UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE FÍSICA

Comparando a performance de diferentes Redes Neurais para a classificação de Galáxias de Baixo Brilho Superficial

Manuel Speranza Torres Veras

Trabalho de Conclusão de Curso apresentado à Comissão de Graduação em Física do Instituto de Física da Universidade Federal do Rio Grande do Sul sob orientação da Profa. Dra. Cristina Furlanetto, como parte dos requisitos para a obtenção do grau de Bacharel em Física - ênfase em Astrofísica.

Porto Alegre, RS, Brasil
Maio, 2022

Agradecimentos

À minha família, em especial aos meus pais, que nunca deixaram de me incentivar e apoiar de todas formas possíveis para que eu pudesse buscar conhecimento e me aventurar no mundo da pesquisa científica.

À Professora Cristina, por acreditar no meu potencial e ter me orientado com tanta dedicação e paciência.

Aos meus colegas de curso, pelo companheirismo e pela ajuda nos momentos de dificuldade. Não vejo a hora de reencontra-los nas filas do RU e do café da física. Em particular, agradeço ao Marcos Pasa, que contribuiu para a implementação da técnica de pré-processamento utilizada neste trabalho.

Aos professores que tive durante esse curso, por terem compartilhado seu conhecimento comigo e mostrado uma fração da infinita beleza da natureza e da matemática.

À UFRGS, por me proporcionar acesso a um ensino gratuito e de excelência.

Ao CNPQ e a FAPERGS, que financiaram minha pesquisa de iniciação científica sobre Aprendizado de Máquina aplicado a Lentes Gravitacionais, que foi fundamental para desenvolver boa parte do conhecimento e maturidade necessários para a realização deste trabalho.

Aos meus amigos, em especial, ao Vitor, Nicolas e Campello, pelo afeto, carinho e por todos rolês ao longo desses anos.

À Professora Ana e ao Professor Clécio, por aceitarem fazer parte da banca examinadora deste trabalho.

Resumo

Nos últimos anos, trabalhos inovadores, impulsionados por levantamentos robustos do céu e método eficientes de detecção, reacenderam o interesse da comunidade científica por Galáxias de Baixo Brilho Superficial, uma classe peculiar de galáxias que apresentam baixa densidade superficial de estrelas e são muito fracas e difusas em imagens ópticas. Se a detecção automática dessas galáxias em levantamentos fotométricos por si só já é uma tarefa complicada, dado seus baixos brilhos superficiais, as buscas por esses objetos sofrem de um problema adicional: a enorme quantidade de artefatos que são detectados nas imagens e também possuem baixo brilho superficial. Com a crescente quantidade de dados astronômicos, a inspeção visual para rejeitar os artefatos detectados se torna impraticável e é essencial desenvolver métodos eficientes para separar galáxias de baixo brilho superficial de artefatos. Métodos de aprendizado de máquina profundo, como as Redes Neurais Convolucionais, são considerados o estado da arte em diversos problemas de classificação de imagens.

Neste trabalho, temos o objetivo de comparar a performance da rede *Vision Tranformers*, que causou forte impacto na literatura recentemente por desafiar o paradigma do estado da arte, com as Redes Neurais Convolucionais para identificar Galáxias de Baixo Brilho Superficial. Para fazer isso, implementamos uma rede *Vision Transformers* e a comparamos com a rede *DeepShadows*, uma Rede Neural Convolucional desenvolvida por Tanoglidis et al. (2021b). Ambas as redes foram treinadas a partir do único conjunto de imagens desses objetos publicamente disponível, composto de 40000 imagens do *Dark Energy Survey*. Verificamos que o nosso modelo alcançou métricas ligeiramente superiores às da *DeepShadows*. Entre essas, quando treinadas no conjunto de dados em formato PNG, o modelo padrão da nossa rede obteve uma *acurácia* de 0.929, 0.98% maior que a da *DeepShadows*. No entanto, ao fazer uma análise das incertezas das métricas de classificação utilizando o método de *bootstrap*, constatamos que o desempenho da nossa rede foi tão bom quanto o da *DeepShadows* dentro do intervalo de 95% de confiança das métricas. Além disso, utilizando imagens do mesmo conjunto de dados, porém no formato FITS, implementamos um método de pré-processamento dos dados de entrada, que visa aperfeiçoar o desempenho das redes. Este método consiste em realizar um ajuste de contraste para ressaltar os objetos de baixo brilho superficial nas imagens. Verificamos que o ajuste de contraste contribuiu para que tanto o modelo padrão da *Vision Transformers* como a *DeepShadows* alcançassem uma performance superior. Ainda assim, os melhores resultados do treinamento com o conjunto de imagens FITS não superam os resultados de ambas as redes quando treinadas no conjunto de PNG. Conforme mostrou Dosovitskiy et al. (2021), com o aumento da quantidade de dados, as redes *Vision Transformers* são capazes de superar o desempenho das Redes Neurais Convolucionais, portanto esse parece ser um panorama especialmente interessante para as *Vision Transformers*.

Abstract

In the last years, innovative works, fueled by robust sky surveys and efficient methods of detection, rekindled the interest of the scientific community for Low Surface Brightness Galaxies, a peculiar class of galaxies with low superficial stellar density that are very faint and diffuse in optical images.

If the automatic detection of these galaxies in photometric surveys is already a complicated task, given their low surface brightness, searches for these objects suffer an aditional problem: the enormous amount of artifacts that are detected in the images and also have low surface brightness. With the growing amount of astronomical data, visual inspection to reject detected artifacts becomes impratical and it is necessary to develop efficient methods to separate Low Surface Brightness Galaxies from artifacts. Deep Learning methods, like Convolutional Neural Networks, are considered the state-of-the-art in several image classification problems.

In this work, we aim at comparing the performance of the Visual Tranformers network, which recently caused a huge impact in the literature for challenging the paradigm of the state-of-the-art, with Convolutional Neural Networks to identify Low Surface Brightness Galaxies. To do that, we implemented a Visual Transformers Network and compared it with the DeepShadows Network, a Convolutional Neural Network developed by Tanoglidis et al. (2021b). Both networks were trained using the only image set of these objects publicly avaliable, composed of 40000 images from the Dark Energy Survey. We verified that our model achieved slightly superior metrics in comparison with DeepShadows. Between those, when trained on the dataset using the PNG format, the standard model of our network achieved an $accuracy = 0.929$, which is 0.98% higher than the one of DeepShadows. However, when computing the uncertainties on the metrics of our method using the bootstrap method, we have noticied that the performance of our network was as good as the one obtained by DeepShadows when considering de 95% confidence interval of the metrics. Besides that, using images of the same dataset, but in the FITS format, we implemented a method to pre-processes the input data, which aims to improve the performance of the networks. This method consists of doing a contrast adjusment to highlight low surface brightness objects in the images. We noticed that the contrast adjustment contribuited both to the ViT standard model and DeepShadows to achieve a higher performance. However, the best training result with the FITS image set did not overcome the results of both networks when applied to the PNG dataset. Like Dosovitskiy et al. (2021) showed, with the increase of the datasets, the Visual Tranformers Networks can overcome the performance of Convolutional Neural Networks, therefore that seems to be an specially interesting paradigm to Vision Transformers.

Lista de Abreviaturas

ANN: Rede Neural Artificial (*Artificial Neural Network*)

CNN: Rede Neural Convolucional (*Convolutional Neural Network*)

DES: *Dark Energy Survey*

LSBG: Galáxia de Baixo Brilho Superficial (*Low Surface Brightness Galaxy)*

LSST: *Legacy Survey of Space and Time on the Vera C. Rubin Observatory*

MLP: Perceptron Multicamadas (*Multilayer Perceptron*)

SGD: Gradiente Descendente Estocástico (*Stochastic Gradient Descent*)

UDG: Galáxia Ultra-Difusa (*Ultra Diffuse Galaxy*)

ViT: *Vision Transformer*

Conteúdo

Capítulo 1

Introdução

Galáxias são sistemas gigantescos formados por matéria escura, gás, poeira, estrelas e seus sistemas planetários, além de buracos negros supermassivos em seus centros. A grande responsável por manter todas essas estruturas de maneira coesa, formando um sistema, é a gravidade, de forma que, embora existam exceções, quase todas estrelas do universo fazem parte de uma galáxia, como é o caso da nossa estrela - o Sol, que pertence à Via Láctea.

No entanto, as galáxias podem ser radicalmente diferentes, variando seu tamanho, brilho, cor, abundância química e outras características. Caracterizar os tipos diferentes de galáxias, ou seja, catalogar as especificidades de cada tipo, é uma importante tarefa dos astrônomos, já que isso nos ajuda a entender não só como o Universo é hoje, mas como ocorreu sua formação e evolução.

Neste trabalho, estudamos um desses tipos específicos de galáxias - as Galáxias de Baixo Brilho Superficial (LSBGs, do inglês *Low Surface Brightness Galaxies*). Esses objetos se destacam por terem algumas características relativamente peculiares: são tipicamente pequenas e de luminosidade muito inferior às outras galáxias como a Via Láctea.

Apesar disso, detectar LSBGs não é uma tarefa fácil: existem efeitos de seleção observacional, justamente por terem baixo brilho. Portanto, é essencial desenvolver métodos eficientes para detectar essas galáxias.

Mais especificamente, com o avanço de levantamentos de grandes áreas do céu que entrarão em operação em breve, tais como Euclid[1] e *Legacy Survey of Space and Time on the Vera C. Rubin Observatory* (LSST)[2], teremos uma enorme quantidade de dados astronômicos a serem analisados. Somente o LSST, que atingirá níveis de brilho superficial sem precedentes, produzirá 20 TB de dados por noite e observará aproximadamente 20 bilhões de galáxias durante os seus 10 anos em atividade[3].

Assim, como poderemos detectar LSBGs quando temos uma quantidade cada vez maior de dados de forma que a mera inspeção visual de objetos se mostra uma maneira bastante limitada? Métodos de detecção automática, ou seja, algoritmos

[1] `https://www.euclid-ec.org/`

[2] `https://www.lsst.org/`

[3] `https://www.lsst.org/scientists/keynumbers`

computacionais com a capacidade de processar grandes quantidades de dados de maneira acurada, podem fornecer uma solução para esse problema.

Notadamente, algoritmos de detecção já estão sendo utilizados em larga escala para identificar objetos astronômicos. Os avanços na computação, nos deram a possibilidade de desenvolver programas de computador para inspecionar quantidades gigantescas de dados e classificar objetos de nosso interesse.

Logo, apesar das LSBGs serem uma classe numerosa de galáxias, ainda não conhecemos elas com a devida profundidade, de forma que quase tudo que sabemos sobre a evolução de galáxias está fundamentada na observação do universo de alto brilho superficial. Por conseguinte, implementar métodos eficientes de detecção desses objetos é essencial para estudá-los e assim entender se o que conhecemos do Universo de alto brilho ainda é válido nos regimes de baixa massa e baixa luminosidade.

A seguir, fazemos uma breve revisão da literatura sobre LSBGs e sua detecção e classificação em imagens astronômicas de grandes áreas utilizando métodos automáticos.

1.1 Galáxias de Baixo Brilho Superficial

Galáxias de Baixo Brilho Superficial são galáxias com brilho superficial central mais fraco que o céu noturno. Assim, tais objetos se destacam por terem uma luminosidade muito inferior às de outras galáxias. De forma mais rigorosa, convencionalmente se define as LSBGs como galáxias com brilho superficial central na banda B, $\mu_0(B)$, maior do que um certo limite, cujo valor na literatura geralmente varia de $\mu_0(B) \geq 22.0\ mag\ arscsec^{-2}$ até $\mu_0(B) \geq 23.0\ mag\ arscsec^{-2}$ (Yi et al., 2022). Assim, justamente por serem pouco luminosas, os efeitos de seleção observacional tornam-se relevantes e sua detecção fica particularmente difícil, de forma que o Universo de baixo brilho superficial continua relativamente inexplorado (Dokkum et al., 2014) e muitas características das LSBGs são ainda pouco conhecidas (Impey and Bothun, 1997), apesar de formarem uma parte significativa da população de galáxias. Mais especificamente, estima-se que de 30% a 60% das galáxias do Universo Local sejam LSBGs (Yi et al., 2022).

A existência dessas galáxias foi proposta por M. J. Disney (1976) (Disney, 1976), mas elas só foram descobertas no final da década de 80 (Bothun et al., 1987; Impey et al., 1988). Nos últimos anos, graças ao avanço na instrumentação e ao desenvolvimento de instrumentos dedicados a explorar objetos de baixo brilho superficial, essas galáxias voltaram a chamar a atenção e reacenderam o interesse da comunidade acadêmica no estudo das LSBGs. Notadamente, com o imageamento profundo do *Dragonfly Telephoto Array*, foi descoberta uma população relativamente grande de LSBGs no aglomerado de Coma. Essas galáxias que foram detectadas tem raios efetivos variando de $R_e = 1.5 - 4.6kpc$ e brilhos superficiais centrais, calculados a partir dos raios efetivos circularizados, $\mu(g, 0) = 24 - 26\ mag\ arcsec^{-2}$ (Dokkum et al., 2014). Essas galáxias foram cunhadas como Galáxias Ultra-Difusas (UDGs,

do inglês *Ultra Diffuse Galaxies*). Tais galáxias possuem tamanho similar à Via Láctea, mas brilho superficial e massa estelar bastante inferiores (Conselice, 2018). Além disso, o brilho das UDGs é similar ao de clássicas galáxias anãs, apesar de terem tamanho aproximadamente 5 vezes maior que o esperado para sua massa (Lim et al., 2018).

De acordo com Prole et al. (2019), o consenso na literatura começou a apontar que as UDGs mantém uma continuidade de propriedades com outras LSBGs no que tange à taxa de formação estelar, tamanho, luminosidade e metalicidade. Mais especificamente, segundo Conselice (2018), existe evidência que as UDGs podem ser um subconjunto das Galáxias de Aglomerados de Baixa Massa (LMCGs, do inglês *Low-Mass Cluster Galaxies*), uma classe de galáxias bem conhecidas que geralmente se localiza em aglomerados de galáxias. Além disso, as UDGs compartilham propriedades estruturais e morfológicas com as Galáxias Anãs Elípticas e Anãs Esferoidais, além de apresentarem um contínuo de propriedades com as mesmas, o que sugere que podem fazer parte de apenas uma população com diferenças de tamanho (Conselice, 2018). Na Figura 1.1, podemos ver o diagrama tamanho-luminosidade de UDGs e LSBGs em comparação com outros sistemas estelares.

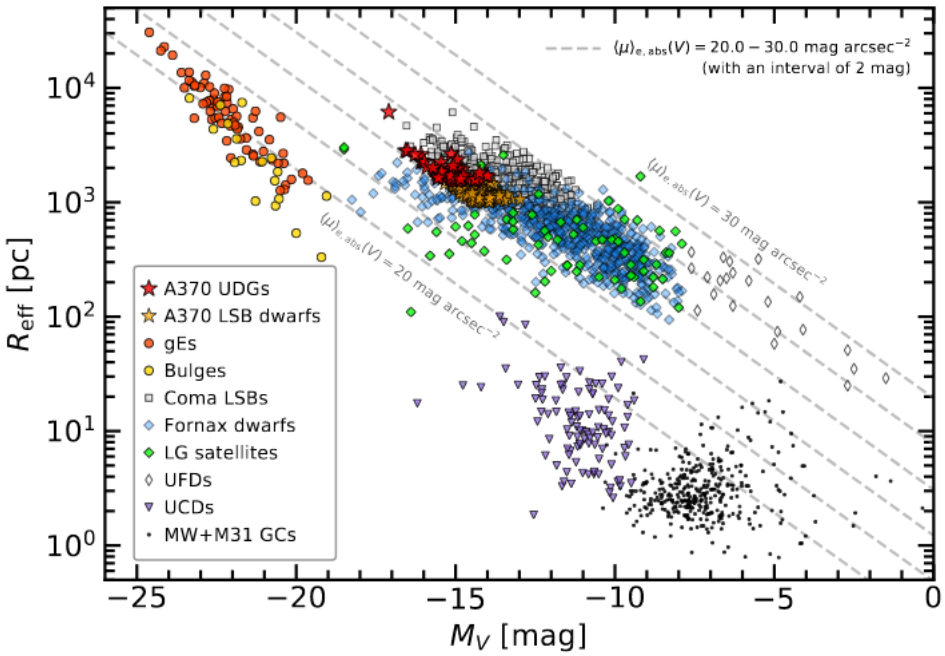

Figura 1.1: Diagrama relação tamanho-luminosidade para LSBGs do aglomerado de Coma (em quadrados cinza), UDGs de Abell 370 (em estrelas vermelhas) e anãs de baixo brilho superficial de Abell 370 (em estrelas laranjas) em comparação com outros sistemas estelares: gigantes elípticas (em círculos vermelhos), bojos de galáxias espirais (em círculos amarelos), anãs de Fornax (diamantes azuis), satélites do Grupo Local (diamantes verdes), anãs ultra-compactas (triângulos roxos), aglomerados globulares (pontos pretos) e anãs ultra-fracas (em diamantes brancos). As linhas pontilhadas em cinza denotam brilhos superficiais constantes. Imagem retirada de Lee et al. (2020).

Segundo Prole et al. (2020), no geral, em termos de massa estelar e metalicidade, as LSBGS são similares às galáxias anãs, no entanto exibem uma grande variedade de tamanhos físicos e massas do halo. Além disso, foram detectadas UDGs aparentemente desprovidas de matéria escura e com propriedades peculiares (van Dokkum et al., 2018). Assim, a variação de tamanho e massa do halo, sugere que podem existir diferentes mecanismos de formação e evolução de UDGs (Prole et al., 2020).

Por um lado, há evidências que tais galáxias podem se formar de maneira secular, impulsionada por processos internos da própria galáxia, não só isoladas em campo, mas também em aglomerados (Amorisco et al., 2018). Mais especificamente, simulações computacionais apontam que a saída de gás causada por mecanismos de *feedbacks* internos seguida da expansão de matéria escura e estelar podem produzir UDGs em halos de tamanho de anãs (Di Cintio et al., 2016). Por outro lado, existe também uma série de mecanismos de formação propostos que envolvem a transformação de galáxias anãs em UDGs por efeitos ambientais (Prole et al., 2019). Por exemplo a existência de UDGs em ambientes de alta densidade indica que algumas são dominadas por matéria escura e que o ambiente pode exercer um papel importante na sua formação e evolução (Lim et al., 2018), de forma que são galáxias que falharam em produzir estrelas. Nesse sentido, vale mencionar Carleton et al. (2019) que mostraram que a perda de massa por efeito de maré e o aquecimento de galáxias anãs pode produzir UDGs.

Outra característica importante das UDGs é que suas características variam conforme o ambiente. Por exemplo, UDGs encontradas em aglomerados de galáxias são predominantemente de sequência vermelha e possuem pouca evidência de interação de maré, o que por sua vez indica que tem uma alta relação massa-luminosidade (Prole et al., 2019). Por outro lado, galáxias anãs de baixo brilho superficial estão localizadas predominantemente em ambientes de baixa densidade e LSBGs vermelhas estão espacialmente correlacionadas com estruturas locais (Prole et al., 2020). Além disso, existe evidência de uma escassez relativa de UDGs próximo ao centro de aglomerados massivos, o que indica que essas galáxias são destruídas muito rapidamente ou não conseguem se formar de maneira tão eficiente nessas regiões (Prole et al., 2019).

Segundo Lim et al. (2018), muitas UDGs têm populações grandes de aglomerados globulares, o que é incomum para galáxias com massa e densidade estelar tão baixas. A importância disso vem do fato que os aglomerados globulares nos permitem estimar a massa total da galáxia usando medidas fotométricas e espectroscópicas, além de revelar características do estágio de formação inicial dessas galáxias (Harris et al., 2017).

Explicar como pode existir um espectro tão grande de tamanhos físicos para as Galáxias de Baixo Brilho Superficial é um desafio para os modelos cosmológicos atuais (Sales et al., 2020). Assim, uma caracterização mais acurada desses objetos pode ser de grande valia para informar os modelos cosmológicos (Tanoglidis et al., 2021a). Mais precisamente, o modelo padrão da cosmologia (ΛCDM) propõe que as galáxias se formam de maneira hierárquica – as menores se agregam formando sistemas maiores e mais complexos. Sabendo que os halos de matéria escura são essenciais para a formação e crescimento das galáxias e que existem diversas ten-

tativas de prever as suas propriedades a partir dos halos de matéria escura que as envolvem, as LSBGs podem ser de grande valia para testar esses modelos que prevem propriedades das galáxias a partir de princípios cosmológicos, uma vez que as LSBGs estão no extremo da relação tamanho-luminosidade.

Assim, a importância de se estudar galáxias desse tipo decorre do fato de que elas têm um papel importante no estudo da formação e evolução da população de galáxias (Impey and Bothun, 1997). Mais especificamente, o conhecimento de propriedades das LSBGs é importante para estudar com mais profundidade os tópicos de função luminosidade de galáxias, distribuição espacial de galáxias de baixa massa, a física de formação de estrelas em ambientes de baixa densidade superficial de gás (Kniazev et al., 2004).

1.2 Detecção Automática de LSBGs

Conforme mencionado na seção 1.1, dentro do escopo das LSBGs, as UDGs destacam-se por sua popularidade na literatura. Assim, em se tratando de detecção automática de UDGs, Prole et al. (2019), partindo de dados do *Kilo-Degree Survey* (KiDS)[4], aplicaram o algoritmo *MTobjects* (Teeninga et al., 2015) e uma série de critérios de seleção, como eliminar objetos com brilho superficial muito elevado ou raio muito grande, pois é improvável que esses sejam UDGs. Dessa forma, os autores obtiveram 212 candidatas a UDGs. Usando critérios de seleção semelhantes, Prole et al. (2020) identificaram 479 LSBGs a partir de imagens do *Hyper Suprime-Cam Subaru Strategic Program* (HSC-SSP)[5]. Por outro lado, usando uma série de algoritmos para subtrair fontes, aplicar filtros e utilizar critérios de seleção em imagens do Legacy Survey[6], Zaritsky et al. (2018) obteve um catálogo de 275 UDGs no aglomerado de Coma.

Além disso, citamos o trabalho de Li et al. (2022), que aplicou um novo método de detecção nas imagens do *Program for Imaging of the PERseus cluster* (PIPER) (Durrell et al., 2019), buscando por sobredensidades em populações de aglomerados globulares intergaláticos, utilizando os processos Cox log-Gaussianos. Como resultado, os autores detectaram objetos confirmados como UDGs com populações de aglomerados globulares conhecidos deste levantamento.

Entre os diversos métodos de detecção automática, destacamos os que utilizam Aprendizado de Máquina. Esta é uma ferramenta poderosa baseada em otimizar funções e assim resolver um problema sem que o algoritmo tenha sido programado explicitamente para tal. Uma vantagem clara dessa abordagem é que, dentro de certos parâmetros, podemos utilizar a mesma estrutura de um algoritmo para lidar com objetos distintos. Dessa forma, podemos utilizar a mesma arquitetura para lidar com diferentes problemas. Por outro lado, ao treinar redes neurais profundas estamos buscando uma função com o menor erro possível, mas entender o que essa

[4] `https://kids.strw.leidenuniv.nl/`

[5] `https://hsc-release.mtk.nao.ac.jp/doc/`

[6] `https://www.legacysurvey.org/`

função está fazendo é uma tarefa extremamente complicada, já que são extremamente complexas, podendo conter milhares ou até milhões de parâmetros ajustados aos dados de treinamento (Yip et al., 2021). Assim, por essa aparente incapacidade de entender como as redes profundas processam os dados, elas são frequentemente chamadas de *"modelos de caixa preta"* (Yip et al., 2021). Apesar disso, nos últimos anos, foram desenvolvidos diversos mecanismos que procuram interpretar de maneira simplificada a estrutura interna desses algoritmos (Buhrmester et al., 2021).

Em termos de classificação automática de LSBGs utilizando Aprendizado de Máquina destacamos o trabalho de Tanoglidis et al. (2021b)[de agora em diante T201b]. Um dos principais resultados desses autores foi a implementação de uma Rede Neural Convolucional (veja 2.3) intitulada *DeepShadows*, que alcançou 92.0% de acurácia de treinamento em uma amostra de 40000 objetos selecionadas através de uma série de algoritmos e inspeção visual das imagens do *Dark Energy Survey*. Além disso, os mesmos autores constataram que a rede *DeepShadows* apresentou um desempenho consideravelmente superior em detrimento de outros métodos de classificação utilizando Inteligência Artificial - o modelo de Floresta Aleatória (*Random Forest*) e Máquina de Vetores de Suporte (*Support Vector Machine*). De fato, as Redes Neurais Convolucionais são um dos algoritmos mais eficientes e populares para abordar problemas de visão computacional e frequentemente figuram entre os melhores modelos para problemas de classificação de imagem[7].

O conjunto de dados disponibilizado publicamente por T201b[8] é formado por 40000 imagens do *Dark Energy Survey* (DES)[9], sendo 20000 LSBGs e as outras 20000 artefatos, que consistem em objetos de baixo brilho, mas que não são LSBGs. Como exemplos de artefatos podemos citar regiões externas de baixo brilho superficial de galáxias mais brilhantes, *ghosts* de estrelas saturadas, problemas de detector, regiões de formação estelar nos braços de grandes galáxias espirais, entre outros objetos. Nas Figuras 1.2 e 1.3, podemos ver exemplos de imagens do conjunto de dados utilizados.

Existem diversos outros trabalhos sobre detecção automática de objetos astronômicos utilizando Inteligência Artificial. Em particular, destacamos Yao-Yu Lin et al. (2021), que aplicaram pela primeira vez a arquitetura de rede neural *Vision Transformer* (ViT) para a classificação morfológica de galáxias e comparam o desempenho dessa com uma Rede Neural Convolucional (CNN, do inglês *Convolutional Neural Network*). Para isso, foram utilizados um conjunto de imagens de galáxias extraídas do *Galaxy Zoo 2 Project*[10]. A melhor acurácia de treinamento obtida pelos autores para a ViT foi de 80.55% comparada a 85.5% da CNN. No entanto, os autores notaram que a ViT alcançou um desempenho particularmente bom ao classificar galáxias menores e pouco brilhantes, o que sugere que essa rede pode ser particularmente interessante para o problema de classificação automática

[7]No site `https://paperswithcode.com/task/image-classification`, podemos ver o estado da arte em problemas de classificação de imagem. Em quase todos conjuntos de dados, as CNNs se destacam pelo alto desempenho.

[8]`https://github.com/dtanoglidis/DeepShadows`

[9]`https://www.darkenergysurvey.org/`

[10]https://data.galaxyzoo.org/

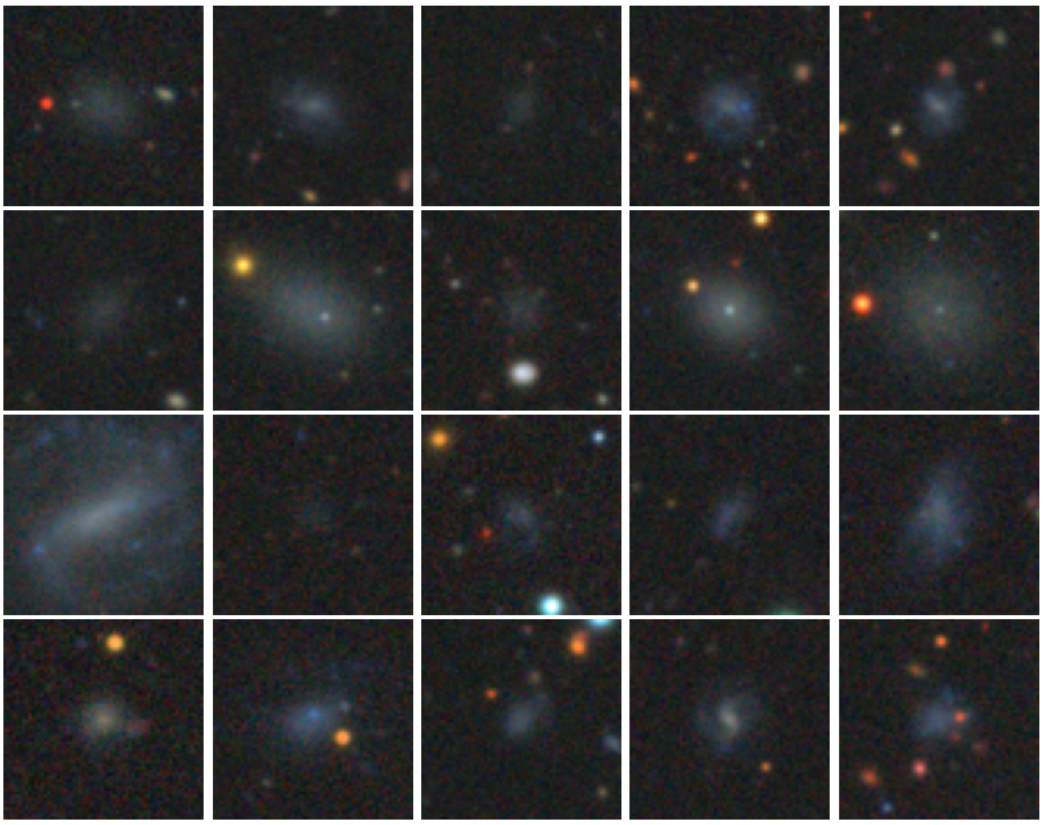

Figura 1.2: Exemplos de LSBGs do conjunto de imagens do DES selecionadas por T201b.

de LSBGs.

1.3 Objetivos e Estrutura

O objetivo desse trabalho é comparar o desempenho de diferentes arquiteturas de redes neurais para a classificação automática de Galáxias de Baixo Brilho Superficial. Motivados pelos excelentes resultados da rede *Vision Tranformers*[11], conforme demonstrado por Dosovitskiy et al. (2021) e baseados na observação de Yao-Yu Lin et al. (2021) que a ViT, implementada em um problema de classificação morfológica teve desempenho particularmente bom em galáxias pequenas e de baixo brilho, formulamos a hipótese que esse modelo poderia ter um desempenho superior as Redes Neurais Convolucionais.

Mais especificamente, utilizamos como *benchmark* a CNN *DeepShadows* e comparamos o seu desempenho com uma rede do tipo *Vision Tranformers*. Para fazer

[11]Destacamos a seção sobre estado da arte do site PapersWithCode `https://paperswithcode.com/task/image-classification`. Na presente data (maio de 2022), a ViT figura como uma das redes mais bem sucedidas em problemas de classificação de imagens.

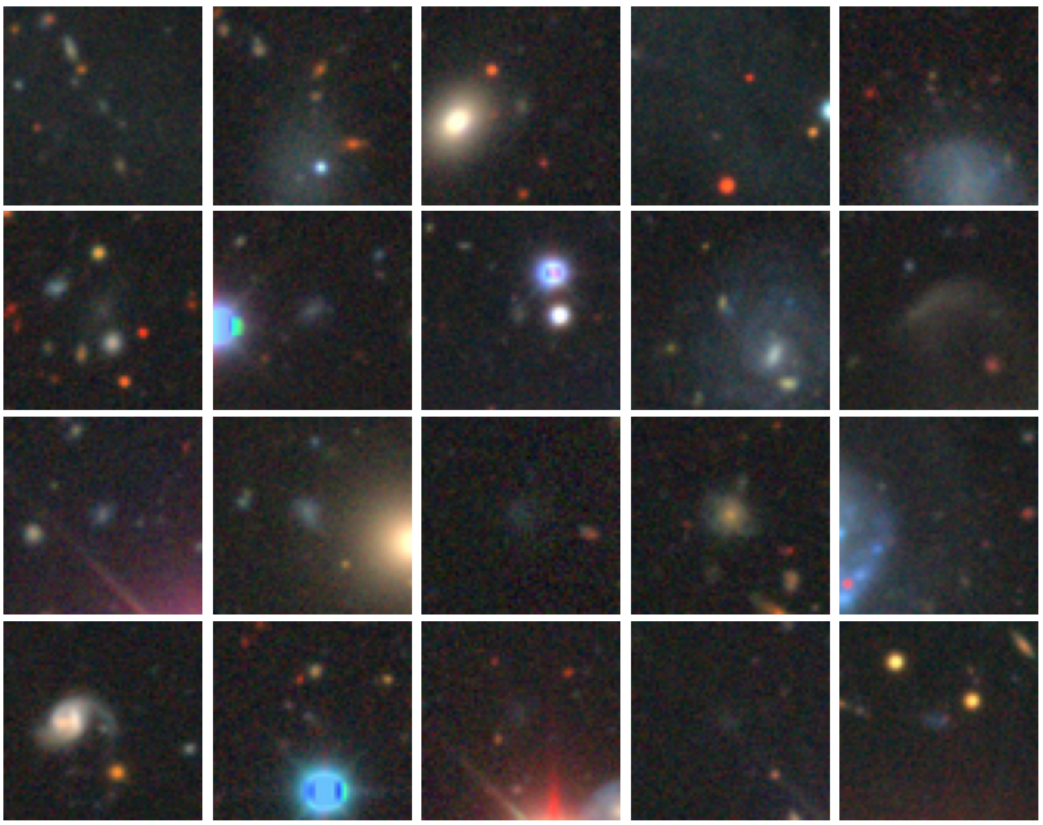

Figura 1.3: Exemplos de artefatos do conjunto de imagens do DES selecionadas por T201b.

isso, selecionamos algumas das métricas de classificação mais comumente usadas na literatura, acurácia (*accuracy*), revocação (*recall*), precisão (*precision*) e AUC, e verificamos qual rede pontua mais em cada.

Esse trabalho foi estruturado da seguinte forma. Na Seção 2.1 introduzimos o conceito de Inteligência Artificial e Aprendizado de Máquina. Na seção 2.2 apresentamos o Modelo Perceptron Multicamadas, uma das redes neurais mais simples e utilizada como camada de outras redes mais complexas. Na Seção 2.3, expomos o modelo das Redes Neurais Convolucionais. Na seção 2.4 introduzimos as redes *Vision Transformers*. Na Seção 2.5 definimos as métricas de classificação utilizadas para avaliar o desempenho das redes. Na seções 3.1 e 3.2, detalhamos a arquitetura da rede *DeepShadows* e ViT, respectivamente. Na seção 3.3 mostramos as características e a forma como foram divididas as amostras de treino, teste e validação. Na seção 3.4, apresentamos os resultados obtidos pela *Vision Transformer* e estimamos as incertezas do modelo utilizando o método de *bootstrap*. Na seção 3.5 introduzimos um método de pré-processamento que utilizamos com o propósito de melhorar o desempenho da rede. Na seção 3.6 comparamos a rede ViT com a CNN *DeepShadows*. No capítulo 4 discutimos os resultados e apresentamos questionamentos e perspectivas futuras desse trabalho.

Capítulo 2

Fundamentação Teórica

Nessa capítulo apresentamos uma breve explicação sobre os conceitos desde Inteligência Artificial até Aprendizado Profundo. Além disso, mostramos a teoria por trás dos modelos *Vision Transformers* e Redes Neurais Convolucionais, assim como o Perceptron Multicamadas, que é utilizado pelos dois anteriores. Também expomos as principais métricas de classificação, que serão utilizadas para avaliar o desempenho das redes no próximo capítulo.

2.1 Inteligência Artificial

A Inteligência Artificial pode ser definida como a tentativa de automatizar tarefas intelectuais normalmente realizadas por humanos (Chollet, 2017). Dentro desse paradigma, podemos incluir um subconjunto de técnicas conhecidas como Aprendizado de Máquina, cujo objetivo consiste em resolver determinado problema sem ter sido programado explicitamente para tal. Em outras palavras, segundo Chollet (2017): *"É a procura por representações úteis de dados de entrada dentro de um espaço pré-definido de possibilidades utilizando um sinal de feedback como guia"*. Vale pontuar que existem duas principais abordagens dentro do Aprendizado de Máquina: o aprendizado supervisionado, que consiste em treinar um algoritmo fornecendo dados que já têm um rótulo pré-definido e o aprendizado não supervisionado, que consiste em analisar e agrupar dados que não têm um rótulo pré-definido.

No âmbito do paradigma do Aprendizado de Máquina, podemos incluir mais um subconjunto: o das Redes Neurais Artificiais (ANNs, do inglês *Artificial Neural Networks*). Como o nome sugere, diversos aspectos dessas redes foram inspiradas no cérebro humano. Em particular, os neurônios artificiais, também chamados de perceptrons, cuja inspiração remete aos neurônios biológicos.

As ANNs podem ser usadas para resolver desde problemas bastante simples até problemas mais complexos (como a classificação de imagens), que requerem muitas camadas de neurônios artificiais. Em se tratando de problemas com alto grau de complexidade, estamos lidando com a chamada Aprendizagem Profunda (*Deep Learning*), que utiliza redes neurais com muitas camadas, o que possibilita

modelar padrões complicados. Convém destacar que as redes ViT e CNN estudadas nesse trabalho estão inseridas dentro desse paradigma de Aprendizado Profundo. Na Figura 2.1, mostramos um esquema com a hierarquia desses conceitos relacionados à Inteligência Artificial.

Figura 2.1: Diagrama mostrando a hierarquia dos conceitos introduzidos na seção.

Um dos exemplos mais clássicos de redes neurais é o Perceptron Multicamadas, particularmente interessante por ser utilizado na arquitetura de redes mais complexas, como as CNNs e ViTs (vale mencionar que todas essas redes são exemplos de algoritmos de aprendizado supervisionado). Uma apresentação mais detalhada deste modelo, baseada em Lanusse et al. (2017), é dada a seguir.

2.2 Modelo Perceptron Multicamadas

Apesar de ser uma das formas mais simples de ANNs, o Perceptron Multicamadas (MLP, do inglês *Multilayer Perceptron*) é uma ferramenta poderosa para modelar problemas. Essa rede pode ser utilizada de maneira eficiente desde aplicações simples como modelar o operador lógico "OU", até tarefas mais complexas como classificar imagens simples, como as da base de dados *Modified National Institute of Standards and Technology database* (MNIST)[1], um conjunto de imagens de dígitos

[1] `http://yann.lecun.com/exdb/mnist/`

escritos a mão. Vale mencionar, os MLPs são aproximadores universais, capazes de aproximar, à qualquer grau de acurácia, uma função Bore1 Mensurável de um espaço de dimensão finita até outro (Hornik et al., 1989).

O MLP pode ser dividido em diversas camadas, cada uma constituída por diversos neurônios, de forma que cada camada se conecta à camada anterior e à seguinte. Dessa forma, a saída (*output*) de cada uma é computada fazendo a combinação linear da saída da camada anterior e de um conjunto de pesos associados. Em seguida, é aplicada uma função de ativação, uma função não-linear utilizada para que a rede consiga modelar comportamentos mais complexos. As funções de ativação mais utilizadas são a ReLU (do inglês *Rectified Linear Unit*), dada por $ReLU(x) = max(0, x)$, e a Função Sigmoide (*Sigmoid Function*), dada por $\sigma(x) = \frac{1}{1+e^{-x}}$.

Usualmente representamos a arquitetura do MLP por meio de um grafo, onde cada neurônio é representado por um vértice e cada aresta representa uma ligação entre os neurônios. Na Figura 2.2, podemos ver uma representação em forma de grafo de um MLP com 4 neurônios de entradas, 2 neurônios de saída e 4 camadas, sendo 1 camada de entrada, 2 camadas escondidas (usadas para processar os dados) e 1 camada de saída.

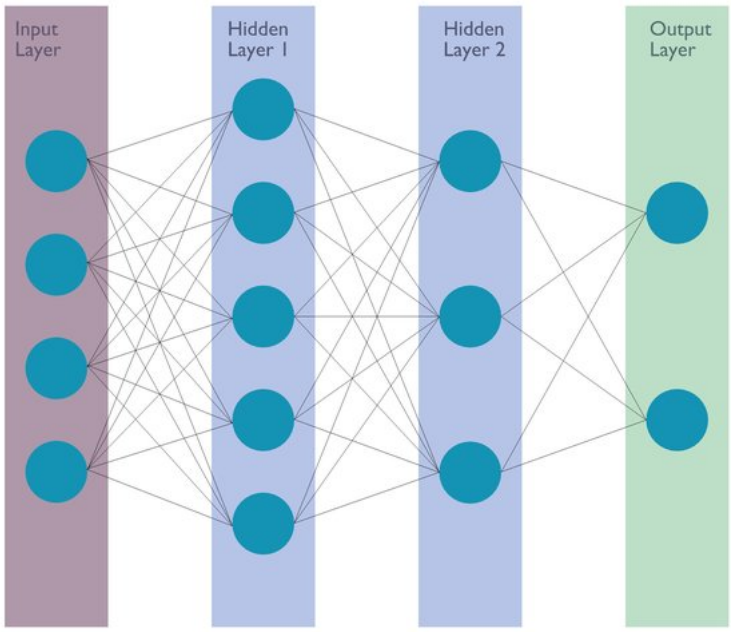

Figura 2.2: Representação de uma rede MLP com 4 camadas (sendo 1 camada de entrada, 2 camadas escondidas e 1 camada de saída) na forma de grafo. Imagem retirada de Yazdani Abyaneh et al. (2018).

De forma mais detalhada, seja L o número de camadas de um MLP, definimos $\mathbf{h}^\ell$ como a saída (*output*) da camada $\ell \in \{1, ..., L\}$ e N_ℓ como o número de neurônios da camada ℓ. Então, se $\ell \geq 1$, podemos escrever a saída de cada camada como função da saída da camada anterior, temos que

$$\mathbf{h}^\ell = f(\mathbf{W}^\ell \mathbf{h}^{\ell-\mathbf{1}} + \mathbf{b}^\ell), \tag{2.1}$$

onde $\mathbf{W}^\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ é a matriz peso dos neurônios da camada ℓ, $\mathbf{b}^\ell \in \mathbb{R}^{N_\ell}$ é um vetor de viéses aditivos (*bias vector*), $\mathbf{h}^\ell$ é o vetor ℓ-ésima camada de neurônios e $f : \mathbb{R}^{N_\ell} \to \mathbb{R}^{N_\ell}$ é a função de ativação.

Particularmente, em problemas de classificação usando Aprendizado de Máquina, dados N exemplos representados por um vetor de características (*feature vector*) $\mathbf{x} = \{x_i : i = 1, 2, .., N\}$ com rótulos (*labels*) $\mathbf{Y} = \{Y^i : i = 1, 2, .., N\}$ associados, definimos $\mathbf{h}^0 = \mathbf{x}$ como a entrada (*input*) da rede neural e $\mathbf{h}^L = \mathbf{Y_{pred}}$ como a saída (*output*) da rede, onde $\mathbf{Y_{pred}} = \{Y^i_{pred} : i = 1, 2, .., N\}$ são os rótulos previstos pela rede para cada exemplo. No caso em que que Y^i toma valores $\{-1, 1\}$ ou $\{1, 0\}$, temos um problema de 2 classes (como é o caso deste trabalho, com as classes LSBG e artefato).

Em seguida, definimos a função perda (*loss function*), utilizada para quantificar o erro entre a saída da rede e os rótulos verdadeiros dos exemplos de entrada, assim é uma forma de indicar o quão longe os rótulos preditos pela rede estão dos rótulos verdadeiros. É importante pontuar que existem diversos tipos de funções perda, de forma que, em geral, para diferentes problemas são utilizadas diferentes funções.

Nosso propósito agora consiste em otimizar os parâmetros $\{\mathbf{W}, \mathbf{b}\}$ para minimizar a função perda, de forma que tenhamos a saída $\mathbf{h}^L = \mathbf{Y_{pred}}$ o mais próximo possível de $\mathbf{Y}$. Ou seja, queremos otimizar uma função, tal que dado um conjunto de exemplos $\mathbf{x}$ como entrada é capaz de prever as classes $\mathbf{Y}$ dos objetos associados.

Para otimizar os parâmetros $\{\mathbf{W}, \mathbf{b}\}$ e minimizar uma dada função perda utiliza-se o algoritmo do Gradiente Descendente Estocástico (SGD, do inglês *Stochastic Gradient Descent*), que atualiza iterativamente os pesos da rede utilizando passos de gradiente calculados sobre um subconjunto aleatório do conjunto de treinamento.

Para calcular esses gradientes é utilizado o algoritmo de retropropagação (*back-propagation*), que toma a derivada da função perda em relação aos parâmetros do modelo. Graças ao fato que o gradiente da camada ℓ pode ser calculado utilizando o gradiente da camada $\ell + 1$, esse algoritmo pode ser aplicado de forma relativamente simples. Assim, vale mencionar que os gradientes do modelo são computados desde a última camada da rede $\ell = L$ e propagados até a primeira camada $\ell = 1$, justamente por isso, o algoritmo é chamado de retroprogação.

Quando treinadas de maneira eficiente, essas redes deveriam apresentar um desempenho cada vez melhor de acordo com o número de camadas. Na prática, isso nem sempre é verdade, uma vez que redes muito profundas podem apresentar o problema de *vanishing gradients*, que consiste em um gradiente que decai excessivamente ao se retropropagar. Assim, os pesos associados às camadas inicias da rede não são atualizados de maneira eficiente. Esse foi um dos principais problemas que as ANNs enfrentavam quando surgiram. Ocorre que hoje, com conjuntos de dados maiores, processamento computacional mais robusto e novos procedimentos eficientes de treino de arquiteturas profundas, as redes profundas se tornaram alguns dos algoritmos mais importantes da computação.

Vale mencionar também dois problemas típicos no processo de treinamento de ANNs: o *overffiting*, que consiste em uma rede demasiadamente adaptada ao conjunto de treinamento, de forma que não consegue generalizar o seu aprendizado para o conjunto de teste, obtendo um desempenho muito inferior ao esperado. Por outro

lado, temos o *underfitting*, que consiste em uma rede que não se adaptou o suficiente ao conjunto de treinamento, de forma que poderia ter aprendido mais.

2.3 Redes Neurais Convolucionais

As Redes Neurais Convolucionais são uma classe de ANNs comumente aplicadas em problemas de visão computacional, área da computação preocupada com o processamento de imagens para gerar informação relevante. Conforme o nome da rede indica, a operação de convolução, que consiste em aplicar filtros sobre uma imagem, está no cerne dessa arquitetura. A saída da operação produz uma série de mapas de características associadas a cada filtro. A Figura 2.3 exemplifica a operação de convolução em uma imagem com 3 canais.

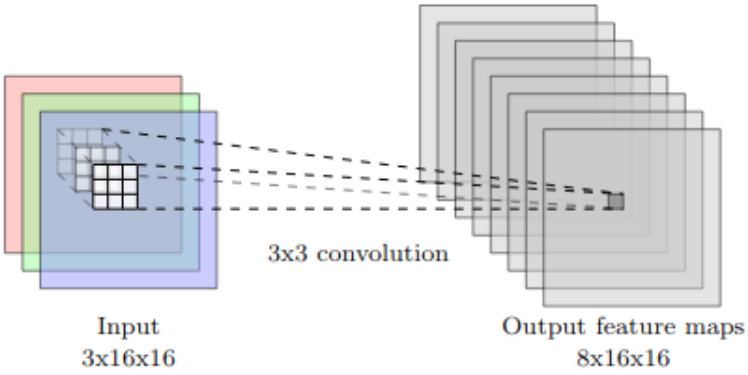

Figura 2.3: Ilustração da operação de convolução em uma imagem. Um filtro de tamanho 3×3 é aplicado em uma imagem de entrada com dimensão 16×16 e 3 canais. A saída da operação é um conjunto de 8 mapas de características de dimensão 16×16 cada um. Imagem retirada de Lanusse et al. (2017).

Na Figura 2.4 podemos ver uma ilustração do funcionamento de uma CNN. Fornecemos uma imagem como entrada e aplicamos a operação de convolução. Em seguida, usamos um filtro de *pooling*, para resumir a informação contida nas imagens, reduzindo a sua dimensão, de forma a reduzir também o processamento computacional. Em geral, utilizam-se várias camadas de convolução intercaladas com camadas de *pooling*, assim a rede converte a imagem de entrada em diversos mapas de características com resolução progressivamente menor e grau de abstração progressivamente maior.

Em princípio, poderíamos sempre utilizar um simples MLP para realizar a classificação de imagens, mas esse processo demandaria muito processamento computacional, tornando-se uma alternativa ineficiente. Além disso, de acordo com Chollet (2017) as CNNs possuem uma série de vantagens:

- Ao contrário do MLP, os padrões aprendidos são invariantes a translação.

- Aprendem hierarquias de padrões. De forma que, conforme se adicionam mais camadas de convolução, as CNNs são capazes de aprender padrões visuais cada vez mais complexos e abstratos.
- Aprendem padrões locais nas imagens, ao contrário das camadas densamente conectadas, associadas ao MLP, que aprendem padrões de maneira global.

Após aplicar última camada de *pooling* aos dados, momento em que foram extraídas as características significativas das imagens, linearizamos os dados e aplicamos um MLP, de forma que a saída final da rede nos fornecerá a classe prevista para cada imagem de entrada.

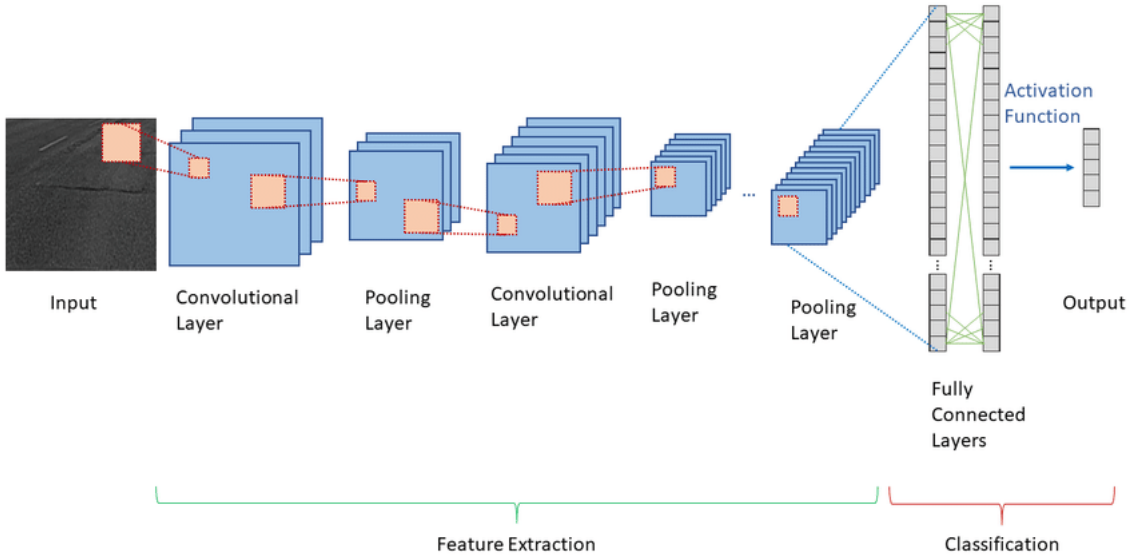

Figura 2.4: Representação da arquitetura de uma CNN. Imagem retirada de Yazdani Abyaneh et al. (2018).

De forma mais detalhada, seja L o número de camadas da rede, então para $\ell \in \{1, 2, .., L\}$ a saída da camada de convolução $\mathbf{h}^{\ell-1}$ é utilizada como entrada da camada de convolução de índice ℓ, onde adotamos a convenção que $\mathbf{h}^0$ é a entrada da rede. Em imagens quadradas, isto é, mesmo número de pixels de altura e largura, a dimensão da saída $\mathbf{h}^{\ell-1}$ é $(N_{\ell-1} \times N_{\ell-1} \times K_{\ell-1})$ onde $N_{\ell-1}$ é o número de pixels de altura/largura e $K_{\ell-1}$ o número de canais (bandas) ou de mapas de características. Para ilustrar isso, perceba que na Figura 2.3 temos $N_{\ell-1} = N_\ell = 16$, $K_{\ell-1} = 3$ para a imagem de entrada e $K_\ell = 8$ para os mapas de características. Além disso, é importante notar que a componente $k \in \{1, .., K_\ell\}$ da camada de convolução $\mathbf{h}^\ell \in \mathbb{R}^{N_\ell \times N_\ell \times K_\ell}$ é dada por:

$$\mathbf{h}_k^\ell = f\left(\sum_{k'} \mathbf{W}_{k',k}^\ell * \mathbf{h}_{k'}^{\ell-1} + \mathbf{b}_k^\ell \right), \tag{2.2}$$

onde f é a função de ativação (usada para introduzir não linearidades na rede, assim como no MLP), $\mathbf{W}_{k',k}^\ell \in \mathbb{R}^{K_\ell \times K_{\ell-1} \times I_\ell \times I_\ell}$ é o núcleo de convolução (*convolution*

kernel), que contém um filtro de tamanho $I_\ell \times I_\ell$ (geralmente restrito a bem poucos pixels, como na Figura 2.3 com $I_\ell = 3$) para cada combinação de canais de entrada e saída. Por fim, assim como no MLP, $\mathbf{b}^\ell$ é a ℓ-ésima componente de um vetor de viéses aditivos (*bias vector*), que será otimizado pelo algoritmo SGD junto com os pesos $\mathbf{W}$.

2.4 Vision Tranformers

A rede *Transformer* foi introduzida em Vaswani et al. (2017), para resolver problemas de linguagem natural, tais como a tradução automática e o resumo de textos. Conforme explicam esses autores, a *Transformer* se destaca por ter sido o primeiro modelo de transdução[2] que, para processar os dados, depende unicamente do mecanismo de atenção própria (*self-attention*), um mecanismo que relaciona posições em um conjunto de dados organizado de maneira sequencial, como em uma frase, para computar uma representação da sequência.

Mais concretamente, quando a rede *Tranformers* é aplicada em problemas de linguagem natural, por exemplo para a tradução de frases, a rede computa a relação de palavras de forma paralelizada, processando as palavras da frase como um todo, ao contrário das redes neurais recorrentes[3], que processam os dados de maneira sequencial, processando palavra por palavra da frase.

A partir da *Transformer*, surgiu um novo tipo de rede: a *Vision Transformer* (ViT), desenvolvida por Dosovitskiy et al. (2021), com funcionamento bastante similar ao de sua predecessora, também baseada no mecanismo de *self-attention*, mas agora adaptado para lidar com problemas de Visão Computacional.

É interessante mencionar que Dosovitskiy et al. (2021) constatou que a ViT, treinada em amostras suficientemente grandes, é capaz de obter resultados ainda melhores que os obtidos pelas CNNs, até então consideradas o estado da arte em problemas de classificação de imagem. Dessa forma, embora a ViT possa não performar tão bem quanto as CNNs quando as amostras são pequenas, a medida que os conjuntos de dados ficam maiores, ela parece se tornar uma alternativa mais interessante. Uma apresentação detalhada da rede ViT, baseada em Dosovitskiy et al. (2021) e Vaswani et al. (2017), é dada a seguir.

Na Figura 2.5 temos uma visão geral da rede. Como entrada temos uma imagem $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$, esta é dividida em uma sequência de frações de imagem (*image patches*) bidimensionais $\mathbf{x}_p \in \mathbb{R}^{N\times(P^2\cdot C)}$, onde H é a altura, W a largura, C o número de canais, (P, P) é a resolução dos *patches* e $N = HW/P^2$ é o número de *patches*. Na Figura 2.6, exemplificamos a divisão de uma imagem em *patches*.

[2]Modelos de Aprendizado de Máquina que mapeiam sequências de entrada em sequências de saída.

[3]Redes neurais que processam dados organizados de maneira sequencial, como frases ou séries temporais. Uma característica importante é a sua capacidade de armazenar memória sobre os dados de entrada, de maneira que ao processar uma entrada, outros dados de entrada anteriores e posteriores ao atual poderão influenciar na saída em questão. Em uma frase por exemplo, a rede processa palavra por palavra, checando a relação de cada uma com as outras.

Em seguida, vamos linearizar esses *patches* obtendo vetores de tamanho fixo D, constante em todas camadas da rede. Dessa forma, para cada *patch* aplicamos uma transformação linear, treinada pela rede, $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$, obtendo o conjunto os vetores $\mathbf{x}_p^j \mathbf{E} \in \mathbb{R}^D$ com $j \in \{1, ..., N\}$. Na Figura 2.5 podemos ver a divisão da imagem de uma galáxia em *patches* e sua linearização em vetores (camada amarela).

Em seguida, definimos a incorporação aprendida (*learnable embedding*) $\mathbf{x}_{class} \in \mathbb{R}^D$, usada para estimar a saída da rede durante o processo treinamento. Além disso, associamos uma incorporação de posição (*position embedding*) $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$, um índice para assinalar em qual parte da imagem cada fração está posicionada, aos vetores $\mathbf{x}_{class}; \mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; ...; \mathbf{x}_p^N \mathbf{E}$.

Com isso, obtemos a incorporação de fração (*patch embedding*) $\mathbf{z}_0$, que guarda o conteúdo e a posição dos *patches* e servirá de entrada para o *Transformers Encoder*. Definimos o *patch embedding* como

$$\mathbf{z}_0 = [\mathbf{x}_{class}; \mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; ...; \mathbf{x}_p^N \mathbf{E}] + \mathbf{E}_{pos}. \tag{2.3}$$

Na Figura 2.5, acima da camada amarela, que simboliza os patches linearizados, podemos ver o *patch embedding*. Temos os vetores, associados aos *patches* e ao *learnable embedding*, ilustrados nas elipses em branco. Ao lado esquerdo de cada uma, estão indicadas, nas elipses coloridas numeradas, as componentes do *position embedding* correspondentes a cada vetor.

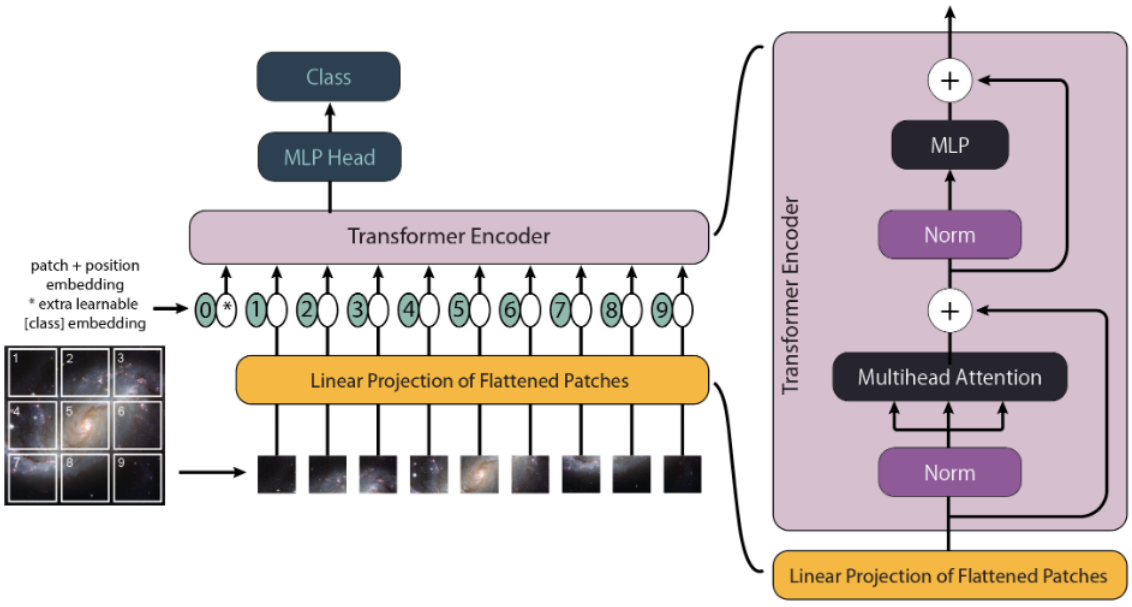

Figura 2.5: Representação da arquitetura de uma rede ViT. A imagem é separada em 9 *patches*, que são transformados em vetores linearizados. Em seguida acrescenta-se um *learnable embedding* e um *position embeding* associado a cada vetor. Por fim, esses dados são processados pelo *Tranformers Encoder* e usados como entrada de um MLP, cuja saída nos dará a classe do objeto em questão. Imagem retirada de Yao-Yu Lin et al. (2021).

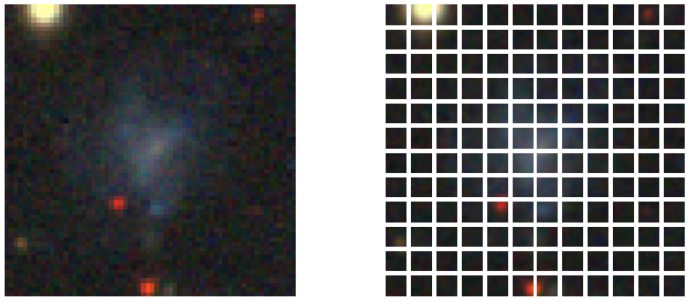

Figura 2.6: Demonstração do processo de divisão de uma imagem em 144 *Patches* para ser processada na ViT.

Em seguida, o *patch embedding* será utilizado como entrada da principal parte do algoritmo, o *Transformer Encoder*[4], que irá processar o *patch embedding* e linearizar os dados para serem enviados como entrada de um Perceptron Multicamadas.

É interessante pontuar que a saída do *Transformer Encoder* associada a incorporação aprendida $\mathbf{z}_L^0$ nos dará a classe prevista pela rede, enquanto as saídas associadas aos outros vetores ($\mathbf{z}_L^j$, onde $j = \{1, .., N\}$) serão ignoradas, já que elas não têm nenhum significado trivial.

Para descrever o funcionamento do *Transformer Encoder*, ilustrada no lado direito da Figura 2.5, iremos introduzir dois mecanismos: a atenção própria (*self-attention*) e a atenção própria multicabeças (*multihead self-attention*).

O mecanismo de *self-attention*, ilustrado no lado esquerdo da Figura 2.7, funciona da seguinte forma. Para cada entrada $\mathbf{z} \in \mathbb{R}^{N \times D}$, associamos três vetores, *query* $\mathbf{q}$, *key* $\mathbf{K}$ e *value* $\mathbf{v}$, de forma que

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{z}\mathbf{U}_{qkv}, \tag{2.4}$$

onde $\mathbf{U}_{qkv} \in \mathbb{R}^{D \times 3D_h}$ é uma matriz de pesos que será treinada pela rede e D_h é a dimensão dos vetores $\mathbf{q}, \mathbf{k}$ e $\mathbf{v}$.

Realizamos o produto escalar entre *query* e *key* transposto, corrigimos pelo fator $\sqrt{D_h}$ e aplicamos a função *softmax* em cada linha da matriz $\mathbf{q}\mathbf{k}^T/\sqrt{D_h}$, onde softmax : $\mathbb{R}^N \to \mathbb{R}^N$ é definida através de sua i-ésima componente por $softmax(\mathbf{x})_i = e^{x_i}/(\sum_{j=1}^{N} e^{x_j})$. Assim obtemos

$$A = softmax(\mathbf{q}\mathbf{k}^T/\sqrt{D_h}), \tag{2.5}$$

onde $A \in \mathbb{R}^{N \times N}$. De forma que os elementos A_{ij} são baseados na similaridade de cada par de elementos da sequência e nas suas respectivas representações $\mathbf{q}^i$ e $\mathbf{k}^j$. Finalmente, definimos a função *self-attention* como

$$SA(\mathbf{q}, \mathbf{k}, \mathbf{v}) = A\mathbf{v}. \tag{2.6}$$

[4]As arquiteturas do tipo encoder-decoder são comumente utilizadas em redes *seq2seq*, que convertem uma sequência de entrada em uma sequência de saída, tais como a rede *Tranformer*. Na arquitetura da ViT, é utilizado um *encoder*, com o propósito de converter imagens em representações latentes de dimensão inferior, idêntico ao da rede *Transformer* e não é utilizado nenhum *decoder*, mecanismo que converte as representações latentes gerada por um *encoder* em uma sequência de saída.

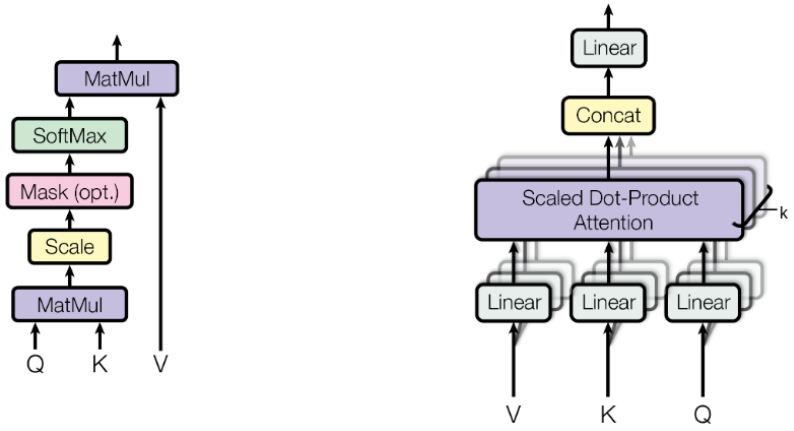

Figura 2.7: No lado esquerdo, temos um esquema mostrando o mecanismo de *self-attention*: realizamos o produto escalar dos query $\mathbf{q}$ e key transposto $\mathbf{k}^T$, ajustamos pelo fator $\sqrt{D_h}$ e aplicamos a função *softmax*, com isso, obtemos a matriz A. Por fim, realizamos a multiplicação matricial de A e $\mathbf{v}$. No lado direito, temos uma ilustração do mecanismo de *multihead self-attention*, os parâmetros de entrada $\mathbf{q}$, $\mathbf{k}$, $\mathbf{v}$ são projetados linearmente, de forma que obtemos k componentes para cada parâmetro de entrada. Então, aplicamos a função *self-attention* a essas componentes, concatenamos as saídas e linearizamos tudo novamente. Imagem adaptada de Vaswani et al. (2017).

No mecanismo *multihead self-attention* (ilustrado no lado direito da Figura 2.7), definimos 3 conjuntos de k matrizes, contendo pesos treinados pela rede, $W_i^{\mathbf{q}}$, $W_i^{\mathbf{k}}$, $W_i^{\mathbf{v}} \in \mathbb{R}^{D \times D_h}$ com $i \in \{1, ..., k\}$. Note que cada conjunto de matrizes está associada aos parâmetros $\mathbf{q}$, $\mathbf{k}$ e $\mathbf{v}$, respectivamente. Em seguida, projetamos linearmente as entradas, ao multiplicar $\mathbf{q}$, $\mathbf{k}$ e $\mathbf{v}$ pelas suas respectivas matrizes associadas, ou seja, obtemos $\mathbf{q}W_i^{\mathbf{q}}, \mathbf{k}W_i^{\mathbf{k}}, \mathbf{v}W_i^{\mathbf{v}}$. Em seguida, aplicamos k vezes a função *self-attention* em paralelo a esses parâmetros $\mathbf{q}W_i^{\mathbf{q}}, \mathbf{k}W_i^{\mathbf{k}}, \mathbf{v}W_i^{\mathbf{v}}$ e linearizamos tudo novamente, ao multiplicar essas saídas concatenadas por outra matriz de pesos treinados pela rede, $W^O \in \mathbb{R}^{k \cdot D_h \times D}$. De forma mais sucinta, temos:

$$MSA(\mathbf{q}, \mathbf{k}, \mathbf{v}) = [head_1; head_2; ...; head_k]W^O, \tag{2.7}$$

onde

$$head_i = SA_i(\mathbf{q}W_i^{\mathbf{q}}, \mathbf{k}W_i^{\mathbf{k}}, \mathbf{v}W_i^{\mathbf{v}}). \tag{2.8}$$

Já o *Transformer Encoder* é composto por camadas *multihead self-attention* alternadas com Perceptron Multicamadas. Podemos visualizar isso no lado direito da Figura 2.5, que mostra a arquitetura do *Transformer Encoder*.

Além disso, utilizamos uma função *NORM* para normalizar os dados e melhorar o treinamento em termos de tempo de processamento computacional (Ba et al., 2016). Esta função é utilizada no *Transformer Encoder* e indicada no lado direito da Figura 2.5, podemos defini-la como

$$NORM(v) = \gamma \frac{v - \mu}{\sigma} + \beta, \tag{2.9}$$

onde μ é a média, σ é o desvio padrão do vetor v, γ e α são parâmetros dados (Xiong et al., 2020).

Dessa forma, a rede *Vision Transformer* é descrita por:

$$\mathbf{z}_0 = [\mathbf{x}_{class}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; ...; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos} \tag{2.10}$$

$$\mathbf{z'}_\ell = MSA(NORM(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1} \tag{2.11}$$

$$\mathbf{z}_\ell = MLP(NORM(\mathbf{z'}_\ell)) + \mathbf{z'}_\ell \tag{2.12}$$

$$\mathbf{y} = NORM(\mathbf{z}_L^0), \tag{2.13}$$

É importante salientar que a ViT é uma rede bastante dependente do tamanho do conjunto de dados de treinamento. Em conjuntos de dados pequenos, sua performance costuma ser pior que o estado da arte das CNNs. Por outro lado, em grandes conjuntos de dados, ela é capaz de obter resultados superiores.

Para demonstrar isso, Dosovitskiy et al. (2021) utilizou um método, chamado de pré-treinamento, que é capaz de melhorar a performance da rede. Esse método consiste em, ao invés de iniciar o treinamento da rede com pesos aleatórios, pré-treinar a rede em um conjunto de dados A, a fim de obter uma estimativa dos pesos que serão utilizados para treinar a rede no conjunto de treinamento B, que é o conjunto de dados que queremos fazer a classificação, ou seja, o conjunto que realmente é de nosso interesse. Essa última etapa de treinamento com o conjunto B é chamada de *fine tunning*.

Assim, Dosovitskiy et al. (2021) realizou o pré-treinamento da ViT em 3 conjuntos de dados (com imagens de diversas classes como animais, alimentos, veículos, entre outros) e tamanhos progressivamente maiores: *ILSVRC-2012 ImageNet* (ou somente *ImageNet*), que contém 1000 classes e 1.3 M imagens, ImageNet-21k (Deng et al., 2009) (conjunto que contém o *ImageNet*), com 21000 classes e 14M imagens, e JFT-300M com 8000 classes e 303M imagens. Por fim, foi feito o *fine tuning* usando o conjunto de dados *ImageNet*.

Conforme indica o lado esquerdo da Figura 2.8, a ViT só alcança uma acurácia superior a rede *Big Transfer* (BiT), uma versão de CNN, estado da arte do conjunto *ImageNet*, quando o pré-treinamento é realizado no JFT-300M, o maior conjunto utilizado para pré-treinamento. Vale pontuar que foram utilizadas diferentes arquiteturas da rede ViT (contendo diferentes números de camadas, números de neurônios associados ao MLP, número de cabeças k associadas ao *multithead self-attention* e tamanho do *patches* – ver tabela 1 de Dosovitskiy et al. (2021)), chamadas ViT-B/32, ViT-B/16, ViT-L/32,ViT-L/16, ViT-H/14.

Paralelamente, foi feito o pré-treinamento em subconjuntos aleatórios com 9M, 30M e 90M de imagens do JFT-300M, além do conjunto inteiro, que contém 24000 classes. Em seguida, foi realizado o *fine tunning* no *ImageNet*. Novamente, as CNNs obtiveram uma melhor performance quando foram utilizados conjuntos menores de pré-treinamento e pior performance quando foram utilizados conjuntos maiores de

pré-treinamento. A saber, a ViT só alcançou resultados melhores que a CNN nos conjuntos com mais de 90M de exemplos. Os resultados desse experimento podem ser visualizados no lado direito da Figura 2.8.

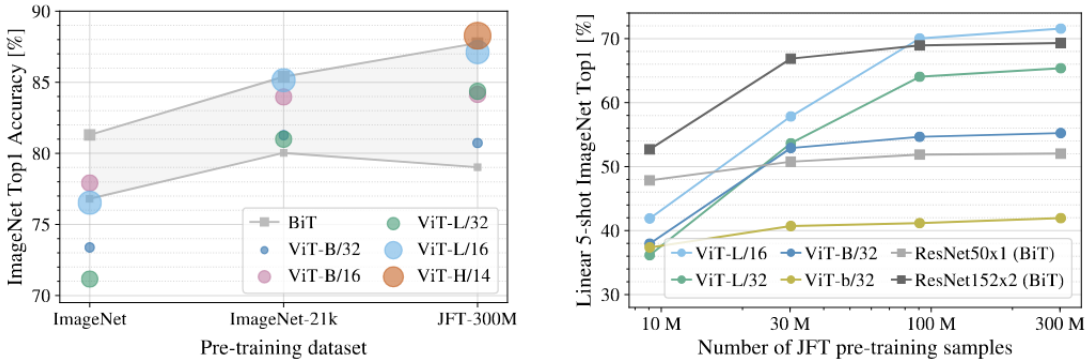

Figura 2.8: Painel esquerdo: Acurácia das redes ViT (diversas arquiteturas) e BiT (uma CNN) quando pré-treinadas nos conjuntos progressivamente maiores ImageNet, ImageNet-21k e JFT-300M. Painel direito: Acurácia das redes ViT (diversas arquiteturas) e BiT (2 arquiteturas) quando pré-treinadas em subconjuntos aleatórios do JFT-300M, usando a técnica de *few-shot learning*, que consiste em treinar uma rede com poucos exemplos. As imagens demonstram que as ViTs são capazes de alcançar resultados cada vez melhores de acordo com o tamanho do conjunto de dados, quando os conjuntos são suficientemente grandes, elas são capazes de superar as CNNs. Imagem retirada de Dosovitskiy et al. (2021).

2.5 Métricas de Classificação

Nesta seção introduzimos as métricas de classificação, utilizadas para quantificar a eficiência de cada algoritmo. Primeiramente iremos apresentar algumas definições preliminares.

Dado um exemplo (no nosso caso uma imagem) de índice i, adotamos a convenção que o rótulo $Y^i = 1$ indica uma LSBG. Analogamente se $Y^i = 0$, dizemos que o exemplo é um artefato.

Analogamente ao caso anterior, $Y^i_{pred} = 1$ denota que a rede classificou o exemplo i como uma LSBG e $Y^i_{pred} = 0$ indica que a rede classificou o exemplo como artefato.

Desta forma, definimos o conjunto de verdadeiros positivos como exemplos positivos corretamente classificados, isto é, LSBGs classificadas como LSBGs. De forma análoga, definimos os verdadeiros negativos como artefatos classificados como artefatos, falsos positivos como artefatos classificados como LSBGs, falsos negativos como LSBGs classificados como artefatos. A partir disso, definimos TP, TN, FP e FN como o número de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos, respectivamente. Na Figura 3.5, podemos ver exemplos de Falsos Negativos.

Definimos um limite (*threshold*) T, que atua como uma probabilidade de corte. Assim, se a saída da rede P^i_{pred} referente ao índice i, a probabilidade do objeto de índice i ser da classe LSBG, for maior que o limite T, então ele é classificado como LSBG. Assim, $P^i_{pred} > T \Rightarrow Y^i_{pred} = 1$, caso contrário o objeto é classificado como artefato, isto é, $P^i_{pred} < T \Rightarrow Y^i_{pred} = 0$.

Neste caso, convencionalmente, adota-se $T = 0.5$, ou seja, classifica-se o objeto de acordo com a classe que a rede prevê como mais provável. No entanto, poderíamos arbitrar qualquer valor $T \in [0, 1]$. Por exemplo, se quissemos ser mais conservadores e apenas classificar como LSBGs aquelas imagens que a rede tem muita confiança, poderíamos arbitrar $T > 0.5$. A partir dessas definições, podemos introduzir as métricas de classificação, comumente utilizadas para medir o desempenho da rede.

A revocação (*recall*), também chamada de taxa de verdadeiros positivos (*True Positive Rate*), é o número de verdadeiros positivos sobre o total de positivos:

$$revocação = TPR = \frac{TP}{TP + FN}. \tag{2.14}$$

Essa métrica corresponde à fração de exemplos positivos que foram identificados corretamente.

Por outro lado, a taxa de falsos positivos (*False Positive Rate*):

$$FPR = \frac{FP}{FP + TN}, \tag{2.15}$$

corresponde à fração de objetos que foram classificados erroneamente como positivos sobre o total de negativos. Na Figura 3.4, podemos ver exemplos de Falsos Positivos.

Conforme mostramos anteriormente, para cada limite T, cada exemplo recebe um rótulo, de forma que para todo T, existem TPR e FPR associados. Assim, definimos a curva ROC (do inglês *Receiving Operating Characteristic*), que traça a relação da taxa de verdadeiros positivos (TPR) com a taxa de falsos positivos (FPR) para cada T. Na Figura 2.9, podemos ver o esboço de uma curva ROC. Para cada ponto T na curva temos um valor de TPR e FPR associados.

A AUROC (do inglês *Area Under the Roc Curve*, também chamada de AUC, é a área abaixo da curva ROC. Uma rede ideal, que acerta 100% dos casos, tem $AUC = 1$. Por outro lado, uma rede incapaz de aprender qualquer padrão tem $AUC \approx 0.5$ (em um problema de 2 classes). Assim. ela só acerta com base no acaso e é incapaz de distinguir as classes. Por fim, uma rede com $AUC = 0$ é uma rede que erra todas as vezes. Nesse caso, sempre que um exemplo é da LSBG, a rede classifica como Artefato e vice-versa. Portanto, ao treinar uma ANN, buscamos uma AUC o mais próximo de 1 possível.

Definimos a precisão (*precision*) como o número de verdadeiros positivos sobre os exemplos classificados como LSBGs:

$$precisão = \frac{TP}{TP + FP}. \tag{2.16}$$

Essa métrica corresponde à fração de identificações positivas que de fato estavam corretas.

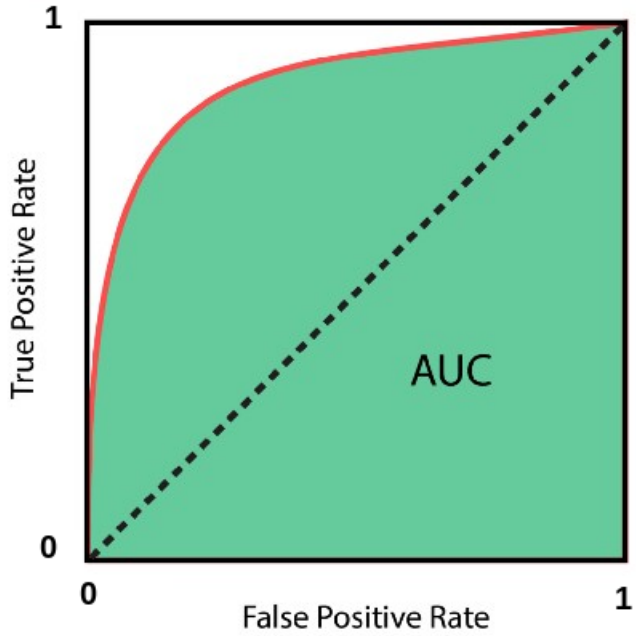

Figura 2.9: Esboço da Curva ROC (em vermelho). No eixo das abcissas temos a taxa de falsos positivos e nas ordenadas a taxa de verdadeiros positivos. A área em verde indica a AUC, área abaixo da curva ROC, uma das métricas utilizadas nesse trabalho. A linha tracejada delimita a área com $AUC = 0.5$, correspondente a um classificador aleatório. Imagem adaptada de `https://towardsdatascience.com/roc-curve-a-complete-introduction-2f2da2e0434c`.

A acurácia (*accuracy*) é definida como a razão de exemplos corretamente classificados sobre o conjunto total:

$$acurácia = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.17}$$

Note que diferente da precisão e revocação, quando calculamos a acurácia, não estamos interessados no tipo de erro da classificação, falso positivo ou falso negativo. Assim, a acurácia mede o desempenho geral da rede.

Capítulo 3

Resultados

Nessa seção detalhamos as arquiteturas das redes ViT e *DeepShadows*, a CNN utilizada como *benchmark* deste trabalho. A partir de uma série de testes empíricos, feitos para tentar determinar os melhores hiperparâmetros para treinar a ViT, apresentamos os valores das melhores métricas de classificação obtidas e suas incertezas associadas, estimadas pelo método de *bootstrap*. Além disso, mostramos uma série de gráficos, utilizados para visualizar o desempenho da rede e comparamos os resultados das redes. Por fim, apresentamos um método de ajuste de contraste, implementado com o objetivo de ressaltar os objeto de baixo brilho e melhorar o desempenho das redes.

3.1 Arquitetura da rede *DeepShadows*

A CNN que escolhemos como *benchmark* foi a rede *DeepShadows*, desenvolvida por T201b. Essa escolha foi motivada pois, até onde sabemos, essa é a única rede disponível publicamente que foi desenvolvida como o mesmo propósito que o nosso: separar LSBGs de artefatos. Além disso tal rede é bem documentada e obteve excelentes resultados.

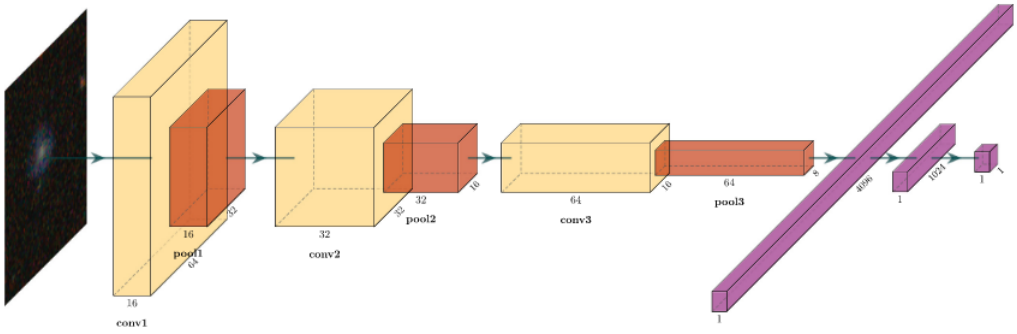

Figura 3.1: Ilustração da arquitetura da rede *DeepShadows*. Imagem retirada de T201b.

Layers	Properties	Stride	Padding	Output shape	Parameters
Input	64 × 64 × 3[a]	–	–	(64, 64, 3)	0
Convolution (2D)	Filters: 16	1 × 1	Same	(64, 64, 16)	448
	Kernel: 3 × 3	–	–	–	–
	Activation: ReLU	–	–	–	–
	Reg: L2 (0.13)	–	–	–	–
Batch normalization	–	–	–	(64, 64, 16)	64
MaxPooling	Kernel: 2 × 2	2 × 2	Valid	(32, 32, 16)	0
Dropout	Rate: 0.4	–	–	(32, 32, 16)	0
Convolution (2D)	Filters: 32	1 × 1	Same	(32, 32, 32)	4640
	Kernel: 3 × 3	–	–	–	–
	Activation: ReLU	–	–	–	–
	Reg: L2 (0.13)	–	–	–	–
Batch normalization	–	–	–	(32, 32, 32)	128
Maxpooling	Kernel: 2 × 2	2 × 2	Valid	(16, 16, 32)	0
Dropout	Rate: 0.4	–	–	(16, 16, 32)	0
Convolution (2D)	Filters: 64	1 × 1	Same	(16, 16, 64)	18 496
	Kernel: 3 × 3	–	–	–	–
	Activation: ReLU	–	–	–	–
	Reg: L2 (0.13)	–	–	–	–
Batch normalization	–	–	–	(16, 16, 64)	256
MaxPooling	Kernel: 2 × 2	2 × 2	Valid	(8, 8, 64)	0
Dropout	Rate: 0.4	–	–	(8, 8, 64)	0
Flatten	–	–	–	(4096)	–
Fully connected	Activation: ReLU	–	–	(1024)	4 195 328
	Reg: L2 (0.12)	–	–	–	–
Fully connected	Activation: Sigmoid	–	–	(1)	1025

Figura 3.2: Arquitetura da rede *DeepShadows*. Imagem retirada de T201b.

Conforme ilustrado pela Figura 3.1, a *DeepShadows* consiste de 3 camadas de convolução (em amarelo), alternadas com camadas de *pooling* (em vermelho). As camadas de convolução usam um núcleo de convolução de dimensão 3×3 e uma função de ativação ReLU (ver seção 2.2), enquanto as camadas de *pooling* usam núcleos de tamanho 2×2. Entre as camadas de convolução e *pooling* é feito uma normalização de *batch* (*batch normalization*), com o objetivo de tornar o treinamento mais rápido e estável. Após essas camadas, é aplicado um MLP com três camadas densamente conectadas (em roxo), sendo que a última camada utiliza a função de ativação sigmoide (ver seção 2.2) e é composta por um único neurônio que retorna como saída o rótulo correspondente a classe das imagens de entrada. Além disso, para evitar o *overfitting* (problema descrito brevemente na seção 2.2), é utilizada a técnica de *dropout*, que consiste em ignorar determinados neurônios da rede selecionados aleatoriamente. Os parâmetros utilizados em cada camada são mostrados na Figura 3.2.

Os principais hiperparâmetros utilizados foram: taxa de aprendizado (*learning Rate*) $LR = 0.1$, utilizada para controlar o tamanho do passo dado pela rede com o objetivo minimizar a função perda. Além disso, foram utilizadas 100 épocas, isto é, o número vezes que todo o conjunto de treinamento é propagado (e retropropagado) pela rede. O tamanho de *batch* utilizado foi 64, ou seja, a rede é atualizada iterativamente a cada 64 imagens. Por fim, a função perda utilizada foi a *binary cross-entropy*[1].

[1] `https://www.tensorflow.org/api_docs/python/tf/keras/losses/BinaryCrossentropy`

3.2 Arquitetura da rede ViT

Para implementar a nossa ViT, cujo código fonte está disponível publicamente[2], nos baseamos no código desenvolvido por Khalid Salama[3], que implementou essa rede para lidar com o conjunto de dados CIFAR-100[4], um conjunto de 60000 imagens com 100 classes. Os principais *frameworks* utilizados foram Tensorflow[5] e Keras[6], essas são ferramentas de código aberto comumente usadas para implementar algoritmos de Aprendizado de Máquina. A rede foi inicializada com pessoas aleatórios e para executar os códigos usamos o GPU do Google Collab[7][8] e para criar parte dos gráficos do trabalho nos baseamos no código desenvolvido por T201b[9].

Para alcançar o melhor resultado possível, testamos diversas variações dos principais parâmetros e hiperparâmetros da rede em busca de uma combinação que otimizasse as métricas de classificação (principalmente a acurácia). Em particular, variamos a taxa de aprendizado usando os seguintes valores $LR = 0.0005, 0.001, 0.002$. Além disso, testamos a rede para camadas de *Transformer* $L = 6, 8, 10, 12$.

Após realizar vários testes, escolhemos os seguintes hiperparâmetros. Para treinar a rede utilizamos uma taxa de aprendizado $LR = 0.002$, 25 épocas e *batch* de tamanho 256. Em termos do fracionamento de imagem em *patches* (ver seção 2.4), utilizamos 6 pixels de altura/largura para os *patches* que foram projetados em vetores de dimensão $D = 64$. Para o *Transformer Encoder*, utilizamos $L = 10$ camadas e $k = 4$ cabeças associadas ao *multihead self-attention*. Por fim, nas camadas densamente conectadas (MLP) usamos 2048 neurônios na primeira camada e 1024 neurônios na segunda camada. A função perda utilizada foi a *Sparse Categorical Cross-entropy*[10].

3.3 Amostras de Treino, Teste e Validação

O conjunto de dados utilizado nesse trabalho consiste de imagens de objetos capturados nos 3 primeiros anos (Y3) do DES que foram selecionadas por Tanoglidis et al. (2021a). Para detectar LSBGs e artefatos capturadas pelo DES, esses autores aplicaram os seguintes critérios. Primeiramente, selecionaram objetos com base no seu tamanho angular (raio de meia luz na banda g, $r_{1/2} > 2.5''$) e brilho superficial

[2] `https://github.com/Manuelstv/VIT-LSBGs`

[3] `https://keras.io/examples/vision/image_classification_with_vision_transformer/`

[4] `https://www.cs.toronto.edu/%7Ekriz/cifar.html`

[5] `https://www.tensorflow.org/`

[6] `https://keras.io/`

[7] `https://colab.research.google.com/`

[8] Vale mencionar que o processo de treino, teste e validação pode ser feito relativamente rápido, em períodos de aproximadamente uma hora.

[9] `https://github.com/dtanoglidis/DeepShadows`

[10] `https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy`

central dentro do raio efetivo na banda g ($\bar{\mu}_{eff}(g) > 24.4 mag/arcsec^2$). Em seguida, utilizaram o algoritmo SVM (*Suport Vector Machines*)[11], para classificar os objetos e realizaram uma inspeção visual sobre objetos classificados como positivos para rejeitar os falsos positivos. Por fim, aplicaram perfis de Sérsic e correção de extinção interestelar aos objetos e adotaram critérios de seleção sobre os parâmetros estruturais obtidos com essas técnicas. A partir desses procedimentos, os autores obtiveram uma amostra final de 20000 LSBGS e 20000 artefatos. As coordenadas dos objetos desta amostra foram disponibilizadas publicamente pelos autores[12].

De posse dessas coordenadas, utilizamos o *DESI Legacy Imaging Surveys Sky Viewer* (Dey et al., 2019)[13] para baixar as imagens da amostra de Tanoglidis et al. (2021a). Mais precisamente, baixamos dois conjuntos de dados, sendo que o primeiro era composto por imagens baixadas no formato PNG, de forma que esse conjunto é idêntico ao de T201b. Para o segundo conjunto, baixamos as imagens no formato FITS, padrão comumente utilizado na astronomia, fundamental para aplicar o método de ajuste de contraste (descrito na 3.5).

Portanto, utilizamos dois conjuntos de dados (com as mesmas imagens, porém em formatos diferentes), cada conjunto é formado por 39996 imagens de dimensão $(64, 64, 3)$, sendo 19996 LSBGs e as outras 20000 artefatos.

Vale pontuar que as imagens (tanto as FITS como PNG) correspondem a uma região do céu de tamanho $30'' \times 30''$. O tamanho inicial das imagens é de 256×256 pixels, que posteriormente é transformado para 64×64 pixels para reduzir o processamento computacional. As imagens possuem 3 canais, que correspondem às bandas g,r e z.

Em ambos conjuntos, foram selecionadas aleatoriamente 15000 LSBGs e 15000 artefatos para compor o conjunto de treino, outras 2500 LSBGs e 2500 artefatos para o conjunto de validação, além de 2496 LSBGs e 2500 artefatos para o conjunto de teste.

Também utilizamos um clássico mecanismo de aumento de dados (*data augmentation*) nas amostras utilizadas pela ViT. Esta técnica consiste em aplicar pequenas transformações nas imagens com o intuito de aumentar a variabilidade dos dados e otimizar o treinamento da rede. Em nosso caso, aplicamos às imagens uma rotação aleatória, um zoom aleatório de no máximo 20% tanto em relação à altura, quanto à largura, e um giro aleatório no sentido horizontal e vertical.

3.4 Resultados da ViT

Como descrito na seção 3.2, testamos diferentes combinações de parâmetros e hiperparâmetros e escolhemos como modelo padrão a rede com $LR = 0.002$ e $L = 10$, pois esta alcançou a maior acurácia e também atingiu uma alta revocação, precisão e AUC. Mais especificamente, aplicado no conjunto de imagens em formato PNG,

[11] Modelo de aprendizado supervisionado comumente utilizado para problemas de classificação.
[12] `https://github.com/dtanoglidis/DeepShadows`
[13] `https://www.legacysurvey.org/`

o modelo padrão alcançou as seguintes métricas de classificação: *acurácia* = 0.929, AUC = 0.979, *revocação* = 0.958, *precisão* = 0.905. A seguir, apresentamos uma série de gráficos para ilustrar os resultados do modelo padrão da ViT.

Na Figura 3.3 mostramos a matriz de confusão para o nosso modelo padrão, que compara as classes previstas pela rede com as classes verdadeiras dos objetos. No conjunto de 5000 imagens de teste, 2500 eram artefatos e destas, 2250 (90%) foram corretamente classificados e outros 250 (10%) eram falsos positivos, objetos classificados como LSBGs, mas que são artefatos (alguns exemplos mostrados na Figura 3.4). Já para as LSBGs do conjunto de teste, 2391 foram classificadas corretamente (96%) e 104 (4%) foram falsos negativos, LSBGs classificadas como artefatos (alguns exemplos mostrados na Figura 3.5).

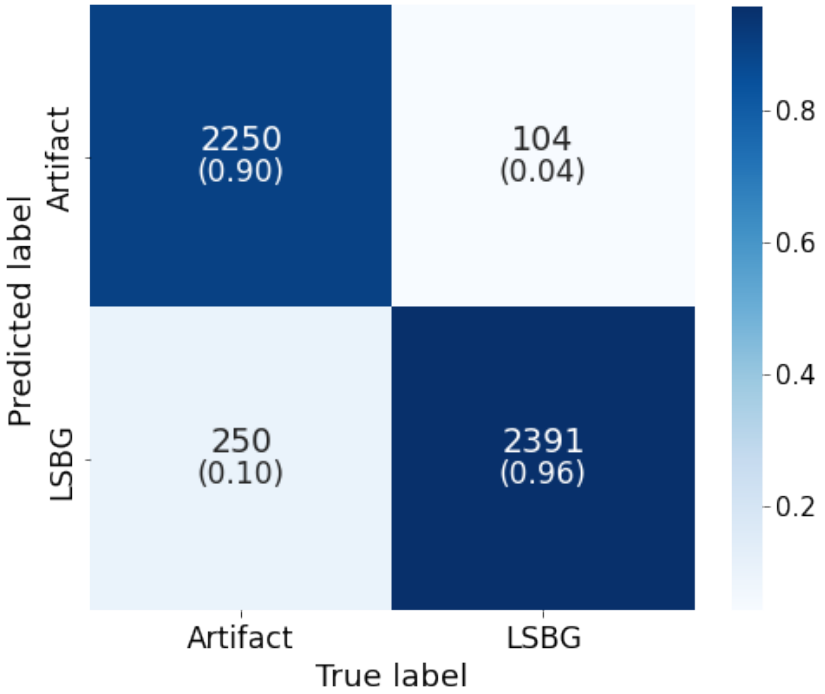

Figura 3.3: Matriz de confusão associada ao modelo padrão da ViT.

A Figura 3.6 mostra a curva ROC da rede (explicada em mais detalhe na seção 2.2), que traça a relação da Taxa de Verdadeiros Positivos (TPR) com a Taxa de Falsos Positivos (FPR) para cada limite T.

Na Figura 3.7, podemos visualizar a evolução da acurácia de treino e acurácia de validação, assim como a evolução da função perda para os conjuntos de treino e validação. É interessante notar que à medida que as funções perda vão diminuindo, as acurácias aumentam, conforme esperado.

Na Figura 3.8, temos um histograma da probabilidade obtida pela rede de um exemplo ser LSBG, P^i_{pred}. Seguimos a convenção comumente utilizada na literatura de usar um limite $T = 50\%$ (ver seção 2.5), para separar as duas classes. As LSBGs estão indicadas em azul e os artefatos em laranja. Um modelo ideal deveria dar uma probabilidade 0 para os artefatos e de 1 para as LSBGs.

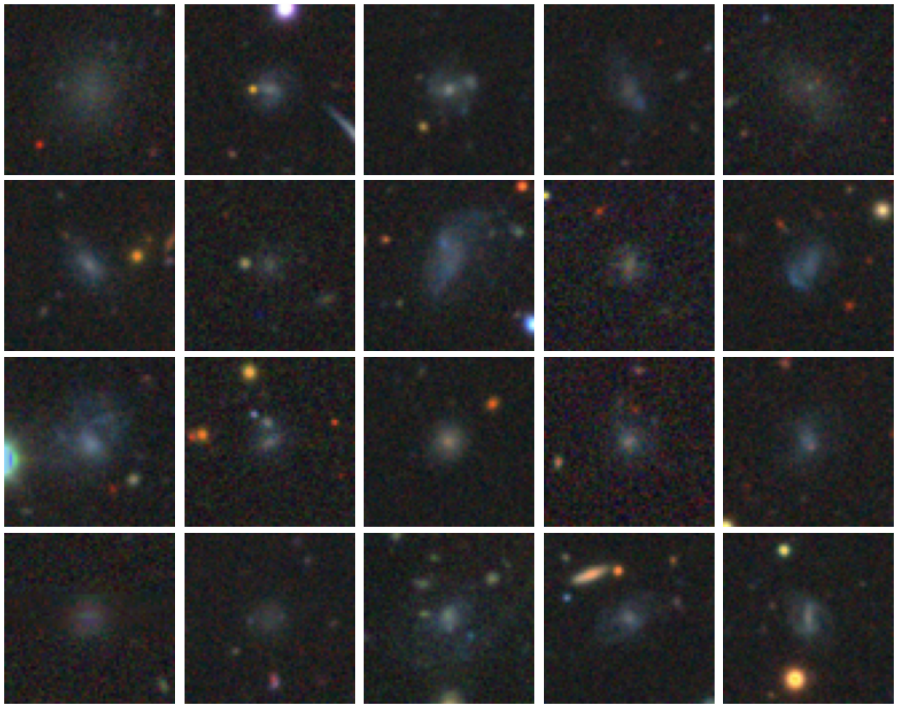

Figura 3.4: Exemplos de Artefatos da amostra de T201b que foram classificados como LSBGs pela ViT (Falsos Positivos).

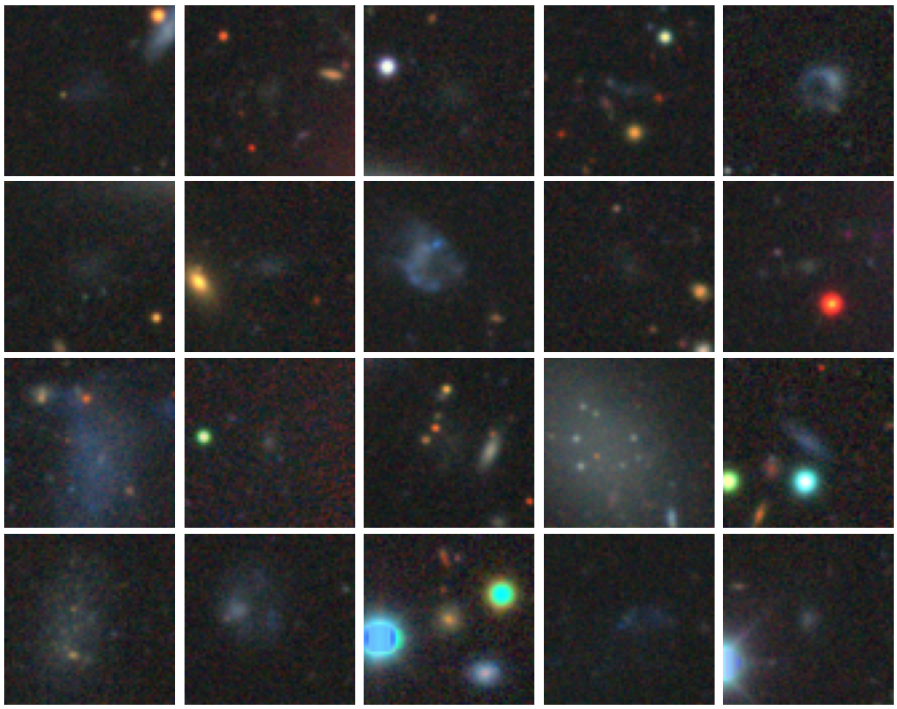

Figura 3.5: Exemplos de LSBGs da amostra de T201b que foram classificados como artefatos pela ViT (Falsos Negativos).

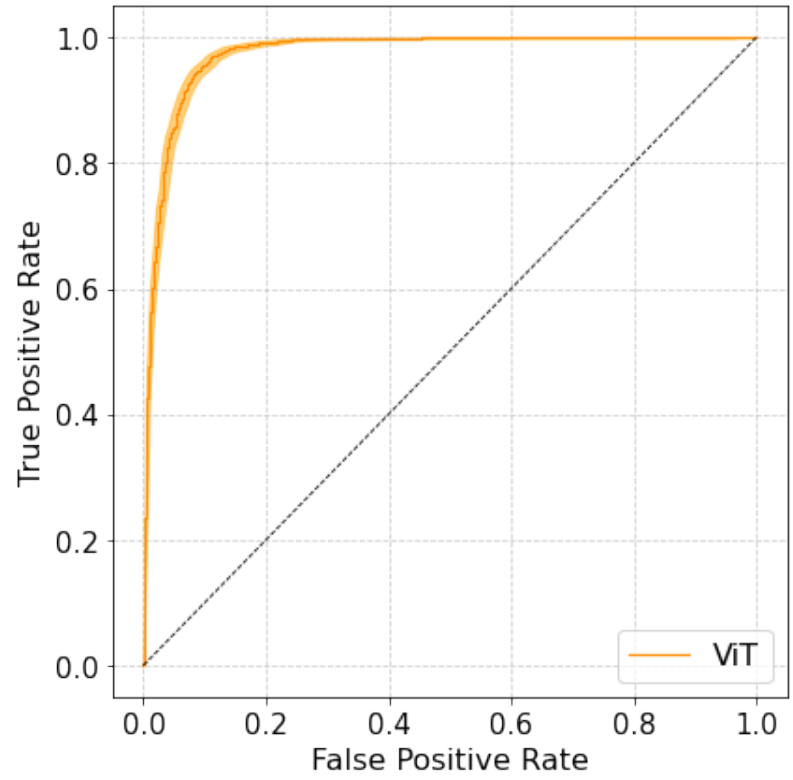

Figura 3.6: Curva ROC associada ao modelo padrão da ViT.

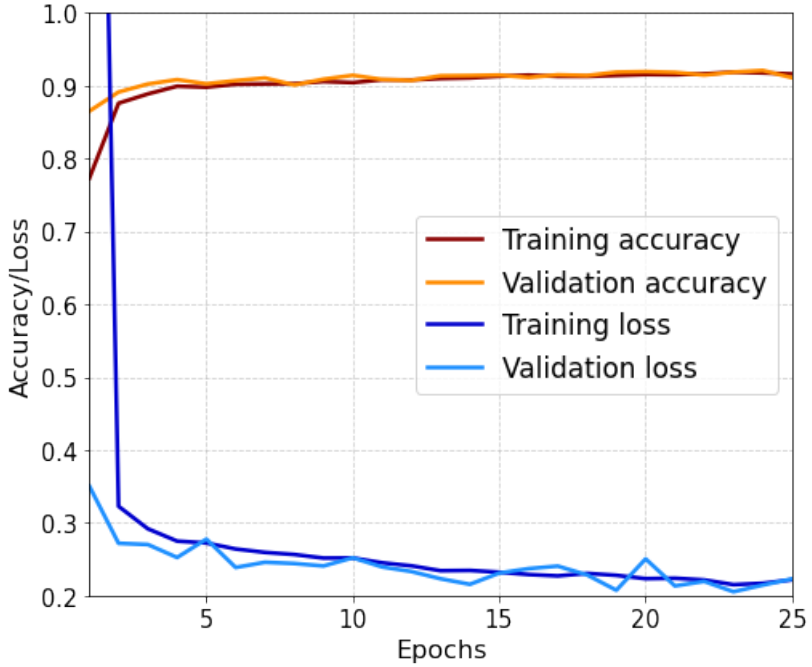

Figura 3.7: Evolução da acurácia e da função perda para a rede ViT.

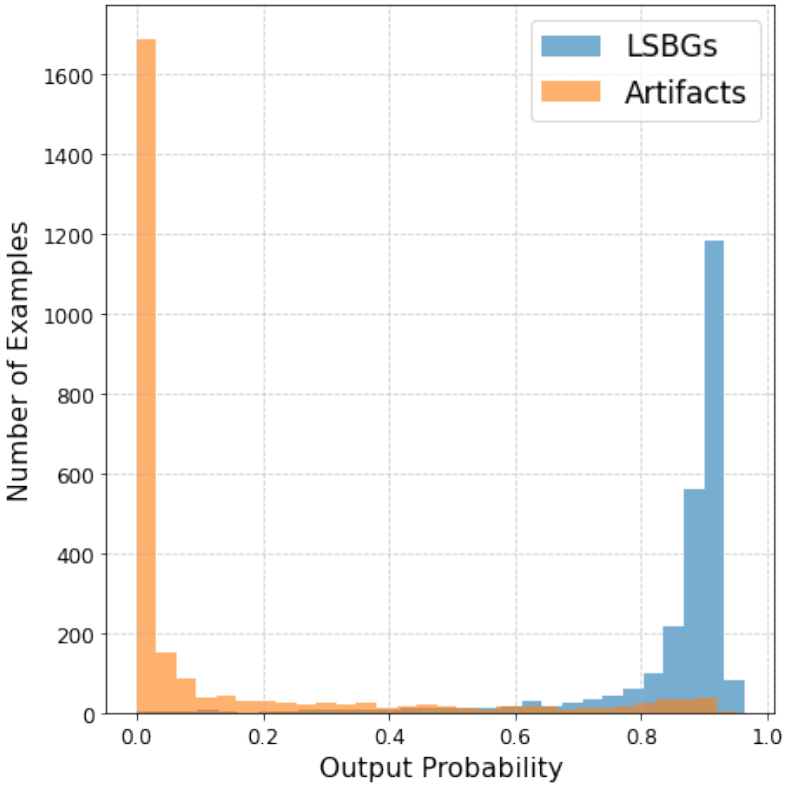

Figura 3.8: Histograma mostrando a probabilidade dos exemplos serem LSBGs para o modelo padrão da ViT.

3.4.1 Estimativa das Incertezas

Para quantificar as incertezas das métricas associadas ao modelo padrão da ViT, utilizamos a abordagem descrita em T201b. Nesta abordagem, são consideradas três fontes de incerteza: (1) Incertezas estatísticas aleatórias ao calcular as métricas no conjunto de teste, (2) Incertezas advindas da divisão aleatória de dados em conjunto de teste, treinamento e validação e (3) Incertezas advindas de exemplos incorretamente classificados no conjunto de treinamento.

A abordagem consiste em usar o método de *bootstrap* para estimar um intervalo de confiança para as métricas. Mais especificamente, fazemos uma re-amostragem (com reposição) do conjunto de testes original, assim obtemos 1000 conjuntos, todos com o mesmo tamanho que o conjunto original, e calculamos as métricas de classificação para cada um. Para estimar as incertezas de forma mais exata, deveríamos treinar a rede do zero e realizar o *bootstrap* sobre os conjuntos de treino e validação também, no entanto, treinar e validar a rede 1000 vezes é computacionalmente inviável. Na Figura 3.9, temos um histograma mostrando as acurácias da rede obtidas nos conjuntos de *bootstrap*.

Dessa forma, encontramos que os intervalos de 95% de confiança para as métricas são: acurácia $= [0.922, 0.936]$, precisão $= [0.893, 0.916]$, revocação $= [0.950, 0.966]$ e AUC $= [0.969, 0.977]$.

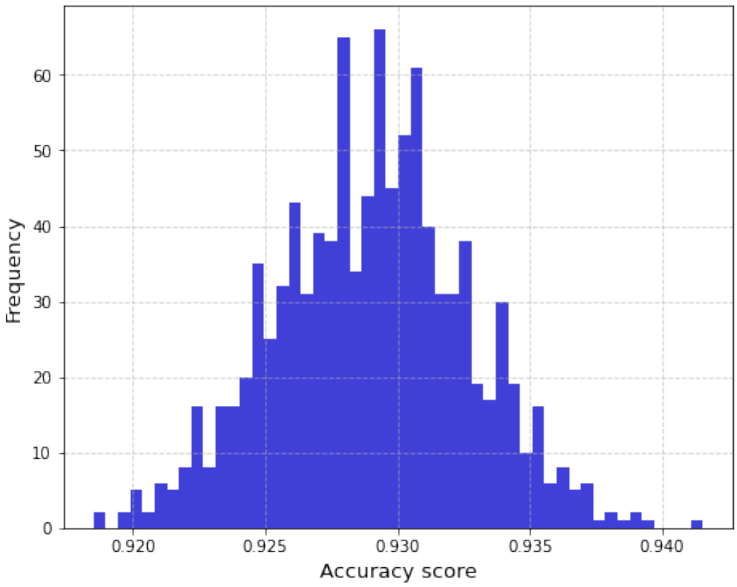

Figura 3.9: Histograma mostrando a acurácia para cada um dos 1000 conjuntos utilizados no método de *bootstrap*.

3.5 Pré-processamento

Com o objetivo de alcançar melhores resultados, aplicamos uma técnica de pré-processamento que consiste em fazer um ajuste de contraste nas imagens, uma tentativa de realçar os objetos de baixo brilho superficial e "saturar" os de alto brilho. Essa abordagem foi inspirada na estratégia de Bom et al. (2022) para o *II Strong Gravitational Lensing Challenge* (IISGLC), um desafio de classificação de imagens de lentes gravitacionais. Esses autores aplicaram uma técnica semelhante em imagens de lentes gravitacionais para realçar arcos gravitacionais embebidos na luz das galáxias-lente. Em seguida, treinaram uma rede neural para classificar as imagens. Como resultado, seu algoritmo de classificação foi o que obteve o maior escore do IISGLC, além disso verificaram que esse método de pré-processamento teve um impacto significativo no desempenho dos modelos.

Para implementar o algoritmo de ajuste de contraste utilizado nesse trabalho em uma imagem $X_i = [x_{m,n}]$, onde $x_{m,n}$ são os pixels da imagem, realizamos os seguintes passos:

Passo 1: Defina constantes $0 \leq a < b \leq 100\%$.

Passo 2: Calcule $P_1 = Percentil(X_i, a)$ e $P_2 = Percentil(X_i, b)$

Passo 3: $X_i \leftarrow (X_i - P_1)/(P_2 - P_1)$[14]

Passo 4: Para todo m e n, se $x_{m,n} < 0$ substitua o valor por 0 e se $x_{m,n} > 1$ substitua o valor por 1[15].

[14] $a \leftarrow b$ denota "a recebe"

[15] Ou seja, nos pixels de valor menor que 0, substituta o valor dos pixels por 0 e nos pixels de valor maior que 1, substituta o valor dos pixels por 1

	acurácia	AUC	revocação	precisão
VIT	0.929	0.975	0.958	0.905
DeepShadows	0.920	0.974	0.944	0.903

Tabela 3.1: Comparação das métricas de classificação entre o modelo padrão da ViT e *DeepShadows* aplicadas no conjunto de imagens PNG.

	Int. acurácia	Int. AUC	Int. revocação	Int. precisão
VIT	$[0.922, 0.936]$	$[0.969, 0.977]$	$[0.950, 0.966]$	$[0.893, 0.916]$
DeepShadows	$[0.912, 0.928]$	$[0.970, 0.974]$	$[0.935, 0.953]$	$[0.891, 0.914]$

Tabela 3.2: Comparação dos intervalos de 95% confiança para as métricas de classificação entre o modelo padrão da ViT e *DeepShadows* usando o conjunto de imagens em formato PNG.

Aplicamos o método para diferentes valores de a e b, tanto para a ViT como para a *DeepShadows*. A comparação dos resultados é mostrada a seguir.

3.6 Comparação das redes ViT e DeepShadows

Para comparar as redes ViT e *DeepShadows*, aplicamos o nosso modelo padrão no conjunto de imagens PNG, já que esse foi o mesmo formato utilizado por T201b. Conforme mostra a Tabela 3.1, o desempenho do modelo padrão da ViT foi ligeiramente superior em todas as métricas avaliadas em comparação com a rede *DeepShadows*. Mais precisamente, a acurácia do nosso modelo foi 0.98% maior, precisão 0.22% maior, revocação 1.48% maior e AUC 0.1% maior.

Para uma melhor comparação entre os métodos devemos considerar as incertezas das métricas de classificação. A Tabela 3.2 mostra os intervalos de 95% de confiança das métricas de classificação do modelo padrão da ViT e *DeepShadows*, determinados pelo método de *bootstrap* descrito na seção 3.4.1. Embora nosso modelo de ViT tenha pontuado melhor em todas as métricas no conjunto de imagens PNG, os limites inferiores dos intervalos de confiança da ViT não superaram os limites superiores da *DeepShadows*. Dessa forma, não podemos afirmar com confiança de 95% que o modelo padrão da ViT teve um desempenho superior a *DeepShadows* no conjunto de imagens PNG.

Além disso, realizamos uma comparação do desempenho de ambas as redes utilizando ou não o método de ajuste de contraste nos dados de entrada. Para implementar esse método de pré-processamento, precisamos utilizar o conjunto de imagens FITS. Note que os resultados da rede *DeepShadows* no conjunto de imagens PNG foram obtidos por T201b. Para obter os resultados dessa arquitetura nas imagens FITS, utilizamos o conjunto de dados composto por essas imagens e executamos o

código desse autor. Como mostrado na tabela 3.3, constatamos que a rede ViT que utilizou dados com ajuste de contraste teve uma acurácia superior em relação a ViT aplicada em dados sem esse algoritmo de pré-processamento das imagens. Analogamente, a rede *DeepShadows* que foi aplicada em dados pré-processados teve um desempenho superior em comparação com a mesma rede sem o pré-processamento. Vale pontuar que nos casos onde os dados de entrada não foram pré-processados pelo método de ajuste de contraste, as imagens foram normalizadas de forma simples. Mais especificamente, essa normalização consistia em, para cada imagem, dividir o valor de todos pixels pelo pixel de maior valor. Conforme podemos ver na Tabela 3.3, quando aplicadas no conjunto de imagens FITS, as redes ViT tiveram uma acurácia superior a CNN *DeepShadows*. Entretanto, os valores das acurácias de ambas as redes obtidas utilizando o conjunto de imagens FITS (independentemente de pré-processamento com ajuste de contraste ou não) são menores do que os limites inferiores dos intervalos de confiança das acurácias obtidas utilizando o conjunto de imagens PNG (tabela 3.2).

	a	b	acurácia	AUC	revocação	precisão
VIT	-	-	0.915	0.969	0.923	0.907
VIT	5%	95%	0.920	0.965	0.957	0.891
DeepShadows	-	-	0.907	0.967	0.976	0.857
DeepShadows	5%	95%	0.915	0.968	0.968	0.966

Tabela 3.3: Comparação das métricas de classificação dos modelos *DeepShadows* e ViT, utilizando ou não o método de ajuste de contraste para o pré-processamento do conjunto de dados com imagens em formato FITS.

Capítulo 4

Conclusão

Dado que futuros levantamentos profundos do céu, tais como Euclid e LSST, produzirão uma quantidade gigantesca de dados, a classificação por métodos meramente visuais de objetos astronômicos será insuficiente para um estudo mais aprofundado destes. Assim, métodos de detecção e classificação automáticas serão absolutamente necessários.

Entre esses métodos de classificação automática destacam-se as Redes Neurais Artificiais, em particular, as Redes Neurais Convolucionais, que têm um histórico de serem o estado da arte em problemas de classificação de imagens.

Neste trabalho abordamos o problema de separar LSBGs de artefatos em imagens astronômicas usando duas redes neurais. Implementamos uma rede *Vision Transformer*, arquitetura que causou forte impacto na literatura desde sua publicação no final de 2020 por desafiar o paradigma das CNNs como estado da arte. Mais especificamente, utilizando o único conjunto de imagens de LSBGs e Artefatos publicamente disponível, com 40000 imagens selecionadas do DES, implementamos uma rede ViT e a comparamos com a CNN *DeepShadows*, definida como nosso *benchmark*.

Utilizando várias métricas de classificação, verificamos que o nosso modelo da ViT obteve um desempenho ligeiramente superior em comparação a Rede Neural Convolucional *DeepShadows*. Em particular, nosso modelo alcançou uma $acurácia = 0.929$, que foi 0.98% maior que a da *DeepShadows*, com $acurácia = 0.920$. No entanto, após fazer uma análise das incertezas das métricas de classificação, constatamos que o desempenho da nossa rede ViT foi tão bom quanto o do *benchmark*, dentro do intervalo de 95% de confiança das métricas.

Assim, esse trabalho sugere que a ViT pode ser uma boa alternativa em relação às CNNs para lidar com a classificação automática de LSBGs. Convém pontuar que conforme obtemos um maior conjunto de dados, é esperado que tenhamos um desempenho ainda superior da ViT em comparação com as CNNs (Dosovitskiy et al., 2021). Dessa forma, conforme levantamentos maiores do céu noturno são disponibilizados, teremos uma amostra cada vez maior de objetos de baixo brilho superficial. Essas circunstâncias parecem apontar um paradigma especialmente interessante para redes *Vision Transformers*.

Além disso, implementamos um algoritmo de pré-processamento de imagens, que

consiste em estabelecer limites superiores e inferiores para o brilho das imagens, ressaltando os objetos de baixo brilho superficial, para melhorar a performance das redes. Nossos resultados indicam que esse algoritmo realmente pode trazer uma melhora no desempenho das redes quando treinadas no conjunto de imagens FITS, uma vez que contribuiu para que tanto o modelo padrão da ViT como a CNN *DeepShadows* alcançassem métricas de classificação superiores. Entretanto, os melhores resultados do treinamento com o conjunto de imagens FITS ainda não superam os resultados de ambas as redes quando treinadas no conjunto de imagens PNG.

4.1 Perspectivas Futuras

Nesse trabalho mostramos que as redes ViT obtém um bom desempenho ao classificar LSBGs, com métricas de classificação ligeiramente superiores aos da CNN *DeepShadows* T201b. Levando em conta que Yao-Yu Lin et al. (2021), ao implementar a ViT em um conjunto de 155951 imagens e 8 classes de galáxias, percebeu que essa rede é particularmente boa em identificar galáxias pequenas e de baixo brilho, uma questão relevante seria: aumentando o tamanho da amostra, conseguiremos um resultado similar ao de Dosovitskiy et al. (2021), que encontraram que o desempenho da ViT melhora com tamanhos crescentes dos conjuntos de treinamento mas a CNN não? Se aplicada a outros objetos astronômicos como seria o desempenho da rede? Para responder tais questionamentos, será fundamental novos estudos aplicando a rede ViT a problemas de classificação de imagens astronômicas.

Por outro lado, a ViT se destaca frente as CNNs quando o conjunto de dados é suficientemente grande (Dosovitskiy et al., 2021). Assim, um dos próximos passos é aumentar a quantidade de imagens para o treino da rede e avaliar os efeitos disso sobre as métricas de classificação. Será que para um conjunto de dados maior seremos capazes de afirmar com maior confiança estatística que a ViT apresenta um desempenho superior? Uma forma relativamente simples de fazer isso é pré-treinar a rede em conjuntos de dados semelhantes e fazer o *fine tunning* no conjunto de dados que utilizamos.

Além disso, mostramos uma melhora do desempenho da rede ao aplicar o Método de Ajuste de Contraste no conjunto de imagens FITS. Assim, é válido levantar o questionamento: como podemos otimizar esse método? Quais os percentis ideais para a classificação? Outros métodos de pré-processamento que visam realçar as LSBGs, como o uso de filtros ou a subtração de objetos de alto brilho, poderiam melhorar o desempenho da rede e o processo de classificação?

Além disso, acreditamos que o nosso modelo padrão ainda pode ser otimizado. Em particular, técnicas para otimizar o limite (*threshold*) T, inicialização da rede com pesos pré-selecionados e outras formas de data augmentation podem melhorar o desempenho do modelo.

Bibliografia

N C Amorisco, A Monachesi, A Agnello, and S D M White. The globular cluster systems of 54 coma ultra-diffuse galaxies: statistical constraints from HST data. *Monthly Notices of the Royal Astronomical Society*, 475:4235–4251, 2018. doi: 10.1093/mnras/sty116. URL `https://doi.org/10.1093%2Fmnras%2Fsty116`.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv e-prints*, art. 1607.06450, 2016. URL `http://dblp.uni-trier.de/db/journals/corr/corr1607.html#BaKH16`.

C. R. Bom, B. M. O. Fraga, L. O. Dias, P. Schubert, M. Blanco Valentin, C. Furlanetto, M. Makler, K. Teles, M. Portes de Albuquerque, and R. Benton Metcalf. Developing a Victorious Strategy to the Second Strong Gravitational Lensing Data Challenge. *arXiv e-prints*, art. 2203.09536, 2022.

Gregory D. Bothun, Christopher D. Impey, David F. Malin, and Jeremy R. Mould. Discovery of a Huge Low-Surface-Brightness Galaxy: A Proto-Disk Galaxy at Low Redshift? *The Astronomical Journal*, 94:23–29, 1987. doi: 10.1086/114443.

Vanessa Buhrmester, David Münch, and Michael Arens. Analysis of explainers of black box deep neural networks for computer vision: A survey. *Machine Learning and Knowledge Extraction*, 3:966–989, 2021. ISSN 2504-4990. doi: 10.3390/make3040048. URL `https://www.mdpi.com/2504-4990/3/4/48`.

Timothy Carleton, Raphaël Errani, Michael Cooper, Manoj Kaplinghat, Jorge Peñarrubia, and Yicheng Guo. The formation of ultra-diffuse galaxies in cored dark matter haloes through tidal stripping and heating. *Monthly Notices of the Royal Astronomical Society*, 485:382–395, 2019.

Francois Chollet. *Deep Learning with Python*. Manning Publications, 2017. ISBN 1617294438; 9781617294433.

Christopher J. Conselice. Ultra-diffuse galaxies are a subset of cluster dwarf elliptical/spheroidal galaxies. *Research Notes of the AAS*, 2:43–44, 2018. doi: 10.3847/2515-5172/aab7f6. URL `https://doi.org/10.3847/2515-5172/aab7f6`.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009. doi: 10.1109/CVPR.2009.5206848.

Arjun Dey, David J. Schlegel, Dustin Lang, Robert Blum, Kaylan Burleigh, Xiaohui Fan, Joseph R. Findlay, Doug Finkbeiner, David Herrera, Sté phanie Juneau, Martin Landriau, Michael Levi, Ian McGreer, Aaron Meisner, Adam D. Myers, John Moustakas, Peter Nugent, Anna Patej, Edward F. Schlafly, Alistair R. Walker, Francisco Valdes, Benjamin A. Weaver, Christophe Yèche, Hu Zou, Xu Zhou, Behzad Abareshi, T. M. C. Abbott, Bela Abolfathi, C. Aguilera, Shadab Alam, Lori Allen, A. Alvarez, James Annis, Behzad Ansarinejad, Marie Aubert, Jacqueline Beechert, Eric F. Bell, Segev Y. BenZvi, Florian Beutler, Richard M. Bielby, Adam S. Bolton, César Briceño, Elizabeth J. Buckley-Geer, Karen Butler, Annalisa Calamida, Raymond G. Carlberg, Paul Carter, Ricard Casas, Francisco J. Castander, Yumi Choi, Johan Comparat, Elena Cukanovaite, Timothée Delubac, Kaitlin DeVries, Sharmila Dey, Govinda Dhungana, Mark Dickinson, Zhejie Ding, John B. Donaldson, Yutong Duan, Christopher J. Duckworth, Sarah Eftekharzadeh, Daniel J. Eisenstein, Thomas Etourneau, Parker A. Fagrelius, Jay Farihi, Mike Fitzpatrick, Andreu Font-Ribera, Leah Fulmer, Boris T. Gänsicke, Enrique Gaztanaga, Koshy George, David W. Gerdes, Satya Gontcho A Gontcho, Claudio Gorgoni, Gregory Green, Julien Guy, Diane Harmer, M. Hernandez, Klaus Honscheid, Lijuan (Wendy) Huang, David J. James, Buell T. Jannuzi, Linhua Jiang, Richard Joyce, Armin Karcher, Sonia Karkar, Robert Kehoe, Kneib Jean-Paul, Andrea Kueter-Young, Ting-Wen Lan, Tod R. Lauer, Laurent Le Guillou, Auguste Le Van Suu, Jae Hyeon Lee, Michael Lesser, Laurence Perreault Levasseur, Ting S. Li, Justin L. Mann, Robert Marshall, C. E. Martínez-Vázquez, Paul Martini, Hélion du Mas des Bourboux, Sean McManus, Tobias Gabriel Meier, Brice Ménard, Nigel Metcalfe, Andrea Muñoz-Gutiérrez, Joan Najita, Kevin Napier, Gautham Narayan, Jeffrey A. Newman, Jundan Nie, Brian Nord, Dara J. Norman, Knut A. G. Olsen, Anthony Paat, Nathalie Palanque-Delabrouille, Xiyan Peng, Claire L. Poppett, Megan R. Poremba, Abhishek Prakash, David Rabinowitz, Anand Raichoor, Mehdi Rezaie, A. N. Robertson, Natalie A. Roe, Ashley J. Ross, Nicholas P. Ross, Gregory Rudnick, Sasha Safonova, Abhijit Saha, F. Javier Sánchez, Elodie Savary, Heidi Schweiker, Adam Scott, Hee-Jong Seo, Huanyuan Shan, David R. Silva, Zachary Slepian, Christian Soto, David Sprayberry, Ryan Staten, Coley M. Stillman, Robert J. Stupak, David L. Summers, Suk Sien Tie, H. Tirado, Mariana Vargas-Magaña, A. Katherina Vivas, Risa H. Wechsler, Doug Williams, Jinyi Yang, Qian Yang, Tolga Yapici, Dennis Zaritsky, A. Zenteno, Kai Zhang, Tianmeng Zhang, Rongpu Zhou, and Zhimin Zhou. Overview of the DESI legacy imaging surveys. *The Astronomical Journal*, 157:168–197, 2019. doi: 10.3847/1538-3881/ab089d. URL `https://doi.org/10.3847%2F1538-3881%2Fab089d`.

Arianna Di Cintio, Chris B. Brook, Aaron A. Dutton, Andrea V. Macciò, Aura Obreja, and Avishai Dekel. NIHAO – XI. Formation of ultra-diffuse galaxies by outflows. *Monthly Notices of the Royal Astronomical Society: Letters*, 466:L1–L6, 2016.

M. J. Disney. Visibility of galaxies. *Nature*, 263(5578):573–575, 1976. doi: 10.1038/263573a0.

Pieter Dokkum, Roberto Abraham, Allison Merritt, Jielai Zhang, Marla Geha, and Charlie Conroy. Forty-seven milky way-sized, extremely diffuse galaxies in the coma cluster. *The Astrophysical Journal*, 798:45, 2014. doi: 10.1088/2041-8205/798/2/L45.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv e-prints*, art. 2010.11929, 2021.

Patrick R. Durrell, William E. Harris, Aaron J. Romanowsky, John Blakeslee, Jean Brodie, Steven Janssens, Thorsten Lisker, Sakurako Okamoto, and Carolin Wittmann. The PIPER Survey: An Initial Look at the Intracluster Globular Cluster Population in the Perseus Cluster. In *American Astronomical Society Meeting Abstracts #233*, volume 233 of *American Astronomical Society Meeting Abstracts*, page 261, 2019.

William E. Harris, John P. Blakeslee, and Gretchen L. H. Harris. Galactic dark matter halos and globular cluster populations. III. extension to extreme environments. *The Astrophysical Journal*, 836:67, 2017. doi: 10.3847/1538-4357/836/1/67. URL `https://doi.org/10.3847%2F1538-4357%2F836%2F1%2F67`.

KM Hornik, M. Stinchcomb, and H. White. Multilayer feedforward networks are universal approximator. *IEEE Transactions on Neural Networks*, 2:45, 1989.

Chris Impey and Greg Bothun. Low surface brightness galaxies. *Annual Review of Astronomy and Astrophysics*, 35:267–307, 1997. doi: 10.1146/annurev.astro.35.1.267. URL `https://doi.org/10.1146/annurev.astro.35.1.267`.

Chris Impey, G. Bothun, and David Malin. Virgo dwarfs - new light on faint galaxies. *The Astrophysical Journal*, 330:634–660, 1988. doi: 10.1086/166500.

Alexei Y. Kniazev, Eva K. Grebel, Simon A. Pustilnik, Alexander G. Pramskij, Tamara F. Kniazeva, Francisco Prada, and Daniel Harbeck. Low surface brightness galaxies in the sloan digital sky survey. i. search method and test sample. *The Astronomical Journal*, 127:704–727, 2004. doi: 10.1086/381061. URL `https://doi.org/10.1086%2F381061`.

Franç ois Lanusse, Quanbin Ma, Nan Li, Thomas E. Collett, Chun-Liang Li, Siamak Ravanbakhsh, Rachel Mandelbaum, and Barnabás Póczos. CMU DeepLens: deep learning for automatic image-based galaxy–galaxy strong lens finding. *Monthly Notices of the Royal Astronomical Society*, 473:3895–3906, 2017. doi: 10.1093/mnras/stx1665. URL `https://doi.org/10.1093%2Fmnras%2Fstx1665`.

Jeong Hwan Lee, Jisu Kang, Myung Gyoon Lee, and In Sung Jang. The nature of ultra-diffuse galaxies in distant massive galaxy clusters: A370 in the hubble frontier fields. *The Astrophysical Journal*, 894:75, 2020. doi: 10.3847/1538-4357/ab8632. URL `https://doi.org/10.38472F1538-43572Fab8632`.

Dayi Li, Gwendolyn M. Eadie, Roberto G. Abraham, Patrick E. Brown, William E. Harris, Steven R. Janssens, Aaron J. Romanowsky, Pieter van Dokkum, and Shany Danieli. Light from the Darkness: Detecting Ultra-Diffuse Galaxies in the Perseus Cluster through Over-densities of Globular Clusters with a Log-Gaussian Cox Process. *arXiv e-prints*, art. 2204.05487, 2022.

Sungsoon Lim, Eric W. Peng, Patrick Côté, Laura V. Sales, Mark den Brok, John P. Blakeslee, and Puragra Guhathakurta. The globular cluster systems of ultra-diffuse galaxies in the coma cluster. *The Astrophysical Journal*, 862: 82, 2018. doi: 10.3847/1538-4357/aacb81. URL https://doi.org/10.3847%2F1538-4357%2Faacb81.

D Prole, R van der Burg, Michael Hilker, and J Davies. Observational properties of ultra-diffuse galaxies in low-density environments: field udgs are predominantly blue and star forming. *Monthly Notices of the Royal Astronomical Society*, 488: 2143–2157, 2019. doi: 10.1093/mnras/stz1843.

D J Prole, R F J van der Burg, M Hilker, and L R Spitler. The quiescent fraction of isolated low surface brightness galaxies: observational constraints. *Monthly Notices of the Royal Astronomical Society*, 500:2049–2062, 2020. doi: 10.1093/mnras/staa3296. URL https://doi.org/10.1093%2Fmnras%2Fstaa3296.

Laura V Sales, Julio F Navarro, Louis Peñ afiel, Eric W Peng, Sungsoon Lim, and Lars Hernquist. The formation of ultradiffuse galaxies in clusters. *Monthly Notices of the Royal Astronomical Society*, 494:1848–1858, 2020. doi: 10.1093/mnras/staa854. URL https://doi.org/10.1093%2Fmnras%2Fstaa854.

D. Tanoglidis, A. Drlica-Wagner, K. Wei, T. S. Li, J. Sánchez, Y. Zhang, A. H. G. Peter, A. Feldmeier-Krause, J. Prat, K. Casey, A. Palmese, C. Sánchez, J. DeRose, C. Conselice, L. Gagnon, T. M. C. Abbott, M. Aguena, S. Allam, S. Avila, K. Bechtol, E. Bertin, S. Bhargava, D. Brooks, D. L. Burke, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, C. Chang, M. Costanzi, L. N. da Costa, J. De Vicente, S. Desai, H. T. Diehl, P. Doel, T. F. Eifler, S. Everett, A. E. Evrard, B. Flaugher, J. Frieman, J. García-Bellido, D. W. Gerdes, R. A. Gruendl, J. Gschwend, G. Gutierrez, W. G. Hartley, D. L. Hollowood, D. Huterer, D. J. James, E. Krause, K. Kuehn, N. Kuropatkin, M. A. G. Maia, M. March, J. L. Marshall, F. Menanteau, R. Miquel, R. L. C. Ogando, F. Paz-Chinchón, A. K. Romer, A. Roodman, E. Sanchez, V. Scarpine, S. Serrano, I. Sevilla-Noarbe, M. Smith, E. Suchyta, G. Tarle, D. Thomas, D. L. Tucker, and A. R. Walker and. Shadows in the dark: Low-surface-brightness galaxies discovered in the dark energy survey. *The Astrophysical Journal Supplement Series*, 252:18, 2021a. doi: 10.3847/1538-4365/abca89. URL https://doi.org/10.3847/1538-4365/abca89.

D. Tanoglidis, A. Ćiprijanović, and A. Drlica-Wagner (T21b). Deepshadows: Separating low surface brightness galaxies from artifacts using deep learning. *Astronomy and Computing*, 35:100469, 2021b. ISSN 2213-1337. doi: https://

doi.org/10.1016/j.ascom.2021.100469. URL https://www.sciencedirect.com/science/article/pii/S2213133721000238.

Paul Teeninga, Ugo Moschini, Scott Trager, and Michael Wilkinson. Improved detection of faint extended astronomical objects through statistical attribute filtering. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, volume 9082, pages 157–168. Springer, 2015. doi: 10.1007/978-3-319-18720-4_14.

Pieter van Dokkum, Shany Danieli, Yotam Cohen, Allison Merritt, Aaron J. Romanowsky, Roberto Abraham, Jean Brodie, Charlie Conroy, Deborah Lokhorst, Lamiya Mowla, Ewan O'Sullivan, and Jielai Zhang. A galaxy lacking dark matter. *Nature*, 555(7698):629–632, 2018. doi: 10.1038/nature25767.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Joshua Yao-Yu Lin, Song-Mao Liao, Hung-Jin Huang, Wei-Ting Kuo, and Olivia Hsuan-Min Ou. Galaxy Morphological Classification with Efficient Vision Transformer. *arXiv e-prints*, art. 2110.01024, 2021.

Amirhossein Yazdani Abyaneh, Ali Hosein Gharari Foumani, and Vahid Pourahmadi. Deep Neural Networks Meet CSI-Based Authentication. *arXiv e-prints*, art. 1812.04715, 2018.

Zhenping Yi, Jia Li, Wei Du, Meng Liu, Zengxu Liang, Yongguang Xing, Jingchang Pan, Yude Bu, Xiaoming Kong, and Hong Wu. Automatic detection of low surface brightness galaxies from Sloan Digital Sky Survey images. *Monthly Notices of the Royal Astronomical Society*, 513:3972–3981, 2022. ISSN 0035-8711. doi: 10.1093/mnras/stac775. URL https://doi.org/10.1093/mnras/stac775.

Kai Hou Yip, Quentin Changeat, Nikolaos Nikolaou, Mario Morvan, Billy Edwards, Ingo P. Waldmann, and Giovanna Tinetti. Peeking inside the black box: Interpreting deep-learning models for exoplanet atmospheric retrievals. *The Astronomical Journal*, 162:195, 2021. doi: 10.3847/1538-3881/ac1744. URL https://doi.org/10.3847/1538-3881/ac1744.

Dennis Zaritsky, Richard Donnerstein, Arjun Dey, Jennifer Kadowaki, Huanian Zhang, Ananthan Karunakaran, David Martínez-Delgado, Mubdi Rahman, and Kristine Spekkens. Systematically measuring ultra-diffuse galaxies (SMUDGes). i. survey description and first results in the coma galaxy cluster and environs. *The*

Astrophysical Journal Supplement Series, 240:1, 2018. doi: 10.3847/1538-4365/aaefe9. URL https://doi.org/10.3847/1538-4365/aaefe9.