 10, que os modelos são comparáveis entre si (as áreas sob as curvas são parecidas), e que os resultados do Emsemble são claramente melhores, AUC maiores, tanto para $r_{auto} < 17$ quanto para $r_{auto} < 17,5$. Outra conclusão ao se inspecionar as figuras em questão é que o Ensemble atinge resultados numericamente melhores no caso de classificação de objetos com $r_{auto} < 17$, do que com objetos com $r_{auto} < 17.5$.

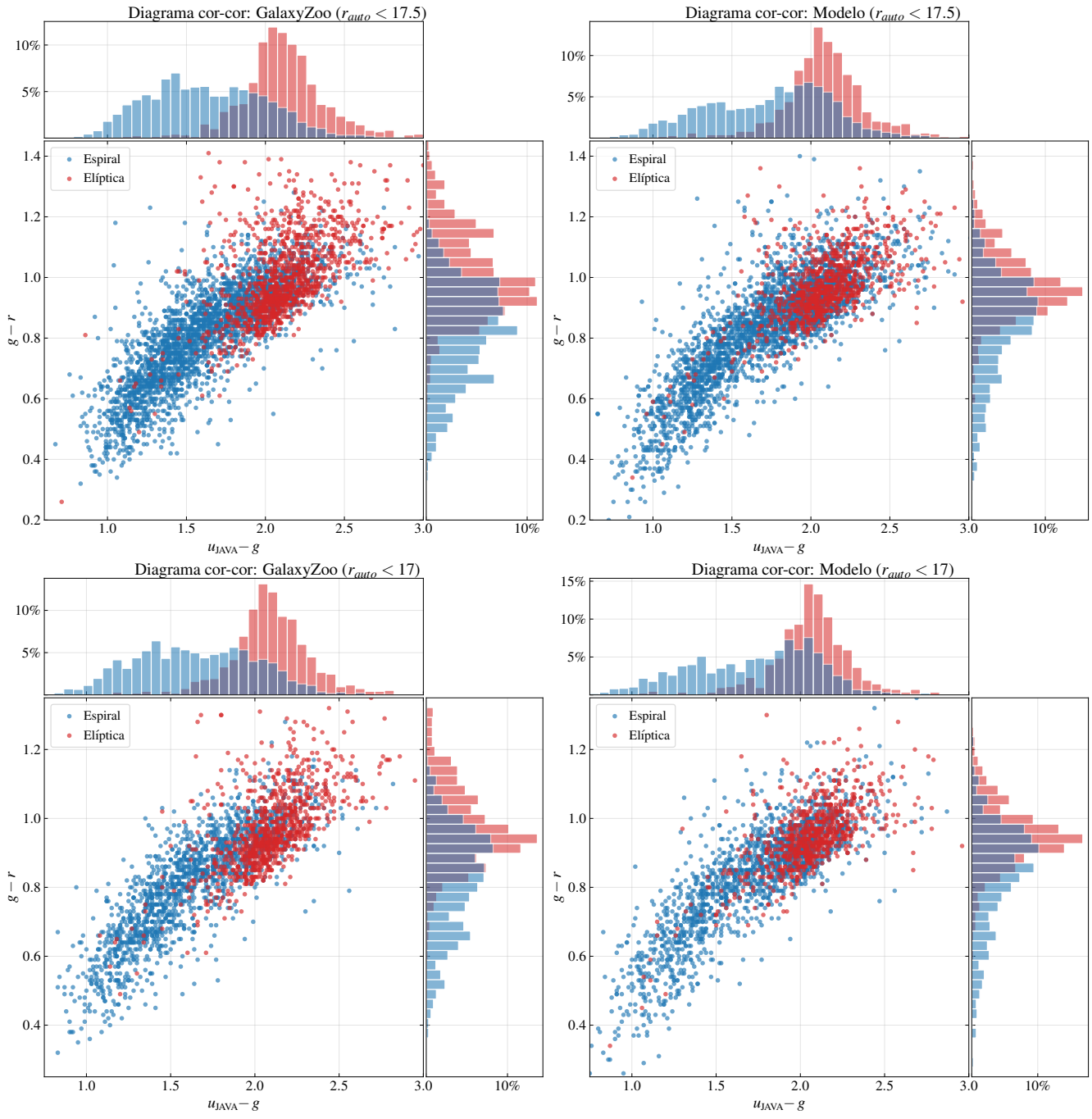


Figura 11: Diagrama cor-cor de $u_{JAVA} - g$ versus $g - r$. À esqueda, classificações visuais do GalaxyZoo nos conjuntos de treino, validação e teste e, à direita, classificações na amostra Blind feitas pelos modelos treinados com $r_{auto} < 17$ em cima e $r_{auto} < 17.5$ em baixo. Em azul, galáxias classificadas como Espirais e, em vermelho, galáxias classificadas como Elípticas.

4.4. Avaliação qualitativa do modelo no conjunto blind: diagramas cor-cor

Um dos objetivos deste trabalho é obter classificações de galáxias elípticas e espirais, que não foram ainda classificadas (nas chamadas amostras *blind*). Isto foi feito para uma amostra contendo 2536 galáxias com magnitude $r_{auto} < 17$ e 3951 galáxias com magnitude $r_{auto} < 17.5$, com distribuições de magnitude e redshift compatíveis com as amostras de treinamento, como mostrado na Seção 2.2.

Uma maneira mais qualitativa de avaliar os resultados da classificação é pela análise do diagrama cor-cor das galáxias, comparando as nossas classificações com a do Galaxy Zoo. Nesta análise, é muito importante lidar com amostras com distribuições de magnitude e redshift similares, como visto na Seção 2.2.

A Figura 11 mostra um diagrama de $u_{JAVA} - g$ versus $g - r$. Os painéis da esquerda mostram classificações visuais obtidas do GalaxyZoo nos conjuntos de treino, validação e teste enquanto que os painéis da direita mostram classificações

feitas pelos nossos modelos descritos neste trabalho, no conjunto Blind. Como esta comparação é feita entre amostras diferentes de galáxias (embora com distribuição de brilho e redshift similares), não é esperado que os gráficos da esquerda sejam idênticos aos da direita, mas que tenham formas parecidas. Logo, o que se pretende mostrar é que a semelhança entre os diagramas da classificação humana com a classificação automática (usando nosso modelo) é um indicativo de uma classificação robusta. De fato, os critérios para definir o que é uma galáxia Espiral e Elíptica podem não ser identicos pois se tratam de uma fronteira artificialmente definida no qual se esperam diversas galáxias com classificação ambígua como no caso de S0 e diversos tipos de galáxias que, para efeito deste estudos foram agrupadas em apenas duas classes. Adicionalmente, as classificações humanas do GalaxyZoo foram realizadas em imagens com outro nível de ruído e resolução.

Um outro resultado mostrado nessas figuras é que as classificações são consistentes tanto para o modelo treinado usando galáxias com $r_{auto} < 17$ quanto para $r_{auto} < 17.5$. Aconselhamos que o modelo treinado em $r_{auto} < 17$ seja utilizado para classificação de galáxias até esta magnitude e o modelo $r_{auto} < 17.5$ seja utilizado em galáxias com magnitudes no intervalo de 17 a 17.5. As Figuras 12 e 13 mostram exemplos de imagens de galáxias classificadas pelos modelos de Inteligência Artificial desenvolvidas neste trabalho no conjunto de dados sem classificacao prévia (*blind*).

5. DISCUSSÃO E CONCLUSÃO

Em contraste com as abordagens de aprendizagem comuns, o que fizemos neste trabalho foi construir modelos a partir dos dados de treinamento, escolher entre os melhores modelos e combiná-los. O resultado principal deste trabalho é, então, a combinação das predições de várias redes com seus respectivos melhores hiperparâmetros e a constatação de que as métricas que avaliam os resultados mostram significativa melhoria quando usamos o método de emsemble, comparado com o uso de apenas um modelo. O objetivo final era o de obter maior acurácia na classificação de galáxias do mapeamento S-PLUS, e este objetivo é atingido através do método de emsemble aqui utilizado. O produto final é a classificação de galáxias no mapeamento S-PLUS em elípticas e espirais utilizando diversas redes. A morfologia das galáxias é uma propriedade fundamental necessária, por exemplo, para estudos de formação e evolução de galáxias.

Exploramos o uso de um *Ensemble*, aumentando a acurária dos resultados e diminuindo a variância das predições e proporcionando melhores resultados em relação ao que é obtido com apenas o melhor classificador indidual. Outra novidade foi testar magnitudes até 17.5, dado que um trabalho anterior [12] tinha classificado galáxias com magnitude até 17 com um classificador apenas.

Comparando este trabalho com o de [12], para o limite de magnitude 17, notamos resultados bastante similares, sendo que aquele trabalho levou em consideração imagens FITS enquanto que este trabalho utilizou imagens RGB, obtidas utilizando o *software* Trilogy [11]. Do ponto de vista computacional, muitas vezes a utilização de imagens RGB é a única opção, pois as imagens em formato FITS podem ocupar um maior espaço na memória de GPU, aumentando o tempo de processamento. Logo, foi importante mostrar aqui que as imagens RGB dão resultados similares quando o Ensemble é utilizado.

Um ponto muito importante neste trabalho é o pré-processamento. Este incluiu padronização, escalonamento e/ou normalização, como descrito na Seção 3.1. O importante é que se faça o procedimento adequado para cada arquitetura. No caso específico deste trabalho, esta foi a maneira que apresentou o melhor desempenho quando combinado com a inicialização dos pesos com pré-treino na base de dados *ImageNet*. Um resultado similar é mostrado no trabalho de [12], em que as redes treinadas usando o pré-treino *ImageNet* ultrapassaram a performance das redes treinadas sem pré-treino, mesmo com um pré-processamento de dados diferente do que foi feito aqui.

Outro ponto importante é que, tanto a escolha da arquitetura, quanto dos hiperparâmetros da rede têm impacto no potencial de predição do modelo. Para reduzir os impactos negativos, utilizamos uma combinação de quatro classificadores. Esta técnica permite reduzir os viéses dos classificadores individuais, garantindo, assim, um maior potencial de generalização, e, por consequência, uma maior capacidade preditiva do modelo.

Este trabalho mostrou a melhoria dos resultados usando método de *Ensemble*. No futuro, este modelo pode ser usado para classificação de galáxias do levantamento S-PLUS.

6. DISPONIBILIDADE DOS DADOS

Nós disponibilizamos publicamente os catálogos de classificação bem como os modelos de *Deep Learning* na página `https://link.natanael.net/sh-5hm-g7`.

7. AGRADECIMENTOS

Os modelos foram implementados usando vários projetos *open-source*, como a linguagem de programação Python [59], o Trilogy [60], as bibliotecas de *deep learning* Tensorflow [61] e Keras [62] e outras bibliotecas de computação científica [63–70]. Este projeto também fez uso de serviços online, como o SkyServer[7] e o S-PLUS Cloud[8], para acesso de imagens e catálogos astronômicos, e o Google Colab[9], para treinamento dos modelos usando GPU's gratuitamente. Gostaríamos de agradecer os colegas Arianna Cortesi, Gefferson Lucatelli, Erik Lima, além do grupo de trabalho de Morfologia, pelas importantes contribuições e discussões sobre o assunto.

[7] `http://skyserver.sdss.org/dr16/en/home.aspx`
[8] `https://splus.cloud`
[9] `https://colab.research.google.com`

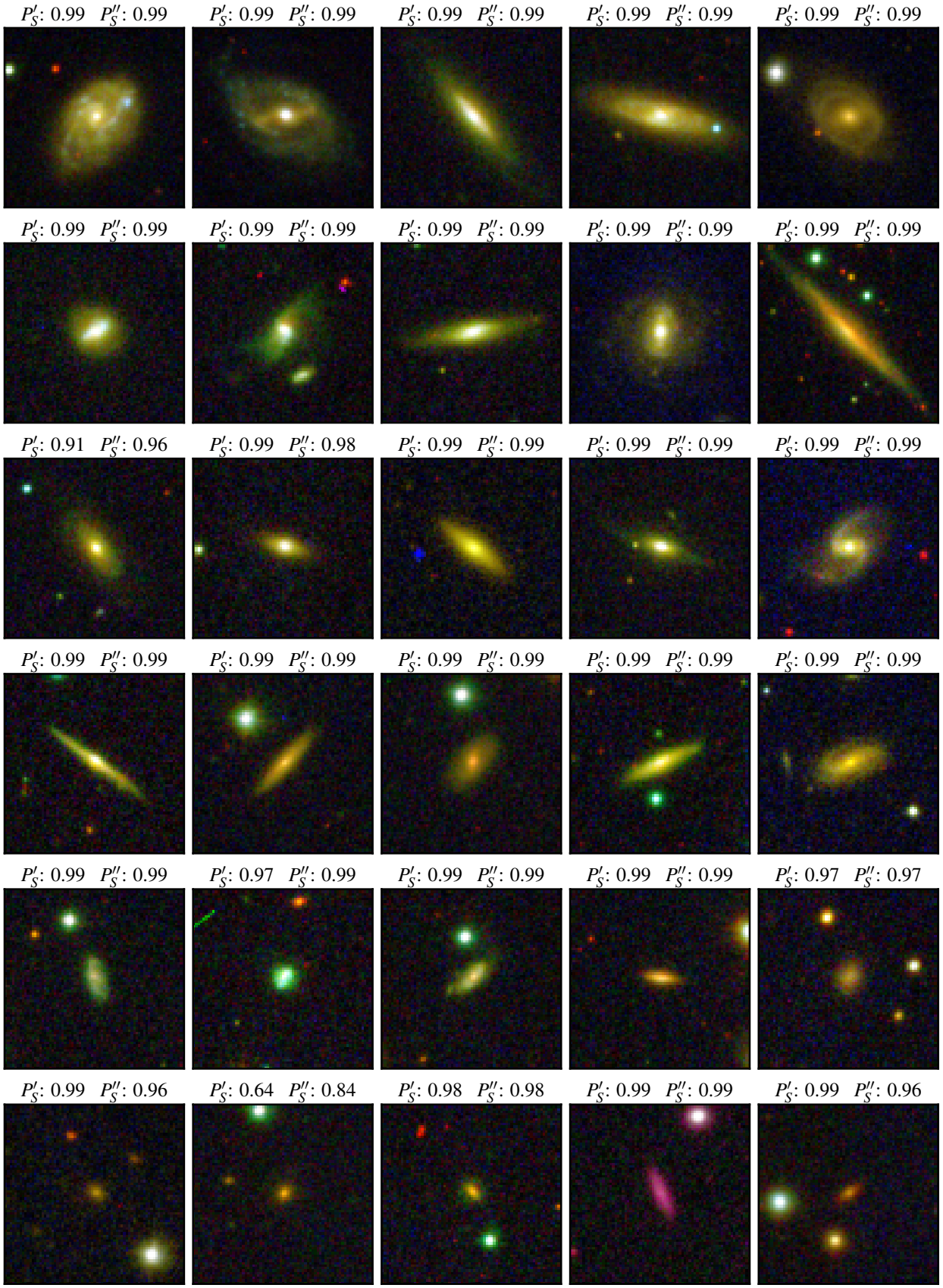


Figura 12: Amostra aleatória com 30 galáxias do conjunto *blind* com classificações dadas pelos modelos. Onde P'_S e P''_S representam a probabilidade da galáxia pertencer à classe espiral, inferida pelo modelo treinado com galáxias no intervalo de magnitude $r_{auto} < 17$ e $r_{auto} < 17.5$, respectivamente. As galáxias estão dispostas em ordem crescente de magnitude e o intervalo de r_{auto} de cada linha é mostrado no lado direito.

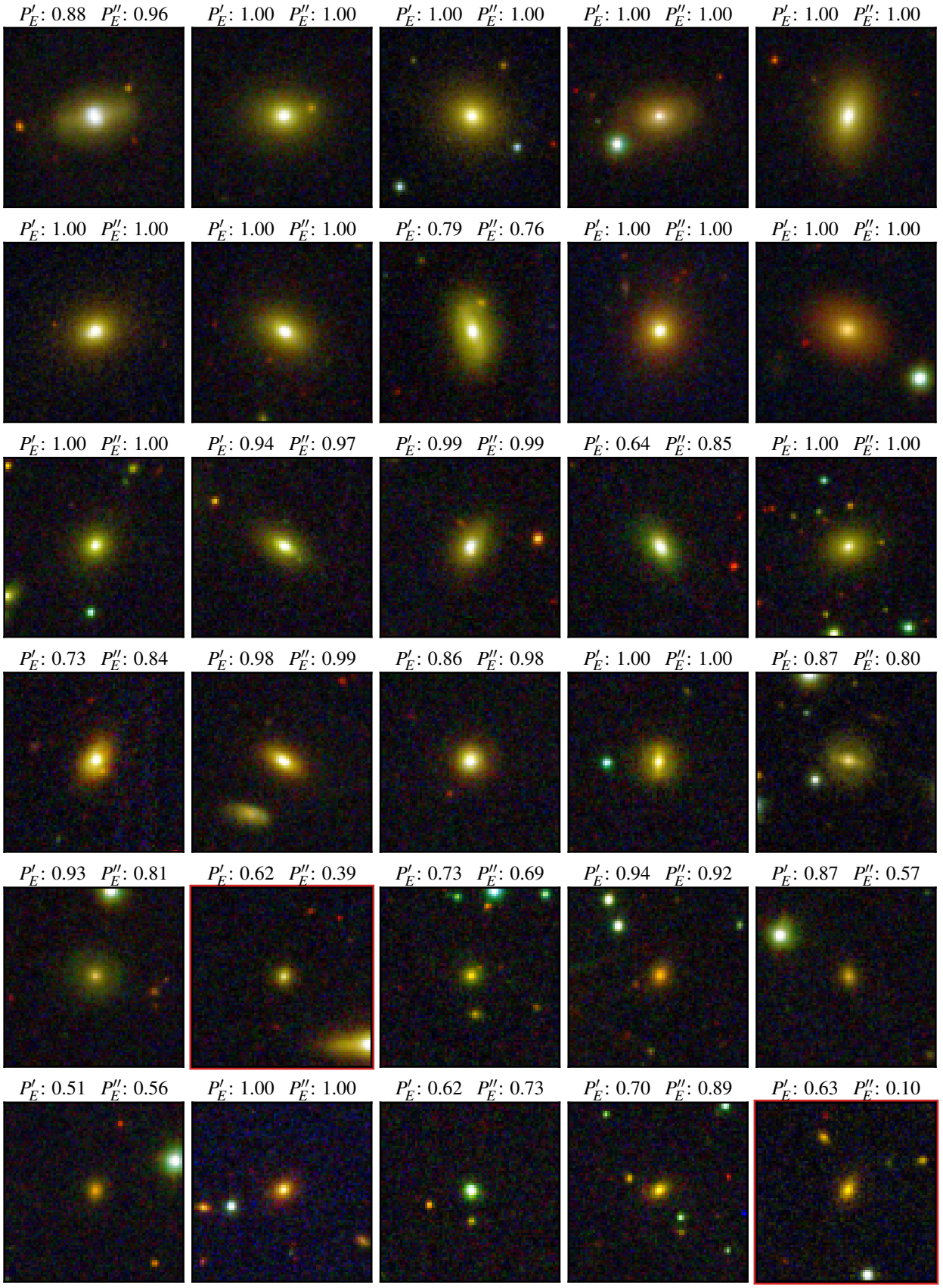


Figura 13: Amostra aleatória com 30 galáxias do conjunto *blind* com classificações dadas pelos modelos. Onde P'_E e P''_E representam a probabilidade da galáxia pertencer à classe elíptica, inferida pelo modelo treinado com galáxias no intervalo de magnitude $r_{auto} < 17$ e $r_{auto} < 17.5$, respectivamente. Painéis com bordas vermelhas assinalam para divergência na classificação dos modelos (para um limiar de 0.5).

Referências Bibliográficas

[1] E. P. Hubble. Extragalactic nebulae. *Astrophysical Journal*, 64:321–369, Dec 1926.

[2] Lucy Fortson, Karen Masters, Robert Nichol, et al. *Galaxy Zoo: Morphological Classification and Citizen Science*, pages 213–236. 2012.

[3] Ross E. Hart, Steven P. Bamford, Kyle W. Willett, et al. Galaxy Zoo: comparing the demographics of spiral arm number and a new method for correcting redshift bias. *Monthly Notices of the Royal Astronomical Society*, 461(4):3663–3682, 07 2016.

[4] Chisato Yamauchi, Shin-ichi Ichikawa, Mamoru Doi, et al. Morphological classification of galaxies using photometric parameters: The concentration index versus the coarseness parameter. *The Astronomical Journal*, 130(4):1545–1557, Oct 2005.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[6] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[7] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[8] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, page 689–692, New York, NY, USA, 2015. Association for Computing Machinery.

[9] C Mendes de Oliveira, T Ribeiro, W Schoenell, et al. The southern photometric local universe survey (s-plus): improved seds, morphologies, and redshifts with 12 optical filters. *Monthly Notices of the Royal Astronomical Society*, 489(1):241–267, Aug 2019.

[10] Gary King and Langche Zeng. Logistic regression in rare events data. *Political Analysis*, 9:137–163, Spring 2001.

[11] Dan Coe, Keiichi Umetsu, Adi Zitrin, Megan Donahue, Elinor Medezinski, Marc Postman, Mauricio Carrasco, Timo Anguita, Margaret J Geller, Kenneth J Rines, et al. Clash: precise new constraints on the mass profile of the galaxy cluster a2261. *The Astrophysical Journal*, 757(1):22, 2012.

[12] C. R. Bom, A. Cortesi, G. Lucatelli, et al. Deep learning assessment of galaxy morphology in s-plus datarelease 1, 2021.

[13] Larry Yaeger, Richard Lyon, and Brandyn Webb. Effective training of a neural network character classifier for word recognition. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS'96, page 807–813, Cambridge, MA, USA, 1996. MIT Press.

[14] Tom McReynolds and David Blythe. Antialiasing. In *Advanced Graphics Programming Using OpenGL*, pages 169–184. Elsevier, 2005.

[15] Alan C. Bovik. Basic gray level image processing. In *The Essential Guide to Image Processing*, pages 43–68. Elsevier, 2009.

[16] Patrice Y. Simard, Dave Steinkraus, and John C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2*, ICDAR '03, page 958, USA, 2003. IEEE Computer Society.

[17] Francois Chollet. *Deep Learning with Python*. Manning Publications Co., USA, 1st edition, 2017.

[18] Naoki Nonaka and Jun Seita. Electrocardiogram classification by modified EfficientNet with data augmentation. In *2020 Computing in Cardiology Conference (CinC)*. Computing in Cardiology, December 2020.

[19] David William Edwards II and Imren Dinc. Classification of protein crystallization images using EfficientNet with data augmentation. In *CSBio '20: Proceedings of the Eleventh International Conference on Computational Systems-Biology and Bioinformatics*. ACM, November 2020.

[20] Ansh Mittal, Anu Soorya, Preeti Nagrath, and D. Jude Hemanth. Data augmentation based morphological classification of galaxies using deep convolutional neural network. *Earth Science Informatics*, 13(3):601–617, December 2019.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[23] Edmar Rezende, Guilherme Ruppert, Tiago Carvalho, Antonio Theophilo, Fabio Ramos, and Paulo de Geus. Malicious software classification using VGG16 deep neural network's bottleneck features. In *Advances in Intelligent Systems and Computing*, pages 51–59. Springer International Publishing, 2018.

[24] Mohamad Aqib Haqmi Abas, Nurlaila Ismail, Ahmad Ihsan Mohd Yassin, and Mohd Nasir Taib. VGG16 for plant image classification with transfer learning and data augmentation. *International Journal of Engineering & Technology*, 7(4.11):90, October 2018.

[25] Ouiza Nait Belaid and Malik Loudini. Classification of brain tumor by combination of pre-trained vgg16 cnn. *Journal of Information Technology Management*, 12(2):13–25, 2020.

[26] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016.

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[29] Masoud Mahdianpari, Bahram Salehi, Mohammad Rezaee, Fariba Mohammadimanesh, and Yun Zhang. Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. *Remote Sensing*, 10(7):1119, July 2018.

[30] Tomoyuki Fujioka, Leona Katsuta, Kazunori Kubota, et al. Classification of breast masses on ultrasound shear wave elastography using convolutional neural networks. *Ultrasonic Imaging*, 42(4-5):213–220, June 2020.

[31] Mark Kriegsmann, Christian Haag, Cleo-Aron Weis, et al. Deep learning for the classification of small-cell and non-small-cell lung cancer. *Cancers*, 12(6):1604, June 2020.

[32] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[33] Pan Zhang, Ling Yang, and Daoliang Li. EfficientNet-b4-ranger: A novel method for greenhouse cucumber disease recognition under natural complex environment. *Computers and Electronics in Agriculture*, 176:105652, September 2020.

[34] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

[35] Zhong Li, Yuele Lin, Arne Elofsson, and Yuhua Yao. Protein

contact map prediction based on resnet and densenet. *BioMed Research International*, 2020, 2020.

[36] Braulio Solano-Rojas, Ricardo Villalón-Fonseca, and Gabriela Marín-Raventós. Alzheimer's disease early detection using a low cost three-dimensional densenet-121 architecture. In *Lecture Notes in Computer Science*, pages 3–15. Springer International Publishing, 2020.

[37] Xia Li, Xi Shen, Yongxia Zhou, Xiuhui Wang, and Tie-Qiang Li. Classification of breast cancer histopathological images using interleaved densenet with senet (idsnet). *PLoS ONE*, 15, 2020.

[38] Shui-Hua Wang and Yu-Dong Zhang. Densenet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 16(2s), June 2020.

[39] Yakov Frayman, Bernard F. Rolfe, and Geoffrey I. Webb. Solving regression problems using competitive ensemble models. In *Lecture Notes in Computer Science*, pages 511–522. Springer Berlin Heidelberg, 2002.

[40] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

[41] Jing Wang, Liu Yang, Zhanqiang Huo, Weifeng He, and Junwei Luo. Multi-label classification of fundus images with EfficientNet. *IEEE Access*, 8:212499–212508, 2020.

[42] Qishen Ha, Bo Liu, and Fuxu Liu. Identifying melanoma images using efficientnet ensemble: Winning solution to the siim-isic melanoma classification challenge, 2020.

[43] Ademola Oladosu, Tony Xu, Philip Ekfeldt, et al. Meta-learning for anomaly classification with set equivariant networks: Application in the milky way, 2020.

[44] Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM*, 41(1):67–95, January 1994.

[45] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, June 1990.

[46] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996.

[47] David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[48] Leo Breiman. Stacked regressions. *Machine Learning*, 24(1):49–64, July 1996.

[49] Padhraic Smyth and David Wolpert. Linearly combining density estimators via stacking. *Machine Learning*, 36(1/2):59–83, 1999.

[50] John Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1990.

[51] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[52] Timothy Dozat. Incorporating nesterov momentum into. 2015.

[53] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond, 2020.

[54] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. On the variance of the adaptive learning rate and beyond, 2012.

[55] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[56] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, et al. Improving neural networks by preventing co-adaptation of feature detectors, 2012.

[57] J A Hanley and B J McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, April 1982.

[58] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.

[59] Guido van Rossum and Jelke de Boer. Interactively testing remote servers using the python programming language. *CWi Quarterly*, 4(4):283–303, 1991.

[60] Dan Coe. Trilogy. `https://www.stsci.edu/~dcoe/trilogy/Intro.html`, 2012.

[61] Martín Abadi, Ashish Agarwal, Paul Barham, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[62] François Chollet et al. Keras. `https://keras.io`, 2015.

[63] A. M. Price-Whelan, B. M. Sipőcz, H. M. Günther, et al. The astropy project: Building an open-science project and status of the v2.0 core package. *The Astronomical Journal*, 156(3):123, August 2018.

[64] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[65] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, et al. scikit-image: image processing in python. *PeerJ*, 2:e453, June 2014.

[66] Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[67] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[68] F. Perez and B. E. Granger. Ipython: A system for interactive scientific computing. *Computing in Science Engineering*, 9(3):21–29, 2007.

[69] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

[70] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science Engineering*, 9(3):90–95, 2007.