

CBPF – Centro Brasileiro de Pesquisas Físicas

Dissertação de Mestrado

The use of Deep Learning to Automated Transient Identification

Phelipe Antonie D'arc De Matos

Orientador
Dr. Clécio Roque De Bom

Rio de Janeiro, RJ
2023

"THE USE OF DEEP LEARNING
TO AUTOMATED TRANSIENT IDENTIFICATION"

PHELIPE ANTONIE DARC DE MATOS

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



Clecio Roque de Bom – Orientador/CBPF



Kary Ann del Carmen Ocaña Gautherot - LNCC



Thiago Signorini Gonçalves – UFRJ

Rio de Janeiro, 29 de agosto de 2023.

Phelipe Antonie D'arc De Matos

The use of Deep Learning to Automated Transient Identification

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

CBPF – Centro Brasileiro de Pesquisas Físicas

Orientador: Dr. Clécio Roque De Bom

Rio de Janeiro, RJ
2023

Darc, Phelipe Antonie

The use of Deep Learning to Automated Transient Identification/ Phelipe Antonie
D'arc De Matos. - 2023

93 f. : il.

Dissertação de Mestrado – CBPF – Centro Brasileiro de Pesquisas Físicas , Rio de Janeiro, RJ, 2023.

Orientador: Dr. Clécio Roque De Bom

1. Transientes Astronômicos. 2. Kilonovas. 3. Astronomia de multi-mensageiro. 4.
Redes Neurais Convolucionais. 5. Inteligência Artificial. I. Título
CDU 02:141:005.7

AGRADECIMENTOS

Agradeço sinceramente a todas as pessoas que contribuíram para a conclusão da minha dissertação. Neste momento, gostaria de expressar minha gratidão especial aos seguintes indivíduos:

Aos meus pais, Eliane D'arc e José Matos, por estarem sempre presentes e apoiarem todas as decisões que me trouxeram até aqui. Agradeço profundamente ao meu irmão, Arthur Antonie, por suas inúmeras perguntas semanais, que me desafiaram e me ajudaram a crescer. À minha avó, Joana D'arc, merece meu especial reconhecimento por seu amor incondicional. Gostaria de agradecer à minha prima, Ana Gabriela, que, mesmo por meio de críticas não construtivas, contribuiu para minha jornada. E a Bruna Pinto, minha paixão e melhor amiga, que apesar de sentir sono durante todas às vezes que falei sobre astronomia, esteve presente em todos os momentos de trabalho árduo e desafios. Ao meu orientador, Clécio R. Bom, devo minha sincera gratidão. Sua orientação e sabedoria foram fundamentais para o meu desenvolvimento como pesquisador. Desde ensinamentos sobre astrofísica, política e história até lições valiosas para a vida, agradeço por sua confiança em mim e por me guiar durante toda a minha jornada acadêmica.

Aos meus amigos da graduação, Úrsula Fonseca e Gabriel Silva, agradeço por me lembrarem diariamente que a física pode ser divertida e que é ainda melhor quando compartilhada, além das partidas de aram. Também desejo expressar minha gratidão aos meus amigos de Campo Grande, Lucas Tezolin, Daniel Vieira e Nathan Macedo, que ao me negarem ajuda e apoio em diversas ocasiões, me mostraram que sou capaz de superar adversidades. Ao meu amigo mais antigo, Fernando Figueiredo, agradeço por me lembrar de que cada pequena conquista deve ser celebrada.

Aos meus amigos que contribuíram imensamente para o meu crescimento como pesquisador, Noura Field, Bernardo Fraga, Luciana Olivia e André Santos, com quem tive o privilégio de trabalhar lado a lado. Agradeço a todos pelas reuniões e conversas na sala do CBPF. Esta dissertação não existiria sem a ajuda de vocês. Em seguida, a CAPES, pelo financiamento deste trabalho, sem o qual a sua execução, assim como a minha permanência no mestrado, não teria sido possível.

Por fim, gostaria de estender minha gratidão a todos os que, de alguma forma, con-

tribuíram para o desenvolvimento desta pesquisa, mesmo que não mencionados nominalmente, sou imensamente grato por ter pessoas tão maravilhosas ao meu redor.

”Traditions are made to be broken”
- Unknown

RESUMO

O presente trabalho propõe o uso de técnicas de aprendizado profundo (*Deep Learning*) para identificar automaticamente transientes astronômicos em imagens provenientes do Dark Energy Survey Gravitational Wave (DESGW) *pipeline* e do *Southern Photometric Local Universe Survey (SPLUs) Transient Extension Project*, conhecido como STEP. Especificamente, nosso objetivo é integrar uma rede neural após o algoritmo de diferenciação de imagens desses projetos, visando acelerar a detecção de transientes, reduzindo o número de imagens que requerem inspeção visual. Para alcançar esse objetivo, utilizamos de redes neurais convolucionais (CNNs), que são um tipo de algoritmo de aprendizado profundo comprovadamente eficaz em tarefas de classificação de imagens. A CNN foi treinada em um conjunto de dados rotulados contendo transientes e não transientes, o que permitiu que ela aprendesse as características visuais de transientes astronômicos. Uma vez treinada, a CNN será integrada ao pipeline de detecção automatizada de transientes, cujo analisará as imagens e identificará automaticamente os transientes em potencial, que serão posteriormente confirmados por inspeção visual. O resultado obtido foi que nos conjuntos de dados do DES-GW e STEP, conseguimos remover 95% e 97% de todos os exemplos de não-transientes, respectivamente. Isso leva a uma detecção de transientes mais rápida e eficiente, já que a rede neural reduz o número de imagens que precisam ser inspecionadas. Dessa forma, os astrônomos podem se concentrar em eventos cientificamente mais relevantes. Além disso, isso pode ajudar a identificar transientes que não são óbvios para o astrônomo e que podem ter sido classificados como falsos negativos. Em geral, a abordagem proposta tem o potencial de melhorar significativamente a capacidade e eficácia da detecção de transientes nas buscas por contrapartida ópticas de ondas gravitacionais, cujo necessitam de uma rápida identificação e caracterização.

Key-Words: Multimessenger Astronomy, Gravitational Waves, Machine Learning, Deep learning, Astrophysical Transients, Artificial Intelligence.

ABSTRACT

The present work proposes the use of deep learning techniques to automatically identify astrophysical transients in image stamps applied to the Dark Energy Survey Gravitational Wave (DESGW) pipeline and the Southern Photometric Local Universe Survey (SPLUs) Transient Extension Project (STEP) datasets. Specifically, we aim to integrate a neural network into the difference imaging pipeline of these projects to accelerate transient detection by reducing the number of images that require visual inspection. To achieve this objective, we use convolutional neural networks (CNNs), which are a type of deep learning algorithm that has proven effective in image classification tasks. The CNN was trained on a labeled dataset of images containing both transients and non-transients, allowing it to learn the characteristic features of astrophysical transients. Once trained, the CNN will be integrated into the difference imaging pipeline, where it will analyze the images and automatically identify potential transients that will then be confirmed by visual inspection. The obtained result was that in the DES-GW and S-PLUS datasets, we were able to remove 95% and 97% of all non-transient examples, respectively. This leads to faster and more efficient transient detection, as the neural network reduces the number of images that need to be inspected. Thus, astronomers can focus on scientifically relevant events. Furthermore, it can help identify transients that are not obvious to the astronomer and may have been classified as false negatives.

Overall, the proposed approach has the potential to significantly improve the capability and effectiveness of transient detection in the DES-GW and S-PLUS surveys.

Key-Words: Multimessenger Astronomy, Gravitational Waves, Machine Learning, Deep learning, Astrophysical Transients, Artificial Intelligence.

LIST OF FIGURES

Figure 1.1 – Kilonova AT 2017gfo a visible example of Gravitational Wave Counterpart	2
Figure 2.1 – Upper Panel: Two white dwarfs collide, causing a Type Ia supernova. Bottom panel: One white dwarf gaining mass from a companion star and eventually exploding into a Type Ia Supernova Credit: NASA/Dana Berry, Sky Works Digital.	6
Figure 2.2 – Two Neutron stars merging and eventually, creating a Kilonova and Gravitational Waves. Credit: NASA/Dana Berry, Sky Works Digital https://svs.gsfc.nasa.gov/12949	7
Figure 2.3 – DECam and the Blanco Telescope at the Cerro Tololo inter american observatory (CTIO).	8
Figure 2.4 – Standard bandpasses for the DECam grizY filters	9
Figure 2.5 – A schematic overview of the complete single epoch processing of DES-GW pipeline	10
Figure 2.6 – On the upper left The interior of the T80-South Telescope and on the upper right T80-South Telescope dome at Cerro Tololo Inter-American Observatory (CTIO) in Chile (Bottom).	12
Figure 2.7 – 12 S-PLUS survey filter curves	13
Figure 2.8 – A schematic overview of the STEP pipeline	14
Figure 3.1 – left to right, the search, template and diff image of a transient example from STEP.	18
Figure 3.2 – Bad subtraction	19
Figure 3.3 – Noisy Artifact	20
Figure 3.4 – Diffraction Spikes	20
Figure 3.5 – Near Field object - STEP Artifact	20
Figure 3.6 – Preexisting point	21
Figure 3.7 – Not an obvious Transient - DESGW	21
Figure 3.8 – Dark spot	21
Figure 3.9 – Unknown artifacts	21

Figure 4.1 – The hierarchy, with a summarized explanation of each branch of AI	26
Figure 4.2 – Schematic of a neuron Network information flow. the input X are weightned by w and summed with the bias	26
Figure 4.3 – A schematic representing an example of a neural network with two inputs, three layers with three neurons in the first layer and two in the remaining layers, and two outputs. The weights of each connection are denoted as w and the bias b.	27
Figure 4.4 – A schematic of the iterative training process of a Neural Network	32
Figure 4.5 – Illustrative figure of Loss function convergence for different learning rates values	33
Figure 4.6 – Example of 2D convolutional operation on a RGB input image, it demonstrates how the convolutional operation for two $3 \times 3 \times 3$ filters act on an Input tensor of shape (w, h, 3)	40
Figure 4.7 – Example of different pooling layers acting on a 4x4 feature map.	41
Figure 4.8 – This figure shows a simple example of how a CNN works. It uses an image of Jupiter to demonstrate the process of applying a convolutional filter, detecting features with an activation function, and condensing information with a pooling layer. Credits: —NASA cpy later	42
Figure 4.9 – Confusion Matrix, the rows represent the predicted labels and the columns represent the actual label	44
Figure 4.10–ROC curve, exemplifying a perfect classifier (blue solid line) and a random classifier (dashed line)	45
Figure 4.11–Precision Recall curve, exemplifying a perfect classifier (yellow solid line) and a random classifier (black dashed line)	46
Figure 5.1 – Illustration of the Augmentation Technique, exemplifying various transformations such as 90° , 180° , and 270° rotations, vertical mirroring (vflip), horizontal mirroring (hflip) and combinations of 270° rotations with mirroring	49
Figure 5.2 – A schematic overview of DESGW automated transient detection pipeline	50
Figure 5.3 – Architectures and hyperparameters configuration for each CNN developed in our thesis for the DES-GW search and discovery pipeline	52
Figure 5.4 – CNN1 - Loss value of the validation and train for each epoch in a 5-fold, and the best model selected.	54
Figure 5.5 – CNN1 - Accuracy value of the validation and Train set for each epoch in a 5-fold	54
Figure 5.6 – CNN2 - Loss value of the validation and train for each epoch in a 5-fold, and the best model selected.	55
Figure 5.7 – CNN2 - Accuracy value of the validation and Train set for each epoch in a 5-fold	55

Figure 5.8 – This figure presents a comparative analysis of a diff image before and after contrast adjustment. On the left, a raw, unprocessed image portrays the diff channel in its original state. The image on the right illustrate the same channel after a contrast adjustment has been applied.	57
Figure 5.9 – A schematic overview of DeepSTEP pipeline	58
Figure 5.10–DeepSTEP - Progression of Loss Values for the validation and training sets across epochs. The loss values represent the measure of the model’s performance, with lower values indicating better fitting to the data. . .	59
Figure 5.11–DeepSTEP - True Positive Rate (TPR) progression across epochs for the validation and training sets.	60
Figure 6.1 – CNN-I - Receiver Operating Characteristics (ROC) curve	63
Figure 6.2 – CNN-I - Confusion matrix of first CNN predictions divided by subclass, with a threshold of 0.501	64
Figure 6.3 – CNN-II - Confusion matrix of second CNN predictions divided by subclass	65
Figure 6.4 – CNN-II - output’s score distribution on testset for each class	65
Figure 6.5 – CNN-II - Probabilities of being a transient + host galaxy compared to Autoscan’s probabilities. The boxes extend from the lower to upper quartile of each group, with the median marked by the line.	66
Figure 6.6 – DeepSTEP - Confusion matrix applied to the test data	67
Figure 6.7 – DeepSTEP - Receiver Operating Characteristics (ROC) curve	68
Figure 6.8 – DeepSTEP - Precision-recall (PR) curve	68
Figure 6.9 – Miss-classified example, which was given a probability of 27% of being a Transient.	69

LIST OF TABLES

Table 1 – DES-GW dataset divided by subclass, where the two main classes are highlighted.	48
Table 2 – Table with the list of Hyperparameters tuned with a lower limit, upper limit and step used to search for the best set of hyperparameter on the first and second CNN	51
Table 3 – DES-GW automated transient detection algorithm testset	61

ABBREVIATIONS

CBPF	Centro Brasileiro de Pesquisas Físicas
AI	Artificial intelligence
DIA	<i>Difference Image Analysis</i>
PSF	<i>Point spread function</i>
S-PLUS	<i>Southern Photometric Local Universe Survey</i>
STEP	<i>S-PLUS Transient extension program</i>
SDSS	<i>Sloan Digital Sky Survey</i>
DES	<i>Dark Energy Survey</i>
DES-GW	<i>Dark Energy Survey - Gravitational Wave group</i>
LIGO	<i>Laser Interferometer Gravitational-Wave Observatory</i>
KAGRA	<i>Kamioka Gravitational Wave Detector</i>
NN	<i>Neural Network</i>
CNN	<i>Convolutional Neural Network</i>
ANN	<i>Artificial Neural Network</i>
AUC	Area under the curve
ROC	Receptor operator characteristic
HUDF	<i>Hubble Ultra Deep Field</i>

CONTENTS

List of Figures	xi	
List of Tables	xv	
Contents	xix	
1	INTRODUCTION	1
2	THE SEARCH FOR ASTRONOMICAL TRANSIENTS	5
2.1	Astronomical Transients	5
2.2	DES-GW Transient Detection pipeline	7
2.2.1	Dark Energy Survey (DES)	7
2.2.2	DES-GW search and discovery pipeline	9
2.3	S-PLUS Transient detection pipeline	11
2.3.1	S-PLUS Transient Extension Program	12
3	DIFFERENCE IMAGE ANALYSIS	17
3.1	Artifacts	18
3.2	Current methods	22
3.3	An overview of STEP and DESGW search and discovery pipelines	22
4	A BRIEF INTRODUCTION TO ARTIFICIAL INTELLIGENCE	25
4.1	Introduction to Neural Networks	26
4.2	Activation functions and Loss functions	28
4.3	Neural Network's Learning Process:	30
4.4	Caveats on training an Artificial Neural Network	35
4.4.1	Exploding gradient and vanishing gradient	35
4.4.2	Overfitting and Underfitting	36
4.5	Convolutional Neural Network	38
4.5.1	Convolutional Layer	38
4.5.2	Pooling Layer	40
4.6	Metrics for Evaluation model's performance	42
4.6.1	Loss Function	43
4.6.2	Confusion Matrix (CM)	43

4.6.3	Receiver Operating Characteristics (ROC) curve	45
4.6.4	Precision-recall (PR) curve	46
5	AUTOMATED TRANSIENT DETECTION	47
5.1	Automated Transient detection in Dark energy survey-GW (DES-GW) search and discovery pipeline	47
5.1.1	DES-GW Data collection	48
5.1.2	Convolutional Neural Network Design	50
5.1.3	Training	51
5.2	Deep STEP - Automated transient detection in the S-PLUS Transient Extension Program	56
5.2.1	DeepSTEP Data Collection	56
5.2.2	Neural Network design	58
5.2.3	Training	58
6	RESULTS	61
6.1	DES-GW Automated transient detection algorithm performance on testset	61
6.1.1	CNN-I Performance	62
6.1.2	CNN-II Performance	63
6.2	DeepSTEP performance on testset	66
7	CONCLUSION	71
REFERENCES		73

CHAPTER 1

INTRODUCTION

Until the middle of the 20th century, the only source, or “messenger”, to observe the distant universe was the electromagnetic interaction carrier, commonly called photons, which are observed in wavelengths ranging from Radio to X-rays and later in γ rays. This limited the understanding of the universe to what could be inferred from the electromagnetic spectrum. However, two fundamental discoveries, the detection of neutrinos and gravitational waves (GW) from astrophysical sources, opened new windows to the Universe allowing for a more comprehensive view of the cosmos.

There are four types of what Astronomers call “messenger”:

- Electromagnetic Radiation
- Gravitational Waves
- Neutrinos
- Cosmic Rays

These observations mark the new era of Multi-messenger astronomy [1]: *The exploration of the Universe through combining information from a multitude of cosmic messengers: electromagnetic radiation, gravitational waves, neutrinos and cosmic rays.* Different cosmic messengers, despite belonging to the same source, are produced by distinct processes and thus carry information about different mechanisms within their source, allowing us to observe the universe in a completely different way.

Some astrophysical phenomena, such as the coalescence of two neutron stars, are expected to produce both GW signal and electromagnetic radiation. Therefore, astronomers have been searching for electromagnetic counterparts to GW events. One of the earliest and most well-known examples of a gravitational wave counterpart is the kilonova (AT 2017gfo) associated with the binary neutron star merger detected by LIGO GW170817 in 2017 shown in 1.1.

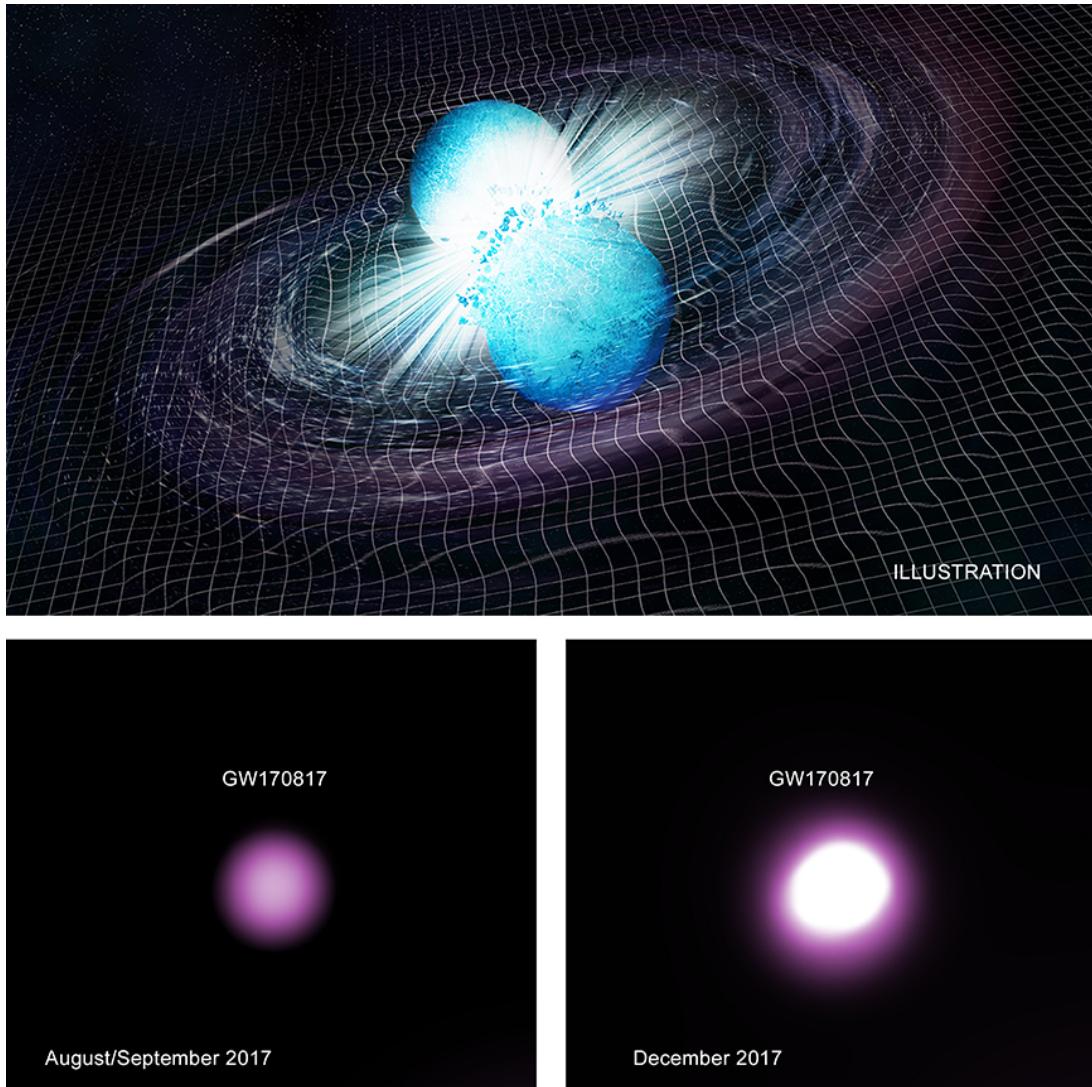


Figure 1.1 – Kilonova (GW170817) a visible example of Gravitational Wave Counterpart - Credit: NASA/CXC/Trinity University/D. Pooley et al. Illustration: NASA/CXC/M. Weiss

Searching for these optical Gravitational waves counterparts is not a simple task. The number of events and the search area generates a considerable amount of data that make it difficult for astrophysicists to analyze all of it in a few hours after the GW alert. Therefore, an algorithm that reduces the need for visual inspection is a useful tool so that astrophysicists can focus on science, discovery, and projection rather than seeing thousands of different images. The main focus of this thesis is the search for transients, astronomical phenomena with durations of fractions of a second to weeks or years, such as kilonovae and supernovae, in the electromagnetic spectrum.

The Dark Energy Survey GW (DESGW) and the S-PLUS Transient Extension Program (STEP) are projects that have been searching for electromagnetic counterparts to gravitational wave events. In both surveys, a transient detection pipeline is used to identify possible transients. The search utilizes Difference Imaging to compare a recent image

(referred to as “search”) of the sky associated with a GW or neutrino detection to a past image (referred to as “Template”) of the same area of the sky. After matching the point spread functions (PSFs) to the search and template images, a subtraction of both images is made, creating a third image called a “difference image”. However, this process creates various types of false positive images. This large number of candidates makes it difficult to follow up on all of them. In this way, this work aims to demonstrate the use of deep learning techniques, such as convolutional neural networks, to reduce the number of candidates that require visual inspection. This dissertation is organized as follows:

Chapter 2 provides a review of optical counterparts of gravitational waves, which are electromagnetic signals produced by astrophysical transients that are associated with gravitational wave events. The chapter explains how the transient detection pipeline works in the DESGW and in the STEP. The chapter covers the basic principles and methods of transient detection, as well as some limitations of the current approach.

Chapter 3 discusses a process known as Difference Image Analysis (DIA), which involves comparing a nightly image with a previously observed sky image to facilitate the identification of new transient sources. The chapter also addresses the main artifacts present in our analysis and the current methods to remove them.

Chapter 4 discusses the basics of neural networks and machine learning, which are the key technologies used in the proposed approach to transient detection. The chapter explains how neural networks learn from data, and provides an overview of the different types of neural networks that are commonly used in machine learning and the most important metrics used when evaluating a binary classification, which are the focus of this study.

Chapter 5 focuses on the automated transient detection algorithm developed in this thesis, in two different programs: the Dark Energy Survey-GW search and discovery pipeline, and the S-PLUS Transient Extend Program. Where we discuss the data collection, training process and neural network architecture design used on each pipeline

Chapter 6 describes the results and evaluation of the proposed approach to transient detection using Convolutional Neural Network.

CHAPTER 2

THE SEARCH FOR ASTRONOMICAL TRANSIENTS

The discovery of the accelerating expansion of the universe [2] using Type Ia supernovae (SNe Ia) has greatly motivated ever larger transient searches in broadband imaging surveys. The associated search pipelines have become increasingly complex in distributing enormous computing tasks needed to rapidly find new transients for spectroscopic follow-ups, and in processing a wide range of data quality.

This chapter provides an overview of the search for Astronomical Transients, presenting the difference imaging pipeline from DES-GW and STEP. This includes an explanation of how the pipeline operates, as well as the essential features that make it an effective tool for identifying transients. Specifically, this chapter focuses on the search for two main types of transients: supernovae and kilonovae.

2.1 Astronomical Transients

The study of transients is a key area of Time-domain-Astronomy, as it focuses on the identification and characterization of objects that exhibit variability on short timescales, i.e., objects with their luminosity varying over time, such as supernovae.

The process of supernova formation depends on the type of supernova, which is classified based on the characteristics of its optical spectra. There are can be broadly separated in:

- Core-Collapse Supernovae (CCSNe)
- Thermal runaway or Type Ia supernovae (SNe Ia)

Core-collapse supernovae occur when a massive star, with a mass greater than 8 times the mass of the Sun, runs out of fuel for nuclear fusion in its core, and the star will no

longer able to support itself by electron degeneracy pressure. This causes the core to collapse inward, which releases a large amount of energy and leads to the formation of a neutron star or black hole. CCSNe are further classified into several subtypes based on the characteristics of their light curve and optical spectra.

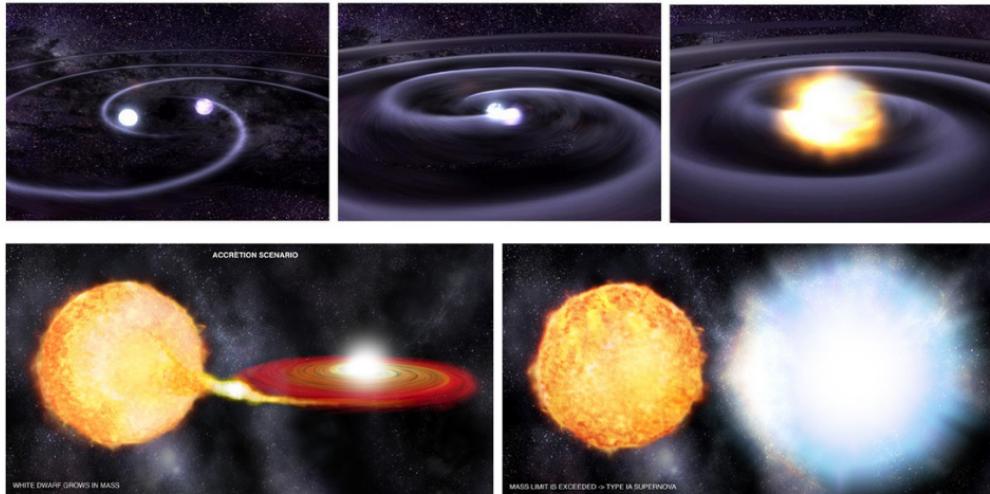


Figure 2.1 – Upper Panel: Two white dwarfs collide, causing a Type Ia supernova.
 Bottom panel: One white dwarf gaining mass from a companion star and eventually exploding into a Type Ia Supernova
Credit: NASA/Dana Berry, Sky Works Digital.

On the other hand, Type Ia supernovae occur when a white dwarf star, which is a stellar core remnant composed mostly of electron-degenerate matter, gains mass from a companion star (Figure 2.1 Upper panel) or merges with another white dwarf (Figure 2.1 Bottom panel). There are several mechanisms by which a supernova of Type Ia can be created, but they all share the same physical process. The white dwarf's mass approaches the Chandrasekhar limit of approximately 1.4 solar masses, it undergoes a runaway nuclear fusion reaction that completely disrupts the star, producing a bright explosion that can be as bright as billions of times the luminosity of our Sun, equivalent to the release of energy of 10^{44} Joules.

Another important type of transient event in astronomy is Kilonovae, which dates back to 1998 when researchers proposed that neutron star mergers lead to observable electromagnetic radiation [3]. However, it was not until the 2010s, with the development of advanced gravitational wave detectors such as LIGO and VIRGO [4], that the first kilonova event was detected and confirmed in 2017 [5].

Different from supernovae, kilonovae are caused by the merger of two compact and dense objects, such as neutron stars or a neutron star and a black hole. The merger

process emits a large amount of radiation and produces gravitational waves detectable by the collaboration LIGO/VIRGO/KAGRA (LVK) [6] and one of the most powerful electromagnetic explosions in the universe. An illustrative picture of how it might work is shown in Figure 2.2.



Figure 2.2 – Two Neutron stars collide, causing a Kilonova and Gravitational Waves. Credit: NASA/Dana Berry, Sky Works Digital <https://svs.gsfc.nasa.gov/12949>.

The goal of this dissertation is to identify possible transients by integrating an AI into the current difference imaging pipeline. The next section will describe the transient detection pipeline used to discover point-source transients in DES-GW and STEP.

2.2 DES-GW Transient Detection pipeline

To identify scientifically valuable transients, imaging surveys have historically adopted a manual approach, employing astronomers to visually inspect images facilitated by a difference imaging algorithm. This work focuses specifically on electromagnetic counterparts that emit in the optical wavelengths and are bright enough to be detected by existing instruments, such as the Dark Energy Camera [7] used in the DES-GW pipeline and the T80-South [8] (T80S) used in the STEP pipeline.

2.2.1 Dark Energy Survey (DES)

The Dark Energy Survey [9] (DES) is a six-year optical imaging campaign with the goal of mapping hundreds of millions of galaxies, detecting numerous supernovae and finding patterns of cosmic structure that could help scientists discover the nature of dark energy, which is accelerating the expansion of our universe.

DES comprises two multi-band imaging surveys:



Figure 2.3 – DECam and the Blanco Telescope at the Cerro Tololo inter american observatory (CTIO).

- Wide area survey: Covers a wide-area of sky (5000 deg^2) in the southern celestial hemisphere in the g, r, i, z and Y bands. Here, the exposure times are long and cumulative to detect a large number of very faint galaxies.
- Time domain survey: Covers a smaller area of the sky (27 deg^2), but exposures are repeated a large number of times at regular, six night intervals over the course of the survey in the g, r, i, z bands, in order to detect transients.

Multi-band imaging is an astronomical technique to take images of the sky across different wavelengths of light. A set of filters are placed in front of the camera to capture light only within specific wavelength ranges, so that only the desired band of light can be observed. Each band covers a different range of wavelengths, spanning from 400nm to 1080nm as shown in Figure 2.4.

Both of these multi-band imaging surveys uses an extremely sensitive 570-Megapixel digital Camera, Dark energy Camera [7]. DECam is mounted on the Blanco 4m telescope (2.3) at the Cerro Tololo inter-American observatory (CTIO) and the data are processed by the DES data management system [10];[11] at the National Center for Supercomputing Applications (NCSA). The DECam is composed of 62 science image CCDs, each with 2.048×4.096 pixels, and 8 CCDs for guiding. The total active area covered by all working CCDs is around 2.7 deg^2

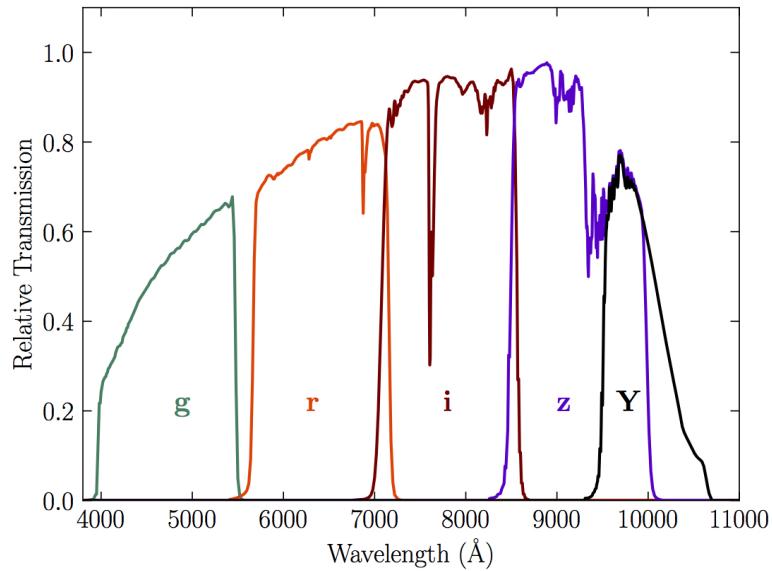


Figure 2.4 – Standard bandpasses for the DECam grizY filters, including atmospheric transmission (airmass = 1.2) and the average instrumental response across the science CCDs. Credit: NOIRLAB SCIENTISTS’ WEBSITE <https://noirlab.edu/science/programs/ctio/filters/Dark-Energy-Camera>.

2.2.2 DES-GW search and discovery pipeline

DES-GW is a search and discovery program for electromagnetic (EM) signatures of GW events, which is integrated with LIGO/VIRGO alerts that triggers the pipeline into identifying these signatures. While there are multiple reasons to search for GW electromagnetic counterparts, the program’s primary goal is to perform a measurement of the Hubble Constant with the best precision afforded by the available data. The DES-GW search and discovery pipeline is based on the DES Single-epoch processing pipeline [12] and a modified version of DES Supernova processing pipeline [13].

The first step of the image preparation is the Single Epoch (SE) pipeline, which occurs each night. It consists of an image correction stage and an object cataloging stage (known as First Cut). A schematic overview of the complete single epoch processing of DES-GW search and discovery pipeline is illustrated in the figure 2.5.

1. Image correction: This steps aims to make the raw images science ready by applying a series of corrections and masking. A brief description of each correction is given below:
 - a) Crosstalk corrections: Crosstalk is an error that occurs when the electrical signals from one CCD detector leaks into an adjacent detection. This can result in errors in the pixel values of the adjacent detector, affecting the quality of the data.

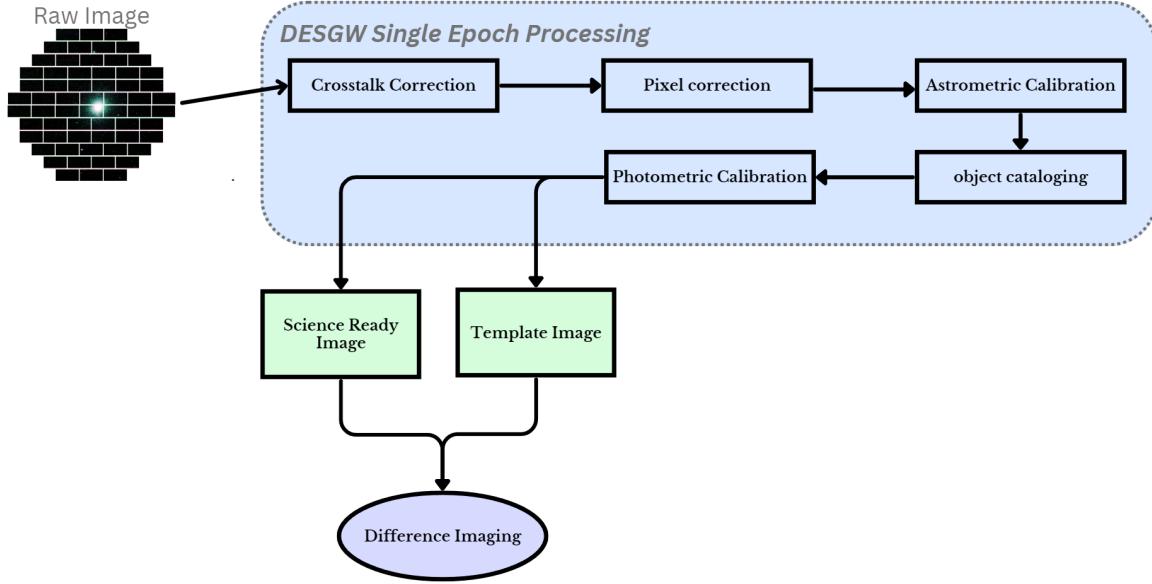


Figure 2.5 – A schematic overview of the complete single epoch processing of DES-GW pipeline

- b) Pixel corrections: A series of pixel-level corrections are applied to the images, including:
 - i. Bad pixel masking [14]: CCDs may contain a few defective pixels that need to be masked out to prevent them from affecting the data. In addition to masking out defective pixels, the SE pipeline also corrects for hot pixels, which are pixels with a high dark current, and for bleed trails, which are caused by bright sources saturating the CCDs.
 - ii. Flat fielding: This corresponds to correcting the combined optical-system and CCD throughput at each pixel so that each pixel on the CCD would respond equally to a source with the same photon flux. Flat fielding removes the effect of the pixel-to-pixel sensitivity variations across the array as well as the effect of dust or scratches on the CCD window
 - iii. A “brighter-fatter” correction: The brighter-fatter effect is caused by charge building up on well-illuminated pixels and pushing away nearby charge, effectively broadening the PSF of bright objects.
 - iv. Pixel non-linearity correction: The pixel response of CCDs is non-linear at low fluxes and near saturation, as determined from flat field images. A non-linearity correction is applied to the images to make the pixel response linear.
 - v. A conversion from DN to electrons: The images are converted from units of Digital Numbers (DN) to electrons, which is the physical unit of the

CCD signal.

- vi. Bias subtraction [7]: The bias in a CCD camera is a DC offset applied to all pixels so that when the voltage in each pixel is converted to a number, the number will always be positive. In an ideal CCD the bias would be the same for every pixel and not change over time. In practice, the bias is slightly different for each pixel. Then a bias-subtraction, to deal with that variability, is made by using a bias image, which is a picture taken with the shutter closed and zero exposure time.

After this stage, a catalog of the brighter objects is created using Source Extractor (SEXtractor) [15], to be used in the astrometric calibration.

- c) Astrometric calibration: This stage uses the positions of these bright objects in the image to determine the pointing and orientation of the telescope. This is done using SCAMP [16], which compares a catalog of known star positions to the positions of the stars in the image.
- 2. The Firstcut processing: This stage calculates an astrometric solution with SCAMP, performs bleed trail masking, fits and subtracts the sky background, divides out the starflat, masks cosmic rays and satellite trails, measures and models the point spread function (PSF), performs object detection and measurement using SEXtractor, and finally performs image quality measurements.

The outputs of SE processing are the inputs to the difference imaging stage, which will be discussed in details in Chapter 3 after an introduction to S-PLUS transient detection pipeline.

2.3 S-PLUS Transient detection pipeline

The Southern Photometric Local Universe Survey [8] (S-PLUS) is an imaging survey that covers a region of 9300 deg^2 of the celestial sphere in 12 optical bands using a dedicated 0.8 m robotic telescope, T80-South. The telescope located at Cerro Tololo Interamerican Observatory (CTIO), Chile.

The S-PLUS is divided into two main sub-surveys:

- The Main Survey (MS): Covering an area of $\approx 8000 \text{ deg}^2$ with a single visit to each field per filter. The MS strategy is motivated by the requirements set by the extragalactic science, with accurate photometric redshifts for objects down to $i=21$ $\sim \text{mag}$ [17], allowing the study of the local large-scale structure, star formation rates and stellar populations.



Figure 2.6 – On the upper left The interior of the T80-South Telescope and on the upper right T80-South Telescope dome at Cerro Tololo Inter-American Observatory (CTIO) in Chile (Bottom).

- Galactic Survey(GS): Covering an area of 1300 deg^2 in the milk Way plane in all 12 filters, including two galactic regions, the bulge and the disk, using the same strategy as the Main survey.

There are other three sub-surveys are the ultra-short survey (USS), the marble field survey (MFS), and the variability fields survey (VFS). The S-PLUS includes 12 bands, consisting of 5 Broad bands similar to those used in DES, with the addition of 7 narrow bands as displayed in Figure 2.7.

The telescope T80S is equipped with an optical imager, T80Cam-S, consisting of a 12-filter system. The detector used is a $9232 \times 9216 \text{ } 10\mu\text{m} - \text{pixel}$ array manufactured by the company e2v. The telescope plate scale at the detector is 0.55 arcsec/pixel, and the field of view (FoV) of the camera is $1.4 \times 1.4 \text{ deg}^2$.

2.3.1 S-PLUS Transient Extension Program

The S-PLUS Transient Extension Program (STEP) pipeline was developed with the objective of identifying, reporting and characterizing transients using data from S-PLUS Main Survey or follow-ups with T80S. This subsection describes the main stages used

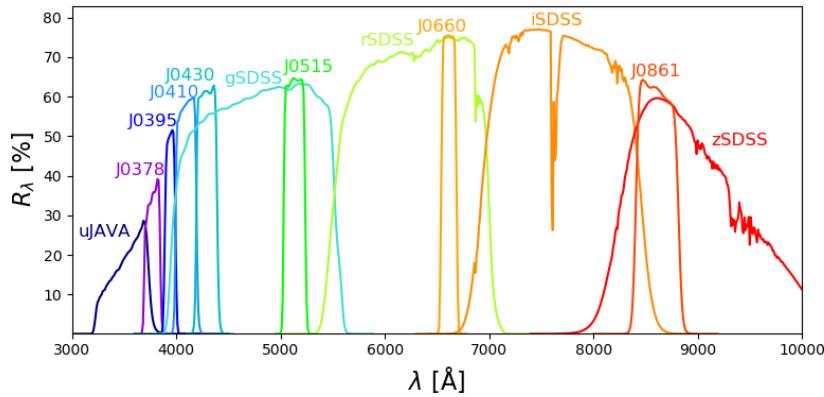


Figure 2.7 – 12 S-PLUS survey filter curves Credit: Noirlab scientists’ website: <https://datalab.noirlab.edu/splus/>.

to prepare the raw images for use in the difference imaging analysis, similarly to the prepossessing steps described for DES-GW search and discovery pipeline.

The pipeline starts each day by collecting raw images from S-PLUS Main Survey, captured using 4 broad band filters (g , r , i , z) and each filter with 33, 40, 46 and 56 seconds of exposure time respectively. These images are stored as fits files, and in addition to the images, a file containing R.A. and Dec. coordinates for the center of each field retrieved is also saved on the CBPF-Server. It is important to highlight that the raw image does pass through a partial reduction pipeline in S-PLUS data flow, this partial reduction of each individual image includes overscan subtraction, trimming, bias subtraction, master flat-field correction, cosmetic corrections (e.g. satellite and airplane trails, cosmic rays and bad pixels masking), and fringing pattern subtraction (which is usually only necessary for the z filter) [8]. Besides all these corrections, the image can still be called Raw, it is only after the STEP reduction pipeline that the image becomes scientifically ready. A schematic overview of the complete transient detection pipeline is shown in Figure 2.8.

After transferring the image to the CBPF’s server, the raw image passes through an initial astrometric calibration. Then, the calibrated image goes into the image corrections stage, which starts by reducing image distortions using a series of open-source software called IRAF [18]. This includes a tool that matches a star catalog (usually obtained from the GAIA DR3 survey) with the stars in T80’s camera data. Distortions in this context refer to any deviation from the ideal imaging properties of the telescope and camera system, such as geometric distortions, pixel irregularities, and variations in sensitivity across the detector. On the next stage, for each collected image, using source extractor [15] (SExtractor) to measure the background levels and generate a catalog of saturated stars. From this catalog and output segmentation maps in SExtractor, we mask out saturated stars and the unique diffraction spike pattern for T80S based on the number of saturated pixels within each star.

Next, SWarp [19], which is a program that resamples and co-adds together FITS

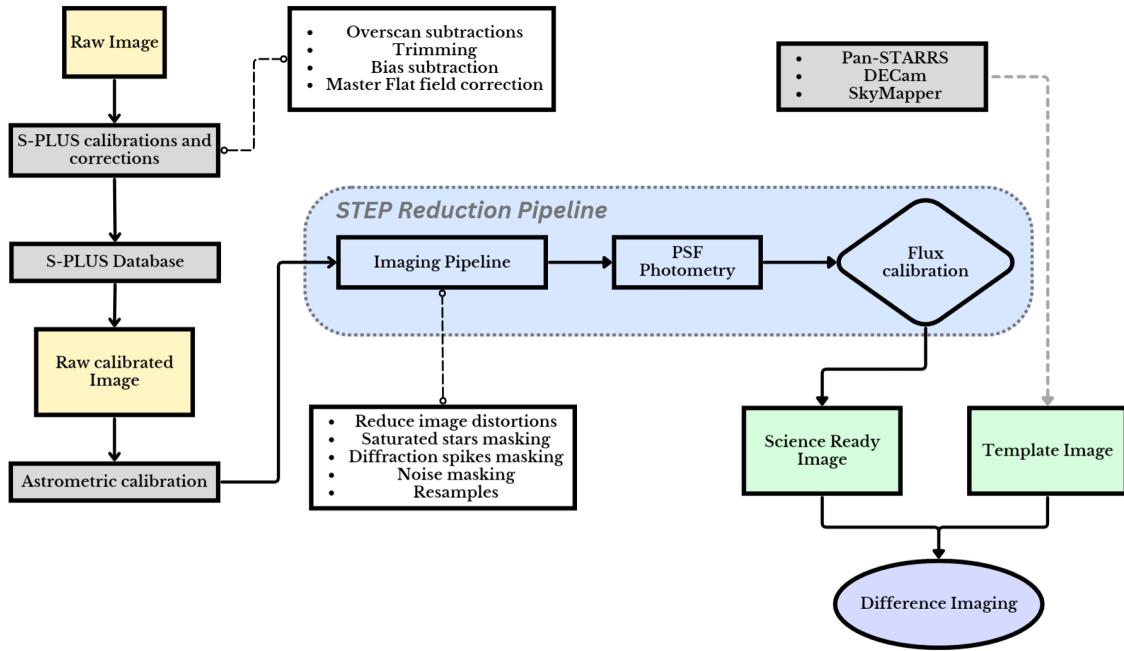


Figure 2.8 – A schematic overview of the STEP pipeline

images, is used to assure that the science and template images have a common and uniform pixel scale of $0.55''$ across the image, a common field center, and a uniform size of 10000×10000 per frame. With all these corrections made, the next step is to perform a PSF photometry using DoPHOTS [20] and flux calibration by comparing to the Pan-STARRS DR2 [21] and SkyMapper [22] photometric catalogs. Point spread function photometry is the measurement of the flux of bright objects by fitting an analytical function called the point spread function (PSF) to each bright point source object. Then, we compare this flux measurement with catalogs of other known surveys like SkyMapper [22] for known stars in order to get a zero point magnitude and calculate the absolute magnitudes of any interesting objects in the data. The process described above is a key step, allowing the identification and characterization of sources in astronomical images.

Template images, as mentioned previously, are images from the sky taken in previous observations. These images are generated on a server in which we give as input a file containing a set of fields with their RA and DEC coordinates, then the template server determines if the files have coverage in Pan-STARRS [21], DECam [23], or SkyMapper [22], and downloads all available imaging covering that field. Thus, template images are made using cutout images available in public releases of different surveys. Once this process is finished, we retrieve those images back to our main server to perform the reduction process. The reduction process for template images is done similarly to those collected from S-PLUS, but first the templates images undergo through sky subtraction, normalization using their reported photometry calibration, and convolution to a uniform PSF shape using SWARP [19] to produce an output template image with the input field center and pixel scale of the T80S camera. Finally, the search image and the template

are ready to go into the Difference imaging pipeline. A detailed discussion about the Difference imaging analysis is presented in Chapter 3.

CHAPTER 3

DIFFERENCE IMAGE ANALYSIS

There are a variety of astrophysical phenomena that can cause variations in the brightness and color of astrophysical objects, including explosive stellar death (supernovae, kilonovae, Gamma Ray Bursts, etc.), less dramatic and powerful stellar variability (flares, pulsations), and variability arising from geometric effects (such as planetary transients and microlensing). The time scales for these phenomena range from seconds to years. When variations are sudden and terminal, as in the case of supernovae, or stochastic and unpredictable, as with stellar flares, they are referred to as "transients." To detect these transients, a process known as Difference Image Analysis (DIA) is employed. DIA involves comparing a nightly image with a previously observed sky image and subtracting them. This technique was first pioneered by Tomaney and Crotts [24] and later formalized by Alard and Lupton [25], and forms the basis of the two transient search pipelines described in Chapter 2.

Templates are typically constructed as stacks of high-quality sky images. These images must then be aligned with the nightly image, typically called the "search image". The alignment and calibration of these images are two critical steps that must be carefully performed to produce a high-quality subtracted image, known as a "difference image"(diff). For a careful calibration, it must account for the fact that the images used to build the template and the search images are taken principally within different atmospheric conditions, generating variations in the quality of the images [26]. Another complication is the construction of a proper template. Typically, templates are built by stacking tens of images taken under favorable sky conditions at different times. This improves the image quality but also mitigates issues related to variability in the astrophysical objects captured in the image. As a result, the template image is of higher quality than the search image, requiring it to be degraded to match the search image's point spread function (PSF) and scaled to match its brightness.

The central point of the difference imaging approach is to find a convolution kernel

“K” that matches the PSFs of two astronomical images, “I” (referred to as the search image) and T (referred to as the template). In terms of least-squares, this is equivalent to solving for a kernel that will minimize the sum:

$$\sum([T \otimes K](x, y) - I(x, y))^2 \quad (3.1)$$

To address the computationally expensive task of matching PSFs, the kernel can be decomposed in terms of simple functions, such as Gaussian functions and the method of the least squares can be used to determine the best values for the kernel. The fitted solution of one search image can be determined in a short computational time.

In both DESGW and STEP pipelines, HOTPANTS [27], an open-source software that enables difference imaging on astronomical image data, is used. This is responsible for the alignment, calibration, and subtraction steps of the process. However, The DESGW pipeline performs the difference imaging step per CCD, thus a separate subtraction between the search image and each template for each CCD is done, and then it combines the difference images, rather than do a single subtraction on one combined template. While this approach is clearly slower than doing a single subtraction, it avoids potentially large PSF variations that could arise when combining templates taken in potentially very different observing conditions. The STEP pipeline uses the traditional approach of performing a subtraction in the full-sized search image and after the subtraction, a crop around each transient candidate is made.

3.1 Artifacts

After the execution of difference imaging, it becomes possible to identify potential transients by selecting images that exhibit bright regions after the subtraction. An example of a transient, with its corresponding search, template, and diff. images, are shown below (Figure 3.1):

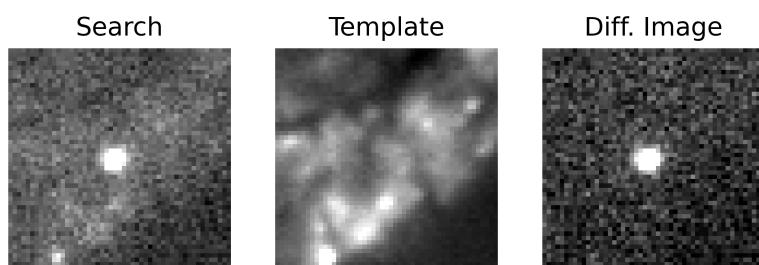


Figure 3.1 – left to right, The search, template and diff. image of a transient example from STEP.

However, this subtraction stage generates false positive candidates, also known as artifacts. Artifacts can arise due to poor PSF matching, imperfect alignment, defective pixels, near-field objects, and other inaccuracies in preprocessing.

Artifacts dominate the difference imaging data, with most images produced daily being false positives. Since the primary objective of these two pipelines (STEP and DESGW) is to quickly find candidate objects for spectroscopic characterization, excluding these artifacts by visual inspection is not a viable option. To reduce the number of artifacts, each transient detection pipeline has unique steps after the difference imaging analysis.

In STEP pipeline (Figure ??), after the difference imaging, the PSF for the remaining possible sources is recalculated, and the zero-point magnitude is calculated. Then, a catalog containing all the remaining candidates, their unique identifiers, celestial coordinates, and magnitudes are derived. However, this catalog is often contaminated with various types of artifacts that need to be removed.

In the DESGW search and discovery pipeline, after DIA, SEXtractor [15] is used on the difference image to identify candidates. Objects detected are then filtered through a set of selection criteria in Table 1 of K.Herner et al. [28]. Despite these selection criteria, several types of artifacts can still be labelled as transient candidates. These artifacts are described below, where "Not an obvious transient" is an artifact exclusively from DESGW, due to the fact that in a follow-up program in order to maximize the probability of finding a new kilonova, it's necessary to remove possible transients that are not easily identified by the diff image.

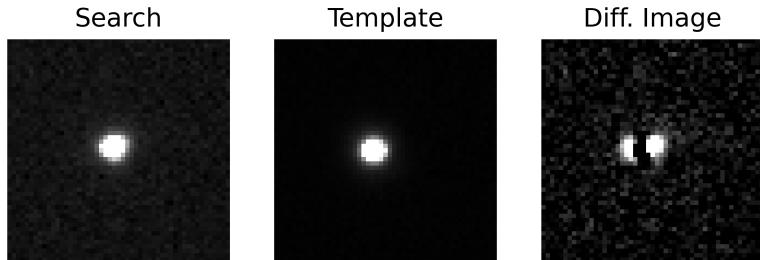


Figure 3.2 – Bad subtraction

- **Bad Subtraction:** The most common example of an artifact is known as a "bad subtraction". When the search and template images are not perfectly aligned, or if there are variations between the point spread functions (PSFs), the subtraction process can create undersubtracted and oversubtracted regions in the difference image.
- **Noisy search/template image:** Another type of artifact occurs when realizations of Poisson noise produce groups of pixels that can resemble an object in the search image. In some cases, groups of under-fluctuations in the template image can also create the appearance of an object after subtraction.

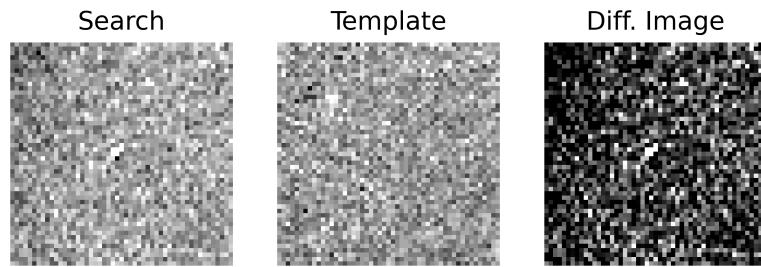


Figure 3.3 – Noisy search/template image

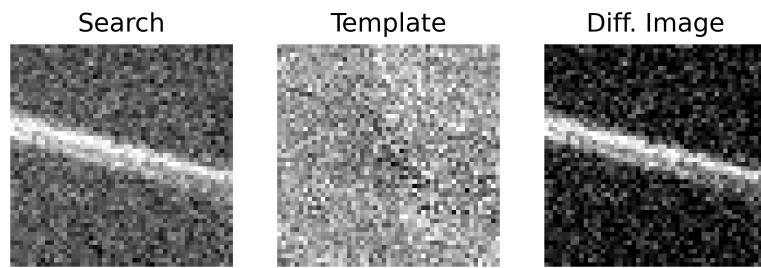


Figure 3.4 – Diffraction Spikes

- **Diffraction Spikes Artifact:** Another common false positive occurs due to diffraction spikes that were not eliminated in the preprocessing and can produce a large bright line in the search image.

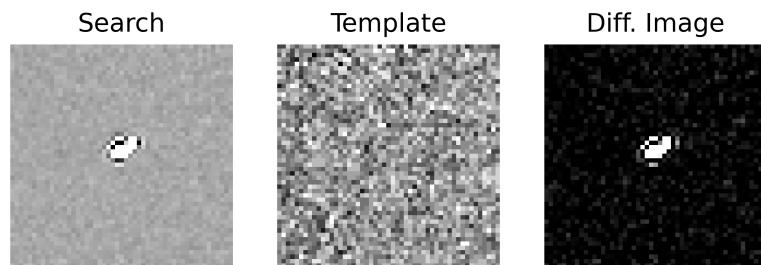


Figure 3.5 – Near Field object

- **Near field object:** A point-like source in the difference image could appear due to a moving object that was detected in a previously empty patch of sky.
- **Preexisting point source:** Another type of artifact is an object that was already visible in the template image but produced a bright spot in the difference image due to changing brightness (e.g., variable Milky Way stars).
- **Not an obvious transient:** These are images that seem to contain a host galaxy, and a new object may appear in the search image. However, the resulting difference image is inconclusive. This class usually contains galaxies with small variability in their centers, rather than supernovae or kilonovae producing an obvious transient.

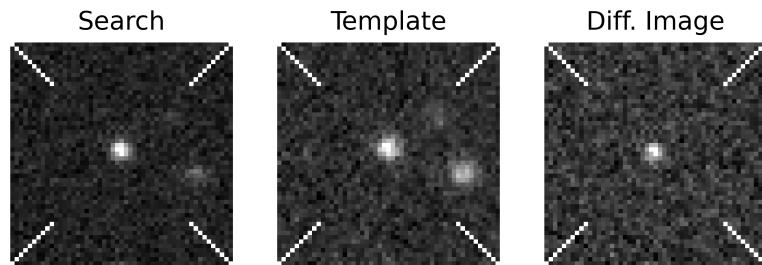


Figure 3.6 – Preexisting point

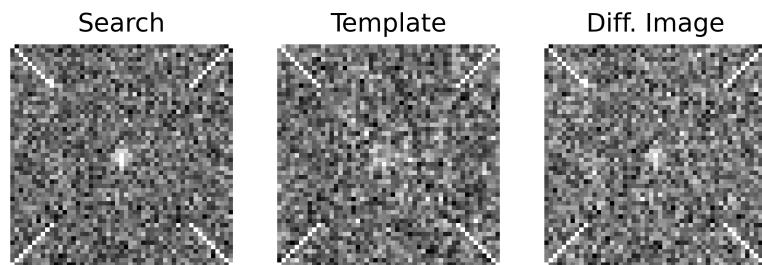


Figure 3.7 – Not an obvious Transient - DESGW

- Unknown artifacts: There are others less common types of false positives, like dark spot and unknown artifact, which the explanation of how they occur is not clear.

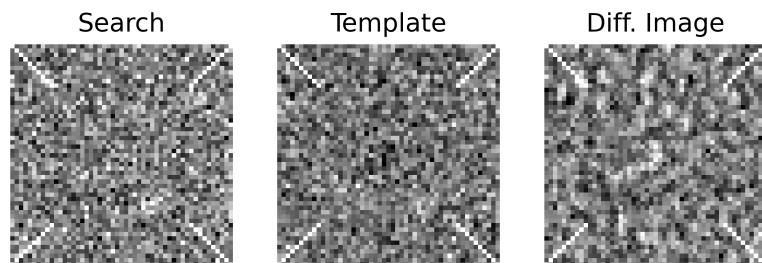


Figure 3.8 – Dark spot artifact

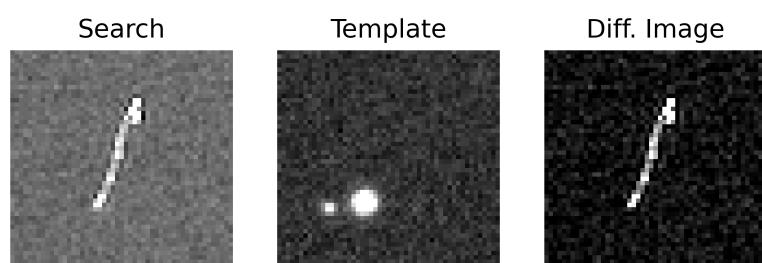


Figure 3.9 – Unknown artifacts

3.2 Current methods

Knowing that most of the transient candidates are artifacts, it is necessary to have a stage focused on removing these artifacts or at least selecting the real transients. This stage uses two main methods described below:

- Autoscan (DESGW) - Autoscan [29] is a machine learning tool, that uses a Random Forest (RF) supervised learning model, specialized in removing Bad Subtractions images from the Dark Energy Survey dataset. Autoscan is based on 38 features derived from the template, search, and difference images. It takes as input features such as the ratio of PSF flux to aperture flux on the template image, the magnitude difference between the detection and the nearest catalog source, the SEXtractor-measured [15] *SPREAD MODEL* of the detection and others custom-built features that were developed to represent quantitatively what astronomers would leverage in visual inspections. Autoscan was trained using only simulated data and returns a score between 0, an obvious artifact, and 1, a high-quality detection, achieving a false negative rate of 5% and a false positive rate of 4% on their validation set.
- Visual inspection (STEP / DESGW) - This is the most straightforward way to select candidates. Essentially, a specialized astronomer identifies real transients by eye on a Webpage.

There are some issues with relying only on the two methods mentioned above. Firstly, Autoscan only removes bad subtractions, and there are several different types of artifacts. Although bad subtraction is the most common type, with the current O4 season, the number of images is expected to rise dramatically, necessitating a faster and more generalized solution. A similar issue arises with visual inspection in the STEP pipeline. Without an artifact-removing algorithm, astronomers must manually examine hundreds/thousands/tens of thousands of images. This thesis aims to address this issue by proposing a Deep learning model that can classify images as either transients or non-transients, thereby reducing the number of images that requires visual inspection.

3.3 An overview of STEP and DESGW search and discovery pipelines

The DESGW search and discovery pipeline, which aims to discover Kilonovae events associated to GW alerts [30], involves a complex data flow that begins with an alert from LIGO/VIRGO containing information about the event, including luminosity distance, classification of the progenitors and a skymap depicting the probability area localization for the GW event and thus, the region with the highest probability of identifying an

optical counterpart to that trigger. If the DESGW team decides to follow-up, then the transient detection pipeline starts with the acquisition of raw images using DECam and ends with the identification and characterization of potential optical gravitational wave counterparts. The pipeline consists of two main stages: the Single Epoch pipeline and the Difference Imaging Pipeline.

The SE pipeline is run each night and includes image correction to prepare raw images for scientific analysis, as well as cataloging of brighter objects for astrometric calibration. The Difference Imaging Pipeline uses a template image and a science-ready search image to perform a pixel by pixel subtraction. The resulting difference image is then passed to a machine learning algorithm called Autoscan and a convolutional neural network classifier (CNN) presented in this dissertation. The candidates identified by these methods are then selected for further analysis through visual inspection on the DESGW webpage.

The STEP pipeline, focused in the discovery of Supernovae, begins by daily downloading raw calibrated images from the S-PLUS Main Survey database. These images then undergo corrections and PSF photometry in the STEP’s reduction stage. The difference imaging analysis is performed next, which involves subtracting the template and search images in the full-sky image. The resulting images are then processed through DeepSTEP, which is another CNN model also presented in this work. Finally, the candidates are selected through visual inspection on the STEP’s Webpage. The output of DeepSTEP is a list of candidates for transient astronomical phenomena. Overall, the STEP pipeline’s goal is to detect transient astronomical phenomena using data from the T80S telescope and also performs follow-ups on LIGO/VIRGO events.

As the field of Artificial Intelligence (AI) continues to evolve and the volume of data increases exponentially, integrating Deep Learning techniques into the pipelines has become increasingly relevant. In order to discuss the two Convolutional Neural Network classifiers developed in this thesis, a brief introduction to Deep Learning, CNNs, and the associated metrics for classifiers will be presented in the following section.

CHAPTER 4

A BRIEF INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) is a field of computer science with synergies to many other areas, such as physics, statistics, and astronomy, that aims to develop intelligent systems that can perform tasks that typically a human can do, such as learning patterns, problem-solving, and even more elementary level.

The term "AI" was first coined in 1956 at the Dartmouth Conference [31]. Since then, the field has grown rapidly. Machine learning (ML) is a subfield of AI that uses models built from Data. These models, also called Data-Driven models, enable computer systems to learn from data without being explicitly programmed.

Finally, Deep learning (DL) is a subset of machine learning and AI that involves the use of artificial neural networks to learn from data (Data Driven). However, Deep learning is different from traditional machine learning in several ways. Traditional machine learning algorithms rely on custom-made features that are fed into the model as inputs, whereas deep learning algorithms can learn these features directly from the raw data, making them more flexible and adaptable to a wide range of tasks, including image recognition, speech recognition, natural language processing, and financial forecasting. For example, convolutional neural networks (CNNs) are commonly used for image recognition tasks because these convolutional layers can filter an image for a particular feature.

The history of Deep learning can be traced back to 1943, when Walter Pitts and Warren McCulloch created a computer model inspired on the neural networks [32] (NN) of the human brain. Neural networks are a key component of deep learning. They are an algorithm consisting of layers of interconnected nodes, or neurons, that process and transmit information. The next subsections introduce the basics of NN, CNN, and the main problems faced when using Deep learning on image classification tasks. Figure 4.1 illustrates the hierarchy and a summary of the subfields of AI.

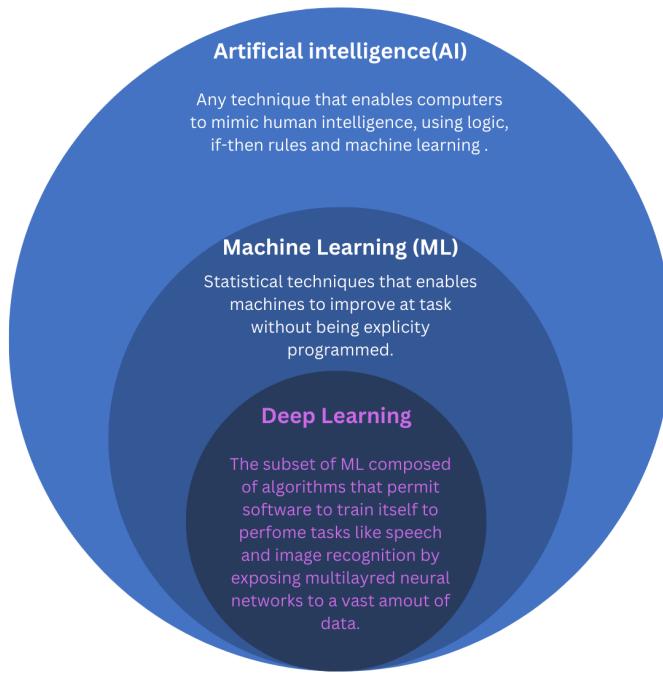


Figure 4.1 – The hierarchy, with a summarized explanation of each branch of AI.

4.1 Introduction to Neural Networks

Neural Networks (NN) are composed of neurons, each individually performing only a single computation. The power of a NN comes instead from the complexity of the connections these neurons can form. The fundamental component of a Neural network is the individual neural, a diagram with a neuron (or unit) and one input is displayed in Figure 4.2.

The input to a neuron is denoted as x , and its connection to the neuron has a weight represented as w . When a value flows through a connection, it's multiplied by the connection's weight. Therefore, the value that reaches the neuron from input x is $w \times x$. A

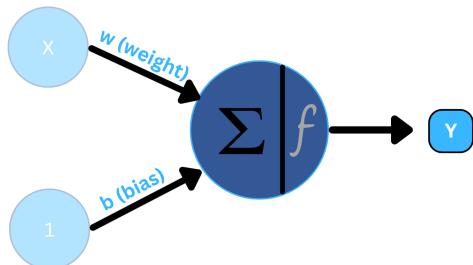


Figure 4.2 – Schematic of a neuron Network information flow. the input X are weighted by w and summed with the bias

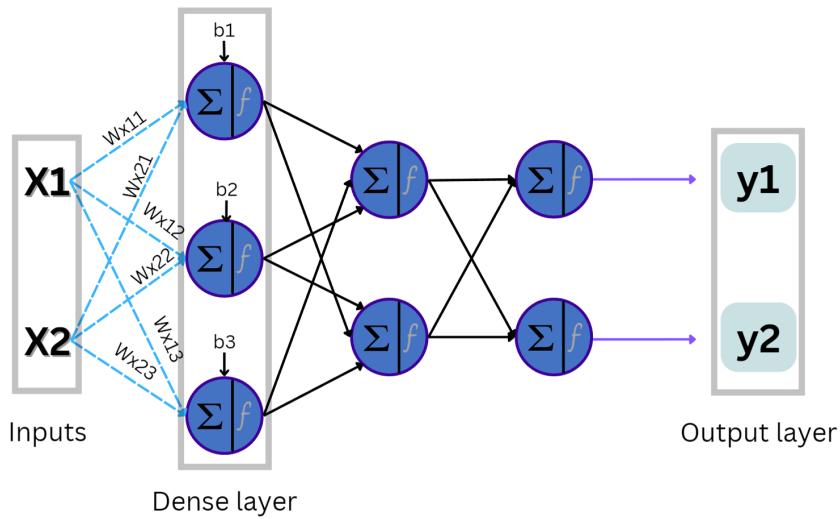


Figure 4.3 – A schematic representing an example of a neural network with two inputs, three layers, with three neurons in the first layer and two in the remaining layers, and two outputs. The weights of each connection are denoted as w and the bias as b .

neural network "learns" by adjusting its weights, including a special weight known as the bias, denoted as \mathbf{b} . Unlike other weights, the bias does not have any associated input data. Instead, for this example, it is defined as 1 in the diagram and is added to the weighted input value to reach the neuron as b (since $1 \times b = b$). The bias enables the neuron to modify its output independent of its inputs.

The value of the neuron's output is denoted as \mathbf{y} . To obtain the output, the neuron sums up all the values it receives through its connections. Although a neuron's activation function can take on various forms, in this example, the activation function is linear and can be expressed as $y = w \times x + b$.

Neural networks typically organize their neurons into layers. A dense layer is formed by grouping linear neurons that share the same set of inputs. Activation functions are utilized to add non-linearity to neurons, replacing the linear function. These functions are applied to the summation and generate the artificial neuron's output signal. There are numerous activation functions available, each better suited for specific types of problems.

Ultimately, the model can be expanded to include multiple inputs, multiple layers of neurons, and different activation functions. Figure 4.3 displays a schematic representing an example of a neural network with two inputs, three layers, with three neurons in the first layer and two in the remaining layers, and two outputs.

In summary, each neuron has multiple input entries for a multilayer neural network denoted as x_1, x_2, \dots, x_n . These inputs are weighted by their respective weights, represented as w_1, w_2, \dots, w_n , and the respective bias b_n is added. An activation function denoted as f_n is then applied to the weighted sum, and the result is passed on to the next layer.

The weights are used to assign different relative importance to each input.

4.2 Activation functions and Loss functions

One of the most widely used activation functions is Rectified Linear Unit or ReLU [33]. When the ReLU activation function is applied to a linear unit, the output becomes $\max(0, w \times x + b)$. This means that any negative value will be set to 0, and positive values will remain the same. In other words, the function is a straight line for positive inputs and 0 for negative inputs, which introduces non-linearity into the network.

There are other slight variations of the Relu function:

- SeLU (Scaled Exponential Linear Unit) [34] is another activation function that has been found to perform well in deep neural networks. The main advantage of SELU is that we can be sure that the output will always be standardized due to its self-normalizing behavior.
- GeLU (Gaussian Error Linear Unit) [35] is a more recent activation function that is based on the Gaussian cumulative distribution function. The GELU nonlinearity weights inputs by their percentile rather than gates inputs by their sign as in ReLUs. Consequently, the GELU can be thought of as a smoother ReLU.

Activation functions are also used in the last layer of a neural network. The activation function in the output layer shapes the output values of the network to match a specific problem which the network is trying to solve. For classification tasks, the three most used activation's function:

- Sigmoid [36] is an activation function that maps input values to a range between 0 and 1. It is often used in the output layer of a neural network for binary classification tasks, where the goal is to produce a probability value between 0 and 1.
- Softmax [36] is an activation function that maps input values to a probability distribution over multiple classes. It is often used in the output layer of a neural network for multi-class classification tasks, where the goal is to produce a probability value for each categorical class that sums to 1 for each input vector.
- Tangent hyperbolic (Tanh) is an activation function that maps input values to a range between -1 and 1. It is similar to the sigmoid function but has a symmetric range around 0.

for regression tasks, some commonly used activation functions include:

- linear activation function, also known as the identity function, represents a straightforward mapping where the output is equal to the input without any transformation

or non-linearity. This activation function is commonly used when the desired output is expected to be a continuous value.

- ReLU (Rectified Linear Unit) [33] is a widely utilized activation function in neural networks, particularly in deep learning models. It introduces non-linearity by mapping negative input values to zero and maintaining positive values unchanged.
- Leaky ReLU [37] is a variation of ReLU that addresses the limitation of possible "dead neurons" (i.e., neurons that never activate) observed in standard ReLU. In Leaky ReLU, negative inputs are assigned a small negative value, thus a non-zero gradient.

Up to this point, we elaborate on some key elements needed to design a neural network architecture and the principles of neuron computation. However, it does not provide any guidance on specifying the particular problem the network needs to solve. This crucial task is accomplished by utilizing a loss function. A loss function is a mathematical function that measures how well a neural network's predictions match the expected values (True values) of the data it is being trained on.

The value of the loss function is used to adjust the network's weights and biases during training, thus choosing the most important features that yield the best results for the specific task it was designed. There are many types of loss functions, each designed for specific problems. Here are some examples of commonly used loss functions:

- Mean Squared Error (MSE) - This is a commonly used loss function for regression problems. It is defined as the average of the squared differences between the predicted output and the true output. The formula for MSE is: $MSE = \frac{1}{n} \times \sum(y_i - y'_i)^2$ where n is the number of data points, y is the true output, and y' is the predicted output.
- Mean absolute Error (MAE) - The MAE loss function is also used to evaluate the performance of regression models, and it is beneficial when dealing with outliers in the dataset, as it is not as sensitive to extreme values as other loss functions. The formula for MAE is as follows: $MAE = \frac{1}{n} \times \sum |y_i - y'_i|$ where n is the number of data points, y_i is the true output, and y'_i is the predicted value. The lower the MAE value, the better the model's predictions match the actual values.
- Binary Cross-Entropy - This is a loss function that is commonly used for binary classification problems. Using the cross-entropy formula, it measures the difference between the predicted output and the true output. The equation for binary cross-entropy is $BCE = \frac{1}{n} \times (y \times \log(y') + (1 - y) \times \log(1 - y'))$ where n is the number of data points, y is the true output (either 0 or 1), and y' is the predicted output (between 0 and 1).

- Categorical Cross-Entropy - This is a loss function that is used for multi-class classification problems. It also measures the difference between the predicted output and the true output using the cross-entropy equation. The categorical cross-entropy can be defined as: $CCE = \frac{-1}{n} \times \sum_i \sum_j^{nclass} (y_{ij} \times \log(y'_{ij}))$ where n is the number of data points, y_{ij} is the true output for the i -th data point and the j -th class, and y'_{ij} is the predicted output for the i th data point and the j th class.

The process of training a neural network is finding ways to reduce the value of the loss function, as will be shown in the next section. This demonstrates the importance of choosing a loss function and an activation function of the output layer in accordance with the objectives of the neural network because all optimization will be done on it. If the loss function is not suitable for the task aimed, the network might learn very well, i.e., greatly reduces the value of the loss, but has poor results in its real application. Some usual combinations of Loss functions and activation functions on the output layer for each type of problem are listed below [36]:

- Binary Classification: For binary classification problems where the output should be a probability value between 0 and 1, the Sigmoid activation function is commonly used in the output layer with The binary cross-entropy as loss function (Sigmoid + Binary cross entropy).
- Multi-class Classification: For multi-class classification problems where the output should be a probability distribution over multiple classes, the softmax activation function is commonly used in the output layer and The categorical cross-entropy as loss function (Softmax + Categorical cross entropy).
- Regression: The linear activation function is commonly used in the output layer for regression problems where the output should be a continuous value. The mean squared error (MSE) loss function is commonly used in this case. However, other loss functions such as mean absolute error (MAE) and mean squared logarithmic error (MSLE) can also be used depending on the nature of the problem (linear/Relu + MAE/MSE).

Overall, the choice of activation function and loss function is one of the most important parts of designing an artificial neural network and should be carefully considered based on the specific requirements of the problem. For this thesis, we developed a binary classifier, then the Sigmoid in the last layers + Binary cross entropy combination was used.

4.3 Neural Network's Learning Process:

In the previous section was described how choosing the activation function and loss function can set the NN to solve a given problem, but it was not present how the NN

learns how to solve it. Training an Artificial Neural Network (ANN) involves iterative updating of the weights and biases of the network to minimize the loss function, which represents the difference between the predicted output and the actual output. This process is accomplished using a technique known as backpropagation, which also involves choosing an optimizer.

The training process can be broken down into four main steps, which are illustrated in Figure 4.4:

1. **Forward propagation (Forward pass):** Starts by randomly initializing the weights and biases of the network, then During the forward pass, the input data is fed through the network, and the output is computed using the current weights and biases.
2. **Loss computation:** The difference between the predicted output and the actual output is computed using a loss function. This value is also known as the error of the model. Then the loss score is used as a feedback signal to update parameter in the backpropagation step
3. **Backward propagation (Backward pass/Backpropagation):** Backpropagation is a key algorithm for training neural networks, as it allows us to calculate the gradient of the loss function with respect to the weights of the network. This gradient is then used to update the weights using an optimization algorithm, such as SGD, Adam or RMSProp, in order to minimize the loss function (error) and improve the network's predictions. This error is then propagated **back** through the network in reverse order, with the gradient of the loss with respect to each weight being calculated at each layer. The calculation of the gradient is done using the chain rule of calculus. The derivative of the loss function is its slope, which provides the direction that is considered for updating (changing) the values of the weights and biases.
4. **Repeat:** The input data is fed again to the NN, but now the output prediction is calculated based on the new weights. This process continues until a minimum in the gradient is found, therefore until the training converges, and the performance of the neural network has reached a plateau and further training will not improve the performance significantly. In practice, a stopping criterion, such as a maximum number of epochs or a minimum change in the loss function, is used to terminate the training process.

Instead of updating the weights after every input is passed through the network, we typically update the weights after a batch of inputs has been processed. This is because updating the weights after every input can be computationally expensive and can lead to overfitting. The size of the batch (also known as Batch Size) is a hyperparameter that

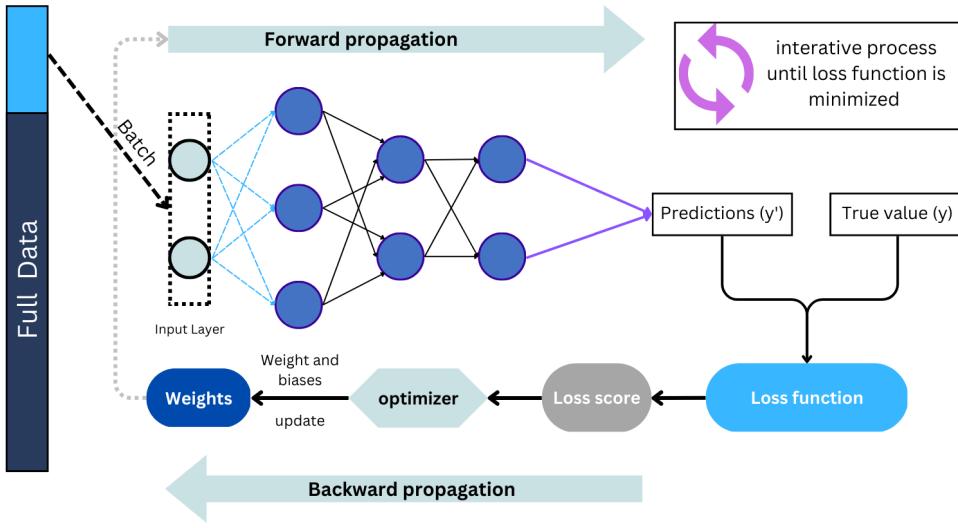


Figure 4.4 – A schematic of the iterative training process of a Neural Network

needs to be chosen carefully, as smaller batches can lead to faster convergence but can also make the optimization process more noisy.

Another important parameter in the training process is learning rate (denoted as α), which determines the step size of the weight (w_i) and bias (b_i) updates. A high learning rate can lead to unstable training (Figure 4.5), while a low learning rate can lead to slow convergence (Figure 4.5). The simplest case of the learning rate usage is on the gradient descent optimization (GD) algorithm,

$$w_i = w_i - \alpha \frac{\partial Loss}{\partial w_i} \quad (4.1)$$

$$b_i = b_i - \alpha \frac{\partial Loss}{\partial b_i}. \quad (4.2)$$

Where w_i denotes the i 'th weight, b_i denotes the i 'th weight bias, α is the learning rate parameter that multiplies the gradient term. A learning rate of 0.01 means that on each training iteration, the weight and bias parameters are updated only 1% of the gradient term. In summary, The hyperparameters of batch size, learning rate, and choice of optimizer play a crucial role in the effectiveness and training performance of the backpropagation algorithm.

An optimizer is a function or an algorithm that updates the weights, biases and learning rates of the neural network. The problem of choosing the right weights for the model is a complicated task, as a deep learning model generally consists of millions of parameters. It's necessary to choose carefully the optimization algorithms that best converges the loss to the global minima. Some of the most frequent optimizers are:

- **Gradient Descent (GD)** [38] is the simplest optimization algorithm used to up-

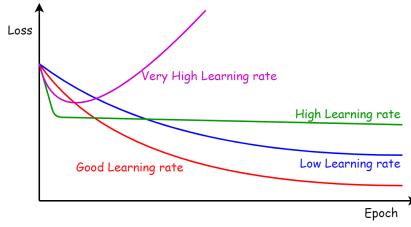


Figure 4.5 – Illustrative figure of Loss function convergence for different learning rates values

date the weights of a network by calculating the negative gradient of the loss function with respect to each weight. In this algorithm, the entire training is used to compute the gradient of the loss function, leading to a slow convergence.

$$w_i^{t+1} = w_i^t - \alpha \cdot g_t, \quad (4.3)$$

where $g_t = \nabla_i L(w_i^t)$, that is, the gradient of the loss function with respect to the i 'th weight on step t , and α is the learning rate.

- **Stochastic Gradient Descent (SGD)** [38] is a popular variation of Gradient descent optimization algorithm. Like GD, it updates the weights using the negative gradient of the loss function for each weight. However, instead of using the entire training dataset, SGD randomly selects a single training sample, computes the gradient based on that example, and performs a parameter update. While SGD can be faster for large datasets, it may have high variance due to the noisy updates, resulting in slower convergence.
- **Batch Gradient Descent (Batch GD)** [38] is another variant of GD where the training dataset is divided into batches. The gradients are computed for each batch, and the model parameters are updated based on the average gradient of all the batches. In other words, Batch GD performs a single update per iteration by considering all the training examples within each batch.
- **Mini-Batch Gradient Descent (Mini-Batch GD)** [38] is a combination of Batch GD and SGD. Instead of calculating the gradient using a single randomly selected sample (like SGD) or the entire training dataset (like GD), Mini-Batch GD computes the gradient based on a small randomly chosen subset of training data, also known as a minibatch. The model parameters are then updated based on the average gradient of the current minibatch. Mini-Batch GD combines the advantages of Batch GD (more stable convergence) and SGD (faster updates).
- **SGD with momentum** [38] is an algorithm that improves the basic GD algorithm by using information from previous gradients to accelerate convergence towards the relevant direction and reduce fluctuations in irrelevant directions. This is achieved

by introducing a new variable called the momentum vector (m_t), which is subtracted from the local gradient term. This momentum term (also known as mean or First momentum) represents the average of the previous gradients. With the objective of simulating some kind of friction and stop the momentum being too high, the algorithm introduces a new hyperparameter β which is defined between 0 (high friction) and 1 (No friction). Where the weight (w_i^{t+1}) is updated by subtracting the learning rate parameter (α) multiplied by a momentum term (m_t), which depends on the previous first momentum (m_{t-1}) and the gradient of the loss function with respect to the i'th weight on step t (g_t).

$$w_i^{t+1} = w_i^t - \alpha \cdot m_t \quad (4.4)$$

$$m_t = \beta \cdot m_{t-1} + (1 - \beta) g_t \quad (4.5)$$

Where the initializing value of m_t is zero.

- **Adaptive Moment Estimation (ADAM)** [39] is an optimizer that computes adaptive learning rates for each parameter in the network, unlike SGD that maintains a single learning rate through training, ADAM optimizer updates the learning rate for each network weight individually. This optimizer was demonstrated to be computationally efficient, has little memory requirements, and is well suited for problems that are large in terms of data and/or parameters. The algorithm updates exponential moving averages of the gradient (m_t) and the squared gradient (v_t) where the hyper-parameters β and β_2 control the exponential decay rates of these moving averages, respectively. The moving averages themselves are estimates of the first momentum (the mean) and the second momentum (the uncentered variance) of the gradients. However, the initializing value of m_t and v_t are, by default, zeros, thus leading to moment estimates that are biased towards zero, especially during the initial epochs, and especially when the decay rates are small (i.e. the β 's are close to 1). Therefore, motivated by this initialization problem, Kingma e.t.al (2014) [39] introduces a bias-corrected estimates m'_t and v'_t . By maintaining an adaptive learning rate based on the second moment estimate, ADAM can converge faster than SGD with momentum, and handle sparse gradients encountered in many machine learning tasks. The weight update is given by:

$$w_i^{t+1} = w_i^t - \alpha \cdot \frac{m'_t}{v'_t} \quad (4.6)$$

$$m'_t = \frac{m_t}{1 - \beta^t} \quad (4.7)$$

$$v'_t = \frac{v_t}{1 - \beta_2^t} \quad (4.8)$$

m'_t and v'_t are bias-corrections to the first and second momentum, which are opti-

mized based on the gradient term.

$$m_t = \beta \cdot m_{t-1} + (1 - \beta)g_t \quad (4.9)$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2)g_t^2 \quad (4.10)$$

Accordingly to Kingma e.tal (2014) [39], ADAM was tested in several different ML's problems leading to the suggested values of $\beta = 0.9$, $\beta_2 = 0.999$ and $\alpha = 0.001$, where β and β_2 are the Exponential decay rates for the moment estimates and α is the learning rate.

4.4 Caveats on training an Artificial Neural Network

Identify transients using AI is a complex task that requires a deep neural network with multiple layers, which in turn involves numerous derivatives. Training a Deep Neural Network can lead to several related challenges, in this section, we will describe each problem and possible solutions.

4.4.1 Exploding gradient and vanishing gradient

There are two main problems that may arise during training that are related to weight derivatives. In a network of n hidden layers, n derivatives will be multiplied together. If the derivatives are large, then the gradient will increase fast as it propagates down the model, resulting in the problem of the exploding gradient. Alternatively, if the derivatives are small, then the gradient will decrease fast as it propagates through the model, resulting in the vanishing gradient problem [38].

In the case of exploding gradients, the accumulation of large derivatives leads to a very unstable model that cannot learn effectively. The large changes in the model's weights make the network highly unstable, and extreme weight values may cause overflow, resulting in NaN (Not a number) weight values that can no longer be updated. On the other hand, the accumulation of small gradients leads to a model that is incapable of learning meaningfully. The weights and biases of the initial layers, which tend to learn the core features from the input data (X), if the gradient reaches zero, this would prevent the network from further training.

To better understand these gradient problems, consider a deep neural network with L layers. The output of the i -th layer is denoted as h_i , and the weights of the i -th layer are denoted as W_i . The activation function of the i -th layer is denoted as f_i and f'_i its derivative. The output of the Neural Network is denoted as y . To update W_i it's necessary

to compute its gradient regarding the loss function.

$$\frac{\partial \text{Loss}}{\partial w_i} = \frac{\partial \text{Loss}}{\partial h_i} \cdot \frac{\partial h_i}{\partial w_i}, \quad (4.11)$$

$$h_i = f(W_i * h_{i-1} + b_i) = f(z_i), \quad (4.12)$$

$$h_{i-1} = f(W_{i-1} * h_{i-2} + b_{i-1}) = f(z_{i-1}), \quad (4.13)$$

W_i and b_i are the weight matrix and bias vector of the i -th layer, respectively, and h_{i-1} is the output of the $(i-1)$ -th layer. Using the chain rule, we can rewrite this as:

$$\frac{\partial \text{Loss}}{\partial w_i} = \frac{\partial \text{Loss}}{\partial h_i} \cdot \frac{\partial h_i}{\partial z_i} \cdot \frac{\partial z_i}{\partial w_i}, \quad (4.14)$$

$$\frac{\partial \text{Loss}}{\partial w_i} = \frac{\partial \text{Loss}}{\partial h_i} \cdot \left(\frac{\partial h_i}{\partial h_{i-1}} \right) \cdot \left(\frac{\partial h_{i-1}}{\partial h_{i-2}} \right) \cdot \dots \cdot \left(\frac{\partial h_2}{\partial h_1} \right) \cdot \left(\frac{\partial h_1}{\partial w_1} \right). \quad (4.15)$$

As we can see, the gradient is a product of the gradients of the activation functions of each layer. If the activation function has a derivative that is ~ 0 , then the gradient will become very small as it is multiplied by each subsequent layer, leading to the problem of vanishing gradients. On the other hand, if the activation function has a derivative with high values, then the gradient will increase until it explodes to very high values.

One solution to the vanish gradient problem is to use normalization techniques, such as batch normalization [40], which standardizes the output of individual layers (also called as activations) for each mini-batch, by subtracting the mini-batch mean and dividing by the mini-batch standard deviation. This process ensures that the mean activation value is close to zero and the standard deviation is close to one, leading to stable training, as stated by Ioffe and Szegedy [40]. Batch normalization introduces two learnable parameters per activation channel, a scale parameter and a shift parameter, after normalizing the activations, these parameters allow the network to learn the optimal scale and shift for each activation channel.

The problem of exploding gradients can be addressed by using techniques such as gradient clipping, which is a method where the error derivative is changed or clipped to a threshold during backward propagation through the network, and the clipped gradients are used to update the weights [38]. The gradient is limited by a hyperparameter c , if the gradient value is larger or equal to c then the gradient vector is rescaled,

$$\mathbf{g} = c \cdot \mathbf{g} / \|g\|. \quad (4.16)$$

Where \mathbf{g} is the gradient vector, and $\|g\|$ is the norm of \mathbf{g} . Since $g / \|g\|$ is a unit vector, after rescaling, it ensures the gradient vector (\mathbf{g}) has norm at most c . This avoids the gradient to explode, note that if $\|g\| < c$, then the gradient maintains his value.

4.4.2 Overfitting and Underfitting

Overfitting and underfitting are two of the most common problems in deep learning. Overfitting occurs when the model is trained excessively on the training data, causing

it to learn irrelevant and specific patterns from the training dataset. As a result, the model performs very well on the training data but poorly on the test data because it could not generalize the learned patterns to unseen data. One indication of overfitting is when the training error value is much lower than the Validation error, indicating that the model is losing the capacity to generalize to the validation set. This is often caused when there are too many model parameters, heterogeneous or unbalanced training data, or when the model is trained for too many epochs. Underfitting occurs when the model is too simple to capture the underlying patterns or features in the data. In this case, the model may perform poorly on both the training and test data since it is unable to capture the complexity of the data. Underfitting is often caused by too few model parameters, insufficient training data, or training the model for too few epochs. The most common techniques to solve the underfitting problem is to increase the model complexity, by adding more layers, neurons, or using a more powerful architecture.

There are a few solutions to address the overfitting problem:

- L1 regularization, also known as Lasso regularization. L1 regularization adds a penalty term to the Loss function that is proportional to the absolute value of the model's weight coefficients. The penalty term aims to reduce the magnitude of the model's weights, effectively eliminating the impact of irrelevant or noisy features.

$$\text{Loss}_{\text{Lasso}} = \text{Loss} + \lambda \sum_{j=1}^N |W_j| \quad (4.17)$$

Where λ is the penalty term, and $|W_j|$ is the absolute value of the model's weight coefficients. As we increase the value of λ , the minimization process causes the value of the coefficient to tend towards zero.

- L2 regularization, also known as Ridge regularization. L2 regularization adds a penalty term to the cost function that is proportional to the squared magnitude of the model's weight coefficients. The penalty term encourages the model to reduce the magnitude of its weights, but unlike L1 regularization, it does not typically result in weights becoming exactly zero. Instead, it shrinks the weights towards zero, effectively reducing their impact on the model's predictions.

$$\text{Loss}_{\text{Ridge}} = \text{Loss} + \lambda \sum_{j=1}^N (W_j)^2 \quad (4.18)$$

Where λ is the penalty term, and $(W_j)^2$ is the squared magnitude of the model's weight coefficients.

- Dropout: A dropout layer randomly drops out some of the neurons in the model during training.

4.5 Convolutional Neural Network

Convolutional Neural Networks (CNNs) [38] are a type of deep neural network (DNN). It has demonstrated enormous potential in image analysis including object detection, recognition and classification. Examples of astrophysics application of CNN include N. M. Cardoso (2021) [41] for galaxy morphology classification, Kim and Brunner (2016) [42] for star-galaxy classification, Bernardo M.O Fraga (2021) [43] for Blazar classification based on multi-frequency spectral energy distribution data and many more. The main difference between a CNN and an ordinary NN is that the CNN was specifically developed to process images as input. The neurons used in a CNN are arranged in three dimensions: width, height, and depth. It is important to note that the term "depth" here refers to the third dimension of an activation volume, rather than the total number of layers in a neural network.

The idea behind using a CNN is based on the principle of convolutions, which is a mathematical operation used for signal processing [44]. By applying convolutional filters to the input image, the CNN is able to extract features such as edges, corners, and textures. These feature extractions are based on the three main types of layers used to build a ConvNet:

- Convolutional Layer: Filters an image for a particular feature and, by using a ReLU activation, can detect that feature within the filtered image.
- Pooling Layer: Condenses the image to enhance the features.
- Fully-Connected Layer: Learns how these features can be used to solve the determined problem, discussed previously in Section 4.1.

4.5.1 Convolutional Layer

The convolutional layer [45] is the core building block of the CNN. It carries the main portion of the network's computational load and filters and detects features from an input image. In the Convolutional Layer, the input image is first convolved with a set of filters, also known as kernels, designed to detect specific patterns in the image. The filter slides over the input image, and at each position, it computes the dot product between the filter and the corresponding region of the input image. The final output from the series of dots products is known as a feature map or convolved feature.

The filter is typically a small matrix, such as a 3×3 or 5×5 matrix, and the number of filters used in the Convolutional Layer is a hyperparameter that can be adjusted during the training process. Each filter in the Convolutional Layer learns to detect a different pattern in the input image, such as edges, corners, or textures.

To illustrate this convolution operation, let's assume we have an input image of size $51 \times 51 \times 3$ (where 51 is the width and height of the image, and 3 is the number of color

channels: red, green, and blue). We also have a Convolutional Layer with a filter size of 5×5 , and let's say we are using a single filter. To apply the filter to the input image, we slide the filter over the image one pixel at a time, computing the dot product between the filter and the corresponding 5×5 region of the input image at each position. We then store the result of each dot product in a new matrix, which represents the output of the Convolutional Layer. Some of the most relevant parameters to define when using a convolutional neural network are:

- Number of filters: The number of filters in a Convolutional Layer is a hyperparameter that determines the number of patterns the layer can convolve from the input volume. Increasing the number of filters increases the capacity of the network to learn complex features, but also increases the number of parameters in the network, which can lead to overfitting. Therefore, the number of filters should be chosen based on the complexity of the task and the available computational resources.
- Stride: The stride is the filter's step size when sliding over the input image (in our example, the stride was set to one). A larger stride reduces the spatial size of the output feature map, which can be useful for reducing the computational cost of the network. However, a larger stride also reduces the spatial resolution of the output feature map and can lead to a loss of important features. A smaller stride can preserve more details in the feature map, but increases the computational cost of the network.
- Padding: Padding refers to the number of zeros added to the border of the input image to ensure that the filter can slide over the entire image. Without padding, the output feature map will be smaller than the input image, which can lead to a loss of information at the borders. There are two common choices of padding: same padding and valid padding. The same padding adds zeros to the border of the input image so that the output feature map has the same spatial size as the input image. Valid padding, on the other hand, does not add any padding, and the output feature map is smaller than the input tensor.

Here's an example of how this would work for the first few positions of the filter on the input image. At position $(0,0)$, we compute the dot product between the filter and the 5×5 region of the input image starting at $(0,0)$, which gives us a single output value. We then move the filter one pixel to the right and compute the dot product between the filter and the 5×5 region of the input image starting at $(1,0)$, which gives us another output value. We continue sliding the filter one pixel at a time until we reach the end of the first row of the input image, then the filter moves down one row and repeat the process until we have computed the dot product for every position of the filter on the input image.

After we have applied the n filters to the input image, we obtain a new matrix with dimensions $51 \times 51 \times n$ per channel, assuming a same padding.

This is because each position of the filter produces a single output value, and we slide the filter over the input image until we reach the edges. The resulting matrix represents the output of the Convolutional Layer for this single filter. The output of the Convolutional Layer is often passed through an activation function, such as the ReLU function, to introduce non-linearity into the model.

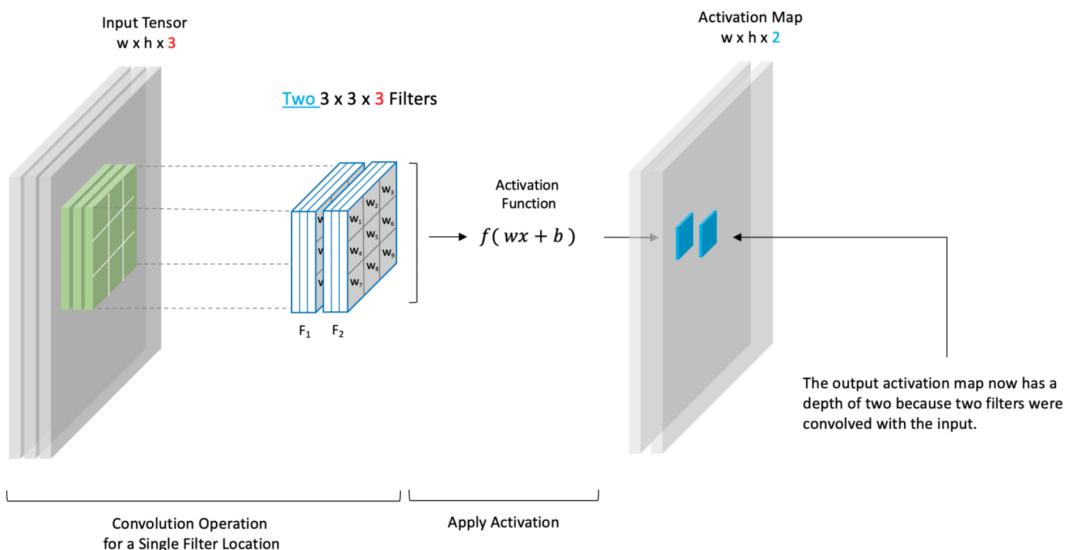


Figure 4.6 – Example of 2D convolutional operation on a RGB input image, it demonstrates how the convolutional operation for two $3 \times 3 \times 3$ filters act on an Input tensor of shape $(w, h, 3)$

Figure 4.6 demonstrates how the convolutional operation for two $3 \times 3 \times 3$ filters act on an Input image (tensor). The input tensor of shape $(w, h, 3)$ is convolved with two kernels, the filter slides over the input image, and at each position, it computes the dot product between the filter and the corresponding region of the input image producing two values on the feature map, one value for each filter that are stacked.

One advantage of using these layers is that there is no need to pre-process the image to identify patterns that can help the model, as CNN can do this automatically. The weights connecting each of these features determine the most important features for solving the presented task.

4.5.2 Pooling Layer

The Convolutional Layer is often followed by a Pooling Layer, which is used to reduce the dimensionality of the feature maps, thus condensing the filtered (detected) information and helping to make the model more robust to variations in the input image.

The most common type of pooling is max pooling, which takes the maximum value within a small region of the feature map and replaces the entire region with that maximum value. This reduces the spatial size of the feature map while preserving the most important features.

The other common type of pooling is average pooling, which instead of taking the maximum value as the max pooling, it takes the average value within a small region of the feature map and replaces the entire region with that mean value.

Global pooling is a pooling layer that operates globally over the entire input feature map instead of using a local window like max pooling or average pooling. Global pooling can be either Global max pooling or Global average pooling. Global max pooling selects the maximum value across the entire feature map, while global average pooling takes the average value across the entire feature map. Global pooling is effective in reducing the dimensionality of the feature map while retaining the most important features.

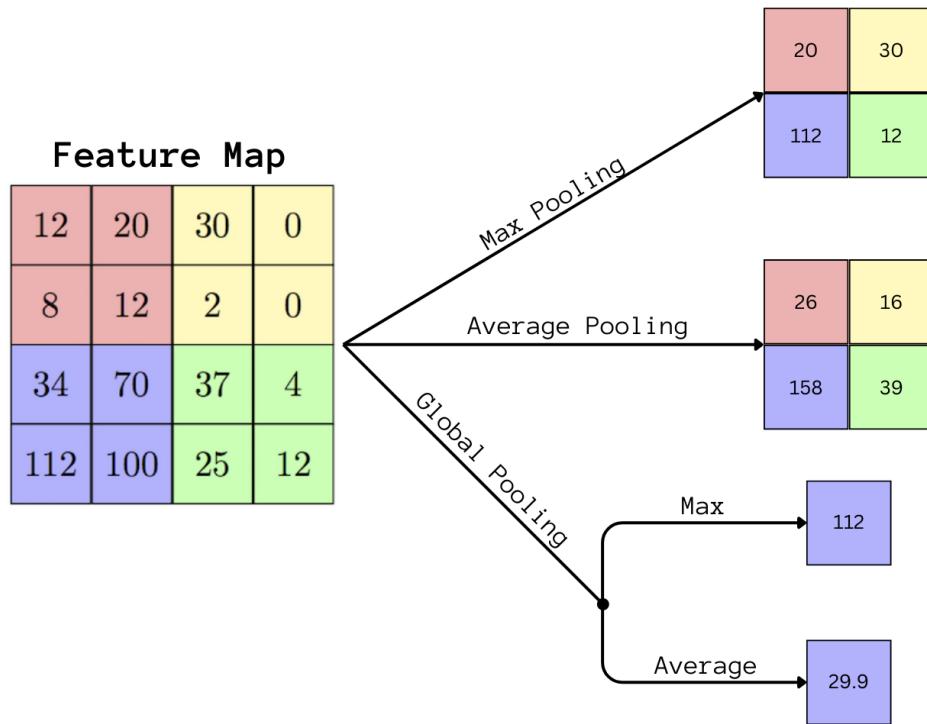


Figure 4.7 – Example of different pooling layers acting on a 4×4 feature map.

Figure 4.7 demonstrates an example of a max pooling, an average pooling, a Global max pooling and global average pooling layer acting on a 4×4 feature map, resulting in a condensed matrix. As described above, max pooling takes the maximum value within a region, while average pooling takes the average value.

For a better understanding of these layers, Figure 4.8 illustrates how an image changes inside a simple convolutional block. An image of Jupiter from the James Webb Telescope

exemplifies the process of applying a convolutional filter, then an activation function is applied to detect the extracted features, followed by a pooling layer that condenses the information. This figure demonstrates how three different kernels acting on the input image can lead to different features being extracted in a convolutional neural network. This is a simple example of how a CNN works, as there are several different types of CNNs with different features and purposes.

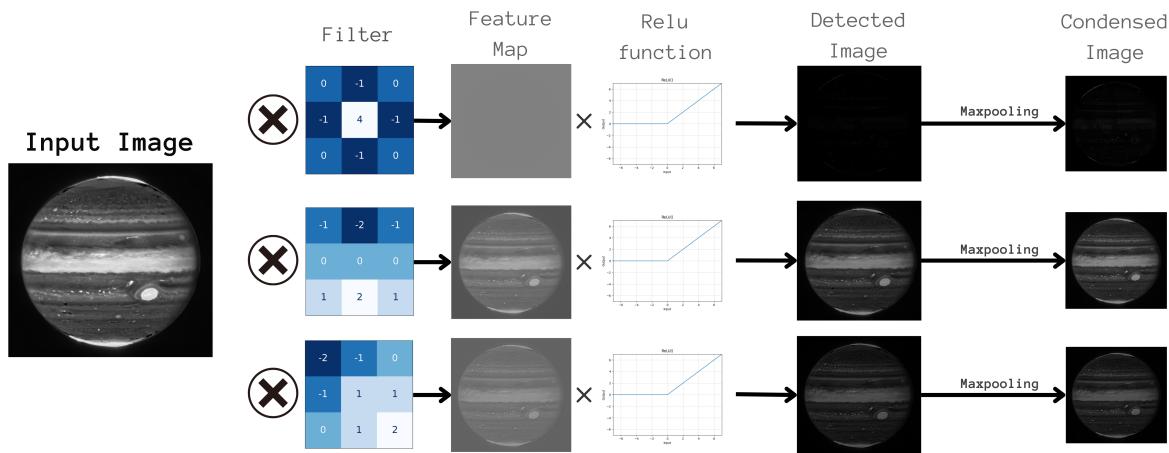


Figure 4.8 – This figure shows a simple example of how a CNN works. It uses an image of Jupiter to demonstrate the process of applying a convolutional filter, detecting features with an activation function, and condensing information with a pooling layer. Credits: NASA

4.6 Metrics for Evaluation model's performance

In deep learning, after a training process, it is essential to validate a trained model's performance using various metrics to ensure its application in real-world scenarios. Typically, the dataset is divided into three subsets: training, validation, and test. The training set is actively used in the training process, as described in section 4.3. The validation set is used to tune the model's hyperparameters and prevent overfitting (section 4.4.2). Overfitting occurs when a model performs well on the training data but poorly on the validation data. That's why a test set composed of unseen data is used to evaluate the model's performance. The test dataset is used to define the model's accuracy, recall, and other metrics, ensuring that the model is ready for scientific usage.

This thesis focused on a binary classification problem, where the main objective is to use images as input and the model outputs a probability of this image belonging to a specific class. In this section, we will describe the most common metrics used in this type of problem.

4.6.1 Loss Function

One way to evaluate the predictive capacity of a model as a classifier during training is by using a Loss function. A commonly used error function for binary classifiers is the Binary Cross-Entropy (BCE) loss function, as described in Chapter 4.2. This loss function measures how well the model predicts the correct label. The BCE loss function compares the output probability for a given image with its true label (either 1 or 0), and penalizes the model for predictions that deviate from the true value, which can be also used as a metric to ensure that the model does not overfit by monitoring the loss function value on a validation set.

If the model's error function on the validation set is significantly higher than its error on the training set, it may indicate that the model is overfitting the training data. While the loss function is a useful metric for evaluating the model's performance, it is insufficient if used alone because it only provides information about the overall performance of the model and not about the prediction of individual classes. Therefore, it is essential to use additional metrics such as accuracy, precision, recall, and F1-score to evaluate the model's performance on each class separately.

4.6.2 Confusion Matrix (CM)

The confusion matrix (CM) provides visual information about the model's predictions and the true labels for each class, facilitating a more detailed analysis of the model's performance. The confusion matrix is a matrix that displays the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for each class. The rows represent the predicted labels, the columns represent the true labels (Figure 4.9), and the matrix elements are the number of instances that fall into each category.

- True positives (TP): The actual value was positive, and the Deep learning model predicted a positive value
- True Negative (TN): The actual value was negative, and the Deep learning model predicted a negative value
- False Positive (FP): The actual value was negative, but the deep learning model predicted a positive value
- False Negative (FN): The actual value was positive, but the deep learning model predicted a negative value

Using the values in the confusion matrix, we can calculate several useful metrics, including accuracy, precision, recall, and F1-score. These metrics provide insightful information about the model's ability to identify transients and artifacts.

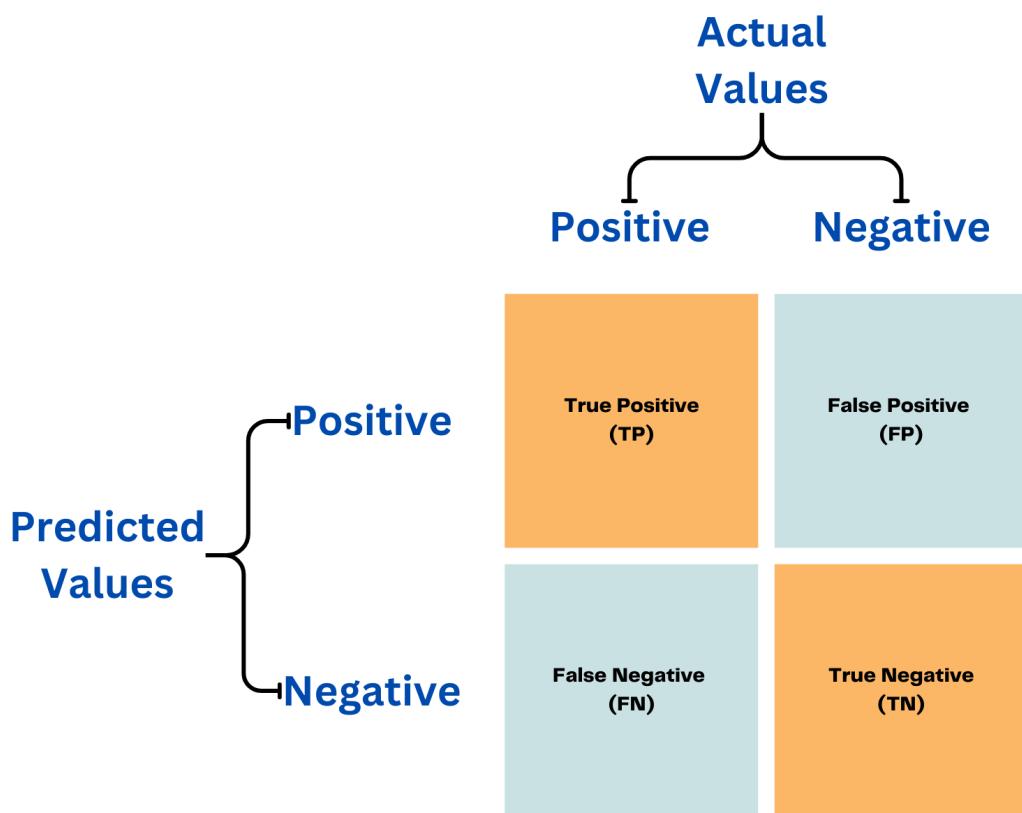


Figure 4.9 – Confusion Matrix, the rows represent the predicted labels and the columns represent the actual label

- **Accuracy:** The percentage of the total predictions that were correct. It is calculated as the number of true positives plus true negatives divided by the total number of predictions, t defines how often the model predicts the correct output. For example, if the model correctly predicts 95 instances out of 100, then the accuracy is 95%.
- **Precision:** : The proportion of the total positive predictions were correct. It provides a metric to calculate the model's ability to classify positive samples. It is calculated as the number of true positives divided by the sum of true positives and false positives. For example, if the model predicts that 100 instances are positive, out of which 90 are actually positive (true positive), then the precision is 90%.
- **Recall (sensitivity or True positive rate):** The proportion of the total actual positive values were correctly classified. It measures the model's ability to detect positive instances. Recall is calculated as the number of true positives divided by the sum of true positives and false negatives. For example, if there are 100 positive instances in the dataset and the model correctly identifies 90 of them, then the recall is 90%.

- **False Positive Rate (FPR):** The fraction of false positive samples to all actual negative samples. It tells us how often the model incorrectly predicts a negative instance as positive. For example, if there are 100 negative instances in the dataset, and the model predicts that 10 of them are positive, then the FPR is 10%.
- **F1-score:** is the harmonic mean of precision and recall, and captures the contribution of both of them in a single score. This means that if a model has high precision but low recall, or vice versa, the F1 score will be low. The higher the F1-score, the better the model. The F1-score equation is:

4.6.3 Receiver Operating Characteristics (ROC) curve

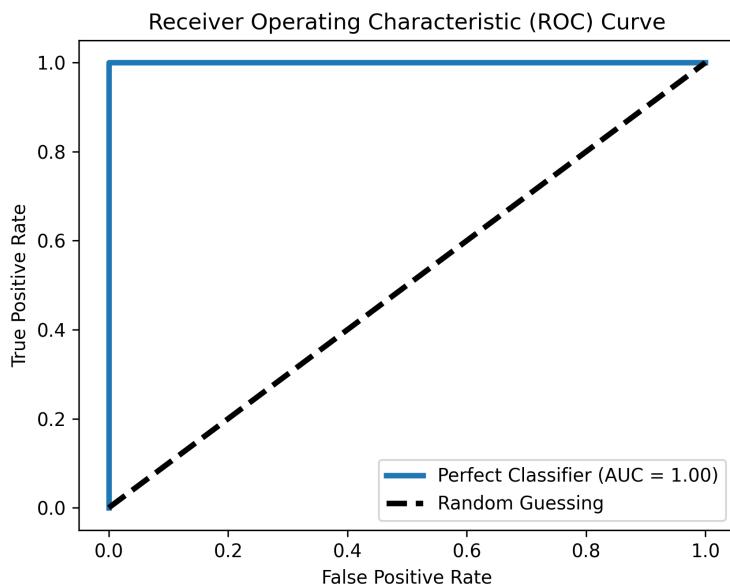


Figure 4.10 – ROC curve, exemplifying a perfect classifier (blue solid line) and a random classifier (dashed line)

A Receiver Operating Characteristics (ROC) curve is a plot of the true positive rate with respect to the false positive rate at different classification thresholds for the output probability. It shows the trade-off between the TPR and FPR of the model as the threshold is varied. If our model presents a random guess to separate artifacts and transients, the line would be a diagonal line, and changing the threshold would have no impact on the curve, as the model would essentially be guessing every image. On the other hand, if the model is a perfect classifier, the TPR would be 1 and the FPR zero for all different thresholds, meaning that the model would find every positive case.

Another important metric that measures the overall performance of a classifier is the area under the ROC curve (AUC) or AUROC value. As the name suggests, it is simply the area measured under the ROC curve. A higher value of AUC represents a better classifier, i.e., If the AUC is 1, the model has perfect discrimination capacity, while an

AUC of 0.5 indicates that the model is guessing the images. The AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance [46].

4.6.4 Precision-recall (PR) curve

Similar to the ROC curve, the precision-recall (PR) curve is another graphical representation of the performance of a binary classification model. It is a plot of the "precision" with respect to the "recall" at different classification thresholds. It shows the trade-off between precision and recall of the model as the threshold is varied. Ideally, a perfect classifier would have a PR curve that passes through the plot's top right corner, representing a precision of 1.0 and a recall of 1.0. The closer the PR curve is to the top right corner is a proxy of the model's quality (Figure 4.11).

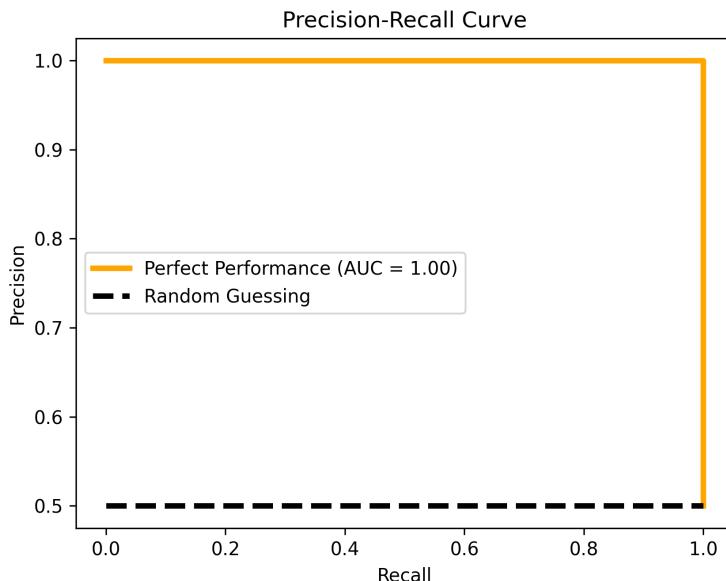


Figure 4.11 – Precision Recall curve, exemplifying a perfect classifier (yellow solid line) and a random classifier (black dashed line)

The area under the precision-recall curve (AUCPRC) is also a metric used to evaluate model performance. It summarizes the plot in a single number, where an AUPRC closer to 1 indicates better performance.

Using a confusion matrix only to evaluate the performance of a model is not the best approach. This is because when a CM is built, a threshold t is defined, which is usually set to 0.5. However, better results can be obtained with other thresholds. That's why the ROC and PR curves are important for analysis. They compare classifiers for several or all possible thresholds, and then we can choose the best threshold that fits our objective.

 CHAPTER 5

AUTOMATED TRANSIENT DETECTION

5.1 Automated Transient detection in Dark energy survey-GW (DES-GW) search and discovery pipeline

Goldstein et al.(2015) [29], created an automated transient classifier based on a Random Forest (RF) supervised learning model [47] to detect transients, and particularly supernovae, in the DES data, named Autoscan. The algorithm is based on 38 features derived from the diff, search and template images. The selection and computation of these features is done in attempting to represent quantitatively what astronomers would leverage in visual inspections. For instance, the feature *diffsum* measures the significance of the detection by summing the pixel values in the center of the diff image; the feature *colmeds*, indicating the CCD used for the detection, is designed to identify artifacts specific to a CCD, like bad rows/columns of pixels; the feature Signal-to-noise ratio, could be used to find noise stamps.

A list of possible transients candidates generated after the DES-GW difference imaging analyses is contaminated with false positives samples. Autoscan was trained on simulated data to remove exclusively bad subtraction samples from the Dark Energy Survey dataset. Despite having a great accuracy, the vast amount of data we expect on O4 season also increases the number of artifacts present, thus, less common types of artifacts that are not removed by Autoscan will become a major problem when visually inspecting these potential transients. Besides that, Autoscan uses hand-engineered features, which are computationally expensive when dealing with a large portion of data.

Convolutional Neural Networks are the natural step to work in parallel with Autoscan and eventually replace it. The use of CNNs in the DES-GW pipeline was first proposed by Shadonay et al. [48], and their work serves as a direct precursor to our own. We share the same set of images with their work; however, we trained our model only on real data. Therefore, our main differences lie in the training set (we only use real follow-up

data), the convolutional neural network architecture (we developed a new and optimized architecture), and, consequently, the results.

Our approach to this problem consists of an algorithm with two CNNs trained on real DECam difference imaging data, along with image processing routines, to eliminate false positives from consideration as transients. The input consists of 51 x 51 images composed of three channels: search (srch), template (tmpl), and difference image (diff), analogous to RGB channels. The resulting output of our algorithm is a score for each image, ranging from 0 to 1, representing the probability of being a transient with a host galaxy. It's worth noting that we specifically look for the presence of a host galaxy in the template image and a new transient in the search image. This choice is made to resemble a real multi-messenger follow-up campaign

5.1.1 DES-GW Data collection

A deep-learning approach requires a large, diverse set of images with accurate labels. The utilized set images are randomly sampled objects detected from applying the DES-GW Search and Discovery Pipeline to DES wide-field data. A team of six experts labeled the images corresponding to the five image types [48]: “host + transient”, “no obvious transient”, “bad subtraction”, “preexisting point source” and “other artifact”.

Additionally, two main extra sources of transients + host galaxy were used. First, we supplemented the data set with a population of the DES wide-field difference imaging data that was given an Autoscan score of at least 0.9. Second, we incorporated transient + host objects identified by visual inspection during the DES-GW follow-up observation of GW190814 and GW200224. In the labeling step, it rejects images where the detected transient could be matched to a high-confidence star in the DES Data Release 1 [23] or GAIA Data Release 2 [49] since the DES-GW pipeline routinely performs these steps. It also rejects images whose detected transient had any masking over the transient, since the flux would be measured inaccurately.

The total collection of images is composed of (??):

Label	Nº samples
Host + Transient	638
Preexisting point source	1050
Non-obvious transient	921
Other artifact	1037
Bad subtraction	9436

Table 1 – DES-GW dataset divided by subclass, where the two main classes are highlighted.

In order to facilitate the classification process, Three preprocessing steps were applied before passing these images to a CNN, two of them were designed to remove low-quality

or noisy images.

1. The first step performs a PSF flux estimation at the center of the search image by subtracting the median value of the images, corresponding roughly to the sky background, from each pixel and weights the result by a Gaussian realization of the PSF from each image, then every image below an empirically determinate threshold were removed.
2. Using SEXtractor [15], we were able to use the flux and flux error of the detection in the difference image and remove images with a signal-to-noise rate below another empirically determined threshold. Both of these thresholds were determined by finding the strictest cut that would allow 99% completeness of all classes not labeled other artifacts in the dataset. Both of these steps aim to eliminate most of the noise detections in the other artifact class, since clear image features are necessary for a CNN to learn. The choice of these threshold cuts is discussed in section 6.
3. A Pixel Normalization was employed by subtracting the maximum pixel value across each image and channel, followed by division by the range (i.e., the difference between the minimum and maximum values). The resulting values were then multiplied by 255 to re-scale the pixel values from 0 to 255.

Most of the remaining data examples are artifacts that are more difficult to remove with simple filters. A small portion of this remaining dataset was selected to compose the test set, which was used to quantify the performance of the first and the second Neural Network.

The images not used for testing were designated to the training/validation set, with the objective of increasing the number of data, a technique known as data augmentations. We applied a total of 8 operations, increasing the training dataset by a factor of eight. These augmentations included rotations and mirroring. All these transformations are shown below (Figure 5.1).

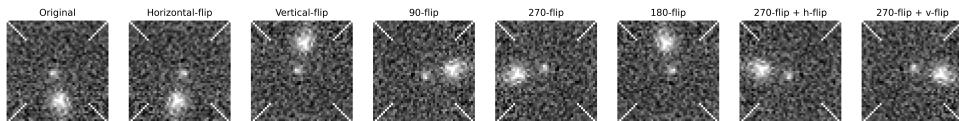


Figure 5.1 – Illustration of the Augmentation Technique, exemplifying various transformations such as 90°, 180°, and 270° rotations, vertical mirroring (vflip), horizontal mirroring (hflip) and combinations of 270° rotations with mirroring

This augmented training dataset was used to train the first CNN of our algorithm, with the goal of exclusively identifying bad subtraction compared to the other classes. Creating one CNN specialized in removing bad subtractions ensures the highest level of accuracy to the algorithm when removing all artifacts from the data set, as opposed to a

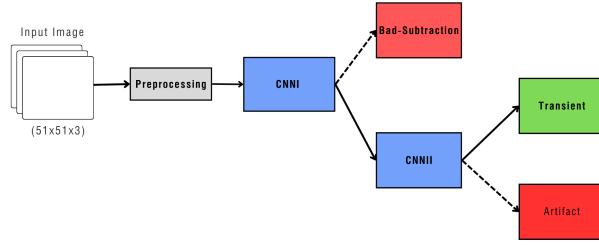


Figure 5.2 – A schematic overview of DESGW automated transient detection pipeline

multi-class classification scheme. That is mostly due to the fact that the bad subtraction is over-sampled, making it difficult to remove other less common artifacts when dealing with a multi-label problem.

Then a second network was trained to identify transients + host galaxy, in the same training images, but this time the bad subtraction class was omitted from the training process. This second CCN outputs a probability of a given image being a transient + host galaxy, so the full algorithm can be summarized as two sequential CNNs, one specialized in removing bad subtraction and the other on finding transients.

5.1.2 Convolutional Neural Network Design

The network architectures used for this work are shown in Figure 5.3, for the first CNN model and the second CNN model, respectively. Both architectures follow a similar structure. In designing the neural networks, we started with the first CNN and developed an architecture that would match the performance of Shadonay et al. (2021) [48]. First we tried to use the state-of-the-art architectures such as VGG [50], ResNet [51] and Efficientnet models [52]. However, these models presented an extreme overfitting, becoming clear that we should choose a shallow architecture with fewer parameters.

Taking that into consideration, we used a custom-built convolutional block as a pillar to our model. The convolutional (conv) block is composed of two convolutional layers, a max pooling layer, followed by a dropout layer. We stacked 4 of these blocks followed by a flatten layer and four fully connected dense layers. The neural network architecture has two main components: a feature extraction step where the convolutional block locates edges and shapes within the images, and a classification step where the fully connected layers weigh the extracted features and reduce them to classifications.

The convolutional layers of the neural networks can be modified by adjusting hyperparameters that determine how the CNN processes each image. In order to determinate the best combination of hyperparameters for our classification task, we made use of a framework called KerasTuner [53]. It is a scalable hyperparameter optimization framework for hyperparameter search, which uses Bayesian Optimization, that follows a probabilistic approach. This approach takes into account already tested combinations and uses this information to sample the next combination of hyperparameters for a test.

For this optimization, we selected 20% of the training set to be used as validation set. we trained our network architecture with different hyperparameters on the training data set and predicted the labels of the images in the validation data set. We perform a search of the number of filters, strides, and number of neurons. The best set of hyperparameters was selected based on the lowest validation loss.

Hyperparameters	Lower limit	Upper limit
Filter	16	512
Kernel size	2	7
Max pooling size	2	6
Number of units	32	512

Table 2 – Table with the list of Hyperparameters tuned with a lower limit, upper limit and step used to search for the best set of hyperparameter on the first and second CNN

The second Neural Network architecture was deliberately kept as close as possible to the first CNN, as we suppose that an architecture that would not overfit for the first classification task would also be good for the classification of transient + host, because we are dealing with similar patterns. However, we notice that the “not an obvious transient” class is very close to a transient, making it difficult for the second CNN to learn the underlying patterns, so we made some modification on the second CNN, building a more complex architecture. The hyperparametrization was also made similarly, using the KerasTuner framework, we were able to see different performance on the validation data for different parameters. We chose the best set of parameters to be that which has the highest true positive rate, meaning the model that correctly labels most of the transient + host samples.

Figure 5.3 illustrates the flow of information through the layers of the network, as well as the hyperparameter settings utilized in our analysis for the first and second CNN.

5.1.3 Training

The previously described models were trained using one of the state-of-art optimizers: the NAdam [54], with a binary-cross entropy loss. We initialize the networks with randomized weights. The model was trained in a Multi-GPU cluster with 8 RTX 3090 with 24 GB of GPU memory each. The model training was implemented in TensorFlow2 [55].

After optimizing the architecture of our network, we opted to perform a 5-fold cross-validation training, it helps in obtaining a more robust estimate of the model’s training performance. By splitting the dataset into 5 subsets (or folds) and training the model 5 times, each time using a different fold as the validation set, one gets multiple performance metrics that can be averaged. This approach provides a more reliable evaluation of the

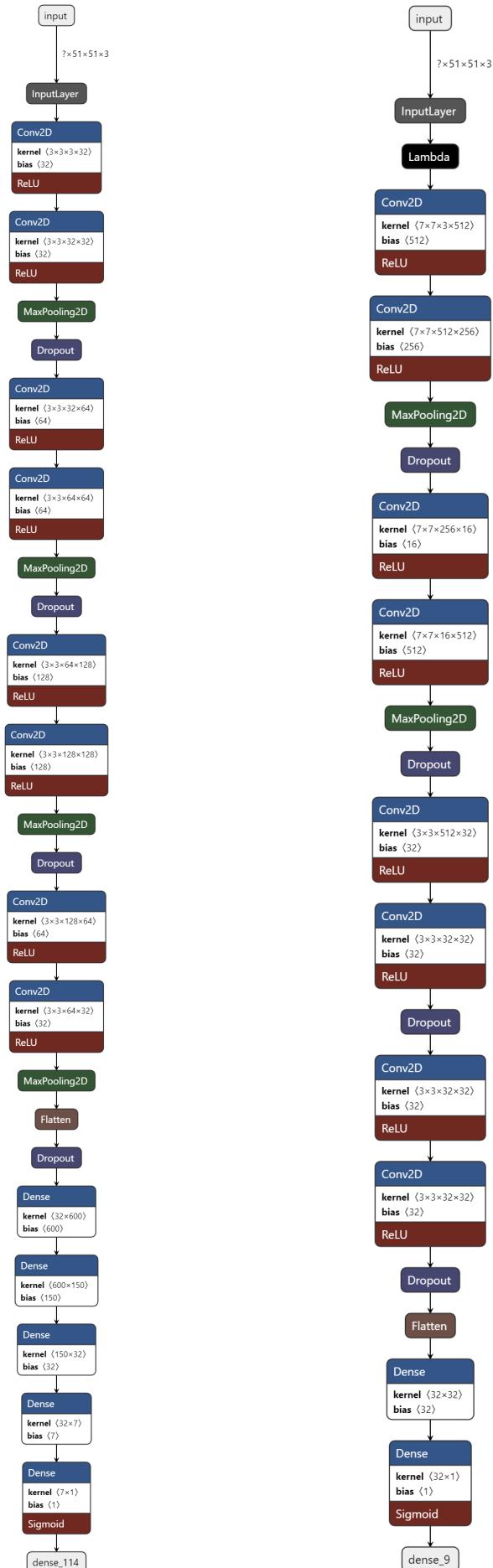


Figure 5.3 – Architectures and hyperparameters configuration for each CNN developed in our thesis for the DES-GW search and discovery pipeline

model’s ability to generalize unseen data. Each fold was trained for 40 epochs, and in each fold the set of weights with the lower validation loss was saved.

It must be pointed out, in the training set for the first CNN, the rate of positive samples and negative samples is close since all non-bad subtraction belongs to the same class (positive, labeled as 1), and the bad subtraction belongs to the negative class (labeled as 0). However, the second CNN addresses an unbalanced classification problem, where the positive class is only composed of transient + host galaxy samples, and the negative class is composed of all the remaining artifacts. Therefore, one additional parameter was used when training the second CNN, the ‘*class weights*’ parameter, which allows assigning different weights to different classes during the training process. By assigning higher weights to the positives classes (weight of 1.80) and lower weights to the negatives classes (weight of 0.75), the model gives more importance to the minority classes during the optimization process, increasing the model’s true positive rate.

The selection of the final model to be implemented in the DES-GW search and discovery pipeline was based on its accuracy in identifying the transient class on the test set, exceeding the performance of other models trained on different folds. This selection approach ensures that our model is capable of removing artifacts, and most important, it maximizes the model’s ability to capture and identify these multi-messenger events in real-world scenarios effectively. This selection rule was used in both CNN’s training.

Figures 5.4 and 5.6 show the performance of the networks during 5-fold training on just the training/validation dataset, while the Results Sections (6) will present the measured performance of the trained networks on the test data set. From Figure 5.5 and 5.7, we observe high training accuracies, but more importantly, we do not observe strong overfitting on both models.

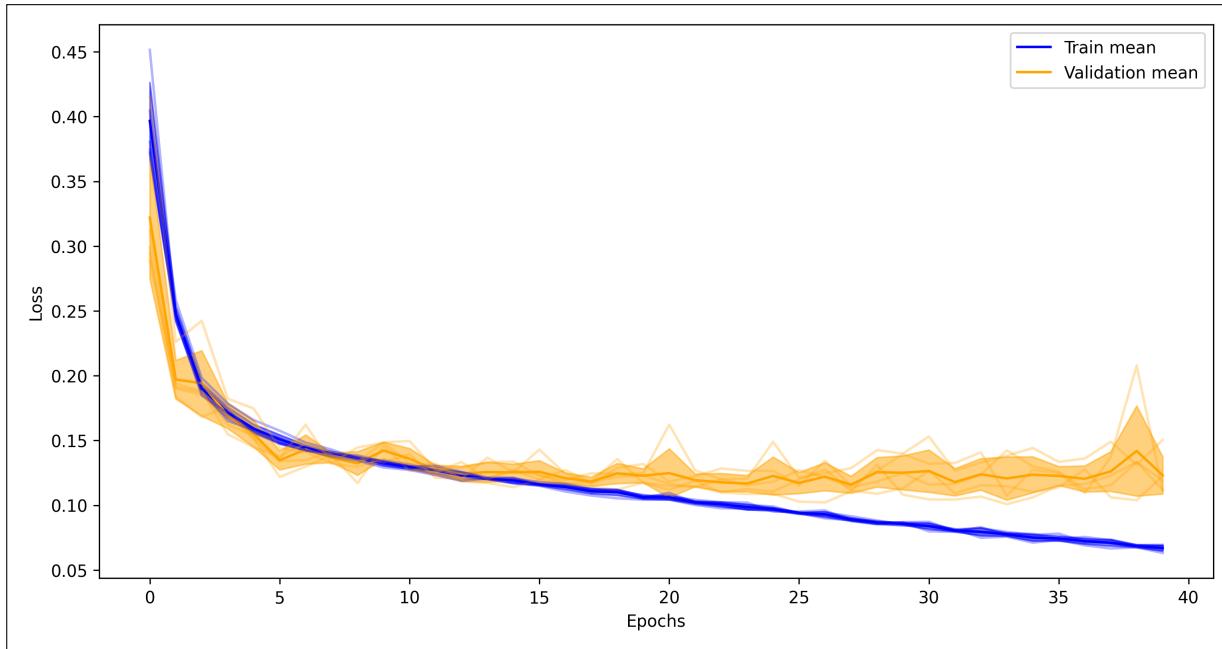


Figure 5.4 – CNN1 - Loss value of the validation and train for each epoch in a 5-fold, and the best model selected.

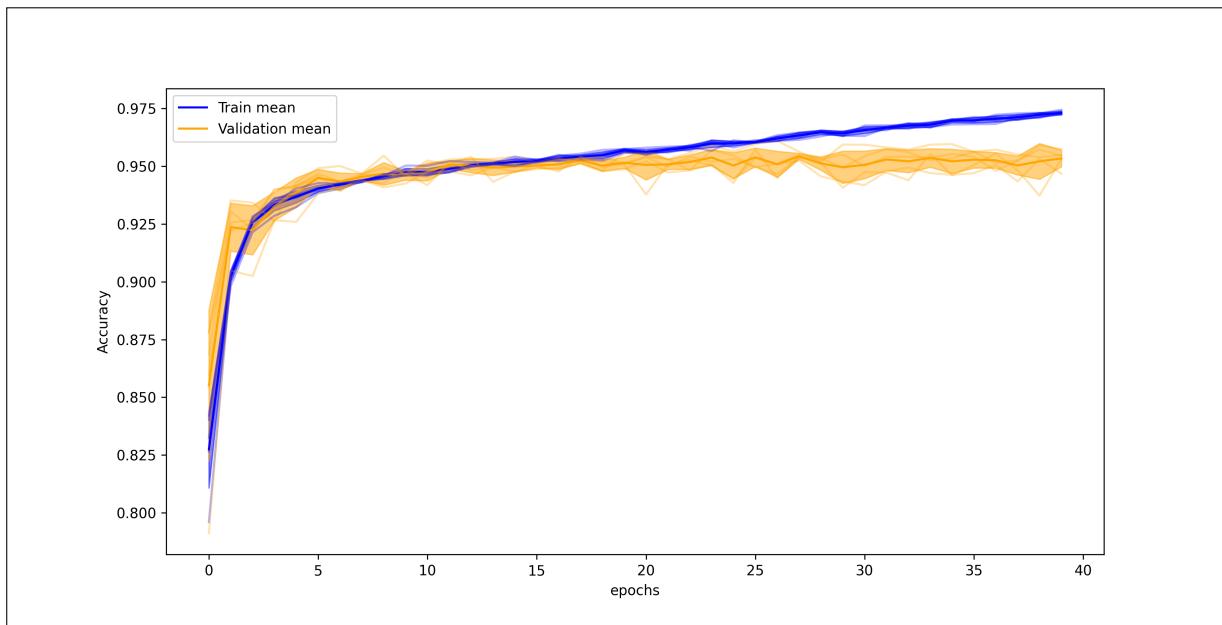


Figure 5.5 – CNN1 - Accuracy value of the validation and Train set for each epoch in a 5-fold

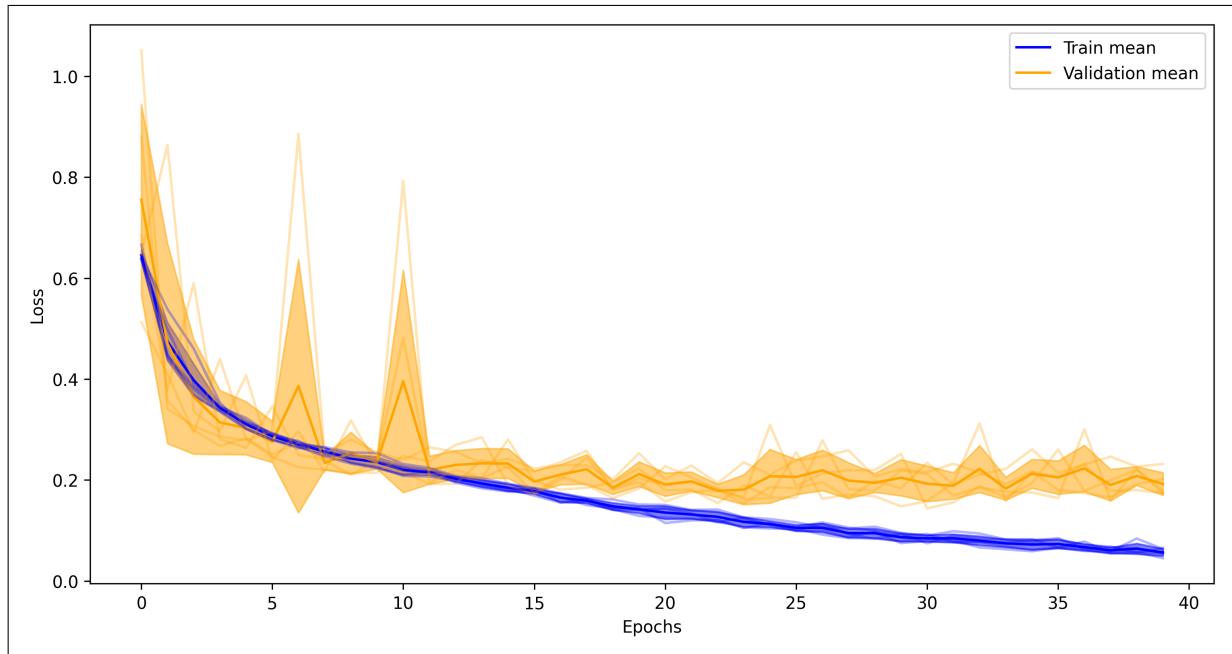


Figure 5.6 – CNN2 - Loss value of the validation and train for each epoch in a 5-fold, and the best model selected.

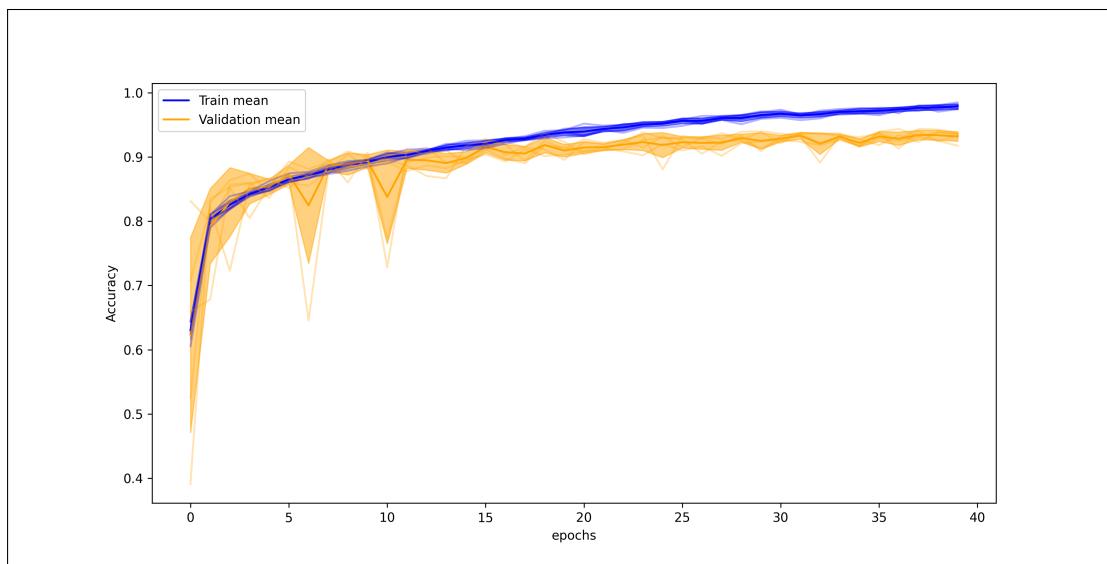


Figure 5.7 – CNN2 - Accuracy value of the validation and Train set for each epoch in a 5-fold

5.2 Deep STEP - Automated transient detection in the S-PLUS Transient Extension Program

The STEP project, described in Chapter 3, utilizes DeepSTEP, a deep learning tool developed in this thesis. DeepSTEP was trained on images from the S-PLUS Main survey that have undergone the STEP reduction pipeline (Section 3) and the difference imaging pipeline (Section 4).

Our approach employs a single Convolutional Neural Network (CNN) to classify images as Transients or Artifacts (Non-Transients). That was trained on real T80S difference imaging data. The output score of our CNN can be interpreted as the probability of a given image being a transient. In this case, unlike the DES-GW automated transient detection algorithm, we are not specifically searching for a host galaxy visually present in the search and template image.

5.2.1 DeepSTEP Data Collection

The training images are well-known objects detected by applying the STEP reduction and difference imaging pipeline to S-PLUS Main Survey data. All these objects were identified during the STEP program through visual inspection. The transients were previously labeled by a specialist, and any other images produced by the difference image that were not labeled as transients were designated as artifacts. Although various types of artifacts are shown in the section 3.1, our main objective is to remove as many false positive examples as possible, regardless of their types.

The images produced by the STEP transient detection pipeline are created using four channels (g , r , i , z), and for each filter, there are three different exposures observed, resulting in a total of 12 different images for each transient observation. Each of these images is composed of the search, template, and difference images, which are simultaneously fed into the network through independent channels in the format of $(51 \times 51 \times 3)$, similar to how red, green, and blue image arrays are fed into CNNs in traditional image analysis.

To ensure the integrity of our results, we avoid data leakage by assigning all 12 images of each supernova observation to the training, validation, or test dataset. This step was taken to prevent any bias in the model's test results that might occur if one of these images was used for training and another for testing and for that same reason, we do not perform a K-fold validation technique.

One of the challenges we faced while working on this dataset is that most of the transient candidates are contaminated by bad subtractions. This has resulted in an unbalanced classification problem. To address this issue, we downsampled the artifact class to have an equal number of transient images. Finally, the total collection of images was partitioned into approximately three subsets, with 67% of the data allocated for training, 13% for

validation, and 20% for testing. Additionally, an extra set of 96 transient images was obtained from two reported transients, at2022rri and SN2022tiv, which were identified by the STEP pipeline during a daily routine.

Deep learning methods are sensitive to visual features by construction. Therefore, we preprocess the data to enhance visualization and facilitate the training process. The preprocessing consists of two main steps: contrast adjustment and normalization.

- First, we perform image contrast adjustment by clipping the image histogram. This involves selecting lower and upper limits and setting every pixel below or above those values to be equal to the nearest bound. For the upper limit, we choose the 99.2 percentile for the search, template, and difference images. As for the lower bound, we choose the 0.01 percentile. An example of the improvement to stamp images by doing the preprocessing is shown in figure 5.8.
- Next, we normalize the images within the range of [0, 1]. This normalization step is a useful practice that improves the convergence of the neural network during training.

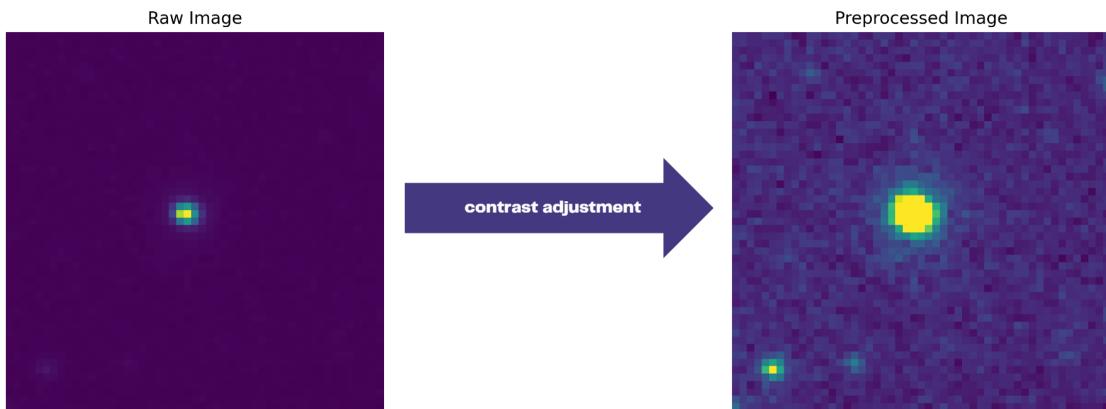


Figure 5.8 – This figure presents a comparative analysis of a diff image before and after contrast adjustment. On the left, a raw, unprocessed image portrays the diff channel in its original state. The image on the right illustrate the same channel after a contrast adjustment has been applied.

For the training dataset, we employed a data augmentation technique to enhance the training process. This technique includes applying rotations 90°, 180°, and 270° rotations, vertical mirroring (vflip), horizontal mirroring (hflip) and combinations of 270° rotations

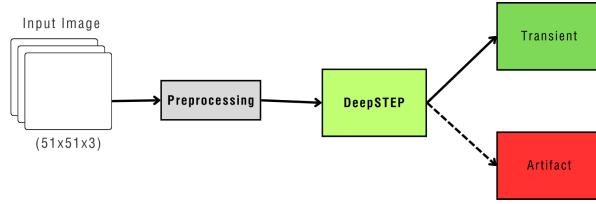


Figure 5.9 – A schematic overview of DeepSTEP pipeline

with mirroring (Same as used for DES-GW’s CNN, illustrated in Figure 5.1). By applying these techniques, we increase the training dataset size by a factor of 8.

5.2.2 Neural Network design

To address this classification problem, we made use of a family of CNN models known as MobileNet [56], which are based on depthwise separable convolutions. This architecture has demonstrated superior speed and accuracy characteristics in our dataset compared to other CNN-based models, such as VGG [50], ResNet [51], Inception [57], and custom-built CNNs (Section 5.1.2).

A Depthwise separable convolution Layer decomposes the convolution operation in two separate stages: depthwise convolution and pointwise convolution. In the depthwise convolution stage, the layer applies a separate kernel, to each input channel independently (search, template and diff), this means that each channel is convolved with its own set of weights and the output feature map retain the same number of channels as the input. Unlike the standard convolutional layer that applies one kernel over the input image, and the output of each channel are summed to produce the final output feature map. In the pointwise convolutional stage, a 1x1 standard convolution layer is applied to the feature map produced by the depthwise conv. layer. It aims to capture the cross-channel correlations and mix the information between different channels, allowing the model to learn from spatial and inter channel information efficiently.

We modify the original MobileNet architecture by replacing the last two layers, which consisted of an average pooling layer and a fully connected layer with one neuron, to a flattening layer and a fully connected layer with one neuron.

5.2.3 Training

We used the same optimizer as in the DES-GW scenario, Adam [39], with a binary cross-entropy loss. We initialized the networks with pre-trained weights from the ImageNet dataset [58] to improve computing time, convergence, overall performance, and training stability, as observed in Bom et al. (2021) [59]. The model was trained on the same Multi-GPU cluster and was implemented in TensorFlow 2 [55]). To determine the optimal set of weights for the model, we trained the network for an unlimited number of epochs until the Validation Loss stopped improving, as shown in Figure 5.10, this strategy

was selected based on the fact that our dataset contains multiple exposures from the same event, thus a standard cross validation would lead to a biased result.

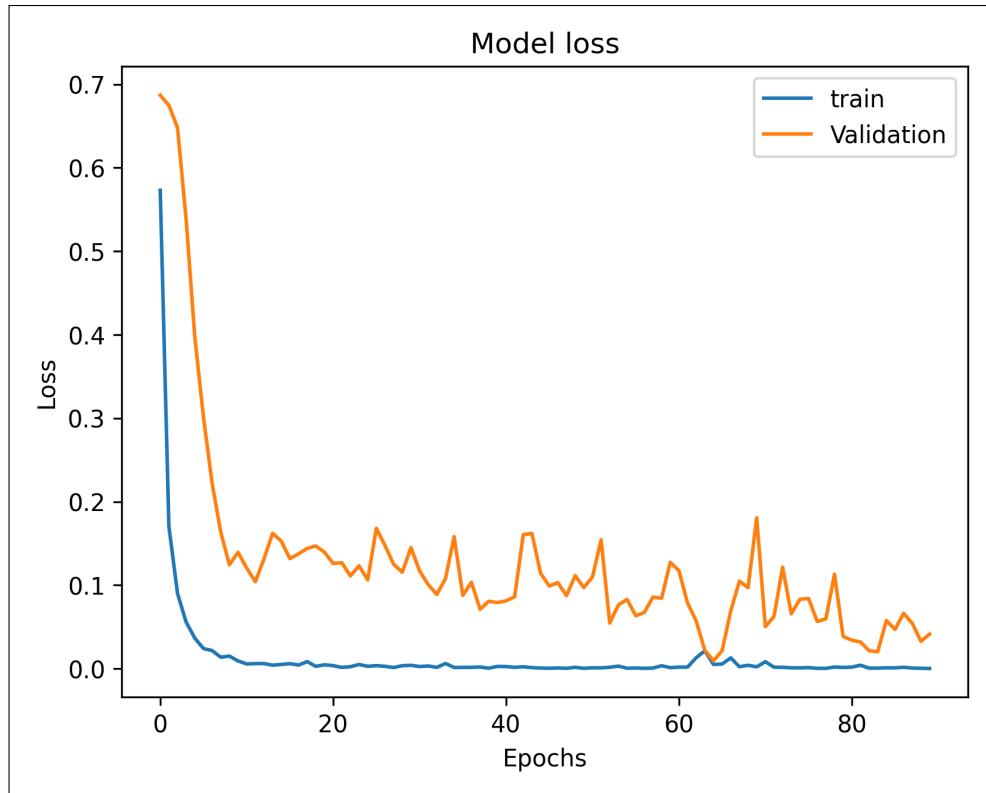


Figure 5.10 – DeepSTEP - Progression of Loss Values for the validation and training sets across epochs. The loss values represent the measure of the model’s performance, with lower values indicating better fitting to the data.

We selected the weights from the epoch with the lowest validation Loss. The orange line represents the validation loss, while the blue line represents the training loss. It is clear that the epoch where the validation loss reaches its minimum value coincides with the epoch where the true positive rate for the validation set is maximized. This weight selection method maximizes performance in finding transients in the validation set (Figure 5.11), which serves as a good indicator of the model’s performance in identifying unseen real transient data. This observation will be further supported and discussed in the results chapter.

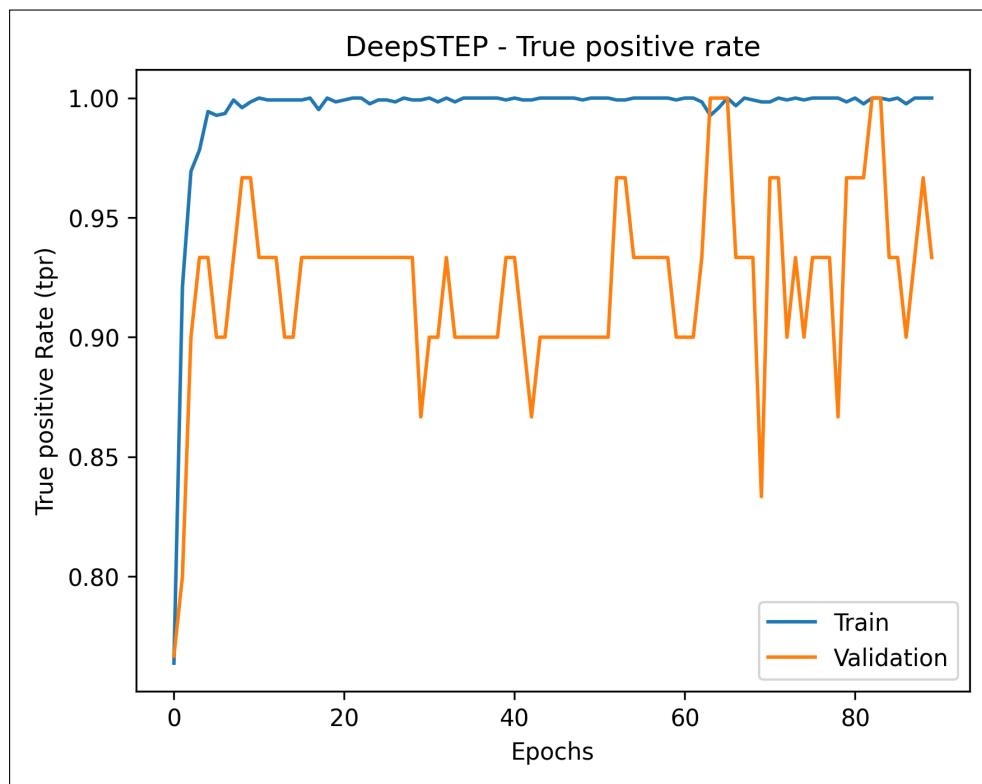


Figure 5.11 – DeepSTEP - True Positive Rate (TPR) progression across epochs for the validation and training sets.

CHAPTER 6

RESULTS

This chapter is divided into two sections. The first focuses on the two CNNs developed for the DES-GW search and discovery pipeline, which were trained on real DECam difference imaging data. The second section focuses on the results for DeepSTEP, which was trained on STEP difference imaging data. We present our results by evaluating each model with a test set and comparing them to previous studies and current tools.

6.1 DES-GW Automated transient detection algorithm performance on testset

The DES-GW Automated transient detection algorithm is composed of three main stages, preprocessing filters stage, first CNN (CNN-I) classification stage and second specialized CNN (CNN-II) classification stage. As described in chapter 5, the images not used for testing were designated to the training set, where we trained the two CNNs with a 5-Fold validation. We select the final model to be the one which outputs the best performance in accurately identifying transient+host samples on the testset.

The test set is composed of :

Label	Nº samples
Host + Transient	168
Preexisting point source	229
Non-obvious transient (Marginal)	238
Other artifact	13
Bad subtraction	730

Table 3 – DES-GW automated transient detection algorithm testset

Note that we share the same set of images with Shandonay et al's works [48]; however, we trained our model only on real data, so our test set is slightly different.

This section is divided into two subsections, one for each CNN test performance. The CNN-I is specialized on removing bad-subtractions. It was trained on two classes, bad-subtraction (label 0) and non-Badsub. (label 1), which includes every other type like non-obvious transient, transient + host galaxy, and other artifacts. This Neural Network act like a filter that passes the images classified as non-bad subtraction to the CNN-II stage. The second CNN is now specialized in finding transients. It was trained on two classes, Transients + host (label 1) and Non-Transients (label 0). The test set (Table 3) is the same for both CNNs, with the presence of the bad subtraction class into the second test, even it was not used by the CNN-II when training.

The choice of using the same test set for both is to ensure the pipeline’s performance in real follow-up scenarios. Because of our limited number of data, it was not possible to have 2 different test data that could be merged to form a larger test data. Our dataset is smaller than our precursor because we opted for using only real data on training. Training with real data helps the neural network learn from a more representative sample, leading to better performance on real-world tasks as it can learn to handle real-world complexities.

The preprocessing threshold for removing low-quality images was set by requiring a completeness closer to 0.99 in order to ensure a high number of positive images, represented by all classes not defined as “other artifacts”, are included. The resulting purity value was 0.98 and completeness of 0.99 at a threshold set to 13.86, such that nearly all passed images are true positives. The threshold of the second preprocessing stage – calculating the signal to noisy ration – was again set by requiring a completeness of 0.99. The resulting purity value of 0.992 at a threshold set to 3.80. Together, these two preprocessing filters remove large fractions of the “other artifacts” class, which includes noise samples and other less common artifacts. This simplifies the classification problem for the deep-learning stage of our approach.

6.1.1 CNN-I Performance

The CNN-I model is applied to the images that passed the preprocessing filters. Using the test set, their output probabilities were used to construct a receiver operating characteristic (ROC) curve. Our CNN-I model performed very closely to the perfect classifier (green dashed line on Figure 6.1), achieving an AUC of 0.991. One of the current methods, Shandonay et al. [48], achieved an AUC of 0.982 (figure 5 in article), the main difference is that our model maintains a lower false positive rate for a longer range of thresholds, thus achieving a maximum true positive rate with a false positive rate less than 0.2. Showing an evidential improvement on our precursor that only achieved a true positive rate of 1.0 when the false positive rate exceeded 0.6.

We selected a threshold that maximizes the accurate labeling of the Transient + host galaxy class on the non-Badsub class. Which also maximize the F1-score, the threshold is set to be 0.501 (Figure 6.1).

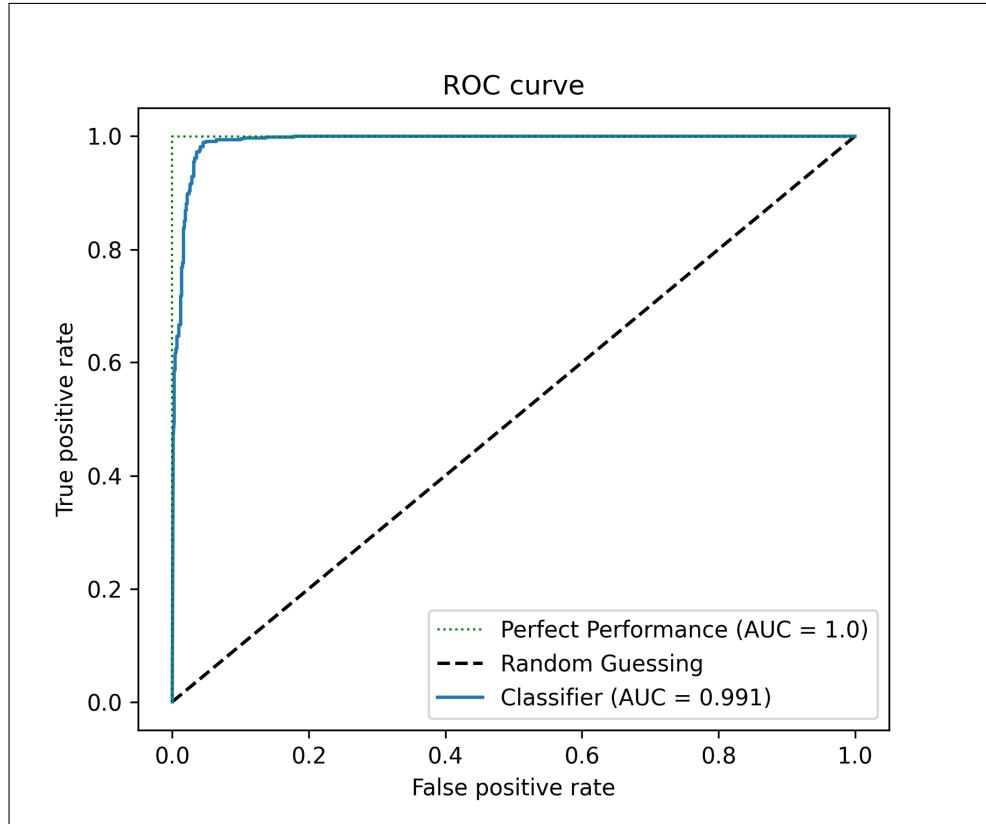


Figure 6.1 – CNN-I - Receiver Operating Characteristics (ROC) curve

Using this threshold, the false negative rate was 1.0% and the false positive rate was 4.5%, and most importantly, none transient + host sample was misclassified. These results show that our model outperforms the Shandonay et al., where it achieves a false negative rate of 3% and a true negative rate of 92%, where our model achieved a true negative rate of 95%.

Autoscan [29] achieves a false positive rate of 2.5% and a false negative rate of 4.0%. Direct comparison of our results is not possible because Autoscan was specifically trained to distinguish between samples of bad subtraction and transients, as explained in section 3. Consequently, it assigns high scores to other types of artifacts, as will be discussed in detail when evaluating CNN-II. When considering the ability to remove bad subtraction alone, CNNI was trained exclusively on real data, with a training sample of 1,300 bad subtractions, while Autoscan was trained with 444 871 bad subtraction samples [29], and even so our model achieved an approximate false positive rate.

6.1.2 CNN-II Performance

The final stage in the DES-GW Automated transient detection algorithm is applied to the images which were not identified as bad subtractions by CNN-I or removed as other artifacts by the preprocessing. The second CNN was developed to be specialized on finding transients + host galaxy (label 1), while also removing the remaining artifacts. It

		Predictions (binary) For Each Class				
		Good	Marginal	Bad Sub Class	Dark Spot + Noisy	Point Source
Predictions	Bad	0.000 0	0.021 5	0.955 697	0.000 0	0.009 2
	Good	1.000 168	0.979 233	0.045 33	1.000 13	0.991 227

Figure 6.2 – CNN-I - Confusion matrix of first CNN predictions divided by subclass, with a threshold of 0.501

has an architecture with 15x times more parameters than the CNN-I.

The best threshold, set to 0.44, was also chosen to be one that maximizes the labeling of the transient + host galaxy class correctly. When evaluating a model in an unbalanced scenario, where the transient class is significantly smaller compared to the artifact class, the accuracy metric alone may be misleading. This is because a classifier can achieve high accuracy by simply predicting the majority class for all instances, completely ignoring the minority class. Using this threshold, we can construct a confusion matrix (Figure 6.3) where it evidences that our model was able to achieved 94.6% of true positive rate, where 159 transients samples of the test set were correctly identified. We also removed 96.9% of the point source artifacts, showing our ability to remove not only bad subtraction but other artifacts. Additionally, we achieved a false positive rate of 10%. When compared to the Shandonay [48] work, we also improved the accuracy from 72% to 90%.

The direct comparison of our work and the precursor of this work is not simple, It should take to account that we are dealing with a similar dataset but not the same test data. however, it's noticeable that the CNN-II model has a superior ability of discriminating between classes, as demonstrated on Figure 6.4, which shows that our model gives high probability scores to the transient class, with 80% of the transient samples falling above a score of 0.75.

Also, from Figure 6.4, nearly all the preexisting point sources are correctly scored below the 0.1 probability threshold, indicating a clear distinction between these two classes. A

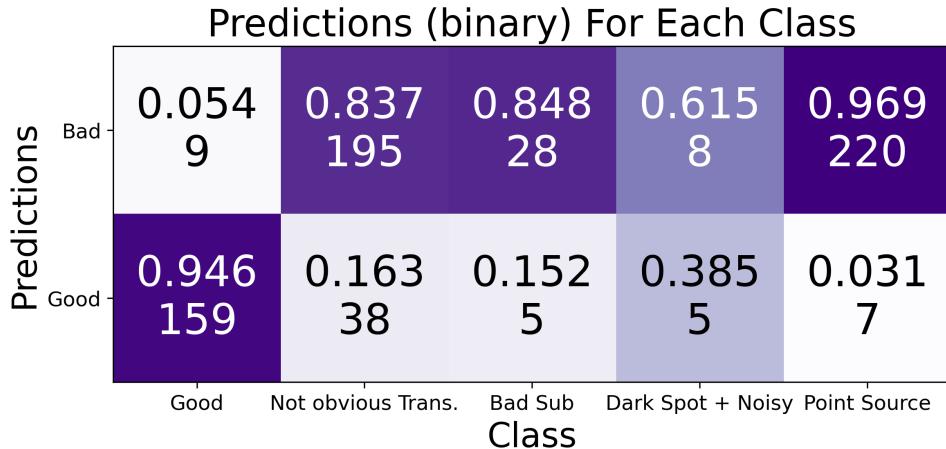


Figure 6.3 – CNN-II - Confusion matrix of second CNN predictions divided by subclass

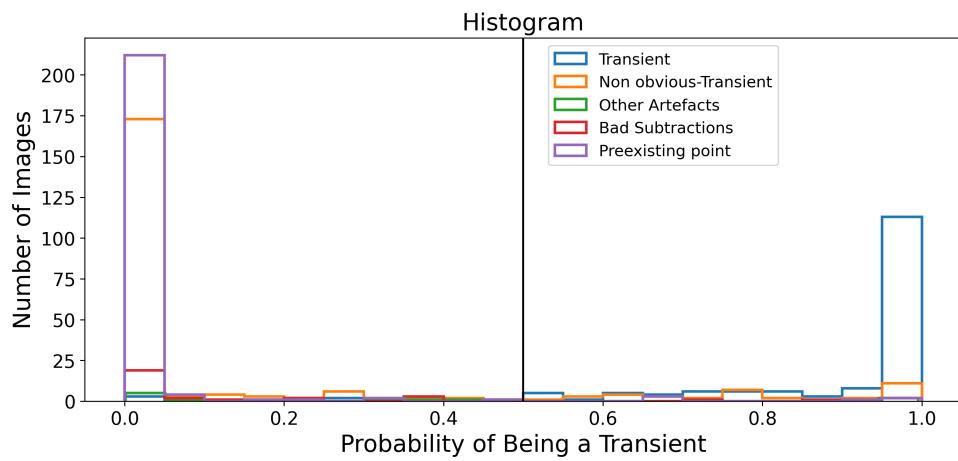


Figure 6.4 – CNN-II - output's score distribution on testset for each class

large portion of the “Not an Obvious Transient” class falls at higher probabilities, which shows the network’s ability to identify useful images that are not distinguishable with visual inspection.

The “not an obvious transient” class is not actually an artifact, it is more accurately described as a hard transient example where we do not have a scientific interest to do a follow-up. Kilonovas are expected to be in the presence of a host galaxy [5]. That is why we designated non-obvious transients as artifacts.

Figure 6.5 confirms another important result, by comparing a probability score given by Autoscan with the probability given by our CNN-II. We demonstrate the ability of our CNN-II in removing samples that Autoscan gives a high score.

Figure 6.5 shows that Autoscan scores shown in green have similar median values around 0.9 and similar upper and lower quartiles ranges, which means the not obvious transient class and preexisting point source class are not distinguished from the transient + host galaxy class. With our image classification algorithm, there is a significant difference in the probabilities assigned to transient + host compared to the other classes with higher

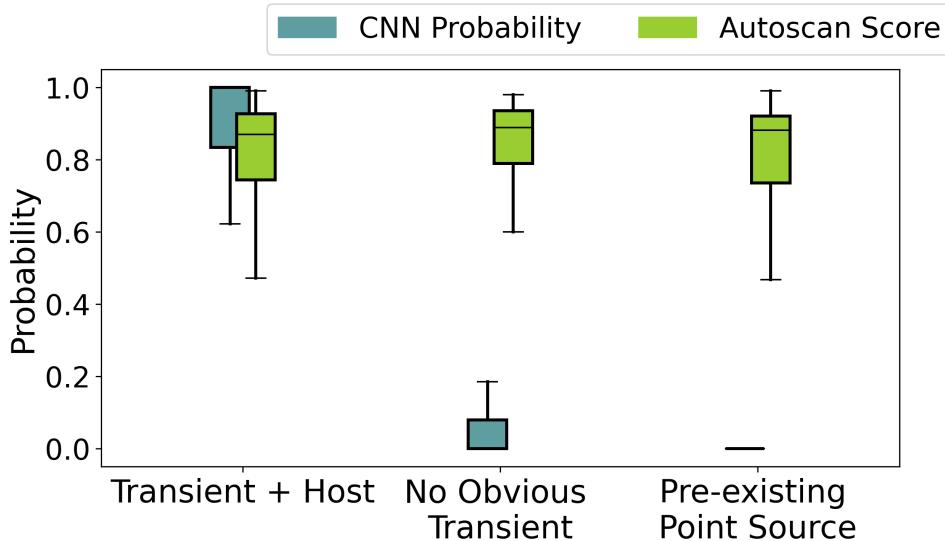


Figure 6.5 – CNN-II - Probabilities of being a transient + host galaxy compared to Autoscan’s probabilities. The boxes extend from the lower to upper quartile of each group, with the median marked by the line.

median values. This corroborated the idea that the output score can be used to identify real transients correctly.

Overall, when we take into account the full pipeline, we began with 168 transients samples, where after both CNN-1+CNN-II, we correctly identified 159, leading to a true positive rate of 94.6%. And we started with 1210 artifacts, and successfully removed 1140, leading to a false positive rate of 5.8 %.

6.2 DeepSTEP performance on testset

The purpose of the test dataset is to assess the model’s performance on unseen data within the pipeline while avoiding any biases. Therefore, four supernova events were selected for the test set instead of randomly choosing transient images. Typically, a supernova event is observed in four bands, each band composed of three different exposures. If each image meets the preprocessing conditions for difference imaging, a supernova event would produce 12 images of the same object.

The output of our convolutional neural network model is a score representing the probability of an image being classified as a Transient. To distinguish between Transients and Artifacts, it is essential to establish a threshold, denoted as ' t '. In this case, the threshold is set to 0.55, which maximizes the F1 score. By defining the threshold, we can construct a confusion matrix presented in Figure 6.6, which visually presents the model’s predictions and the true labels for each class.

In the test dataset, the DeepSTEP model exhibits a false positive rate of only 3.1%, indicating its first favorable result. During daily visual observations of transients, astrono-

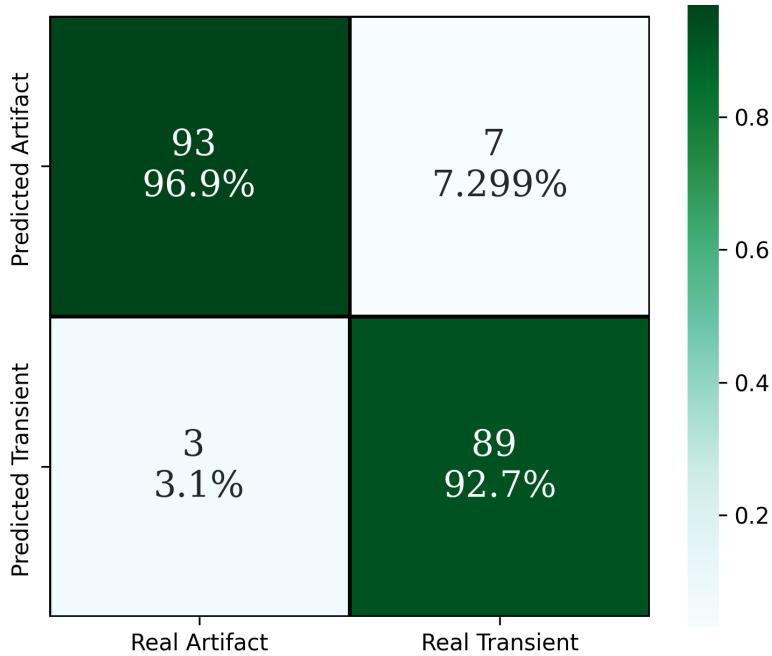


Figure 6.6 – DeepSTEP - Confusion matrix applied to the test data

mers could spend hours inspecting every transient candidate. By integrating DeepSTEP into the pipeline, with a false positive rate of 3.1%, It would reduce the number of images to be inspected to around a dozen images, consequently spending a few minutes to inspect all possible candidates, significantly saving time and effort.

Furthermore, DeepSTEP achieves a true positive rate of 92.7%, indicating its ability to correctly identify transients. Our model also demonstrates a high precision, indicating that 96.7% of all predicted positive classes in the test set were actual transients. This highlights the suitability of DeepSTEP to work with visual inspection, as our model effectively filters out artifacts while preserving transients.

The performance of a model can also be evaluated by analyzing the ROC curve presented in Figure 6.7 and the precision-recall curve shown in Figure 6.8. The ROC curve shows the trade-off between the TPR and FPR of the model as the threshold is varied. The Area in the image refers to the area under the curve (AUC). Remarkably, the DeepSTEP model achieves an AUC of 0.992, This means that the model possesses a 99.2% chance of correctly distinguishing between the positive and negative samples that were randomly chosen.

Finally, the last metric we used to evaluate our model and determine the threshold is the precision-recall curve, it shows the trade-off between precision and recall. A higher precision indicates that the model is presenting a high confidence sample, i.e. high purity, while a higher recall is defined as high completeness. To obtain this balance, we choose the

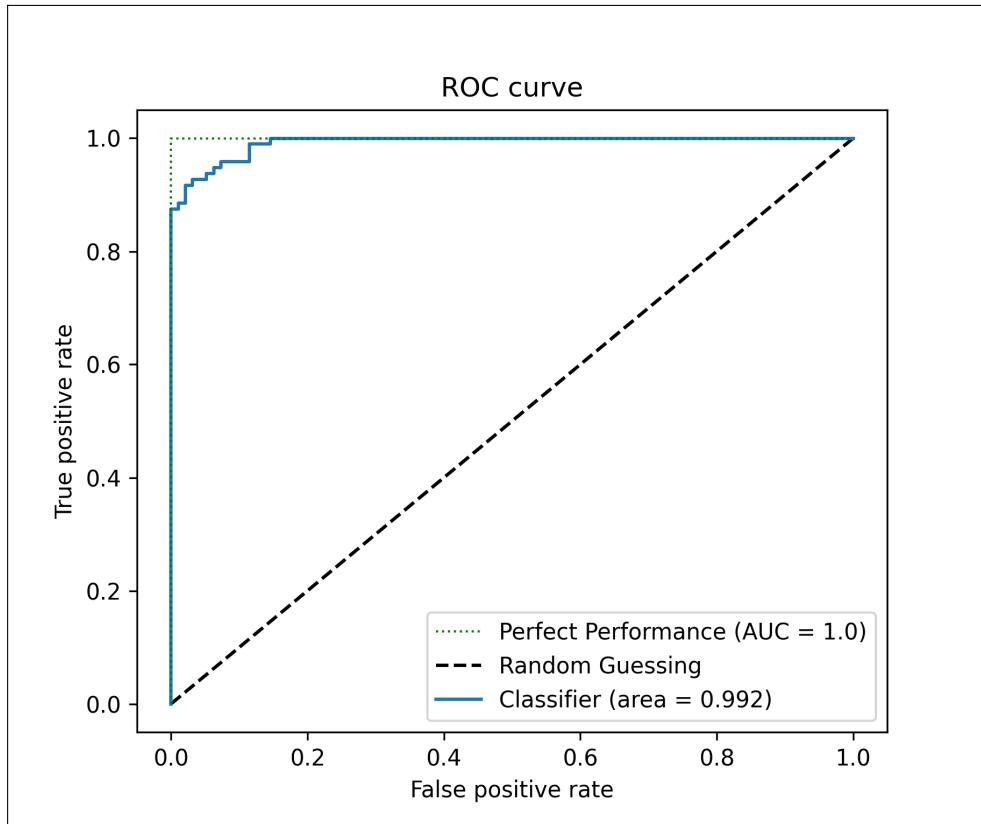


Figure 6.7 – DeepSTEP - Receiver Operating Characteristics (ROC) curve

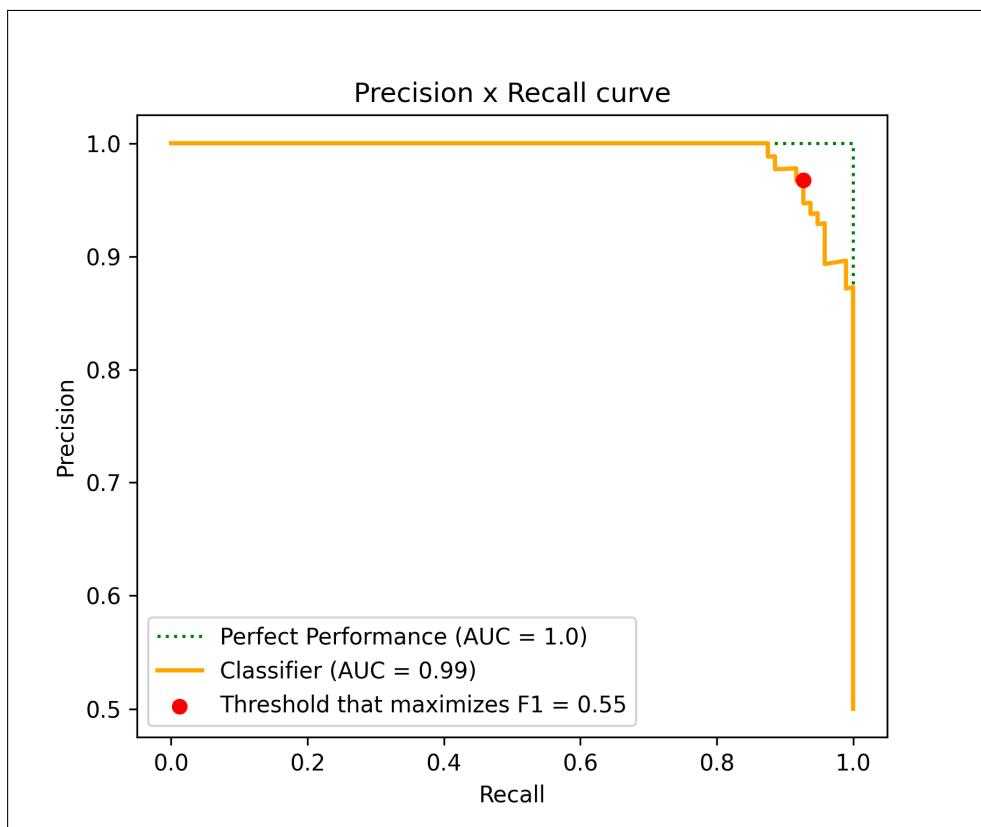


Figure 6.8 – DeepSTEP - Precision-recall (PR) curve

threshold that maximizes the harmonic mean between recall and precision, as it provides a balanced measure of both metrics.

DeepSTEP achieved an AUPRC (Area Under Precision-Recall Curve) of 0.99, which aligns with the AUC from the ROC curve. This indicates the model's capability to accurately identify transients with high purity and maintain a high completeness. Such results hold significant value in our analysis, particularly in scenarios where identifying positive examples is crucial, such as detecting potential multimessengers.

We can also analyze some miss-classified examples from the Transient set, which exhibit instances where the model incorrectly assigns labels. Figure 6.9 presents an example of a misclassified transient with a score of 27%. The corresponding difference image reveals a narrow point spread function (PSF) and the proximity of the transient object to a highly brightened surface. This observation leads to the conclusion that it is not an easily identifiable transient (not an obvious transient).

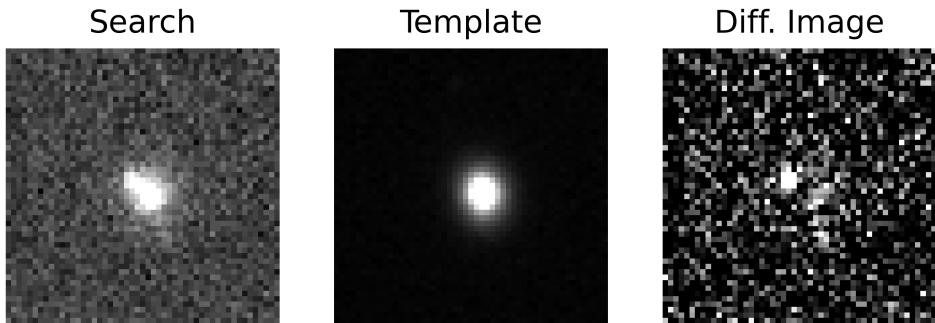


Figure 6.9 – Miss-classified example, which was given a probability of 27% of being a Transient.

These miss-classified examples hold limited importance, as challenging transient cases are typically not prioritized for follow-up. Therefore, our model demonstrates the capability to detect transient targets effectively. Even in cases where it may misclassify, they often represent complex examples that are not of significant interest.

Overall, the DeepSTEP integrated on the STEP pipelines consists in a deep learning tool, with 3.229.889 parameters, capable of rapidly classifying an image within seconds, providing the ID image and the associated probability of being a transient and sent it to the web page.

CHAPTER 7

CONCLUSION

The continuing construction and development of ground-based and space-based sky surveys ranging from gamma rays and X-rays, ultraviolet, optical, and infrared to radio bands is bringing astronomy into the big data era. More than ever, we need an alternative to deal with large amounts of data. One prominent example is the Legacy Survey of Space and Time (LSST) [60], which is a planned 10-year survey of the southern sky that will take place at the [Vera C. Rubin Observatory](#). LSST is expected to produce 20 terabytes (TB) of raw data per day¹. This highlights the scale of data production in modern astronomy and the urgent need for new approaches to manage and extract meaningful information from these extensive datasets. Especially in time-domain imaging surveys accompanied by a spectroscopic follow-up program, challenges related to data volume and purity are compounded by time pressure to produce lists of the most promising targets for follow-up observations before they become too faint to observe or fall outside a window of scientific utility. To address these challenges, astronomers are actively exploring methods to automate data processing and analysis using techniques such as artificial intelligence. The decisions made by Machine Learning algorithms are automatic, reproducible, and sufficiently fast to process streaming data in real time. Importantly, given sufficient computing resources, they remain fast and consistent despite increasing data production rates.

Therefore, in this dissertation, we utilized Convolutional Neural Networks to address the time-consuming problem of removing Artifacts during visual inspections for potential transient candidates and optical gravitational wave counterparts. This work was motivated by the desire to improve the efficiency and speed of multimessenger follow-ups.

The DES-GW automated detection algorithm achieved a false positive rate of 5.8%, with a false negative rate of 3%. By integrating our tool into the DES-GW Search and Discovery pipeline, we have prepared the follow-up campaign for the increased event rate,

¹ <https://www.lsst.org/about/dm>

which we expect to lead to a higher probability of detecting the next multimessenger counterpart. The implementation of DeepSTEP on the S-PLUS Transient Extension Program (STEP) has demonstrated substantial time savings during image visual inspections, reducing the process from hours to minutes, evidenced by the model's ability to correctly remove 96.9% of all artifacts in our test dataset. In addition, DeepSTEP achieved an AUC of 0.992, proving its reliability and usefulness in identifying astronomical transients.

In conclusion, while traditional scientific approaches and human interpretation play a crucial role in astronomy, the era of massive astronomical data necessitates the development and application of advanced algorithms and computational methods. These advancements in transient search algorithms and techniques will facilitate the identification of useful and interesting astronomical objects more effectively and can act as powerful discovery engines in the vast sea of astronomical data. The subsequent phase of our work involves generalizing the efficacy of our algorithm by identifying more challenging scenarios involving transients, particularly those in proximity to high-brightness surfaces. Also, we intend to improve our models by incorporating additional training examples, which will be detected in future observations.

REFERENCES

- [1] Imre Bartos and Marek Kowalski. Multimessenger astronomy. In Multimessenger Astronomy, 2399-2891, pages 1–1 to 1–18. IOP Publishing, 2017.
- [2] Adam G. Riess, Alexei V. Filippenko, Peter Challis, Alejandro Clocchiatti, Alan Diercks, Peter M. Garnavich, Ron L. Gilliland, Craig J. Hogan, Saurabh Jha, Robert P. Kirshner, B. Leibundgut, M. M. Phillips, David Reiss, Brian P. Schmidt, Robert A. Schommer, R. Chris Smith, J. Spyromilio, Christopher Stubbs, Nicholas B. Suntzeff, and John Tonry. Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant. , 116(3):1009–1038, September 1998.
- [3] Li-Xin Li and Bohdan Paczyński. Transient events from neutron star mergers. The Astrophysical Journal, 507(1):L59–L62, nov 1998.
- [4] B. P. Abbott, R. Abbott, T. D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, V. B. Adya, C. Affeldt, M. Afrough, B. Agarwal, M. Agathos, K. Agatsuma, N. Aggarwal, O. D. Aguiar, L. Aiello, A. Ain, P. Ajith, B. Allen, G. Allen, A. Allocca, P. A. Altin, A. Amato, A. Ananyeva, S. B. Anderson, W. G. Anderson, S. V. Angelova, S. Antier, S. Appert, K. Arai, M. C. Araya, J. S. Areeda, N. Arnaud, K. G. Arun, S. Ascenzi, G. Ashton, M. Ast, S. M. Aston, P. Astone, D. V. Atallah, P. Aufmuth, C. Aulbert, K. AultONeal, C. Austin, A. Avila-Alvarez, S. Babak, P. Bacon, M. K. M. Bader, S. Bae, M. Bailes, P. T. Baker, F. Baldaccini, G. Ballardin, S. W. Ballmer, S. Banagiri, J. C. Barayoga, S. E. Barclay, B. C. Barish, D. Barker, K. Barkett, F. Barone, B. Barr, L. Barsotti, M. Barsuglia, D. Barta, S. D. Barthelmy, J. Bartlett, I. Bartos, R. Bassiri, A. Basti, J. C. Batch, M. Bawaj, J. C. Bayley, M. Bazzan, B. Bécsy, C. Beer, M. Bejger, I. Belahcene, A. S. Bell, B. K. Berger, G. Bergmann, S. Bernuzzi, J. J. Bero, C. P. L. Berry, D. Bersanetti, A. Bertolini, J. Betzwieser, S. Bhagwat, R. Bhandare, I. A. Bilenko, G. Billingsley, C. R. Billman, J. Birch, R. Birney, O. Birnholtz, S. Biscans, S. Biscoveanu, A. Bisht, M. Bitossi, C. Biwer, M. A. Bizouard, J. K. Blackburn, J. Blackman, C. D. Blair, D. G. Blair, R. M. Blair, S. Bloemen, O. Bock, N. Bode, M. Boer, G. Bogaert, A. Bohe, F. Bondu, E. Bonilla, R. Bonnand, B. A. Boom, R. Bork, V. Boschi, S. Bose, K. Bos-

sie, Y. Bouffanais, A. Bozzi, C. Bradaschia, P. R. Brady, M. Branchesi, J. E. Brau, T. Briant, A. Brillet, M. Brinkmann, V. Brisson, P. Brockill, J. E. Broida, A. F. Brooks, D. A. Brown, D. D. Brown, S. Brunett, C. C. Buchanan, A. Buikema, T. Bulik, H. J. Bulten, A. Buonanno, D. Buskulic, C. Buy, R. L. Byer, M. Cabero, L. Cadonati, G. Cagnoli, C. Cahillane, J. Calderón Bustillo, T. A. Callister, E. Calloni, J. B. Camp, M. Canepa, P. Canizares, K. C. Cannon, H. Cao, J. Cao, C. D. Capano, E. Capocasa, F. Carbognani, S. Caride, M. F. Carney, G. Carullo, J. Casanueva Diaz, C. Casentini, S. Caudill, M. Cavaglià, F. Cavalier, R. Cavalieri, G. Cella, C. B. Cepeda, P. Cerdá-Durán, G. Cerretani, E. Cesarini, S. J. Chamberlin, M. Chan, S. Chao, P. Charlton, E. Chase, E. Chassande-Mottin, D. Chatterjee, K. Chatzioannou, B. D. Cheeseboro, H. Y. Chen, X. Chen, Y. Chen, H.-P. Cheng, H. Chia, A. Chincarini, A. Chiummo, T. Chmiel, H. S. Cho, M. Cho, J. H. Chow, N. Christensen, Q. Chu, A. J. K. Chua, S. Chua, A. K. W. Chung, S. Chung, G. Ciani, R. Ciolfi, C. E. Cirelli, A. Cirone, F. Clara, J. A. Clark, P. Clearwater, F. Cleva, C. Cocchieri, E. Coccia, P.-F. Cohadon, D. Cohen, A. Colla, C. G. Collette, L. R. Cominsky, M. Constancio, L. Conti, S. J. Cooper, P. Corban, T. R. Corbitt, I. Cordero-Carrión, K. R. Corley, N. Cornish, A. Corsi, S. Cortese, C. A. Costa, M. W. Coughlin, S. B. Coughlin, J.-P. Coulon, S. T. Countryman, P. Couvares, P. B. Covas, E. E. Cowan, D. M. Coward, M. J. Cowart, D. C. Coyne, R. Coyne, J. D. E. Creighton, T. D. Creighton, J. Cripe, S. G. Crowder, T. J. Cullen, A. Cumming, L. Cunningham, E. Cuoco, T. Dal Canton, G. Dálya, S. L. Danilishin, S. D'Antonio, K. Danzmann, A. Dasgupta, C. F. Da Silva Costa, V. Dattilo, I. Dave, M. Davier, D. Davis, E. J. Daw, B. Day, S. De, D. DeBra, J. Degallaix, M. De Laurentis, S. Deléglise, W. Del Pozzo, N. Demos, T. Denker, T. Dent, R. De Pietri, V. Dergachev, R. De Rosa, R. T. DeRosa, C. De Rossi, R. DeSalvo, O. de Varona, J. Devenson, S. Dhurandhar, M. C. Díaz, T. Dietrich, L. Di Fiore, M. Di Giovanni, T. Di Girolamo, A. Di Lieto, S. Di Pace, I. Di Palma, F. Di Renzo, Z. Doctor, V. Dolique, F. Donovan, K. L. Dooley, S. Doravari, I. Dorrington, R. Douglas, M. Dovale Álvarez, T. P. Downes, M. Drago, C. Dreissigacker, J. C. Driggers, Z. Du, M. Ducrot, R. Dudi, P. Dupej, S. E. Dwyer, T. B. Edo, M. C. Edwards, A. Effler, H.-B. Eggenstein, P. Ehrens, J. Eichholz, S. S. Eikenberry, R. A. Eisenstein, R. C. Essick, D. Estevez, Z. B. Etienne, T. Etzel, M. Evans, T. M. Evans, M. Factourovich, V. Fafone, H. Fair, S. Fairhurst, X. Fan, S. Farinon, B. Farr, W. M. Farr, E. J. Fauchon-Jones, M. Favata, M. Fays, C. Fee, H. Fehrmann, J. Feicht, M. M. Fejer, A. Fernandez-Galiana, I. Ferrante, E. C. Ferreira, F. Ferrini, F. Fidecaro, D. Finstad, I. Fiori, D. Fiorucci, M. Fishbach, R. P. Fisher, M. Fitz-Axen, R. Flaminio, M. Fletcher, H. Fong, J. A. Font, P. W. F. Forsyth, S. S. Forsyth, J.-D. Fournier, S. Frasca, F. Frasconi, Z. Frei, A. Freise, R. Frey, V. Frey, E. M. Fries, P. Fritschel, V. V. Frolov, P. Fulda, M. Fyffe, H. Gabbard, B. U. Gadre, S. M. Gaebel, J. R. Gair, L. Gammaletti, M. R. Ganija, S. G. Gaonkar, C. Garcia-Quiros, F. Garufi, B. Gateley, S. Gau-

dio, G. Gaur, V. Gayathri, N. Gehrels, G. Gemme, E. Genin, A. Gennai, D. George, J. George, L. Gergely, V. Germain, S. Ghonge, Abhirup Ghosh, Archisman Ghosh, S. Ghosh, J. A. Giaime, K. D. Giardina, A. Giazotto, K. Gill, L. Glover, E. Goetz, R. Goetz, S. Gomes, B. Goncharov, G. González, J. M. Gonzalez Castro, A. Gopakumar, M. L. Gorodetsky, S. E. Gossan, M. Gosselin, R. Gouaty, A. Grado, C. Graef, M. Granata, A. Grant, S. Gras, C. Gray, G. Greco, A. C. Green, E. M. Gretarsson, P. Groot, H. Grote, S. Grunewald, P. Gruning, G. M. Guidi, X. Guo, A. Gupta, M. K. Gupta, K. E. Gushwa, E. K. Gustafson, R. Gustafson, O. Halim, B. R. Hall, E. D. Hall, E. Z. Hamilton, G. Hammond, M. Haney, M. M. Hanke, J. Hanks, C. Hanna, M. D. Hannam, O. A. Hannuksela, J. Hanson, T. Hardwick, J. Harms, G. M. Harry, I. W. Harry, M. J. Hart, C.-J. Haster, K. Haughian, J. Healy, A. Heidmann, M. C. Heintze, H. Heitmann, P. Hello, G. Hemming, M. Hendry, I. S. Heng, J. Hennig, A. W. Heptonstall, M. Heurs, S. Hild, T. Hinderer, W. C. G. Ho, D. Hoak, D. Hofman, K. Holt, D. E. Holz, P. Hopkins, C. Horst, J. Hough, E. A. Houston, E. J. Howell, A. Hreibi, Y. M. Hu, E. A. Huerta, D. Huet, B. Hughey, S. Husa, S. H. Huttner, T. Huynh-Dinh, N. Indik, R. Inta, G. Intini, H. N. Isa, J.-M. Isac, M. Isi, B. R. Iyer, K. Izumi, T. Jacqmin, K. Jani, P. Jaradowski, S. Jawahar, F. Jiménez-Forteza, W. W. Johnson, N. K. Johnson-McDaniel, D. I. Jones, R. Jones, R. J. G. Jonker, L. Ju, J. Junker, C. V. Kalaghatgi, V. Kalogera, B. Kamai, S. Kandhasamy, G. Kang, J. B. Kanner, S. J. Kapadia, S. Karki, K. S. Karvinen, M. Kasprzack, W. Kastaun, M. Katolik, E. Katsavounidis, W. Katzman, S. Kaufer, K. Kawabe, F. Kéfélian, D. Keitel, A. J. Kemball, R. Kennedy, C. Kent, J. S. Key, F. Y. Khalili, I. Khan, S. Khan, Z. Khan, E. A. Khazanov, N. Kijbunchoo, Chunglee Kim, J. C. Kim, K. Kim, W. Kim, W. S. Kim, Y.-M. Kim, S. J. Kimbrell, E. J. King, P. J. King, M. Kinley-Hanlon, R. Kirchhoff, J. S. Kissel, L. Kleybolte, S. Klimenko, T. D. Knowles, P. Koch, S. M. Koehlenbeck, S. Koley, V. Kondrashov, A. Kontos, M. Korobko, W. Z. Korth, I. Kowalska, D. B. Kozak, C. Krämer, V. Kringel, B. Krishnan, A. Królak, G. Kuehn, P. Kumar, R. Kumar, S. Kumar, L. Kuo, A. Kutynia, S. Kwang, B. D. Lackey, K. H. Lai, M. Landry, R. N. Lang, J. Lange, B. Lantz, R. K. Lanza, S. L. Larson, A. Lartaux-Vollard, P. D. Lasky, M. Laxen, A. Lazzarini, C. Lazzaro, P. Leaci, S. Leavey, C. H. Lee, H. K. Lee, H. M. Lee, H. W. Lee, K. Lee, J. Lehmann, A. Lenon, E. Leon, M. Leonard, N. Leroy, N. Letendre, Y. Levin, T. G. F. Li, S. D. Linker, T. B. Littenberg, J. Liu, X. Liu, R. K. L. Lo, N. A. Lockerbie, L. T. London, J. E. Lord, M. Lorenzini, V. Loriette, M. Lormand, G. Losurdo, J. D. Lough, C. O. Lousto, G. Lovelace, H. Lück, D. Lumaca, A. P. Lundgren, R. Lynch, Y. Ma, R. Macas, S. Macfoy, B. Machenschalk, M. MacInnis, D. M. Macleod, I. Magaña Hernandez, F. Magaña-Sandoval, L. Magaña Zertuche, R. M. Magee, E. Majorana, I. Maksimovic, N. Man, V. Mandic, V. Mangano, G. L. Mansell, M. Manske, M. Mantovani, F. Marchesoni, F. Marion, S. Márka, Z. Márka, C. Markakis, A. S. Markosyan, A. Markowitz, E. Maros, A. Marquina, P. Marsh,

- F. Martelli, L. Martellini, I. W. Martin, R. M. Martin, D. V. Martynov, J. N. Marx, K. Mason, E. Massera, A. Masserot, T. J. Massinger, M. Masso-Reid, S. Mastrogiovanni, A. Matas, F. Matichard, L. Matone, N. Mavalvala, N. Mazumder, R. McCarthy, D. E. McClelland, S. McCormick, L. McCuller, S. C. McGuire, G. McIntyre, J. McIver, D. J. McManus, L. McNeill, T. McRae, S. T. McWilliams, D. Meacher, G. D. Meadors, M. Mehmet, J. Meidam, E. Mejuto-Villa, A. Melatos, G. Mendell, R. A. Mercer, E. L. Merilh, M. Merzougui, S. Meshkov, C. Messenger, C. Messick, R. Metzdorff, P. M. Meyers, H. Miao, C. Michel, H. Middleton, E. E. Mikhailov, L. Milano, A. L. Miller, B. B. Miller, J. Miller, M. Millhouse, M. C. Milovich-Goff, O. Minazzoli, Y. Minenkov, J. Ming, C. Mishra, S. Mitra, V. P. Mitrofanov, G. Mitzelmakher, R. Mittleman, D. Moffa, A. Moggi, K. Mogushi, M. Mohan, S. R. P. Mohapatra, I. Molina, M. Montani, C. J. Moore, D. Moraru, G. Moreno, S. Morisaki, S. R. Morriss, B. Mours, C. M. Mow-Lowry, G. Mueller, A. W. Muir, Arunava Mukherjee, D. Mukherjee, S. Mukherjee, N. Mukund, A. Mullavey, J. Munch, E. A. Muñiz, M. Muratore, P. G. Murray, A. Nagar, K. Napier, I. Nardecchia, L. Naticchioni, R. K. Nayak, J. Neilson, G. Nelemans, T. J. N. Nelson, M. Nery, A. Neunzert, L. Nevin, J. M. Newport, G. Newton, K. K. Y. Ng, P. Nguyen, T. T. Nguyen, D. Nichols, A. B. Nielsen, S. Nissanke, A. Nitz, A. Noack, F. Nocera, D. Nolting, C. North, L. K. Nuttall, J. Oberling, G. D. O'Dea, G. H. Ogin, J. J. Oh, S. H. Oh, F. Ohme, M. A. Okada, M. Oliver, P. Oppermann, Richard J. Oram, B. O'Reilly, R. Ormiston, L. F. Ortega, R. O'Shaughnessy, S. Ossokine, D. J. Ottaway, H. Overmier, B. J. Owen, A. E. Pace, J. Page, M. A. Page, A. Pai, S. A. Pai, J. R. Palamos, O. Palashov, C. Palomba, A. Pal-Singh, Howard Pan, Huang-Wei Pan, B. Pang, P. T. H. Pang, C. Pankow, F. Pannarale, B. C. Pant, F. Paoletti, A. Paoli, M. A. Papa, A. Parida, W. Parker, D. Pascucci, A. Pasqualetti, R. Passaquieti, D. Passuello, M. Patil, B. Patricelli, B. L. Pearlstone, M. Pedraza, R. Pedurand, L. Pekowsky, A. Pele, S. Penn, C. J. Perez, A. Perreca, L. M. Perri, H. P. Pfeiffer, M. Phelps, O. J. Piccinni, M. Pichot, F. Piergiovanni, V. Pierro, G. Pillant, L. Pinard, I. M. Pinto, M. Pirello, M. Pitkin, M. Poe, R. Poggiani, P. Popolizio, E. K. Porter, A. Post, J. Powell, J. Prasad, J. W. W. Pratt, G. Pratten, V. Predoi, T. Prestegard, M. Prijatelj, M. Principe, S. Priovitera, R. Prix, G. A. Prodi, L. G. Prokhorov, O. Puncken, M. Punturo, P. Puppo, M. Pürller, H. Qi, V. Quetschke, E. A. Quintero, R. Quitzow-James, F. J. Raab, D. S. Rabeling, H. Radkins, P. Raffai, S. Raja, C. Rajan, B. Rajbhandari, M. Rakhamanov, K. E. Ramirez, A. Ramos-Buades, P. Rapagnani, V. Raymond, M. Razzano, J. Read, T. Regimbau, L. Rei, S. Reid, D. H. Reitze, W. Ren, S. D. Reyes, F. Ricci, P. M. Ricker, S. Rieger, K. Riles, M. Rizzo, N. A. Robertson, R. Robie, F. Robinet, A. Rocchi, L. Rolland, J. G. Rollins, V. J. Roma, J. D. Romano, R. Romano, C. L. Romel, J. H. Romie, D. Rosińska, M. P. Ross, S. Rowan, A. Rüdiger, P. Ruggi, G. Rutins, K. Ryan, S. Sachdev, T. Sadecki, L. Sadeghian, M. Sakellariadou, L. Salconi,

M. Saleem, F. Salemi, A. Samajdar, L. Sammut, L. M. Sampson, E. J. Sanchez, L. E. Sanchez, N. Sanchis-Gual, V. Sandberg, J. R. Sanders, B. Sassolas, B. S. Sathyaprakash, P. R. Saulson, O. Sauter, R. L. Savage, A. Sawadsky, P. Schale, M. Scheel, J. Scheuer, J. Schmidt, P. Schmidt, R. Schnabel, R. M. S. Schofield, A. Schönbeck, E. Schreiber, D. Schuette, B. W. Schulte, B. F. Schutz, S. G. Schwalbe, J. Scott, S. M. Scott, E. Seidel, D. Sellers, A. S. Sengupta, D. Sentenac, V. Sequino, A. Sergeev, D. A. Shaddock, T. J. Shaffer, A. A. Shah, M. S. Shahriar, M. B. Shaner, L. Shao, B. Shapiro, P. Shawhan, A. Sheperd, D. H. Shoemaker, D. M. Shoemaker, K. Siellez, X. Siemens, M. Sieniawska, D. Sigg, A. D. Silva, L. P. Singer, A. Singh, A. Singhal, A. M. Sintes, B. J. J. Slagmolen, B. Smith, J. R. Smith, R. J. E. Smith, S. Somalia, E. J. Son, J. A. Sonnenberg, B. Sorazu, F. Sorrentino, T. Souradeep, A. P. Spencer, A. K. Srivastava, K. Staats, A. Staley, M. Steinke, J. Steinlechner, S. Steinlechner, D. Steinmeyer, S. P. Stevenson, R. Stone, D. J. Stops, K. A. Strain, G. Stratta, S. E. Strigin, A. Strunk, R. Sturani, A. L. Stuver, T. Z. Summerscales, L. Sun, S. Sunil, J. Suresh, P. J. Sutton, B. L. Swinkels, M. J. Szczepańczyk, M. Tacca, S. C. Tait, C. Talbot, D. Talukder, D. B. Tanner, M. Tápai, A. Taracchini, J. D. Tasson, J. A. Taylor, R. Taylor, S. V. Tewari, T. Theeg, F. Thies, E. G. Thomas, M. Thomas, P. Thomas, K. A. Thorne, K. S. Thorne, E. Thrane, S. Tiwari, V. Tiwari, K. V. Tokmakov, K. Toland, M. Tonelli, Z. Tornasi, A. Torres-Forné, C. I. Torrie, D. Töyrä, F. Travasso, G. Traylor, J. Trinastic, M. C. Tringali, L. Trozzo, K. W. Tsang, M. Tse, R. Tso, L. Tsukada, D. Tsuna, D. Tuyenbayev, K. Ueno, D. Ugolini, C. S. Unnikrishnan, A. L. Urban, S. A. Usman, H. Vahlbruch, G. Vajente, G. Valdes, M. Vallisneri, N. van Bakel, M. van Beuzekom, J. F. J. van den Brand, C. Van Den Broeck, D. C. Vander-Hyde, L. van der Schaaf, J. V. van Heijningen, A. A. van Veggel, M. Vardaro, V. Varma, S. Vass, M. Vasúth, A. Vecchio, G. Vedovato, J. Veitch, P. J. Veitch, K. Venkateswara, G. Venugopalan, D. Verkindt, F. Vetrano, A. Viceré, A. D. Viets, S. Vinciguerra, D. J. Vine, J.-Y. Vinet, S. Vitale, T. Vo, H. Vocca, C. Vorvick, S. P. Vyatchanin, A. R. Wade, L. E. Wade, M. Wade, R. Walet, M. Walker, L. Wallace, S. Walsh, G. Wang, H. Wang, J. Z. Wang, W. H. Wang, Y. F. Wang, R. L. Ward, J. Warner, M. Was, J. Watchi, B. Weaver, L.-W. Wei, M. Weinert, A. J. Weinstein, R. Weiss, L. Wen, E. K. Wessel, P. Weßels, J. Westerweck, T. Westphal, K. Wette, J. T. Whelan, S. E. Whitcomb, B. F. Whiting, C. Whittle, D. Wilken, D. Williams, R. D. Williams, A. R. Williamson, J. L. Willis, B. Willke, M. H. Wimmer, W. Winkler, C. C. Wipf, H. Wittel, G. Woan, J. Woehler, J. Wofford, K. W. K. Wong, J. Worden, J. L. Wright, D. S. Wu, D. M. Wysocki, S. Xiao, H. Yamamoto, C. C. Yancey, L. Yang, M. J. Yap, M. Yazback, Hang Yu, Haocun Yu, M. Yvert, A. Zadrożny, M. Zanolin, T. Zelenova, J.-P. Zendri, M. Zevin, L. Zhang, M. Zhang, T. Zhang, Y.-H. Zhang, C. Zhao, M. Zhou, Z. Zhou, S. J. Zhu, X. J. Zhu, A. B. Zimmerman, M. E. Zucker, and J. Zweizig and. GW170817: Observation of gravitational waves from a binary

- neutron star inspiral. Physical Review Letters, 119(16), oct 2017.
- [5] Marcelle Soares-Santos, DE Holz, J Annis, R Chornock, K Herner, Eric Berger, D Brout, H-Y Chen, R Kessler, M Sako, et al. The electromagnetic counterpart of the binary neutron star merger ligo/virgo gw170817. i. discovery of the optical counterpart using the dark energy camera. The Astrophysical Journal Letters, 848(2):L16, 2017.
- [6] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, V. B. Adya, C. Affeldt, M. Agathos, K. Agatsuma, N. Aggarwal, O. D. Aguiar, L. Aiello, A. Ain, P. Ajith, T. Akutsu, G. Allen, A. Allocata, M. A. Aloy, P. A. Altin, A. Amato, A. Ananyeva, S. B. Anderson, W. G. Anderson, M. Ando, S. V. Angelova, S. Antier, S. Appert, K. Arai, Koya Arai, Y. Arai, S. Araki, A. Araya, M. C. Araya, J. S. Areeda, M. Arène, N. Aritomi, N. Arnaud, K. G. Arun, S. Ascenzi, G. Ashton, Y. Aso, S. M. Aston, P. Astone, F. Aubin, P. Aufmuth, K. AultO-Neal, C. Austin, V. Avendano, A. Avila-Alvarez, S. Babak, P. Bacon, F. Badaracco, M. K. M. Bader, S. W. Bae, Y. B. Bae, L. Baiotti, R. Bajpai, P. T. Baker, F. Baldaccini, G. Ballardin, S. W. Ballmer, S. Banagiri, J. C. Barayoga, S. E. Barclay, B. C. Barish, D. Barker, K. Barkett, S. Barnum, F. Barone, B. Barr, L. Barsotti, M. Barsuglia, D. Barta, J. Bartlett, M. A. Barton, I. Bartos, R. Bassiri, A. Basti, M. Bawaj, J. C. Bayley, M. Bazzan, B. Bécsy, M. Bejger, I. Belahcene, A. S. Bell, D. Beniwal, B. K. Berger, G. Bergmann, S. Bernuzzi, J. J. Bero, C. P. L. Berry, D. Bersanetti, A. Bertolini, J. Betzwieser, R. Bhandare, J. Bidler, I. A. Bilenko, S. A. Bilgili, G. Billingsley, J. Birch, R. Birney, O. Birnholtz, S. Biscans, S. Biscoveanu, A. Bisht, M. Bitossi, M. A. Bizouard, J. K. Blackburn, C. D. Blair, D. G. Blair, R. M. Blair, S. Bloemen, N. Bode, M. Boer, Y. Boetzel, G. Bogaert, F. Bondu, E. Bonilla, R. Bonnand, P. Booker, B. A. Boom, C. D. Booth, R. Bork, V. Boschi, S. Bose, K. Bossie, V. Bossilkov, J. Bosveld, Y. Bouffanais, A. Bozzi, C. Bradascia, P. R. Brady, A. Bramley, M. Branchesi, J. E. Brau, T. Briant, J. H. Briggs, F. Brighenti, A. Brillet, M. Brinkmann, V. Brisson, P. Brockill, A. F. Brooks, D. A. Brown, D. D. Brown, S. Brunett, A. Buikema, T. Bulik, H. J. Bulten, A. Buonanno, D. Buskulic, C. Buy, R. L. Byer, M. Cabero, L. Cadonati, G. Cagnoli, C. Cahillane, J. Calderón Bustillo, T. A. Callister, E. Calloni, J. B. Camp, W. A. Campbell, M. Canepa, K. Cannon, K. C. Cannon, H. Cao, J. Cao, E. Capocasa, F. Carbognani, S. Caride, M. F. Carney, G. Carullo, J. Casanueva Diaz, C. Casentini, S. Caudill, M. Cavaglià, F. Cavalier, R. Cavalieri, G. Celli, P. Cerdá-Durán, G. Cerretani, E. Cesarini, O. Chaibi, K. Chakravarti, S. J. Chamberlin, M. Chan, M. L. Chan, S. Chao, P. Charlton, E. A. Chase, E. Chassande-Mottin, D. Chatterjee, M. Chaturvedi, K. Chatzioannou, B. D. Cheeseboro, C. S. Chen, H. Y. Chen, K. H. Chen, X. Chen, Y. Chen, Y. R. Chen, H.-P. Cheng, C. K. Cheong, H. Y. Chia, A. Chinca-

rini, A. Chiummo, G. Cho, H. S. Cho, M. Cho, N. Christensen, H. Y. Chu, Q. Chu, Y. K. Chu, S. Chua, K. W. Chung, S. Chung, G. Ciani, A. A. Ciobanu, R. Ciolfi, F. Cipriano, A. Cirone, F. Clara, J. A. Clark, P. Clearwater, F. Cleva, C. Cocchieri, E. Coccia, P.-F. Cohadon, D. Cohen, R. Colgan, M. Colleoni, C. G. Collette, C. Collins, L. R. Cominsky, M. Constancio, L. Conti, S. J. Cooper, P. Corban, T. R. Corbitt, I. Cordero-Carrión, K. R. Corley, N. Cornish, A. Corsi, S. Cortese, C. A. Costa, R. Cotesta, M. W. Coughlin, S. B. Coughlin, J.-P. Coulon, S. T. Countryman, P. Couvares, P. B. Covas, E. E. Cowan, D. M. Coward, M. J. Cowart, D. C. Coyne, R. Coyne, J. D. E. Creighton, T. D. Creighton, J. Cripe, M. Croquette, S. G. Crowder, T. J. Cullen, A. Cumming, L. Cunningham, E. Cuoco, T. Dal Canton, G. Dálya, S. L. Danilishin, S. D'Antonio, K. Danzmann, A. Dasgupta, C. F. Da Silva Costa, L. E. H. Datrier, V. Dattilo, I. Dave, M. Davier, D. Davis, E. J. Daw, D. DeBra, M. Deenadayalan, J. Degallaix, M. De Laurentis, S. Deléglise, W. Del Pozzo, L. M. DeMarchi, N. Demos, T. Dent, R. De Pietri, J. Derby, R. De Rosa, C. De Rossi, R. DeSalvo, O. de Varona, S. Dhurandhar, M. C. Díaz, T. Dietrich, L. Di Fiore, M. Di Giovanni, T. Di Girolamo, A. Di Lieto, B. Ding, S. Di Pace, I. Di Palma, F. Di Renzo, A. Dmitriev, Z. Doctor, K. Doi, F. Donovan, K. L. Dooley, S. Doravari, I. Dorrington, T. P. Downes, M. Drago, J. C. Driggers, Z. Du, J.-G. Ducoin, P. Dupej, S. E. Dwyer, P. J. Easter, T. B. Edo, M. C. Edwards, A. Effler, S. Eguchi, P. Ehrens, J. Eichholz, S. S. Eikenberry, M. Eisenmann, R. A. Eisenstein, Y. Enomoto, R. C. Essick, H. Estelles, D. Estevez, Z. B. Etienne, T. Etzel, M. Evans, T. M. Evans, V. Fafone, H. Fair, S. Fairhurst, X. Fan, S. Farinon, B. Farr, W. M. Farr, E. J. Fauchon-Jones, M. Favata, M. Fays, M. Fazio, C. Fee, J. Feicht, M. M. Fejer, F. Feng, A. Fernandez-Galiana, I. Ferrante, E. C. Ferreira, T. A. Ferreira, F. Ferrini, F. Fidecaro, I. Fiori, D. Fiorucci, M. Fishbach, R. P. Fisher, J. M. Fishner, M. Fitz-Axen, R. Flaminio, M. Fletcher, E. Flynn, H. Fong, J. A. Font, P. W. F. Forsyth, J.-D. Fournier, S. Frasca, F. Frasconi, Z. Frei, A. Freise, R. Frey, V. Frey, P. Fritschel, V. V. Frolov, Y. Fujii, M. Fukunaga, M. Fukushima, P. Fulda, M. Fyffe, H. A. Gabbard, B. U. Gadre, S. M. Gaebel, J. R. Gair, L. Gammaitoni, M. R. Ganija, S. G. Gaonkar, A. Garcia, C. García-Quirós, F. Garufi, B. Gateley, S. Gaudio, G. Gaur, V. Gayathri, G. G. Ge, G. Gemme, E. Genin, A. Gennai, D. George, J. George, L. Gergely, V. Germain, S. Ghonge, Abhirup Ghosh, Archisman Ghosh, S. Ghosh, B. Giacomazzo, J. A. Giaime, K. D. Giardina, A. Giazotto, K. Gill, G. Giordano, L. Glover, P. Godwin, E. Goetz, R. Goetz, B. Goncharov, G. González, J. M. Gonzalez Castro, A. Gopakumar, M. L. Gorodetsky, S. E. Gossan, M. Gosselin, R. Gouaty, A. Grado, C. Graef, M. Granata, A. Grant, S. Gras, P. Grassia, C. Gray, R. Gray, G. Greco, A. C. Green, R. Green, E. M. Gretarsson, P. Groot, H. Grote, S. Grunewald, P. Gruning, G. M. Guidi, H. K. Gulati, Y. Guo, A. Gupta, M. K. Gupta, E. K. Gustafson, R. Gustafson, L. Haegel, A. Hagiwara, S. Haino, O. Halim, B. R. Hall, E. D. Hall, E. Z. Hamil-

ton, G. Hammond, M. Haney, M. M. Hanke, J. Hanks, C. Hanna, M. D. Hannam, O. A. Hannuksela, J. Hanson, T. Hardwick, K. Haris, J. Harms, G. M. Harry, I. W. Harry, K. Hasegawa, C.-J. Haster, K. Haughian, H. Hayakawa, K. Hayama, F. J. Hayes, J. Healy, A. Heidmann, M. C. Heintze, H. Heitmann, P. Hello, G. Hemming, M. Hendry, I. S. Heng, J. Hennig, A. W. Heptonstall, M. Heurs, S. Hild, Y. Himmoto, T. Hinderer, Y. Hiranuma, N. Hirata, E. Hirose, D. Hoak, S. Hochheim, D. Hofman, A. M. Holgado, N. A. Holland, K. Holt, D. E. Holz, Z. Hong, P. Hopkins, C. Horst, J. Hough, E. J. Howell, C. G. Hoy, A. Hreibi, B. H. Hsieh, G. Z. Huang, P. W. Huang, Y. J. Huang, E. A. Huerta, D. Huet, B. Hughey, M. Hulkó, S. Husa, S. H. Huttner, T. Huynh-Dinh, B. Idzkowski, A. Iess, B. Ikenoue, S. Imam, K. Inayoshi, C. Ingram, Y. Inoue, R. Inta, G. Intini, K. Ioka, B. Irwin, H. N. Isa, J.-M. Isac, M. Isi, Y. Itoh, B. R. Iyer, K. Izumi, T. Jacqmin, S. J. Jadhav, K. Jani, N. N. Janthalur, P. Jaradowski, A. C. Jenkins, J. Jiang, D. S. Johnson, A. W. Jones, D. I. Jones, R. Jones, R. J. G. Jonker, L. Ju, K. Jung, P. Jung, J. Junker, T. Kajita, C. V. Kalaghatgi, V. Kalogera, B. Kamai, M. Kamiizumi, N. Kanda, S. Kandhasamy, G. W. Kang, J. B. Kanner, S. J. Kapadia, S. Karki, K. S. Karvinen, R. Kashyap, M. Kasprzack, S. Katsanevas, E. Katsavounidis, W. Katzman, S. Kaufer, K. Kawabe, K. Kawaguchi, N. Kawai, T. Kawasaki, N. V. Keerthana, F. Kéfélian, D. Keitel, R. Kennedy, J. S. Key, F. Y. Khalili, H. Khan, I. Khan, S. Khan, Z. Khan, E. A. Khazanov, M. Khursheed, N. Kijbunchoo, Chunglee Kim, C. Kim, J. C. Kim, J. Kim, K. Kim, W. Kim, W. S. Kim, Y.-M. Kim, C. Kimball, N. Kimura, E. J. King, P. J. King, M. Kinley-Hanlon, R. Kirchhoff, J. S. Kissel, N. Kita, H. Kitazawa, L. Kleybolte, J. H. Klika, S. Klimenko, T. D. Knowles, E. Knyazev, P. Koch, S. M. Koehlenbeck, G. Koekoek, Y. Kojima, K. Kokeyama, S. Koley, K. Komori, V. Kondrashov, A. K. H. Kong, A. Kontos, N. Koper, M. Korobko, W. Z. Korth, K. Kotake, I. Kowalska, D. B. Kozak, C. Kozakai, R. Kozu, V. Kringle, N. Krishnendu, A. Królak, G. Kuehn, A. Kumar, P. Kumar, Rahul Kumar, R. Kumar, S. Kumar, J. Kume, C. M. Kuo, H. S. Kuo, L. Kuo, S. Kuroyanagi, K. Kusayanagi, A. Kutytnia, K. Kwak, S. Kwang, B. D. Lackey, K. H. Lai, T. L. Lam, M. Landry, B. B. Lane, R. N. Lang, J. Lange, B. Lantz, R. K. Lanza, A. Lartaux-Vollard, P. D. Lasky, M. Laxen, A. Lazzarini, C. Lazzaro, P. Leaci, S. Leavey, Y. K. Lecoeuche, C. H. Lee, H. K. Lee, H. M. Lee, H. W. Lee, J. Lee, K. Lee, R. K. Lee, J. Lehmann, A. Lenon, M. Leonardi, N. Leroy, N. Letendre, Y. Levin, J. Li, K. J. L. Li, T. G. F. Li, X. Li, C. Y. Lin, F. Lin, F. L. Lin, L. C. C. Lin, F. Linde, S. D. Linker, T. B. Littenberg, G. C. Liu, J. Liu, X. Liu, R. K. L. Lo, N. A. Lockerbie, L. T. London, A. Longo, M. Lorenzini, V. Loriette, M. Lormand, G. Losurdo, J. D. Lough, C. O. Lousto, G. Lovelace, M. E. Lower, H. Lück, D. Lumaca, A. P. Lundgren, L. W. Luo, R. Lynch, Y. Ma, R. Macas, S. Macfoy, M. MacInnis, D. M. Macleod, A. Macquet, F. Magaña-Sandoval, L. Magaña Zertuche, R. M. Magee, E. Majorana, I. Maksimo-

vic, A. Malik, N. Man, V. Mandic, V. Mangano, G. L. Mansell, M. Manske, M. Mantovani, F. Marchesoni, M. Marchio, F. Marion, S. Márka, Z. Márka, C. Markakis, A. S. Markosyan, A. Markowitz, E. Maros, A. Marquina, S. Marsat, F. Martelli, I. W. Martin, R. M. Martin, D. V. Martynov, K. Mason, E. Massera, A. Masserot, T. J. Massinger, M. Masso-Reid, S. Mastrogiovanni, A. Matas, F. Matichard, L. Matone, N. Mavalvala, N. Mazumder, J. J. McCann, R. McCarthy, D. E. McClelland, S. McCormick, L. McCuller, S. C. McGuire, J. McIver, D. J. McManus, T. McRae, S. T. McWilliams, D. Meacher, G. D. Meadors, M. Mehmet, A. K. Mehta, J. Meidam, A. Melatos, G. Mendell, R. A. Mercer, L. Mereni, E. L. Merilh, M. Merzougui, S. Meshkov, C. Messenger, C. Messick, R. Metzdorff, P. M. Meyers, H. Miao, C. Michel, Y. Michimura, H. Middleton, E. E. Mikhailov, L. Milano, A. L. Miller, A. Miller, M. Millhouse, J. C. Mills, M. C. Milovich-Goff, O. Minazzoli, Y. Minenkov, N. Mio, A. Mishkin, C. Mishra, T. Mistry, S. Mitra, V. P. Mitrofanov, G. Mitselmakher, R. Mittleman, O. Miyakawa, A. Miyamoto, Y. Miyazaki, K. Miyo, S. Miyoki, G. Mo, D. Moffa, K. Mogushi, S. R. P. Mohapatra, M. Montani, C. J. Moore, D. Moraru, G. Moreno, S. Morisaki, Y. Moriwaki, B. Mouris, C. M. Mow-Lowry, Arunava Mukherjee, D. Mukherjee, S. Mukherjee, N. Mukund, A. Mullavey, J. Munch, E. A. Muñiz, M. Muratore, P. G. Murray, K. Nagano, S. Nagano, A. Nagar, K. Nakamura, H. Nakano, M. Nakano, R. Nakashima, I. Nardecchia, T. Narikawa, L. Naticchioni, R. K. Nayak, R. Negishi, J. Neilson, G. Nelemans, T. J. N. Nelson, M. Nery, A. Neunzert, K. Y. Ng, S. Ng, P. Nguyen, W. T. Ni, D. Nichols, A. Nishizawa, S. Nissanke, F. Nocera, C. North, L. K. Nuttall, M. Obergaulinger, J. Oberling, B. D. O'Brien, Y. Obuchi, G. D. O'Dea, W. Ogaki, G. H. Ogin, J. J. Oh, S. H. Oh, M. Ohashi, N. Ohishi, M. Ohkawa, F. Ohme, H. Ohta, M. A. Okada, K. Okutomi, M. Oliver, K. Oohara, C. P. Ooi, P. Oppermann, Richard J. Oram, B. O'Reilly, R. G. Ormiston, L. F. Ortega, R. O'Shaughnessy, S. Oshino, S. Ossokine, D. J. Ottaway, H. Overmier, B. J. Owen, A. E. Pace, G. Pagano, M. A. Page, A. Pai, S. A. Pai, J. R. Palamos, O. Palashov, C. Palomba, A. Pal-Singh, Huang-Wei Pan, K. C. Pan, B. Pang, H. F. Pang, P. T. H. Pang, C. Pankow, F. Pannarale, B. C. Pant, F. Paoletti, A. Paoli, M. A. Papa, A. Parida, J. Park, W. Parker, D. Pascucci, A. Pasqualetti, R. Passaquieti, D. Passuello, M. Patil, B. Patricelli, B. L. Pearlstone, C. Pedersen, M. Pedraza, R. Pedurand, A. Pele, F. E. Peña Arellano, S. Penn, C. J. Perez, A. Perreca, H. P. Pfeiffer, M. Phelps, K. S. Phukon, O. J. Piccinni, M. Pichot, F. Piergiovanni, G. Pillant, L. Pinard, I. Pinto, M. Pirello, M. Pitkin, R. Poggiani, D. Y. T. Pong, S. Ponrathnam, P. Popolizio, E. K. Porter, J. Powell, A. K. Prajapati, J. Prasad, K. Prasai, R. Prasanna, G. Pratten, T. Prestegard, S. Privitera, G. A. Prodi, L. G. Prokhorov, O. Puncken, M. Punturo, P. Puppo, M. Pürrer, H. Qi, V. Quetschke, P. J. Quinonez, E. A. Quintero, R. Quitzow-James, F. J. Raab, H. Radkins, N. Radulescu, P. Raffai, S. Raja, C. Rajan, B. Rajbhandari, M. Rakhmanov, K. E. Ramirez, A. Ramos-

Buades, Javed Rana, K. Rao, P. Rapagnani, V. Raymond, M. Razzano, J. Read, T. Regimbau, L. Rei, S. Reid, D. H. Reitze, W. Ren, F. Ricci, C. J. Richardson, J. W. Richardson, P. M. Ricker, K. Riles, M. Rizzo, N. A. Robertson, R. Robie, F. Robinet, A. Rocchi, L. Rolland, J. G. Rollins, V. J. Roma, M. Romanelli, R. Romano, C. L. Romel, J. H. Romie, K. Rose, D. Rosińska, S. G. Rosofsky, M. P. Ross, S. Rowan, A. Rüdiger, P. Ruggi, G. Rutins, K. Ryan, S. Sachdev, T. Sadecki, N. Sago, S. Saito, Y. Saito, K. Sakai, Y. Sakai, H. Sakamoto, M. Sakellariadou, Y. Sakuno, L. Salconi, M. Saleem, A. Samajdar, L. Sammut, E. J. Sanchez, L. E. Sanchez, N. Sanchis-Gual, V. Sandberg, J. R. Sanders, K. A. Santiago, N. Sarin, B. Sassolas, B. S. Sathyaparakash, S. Sato, T. Sato, O. Sauter, R. L. Savage, T. Sawada, P. Schale, M. Scheel, J. Scheuer, P. Schmidt, R. Schnabel, R. M. S. Schofield, A. Schönbeck, E. Schreiber, B. W. Schulte, B. F. Schutz, S. G. Schwalbe, J. Scott, S. M. Scott, E. Seidel, T. Sekiguchi, Y. Sekiguchi, D. Sellers, A. S. Sengupta, N. Sennett, D. Sentenac, V. Sequino, A. Sergeev, Y. Setyawati, D. A. Shaddock, T. Shaffer, M. S. Shahriar, M. B. Shaner, L. Shao, P. Sharma, P. Shawhan, H. Shen, S. Shibagaki, R. Shimizu, T. Shimoda, K. Shimode, R. Shink, H. Shinkai, T. Shishido, A. Shoda, D. H. Shoemaker, D. M. Shoemaker, S. ShyamSundar, K. Siellez, M. Sieniawska, D. Sigg, A. D. Silva, L. P. Singer, N. Singh, A. Singhal, A. M. Sintes, S. Sitmukhambetov, V. Skliris, B. J. J. Slagmolen, T. J. Slaven-Blair, J. R. Smith, R. J. E. Smith, S. Somalia, K. Somiya, E. J. Son, B. Sorazu, F. Sorrentino, H. Sotani, T. Souradeep, E. Sowell, A. P. Spencer, A. K. Srivastava, V. Srivastava, K. Staats, C. Stachie, M. Standke, D. A. Steer, M. Steinke, J. Steinlechner, S. Steinlechner, D. Steinmeyer, S. P. Stevenson, D. Stocks, R. Stone, D. J. Stops, K. A. Strain, G. Stratta, S. E. Stringin, A. Strunk, R. Sturani, A. L. Stuver, V. Sudhir, R. Sugimoto, T. Z. Summerscales, L. Sun, S. Sunil, J. Suresh, P. J. Sutton, Takamasa Suzuki, Toshikazu Suzuki, B. L. Swinkels, M. J. Szczepańczyk, M. Tacca, H. Tagoshi, S. C. Tait, H. Takahashi, R. Takahashi, A. Takamori, S. Takano, H. Takeda, M. Takeda, C. Talbot, D. Talukder, H. Tanaka, Kazuyuki Tanaka, Kenta Tanaka, Taiki Tanaka, Takahiro Tanaka, S. Tanioka, D. B. Tanner, M. Tápai, E. N. Tapia San Martin, A. Taracchini, J. D. Tasson, R. Taylor, S. Telada, F. Thies, M. Thomas, P. Thomas, S. R. Thondapu, K. A. Thorne, E. Thrane, Shubhanshu Tiwari, Srishti Tiwari, V. Tiwari, K. Toland, T. Tomaru, Y. Tomigami, T. Tomura, M. Tonelli, Z. Tornasi, A. Torres-Forné, C. I. Torrie, D. Töyrä, F. Travasso, G. Traylor, M. C. Tringali, A. Trovato, L. Trozzo, R. Trudeau, K. W. Tsang, T. T. L. Tsang, M. Tse, R. Tso, K. Tsubono, S. Tsuchida, L. Tsukada, D. Tsuna, T. Tsuzuki, D. Tuyenbayev, N. Uchikata, T. Uchiyama, A. Ueda, T. Uehara, K. Ueno, G. Ueshima, D. Ugolini, C. S. Unnikrishnan, F. Uraguchi, A. L. Urban, T. Ushiba, S. A. Usman, H. Vahlbruch, G. Vajente, G. Valdes, N. van Bakel, M. van Beuzekom, J. F. J. van den Brand, C. Van Den Broeck, D. C. Vander-Hyde, L. van der Schaaf, J. V. van Heijningen, M. H. P. M. van Putten, A. A. van Veggel,

- M. Vardaro, V. Varma, S. Vass, M. Vasíth, A. Vecchio, G. Vedovato, J. Veitch, P. J. Veitch, K. Venkateswara, G. Venugopalan, D. Verkindt, F. Vetrano, A. Viceré, A. D. Viets, D. J. Vine, J.-Y. Vinet, S. Vitale, Francisco Hernandez Vivanco, T. Vo, H. Vocca, C. Vorvick, S. P. Vyatchanin, A. R. Wade, L. E. Wade, M. Wade, R. Wallet, M. Walker, L. Wallace, S. Walsh, G. Wang, H. Wang, J. Wang, J. Z. Wang, W. H. Wang, Y. F. Wang, R. L. Ward, Z. A. Warden, J. Warner, M. Was, J. Watchi, B. Weaver, L.-W. Wei, M. Weinert, A. J. Weinstein, R. Weiss, F. Wellmann, L. Wen, E. K. Wessel, P. Weßels, J. W. Westhouse, K. Wette, J. T. Whelan, B. F. Whiting, C. Whittle, D. M. Wilken, D. Williams, A. R. Williamson, J. L. Willis, B. Willke, M. H. Wimmer, W. Winkler, C. C. Wipf, H. Wittel, G. Woan, J. Woehler, J. K. Wofford, J. Worden, J. L. Wright, C. M. Wu, D. S. Wu, H. C. Wu, S. R. Wu, D. M. Wysocki, L. Xiao, W. R. Xu, T. Yamada, H. Yamamoto, Kazuhiro Yamamoto, Kohei Yamamoto, T. Yamamoto, C. C. Yancey, L. Yang, M. J. Yap, M. Yazback, D. W. Yeeles, K. Yokogawa, J. Yokoyama, T. Yokozawa, T. Yoshioka, Hang Yu, Haocun Yu, S. H. R. Yuen, H. Yuzurihara, M. Yvert, A. K. Zadrożny, M. Zanolin, S. Zeidler, T. Zelenova, J.-P. Zendri, M. Zevin, J. Zhang, L. Zhang, T. Zhang, C. Zhao, Y. Zhao, M. Zhou, Z. Zhou, X. J. Zhu, Z. H. Zhu, A. B. Zimmerman, M. E. Zucker, and J. Zweizig and. Prospects for observing and localizing gravitational-wave transients with advanced LIGO, advanced virgo and KAGRA. *Living Reviews in Relativity*, 23(1), sep 2020.
- [7] B. Flaugher, H. T. Diehl, K. Honscheid, T. M. C. Abbott, O. Alvarez, R. Angstadt, J. T. Annis, M. Antonik, O. Ballester, L. Beaufre, G. M. Bernstein, R. A. Bernstein, B. Bigelow, M. Bonati, D. Boprie, D. Brooks, E. J. Buckley-Geer, J. Campa, L. Cardiel-Sas, F. J. Castander, J. Castilla, H. Cease, J. M. Cela-Ruiz, S. Chappa, E. Chi, C. Cooper, L. N. da Costa, E. Dede, G. Derylo, D. L. DePoy, J. de Vicente, P. Doel, A. Drlica-Wagner, J. Eiting, A. E. Elliott, J. Emes, J. Estrada, A. Fausti Neto, D. A. Finley, R. Flores, J. Frieman, D. Gerdes, M. D. Gladders, B. Gregory, G. R. Gutierrez, J. Hao, S. E. Holland, S. Holm, D. Huffman, C. Jackson, D. J. James, M. Jonas, A. Karcher, I. Karliner, S. Kent, R. Kessler, M. Kozlovsky, R. G. Kron, D. Kubik, K. Kuehn, S. Kuhlmann, K. Kuk, O. Lahav, A. Lathrop, J. Lee, M. E. Levi, P. Lewis, T. S. Li, I. Mandrichenko, J. L. Marshall, G. Martinez, K. W. Merritt, R. Miquel, F. Muñoz, E. H. Neilsen, R. C. Nichol, B. Nord, R. Ogando, J. Olsen, N. Palaio, K. Patton, J. Peoples, A. A. Plazas, J. Rauch, K. Reil, J. P. Rheault, N. A. Roe, H. Rogers, A. Roodman, E. Sanchez, V. Scarpine, R. H. Schindler, R. Schmidt, R. Schmitt, M. Schubnell, K. Schultz, P. Schurter, L. Scott, S. Serrano, T. M. Shaw, R. C. Smith, M. Soares-Santos, A. Stefanik, W. Stuermer, E. Suchyta, A. Sypniewski, G. Tarle, J. Thaler, R. Tighe, C. Tran, D. Tucker, A. R. Walker, G. Wang, M. Watson, C. Weaverdyck, W. Wester, R. Woods, B. Yanny, and DES Collaboration. The Dark Energy Camera. , 150(5):150, November 2015.

- [8] C Mendes de Oliveira, T Ribeiro, W Schoenell, A Kanaan, R A Overzier, A Molino, L Sampedro, P Coelho, C E Barbosa, A Cortesi, M V Costa-Duarte, F R Herpich, J A Hernandez-Jimenez, V M Placco, H S Xavier, L R Abramo, R K Saito, A L Chies-Santos, A Ederoclite, R Lopes de Oliveira, D R Gonçalves, S Akras, L A Almeida, F Almeida-Fernandes, T C Beers, C Bonatto, S Bonoli, E S Cypriano, E Vinicius-Lima, R S de Souza, G Fabiano de Souza, F Ferrari, T S Gonçalves, A H Gonzalez, L A Gutiérrez-Soto, E A Hartmann, Y Jaffe, L O Kerber, C Lima-Dias, P A A Lopes, K Menendez-Delmestre, L M I Nakazono, P M Novais, R A Ortega-Minakata, E S Pereira, H D Perottoni, C Queiroz, R R R Reis, W A Santos, T Santos-Silva, R M Santucci, C L Barbosa, Beatriz B Siffert, L Sodré, S Torres-Flores, P Westera, D D Whitten, J S Alcaniz, Javier Alonso-García, S Alencar, A Alvarez-Candal, P Amram, L Azanha, R H Barbá, P H Bernardinelli, M Borges Fernandes, V Branco, D Brito-Silva, M L Buzzo, J Caffer, A Campillay, Z Cano, J M Carvano, M Castejon, R Cid Fernandes, M L L Dantas, S Daflon, G Damke, R de la Reza, L J de Melo de Azevedo, D F De Paula, K G Diem, R Donnerstein, O L Dors, R Dupke, S Eikenberry, Carlos G Escudero, Favio R Faifer, H Farías, B Fernandes, C Fernandes, S Fontes, A Galarza, N S T Hirata, L Katena, J Gregorio-Hetem, J D Hernández-Fernández, L Izzo, M Jaque Arancibia, V Jatenco-Pereira, Y Jiménez-Teja, D A Kann, A C Krabbe, C Labayru, D Lazzaro, G B Lima Neto, Amanda R Lopes, R Magalhães, M Makler, R de Menezes, J Miralda-Escudé, R Monteiro-Oliveira, A D Montero-Dorta, N Muñoz-Elgueta, R S Nemmen, J L Nilo Castellón, A S Oliveira, D Ortíz, E Pattaro, C B Pereira, B Quint, L Riguccini, H J Rocha Pinto, I Rodrigues, F Roig, S Rossi, Kanak Saha, R Santos, A Schnorr Müller, Leandro A Sesto, R Silva, Analia V Smith Castelli, R Teixeira, E Telles, R C Thom de Souza, C Thöne, M Trevisan, A de Ugarte Postigo, F Urrutia-Viscarra, C H Veiga, M Vika, A Z Vitorelli, A Werle, S V Werner, and D Zaritsky. The southern photometric local universe survey (s-PLUS): improved SEDs, morphologies, and redshifts with 12 optical filters. *Monthly Notices of the Royal Astronomical Society*, 489(1):241–267, aug 2019.
- [9] Dark Energy Survey Collaboration, T. Abbott, F. B. Abdalla, J. Aleksić, S. Allam, A. Amara, D. Bacon, E. Balbinot, M. Banerji, K. Bechtol, A. Benoit-Lévy, G. M. Bernstein, E. Bertin, J. Blazek, C. Bennett, S. Bridle, D. Brooks, R. J. Brunner, E. Buckley-Geer, D. L. Burke, G. B. Caminha, D. Capozzi, J. Carlsen, A. Carnero-Rosell, M. Carollo, M. Carrasco-Kind, J. Carretero, F. J. Castander, L. Clerkin, T. Collett, C. Conselice, M. Crocce, C. E. Cunha, C. B. D’Andrea, L. N. da Costa, T. M. Davis, S. Desai, H. T. Diehl, J. P. Dietrich, S. Dodelson, P. Doel, A. Drlica-Wagner, J. Estrada, J. Etherington, A. E. Evrard, J. Fabbri, D. A. Finley, B. Flaugher, R. J. Foley, P. Fosalba, J. Frieman, J. García-Bellido, E. Gaztanaga, D. W. Gerdes, T. Giannantonio, D. A. Goldstein, D. Gruen, R. A.

- Gruendl, P. Guarnieri, G. Gutierrez, W. Hartley, K. Honscheid, B. Jain, D. J. James, T. Jeltema, S. Jouvel, R. Kessler, A. King, D. Kirk, R. Kron, K. Kuehn, N. Kuropatkin, O. Lahav, T. S. Li, M. Lima, H. Lin, M. A. G. Maia, M. Makler, M. Manera, C. Maraston, J. L. Marshall, P. Martini, R. G. McMahon, P. Melchior, A. Merson, C. J. Miller, R. Miquel, J. J. Mohr, X. Morice-Atkinson, K. Naidoo, E. Neilsen, R. C. Nichol, B. Nord, R. Ogando, F. Ostrovski, A. Palmese, A. Papadopoulos, H. V. Peiris, J. Peoples, W. J. Percival, A. A. Plazas, S. L. Reed, A. Refregier, A. K. Romer, A. Roodman, A. Ross, E. Rozo, E. S. Rykoff, I. Sadeh, M. Sako, C. Sánchez, E. Sanchez, B. Santiago, V. Scarpine, M. Schubnell, I. Sevilla-Noarbe, E. Sheldon, M. Smith, R. C. Smith, M. Soares-Santos, F. Sobreira, M. Soumagnac, E. Suchyta, M. Sullivan, M. Swanson, G. Tarle, J. Thaler, D. Thomas, R. C. Thomas, D. Tucker, J. D. Vieira, V. Vikram, A. R. Walker, R. H. Wechsler, J. Weller, W. Wester, L. Whiteway, H. Wilcox, B. Yanny, Y. Zhang, and J. Zuntz. The Dark Energy Survey: more than dark energy - an overview. , 460(2):1270–1299, August 2016.
- [10] Joseph J. Mohr, Robert Armstrong, Emmanuel Bertin, Greg Daues, Shantanu Desai, Michelle Gower, Robert Gruendl, William Hanlon, Nikolay Kuropatkin, Huan Lin, John Marriner, Donald Petracic, Ignacio Sevilla, Molly Swanson, Todd Tomashek, Douglas Tucker, and Brian Yanny. The Dark Energy Survey data processing and calibration system. In Nicole M. Radziwill and Gianluca Chiozzi, editors, Software and Cyberinfrastructure for Astronomy II, volume 8451, page 84510D. International Society for Optics and Photonics, SPIE, 2012.
- [11] S. Desai, R. Armstrong, J. J. Mohr, D. R. Semler, J. Liu, E. Bertin, S. S. Allam, W. A. Barkhouse, G. Bazin, E. J. Buckley-Geer, M. C. Cooper, S. M. Hansen, F. W. High, H. Lin, Y. T. Lin, C. C. Ngeow, A. Rest, J. Song, D. Tucker, and A. Zenteno. The Blanco Cosmology Survey: Data Acquisition, Processing, Calibration, Quality Diagnostics, and Data Release. , 757(1):83, September 2012.
- [12] E. Morganson, R. A. Gruendl, F. Menanteau, M. Carrasco Kind, Y. C. Chen, G. Daues, A. Drlica-Wagner, D. N. Friedel, M. Gower, M. W. G. Johnson, M. D. Johnson, R. Kessler, F. Paz-Chinchón, D. Petracick, C. Pond, B. Yanny, S. Allam, R. Armstrong, W. Barkhouse, K. Bechtol, A. Benoit-Lévy, G. M. Bernstein, E. Bertin, E. Buckley-Geer, R. Covarrubias, S. Desai, H. T. Diehl, D. A. Goldstein, D. Gruen, T. S. Li, H. Lin, J. Marriner, J. J. Mohr, E. Neilsen, C. C. Ngeow, K. Paech, E. S. Rykoff, M. Sako, I. Sevilla-Noarbe, E. Sheldon, F. Sobreira, D. L. Tucker, W. Wester, and DES Collaboration. The Dark Energy Survey Image Processing Pipeline. , 130(989):074501, July 2018.
- [13] R. Kessler, J. Marriner, M. Childress, R. Covarrubias, C. B. D’Andrea, D. A. Finley, J. Fischer, R. J. Foley, D. Goldstein, R. R. Gupta, K. Kuehn, M. Marcha, R. C.

- Nichol, A. Papadopoulos, M. Sako, D. Scolnic, M. Smith, M. Sullivan, W. Wester, F. Yuan, T. Abbott, F. B. Abdalla, S. Allam, A. Benoit-Lévy, G. M. Bernstein, E. Bertin, D. Brooks, A. Carnero Rosell, M. Carrasco Kind, F. J. Castander, M. Crocce, L. N. da Costa, S. Desai, H. T. Diehl, T. F. Eifler, A. Fausti Neto, B. Flaugher, J. Frieman, D. W. Gerdes, D. Gruen, R. A. Gruendl, K. Honscheid, D. J. James, N. Kuropatkin, T. S. Li, M. A. G. Maia, J. L. Marshall, P. Martini, C. J. Miller, R. Miquel, B. Nord, R. Ogando, A. A. Plazas, K. Reil, A. K. Romer, A. Roodman, E. Sanchez, I. Sevilla-Noarbe, R. C. Smith, M. Soares-Santos, F. Sobreira, G. Tarle, J. Thaler, R. C. Thomas, D. Tucker, A. R. Walker, and (The DES Collaboration). The difference imaging pipeline for the transient search in the dark energy survey. *The Astronomical Journal*, 150(6):172, nov 2015.
- [14] G. M. Bernstein, T. M. C. Abbott, R. Armstrong, D. L. Burke, H. T. Diehl, R. A. Gruendl, M. D. Johnson, T. S. Li, E. S. Rykoff, A. R. Walker, W. Wester, and B. Yanny. Photometric characterization of the dark energy camera. *Publications of the Astronomical Society of the Pacific*, 130(987):054501, apr 2018.
- [15] E. Bertin and S. Arnouts. SExtractor: Software for source extraction. , 117:393–404, June 1996.
- [16] E. Bertin. Automatic Astrometric and Photometric Calibration with SCAMP. In C. Gabriel, C. Arviset, D. Ponz, and S. Enrique, editors, *Astronomical Data Analysis Software and Systems XV*, volume 351 of *Astronomical Society of the Pacific Conference Series*, page 112, July 2006.
- [17] F. Almeida-Fernandes, L. SamPedro, F. R. Herpich, A. Molino, C. E. Barbosa, M. L. Buzzo, R. A. Overzier, E. V. R. de Lima, L. M. I. Nakazono, G. B. Oliveira Schwarz, H. D. Perottoni, G. F. Bolutavicius, L. A. Gutiérrez-Soto, T. Santos-Silva, A. Z. Vitorelli, A. Werle, D. D. Whitten, M. V. Costa Duarte, C. R. Bom, P. Coelho, L. Sodré, V. M. Placco, G. S. M. Teixeira, J. Alonso-García, C. L. Barbosa, T. C. Beers, C. J. Bonatto, A. L. Chies-Santos, E. A. Hartmann, R. Lopes de Oliveira, F. Navarete, A. Kanaan, T. Ribeiro, W. Schoenell, and C. Mendes de Oliveira. Data Release 2 of S-PLUS: Accurate template-fitting based photometry covering 1000 deg² in 12 optical filters. , 511(3):4590–4618, April 2022.
- [18] Doug Tody. The Iraf Data Reduction And Analysis System. In David L. Crawford, editor, *Instrumentation in Astronomy VI*, volume 0627, pages 733 – 748. International Society for Optics and Photonics, SPIE, 1986.
- [19] Emmanuel Bertin. SWarp: Resampling and Co-adding FITS Images Together. *Astrophysics Source Code Library*, record ascl:1010.068, October 2010.

- [20] Paul L. Schechter, Mario Mateo, and Abhijit Saha. DoPHOT, A CCD Photometry Program: Description and Tests. , 105:1342, November 1993.
- [21] Heather Flewelling. Pan-STARRS Data Release 2. In American Astronomical Society Meeting Abstracts #231, volume 231 of American Astronomical Society Meeting Abstracts, page 436.01, January 2018.
- [22] Christian Wolf, Christopher A. Onken, Lance C. Luvaal, Brian P. Schmidt, Michael S. Bessell, Seo-Won Chang, Gary S. Da Costa, Dougal Mackey, Tony Martin-Jones, Simon J. Murphy, and et al. Skymapper southern survey: First data release (dr1). Publications of the Astronomical Society of Australia, 35:e010, 2018.
- [23] T. M. C. Abbott, F. B. Abdalla, S. Allam, A. Amara, J. Annis, J. Asorey, S. Avila, O. Ballester, M. Banerji, W. Barkhouse, L. Baruah, M. Baumer, K. Bechtol, M. R. Becker, A. Benoit-Lé vy, G. M. Bernstein, E. Bertin, J. Blazek, S. Bocquet, D. Brooks, D. Brout, E. Buckley-Geer, D. L. Burke, V. Busti, R. Campisano, L. Cardiel-Sas, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, F. J. Castander, R. Cawthon, C. Chang, X. Chen, C. Conselice, G. Costa, M. Crocce, C. E. Cunha, C. B. D’Andrea, L. N. da Costa, R. Das, G. Dauves, T. M. Davis, C. Davis, J. De Vicente, D. L. DePoy, J. DeRose, S. Desai, H. T. Diehl, J. P. Dietrich, S. Dodelson, P. Doel, A. Drlica-Wagner, T. F. Eifler, A. E. Elliott, A. E. Evrard, A. Farahi, A. Fausti Neto, E. Fernandez, D. A. Finley, B. Flaugher, R. J. Foley, P. Fosalba, D. N. Friedel, J. Frieman, J. García-Bellido, E. Gaztanaga, D. W. Gerdes, T. Giannantonio, M. S. S. Gill, K. Glazebrook, D. A. Goldstein, M. Gower, D. Gruen, R. A. Gruendl, J. Gschwend, R. R. Gupta, G. Gutierrez, S. Hamilton, W. G. Hartley, S. R. Hinton, J. M. Hislop, D. Hollowood, K. Honscheid, B. Hoyle, D. Huterer, B. Jain, D. J. James, T. Jeltema, M. W. G. Johnson, M. D. Johnson, T. Kacprzak, S. Kent, G. Khullar, M. Klein, A. Kovacs, A. M. G. Koziol, E. Krause, A. Kremin, R. Kron, K. Kuehn, S. Kuhlmann, N. Kuropatkin, O. Lahav, J. Lasker, T. S. Li, R. T. Li, A. R. Liddle, M. Lima, H. Lin, P. López-Reyes, N. MacCrann, M. A. G. Maia, J. D. Maloney, M. Manera, M. March, J. Marriner, J. L. Marshall, P. Martini, T. McClintock, T. McKay, R. G. McMahon, P. Melchior, F. Menanteau, C. J. Miller, R. Miquel, J. J. Mohr, E. Morganson, J. Mould, E. Neilsen, R. C. Nichol, F. Nogueira, B. Nord, P. Nugent, L. Nunes, R. L. C. Ogando, L. Old, A. B. Pace, A. Palmese, F. Paz-Chinchón, H. V. Peiris, W. J. Percival, D. Petracick, A. A. Plazas, J. Poh, C. Pond, A. Porredon, A. Pujol, A. Refregier, K. Reil, P. M. Ricker, R. P. Rollins, A. K. Romer, A. Roodman, P. Rooney, A. J. Ross, E. S. Rykoff, M. Sako, M. L. Sanchez, E. Sanchez, B. Santiago, A. Saro, V. Scarpine, D. Scolnic, S. Serrano, I. Sevilla-Noarbe, E. Sheldon, N. Shipp, M. L. Silveira, M. Smith, R. C. Smith, J. A. Smith, M. Soares-Santos, F. Sobreira, J. Song, A. Stebbins, E. Suchyta, M. Sullivan, M. E. C. Swanson, G. Tarle, J. Thaler, D. Thomas, R. C. Thomas, M. A. Troxel, D. L. Tucker, V. Vikram, A. K. Vivas,

- A. R. Walker, R. H. Wechsler, J. Weller, W. Wester, R. C. Wolf, H. Wu, B. Yanny, A. Zenteno, Y. Zhang, J. Zuntz, S. Juneau, M. Fitzpatrick, R. Nikutta, D. Nidever, K. Olsen, A. Scott, and and. The dark energy survey: Data release 1. *The Astrophysical Journal Supplement Series*, 239(2):18, nov 2018.
- [24] Austin B. Tomaney and Arlin P. S. Croots. Expanding the realm of microlensing surveys with difference image photometry. *The Astronomical Journal*, 112:2872, dec 1996.
- [25] C. Alard and Robert H. Lupton. A method for optimal image subtraction. *The Astrophysical Journal*, 503(1):325–331, aug 1998.
- [26] Barak Zackay, Eran O. Ofek, and Avishay Gal-Yam. PROPER IMAGE SUBTRACTION—OPTIMAL TRANSIENT DETECTION, PHOTOMETRY, AND HYPOTHESIS TESTING. *The Astrophysical Journal*, 830(1):27, oct 2016.
- [27] Andrew Becker. HOTPANTS: High Order Transform of PSF ANd Template Subtraction. Astrophysics Source Code Library, record ascl:1504.004, April 2015.
- [28] K. Herner, J. Annis, D. Brout, M. Soares-Santos, R. Kessler, M. Sako, R. Butler, Z. Doctor, A. Palmese, S. Allam, D. L. Tucker, F. Sobreira, B. Yanny, H. T. Diehl, J. Frieman, N. Glaeser, A. Garcia, N. F. Sherman, K. Bechtol, E. Berger, H. Y. Chen, C. J. Conselice, E. Cook, P. S. Cowperthwaite, T. M. Davis, A. Drlica-Wagner, B. Farr, D. Finley, R. J. Foley, J. Garcia-Bellido, M. S. S. Gill, R. A. Gruendl, D. E. Holz, N. Kuropatkin, H. Lin, J. Marriner, J. L. Marshall, T. Matheson, E. Neilsen, F. Paz-Chinchón, M. Sauseda, D. Scolnic, P. K. G. Williams, S. Avila, E. Bertin, E. Buckley-Geer, D. L. Burke, A. Carnero Rosell, M. Carrasco-Kind, J. Carretero, L. N. da Costa, J. De Vicente, S. Desai, P. Doel, T. F. Eifler, S. Everett, P. Fosalba, E. Gaztanaga, D. W. Gerdes, J. Gschwend, G. Gutierrez, W. G. Hartley, D. L. Hollowood, K. Honscheid, D. J. James, E. Krause, K. Kuehn, O. Lahav, T. S. Li, M. Lima, M. A. G. Maia, M. March, F. Menanteau, R. Miquel, A. A. Plazas, E. Sanchez, V. Scarpine, M. Schubnell, S. Serrano, I. Sevilla-Noarbe, M. Smith, E. Suchyta, G. Tarle, W. Wester, and Y. Zhang. Optical follow-up of gravitational wave triggers with DECam during the first two LIGO/VIRGO observing runs. *Astronomy and Computing*, 33:100425, October 2020.
- [29] D. A. Goldstein, C. B. D’Andrea, J. A. Fischer, R. J. Foley, R. R. Gupta, R. Kessler, A. G. Kim, R. C. Nichol, P. E. Nugent, A. Papadopoulos, M. Sako, M. Smith, M. Sullivan, R. C. Thomas, W. Wester, R. C. Wolf, F. B. Abdalla, M. Banerji, A. Benoit-Lé vy, E. Bertin, D. Brooks, A. Carnero Rosell, F. J. Castander, L. N. da Costa, R. Covarrubias, D. L. DePoy, S. Desai, H. T. Diehl, P. Doel, T. F. Eifler, A. Fausti Neto, D. A. Finley, B. Flaugher, P. Fosalba, J. Frieman, D. Gerdes,

- D. Gruen, R. A. Gruendl, D. James, K. Kuehn, N. Kuropatkin, O. Lahav, T. S. Li, M. A. G. Maia, M. Makler, M. March, J. L. Marshall, P. Martini, K. W. Merritt, R. Miquel, B. Nord, R. Ogando, A. A. Plazas, A. K. Romer, A. Roodman, E. Sanchez, V. Scarpine, M. Schubnell, I. Sevilla-Noarbe, R. C. Smith, M. Soares-Santos, F. Sobreira, E. Suchyta, M. E. C. Swanson, G. Tarle, J. Thaler, and A. R. Walker. AUTOMATED TRANSIENT IDENTIFICATION IN THE DARK ENERGY SURVEY. *The Astronomical Journal*, 150(3):82, aug 2015.
- [30] Kenneth Herner, James Annis, Alyssa Garcia, Marcelle Soares-Santos, Dillon Brout, Noemi Glaeser, Nora Sherman, Richard Kessler, Robert Morgan, Antonella Palmese, Francisco Paz-Chinchon, Amber Lenon, and Tristan Bachmann. The updated DESGW processing pipeline for the third LIGO/VIRGO observing run. In *European Physical Journal Web of Conferences*, volume 245 of *European Physical Journal Web of Conferences*, page 01008, November 2020.
- [31] John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI Magazine*, 27(4):12, Dec. 2006.
- [32] Frederic B. Fitch. Warren s. mcculloch and walter pitts. a logical calculus of the ideas immanent in nervous activity. *bulletin of mathematical biophysics*, vol. 5 (1943), pp. 115–133. *The Journal of Symbolic Logic*, 9(2):49–50, 1944.
- [33] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, jan 2015.
- [34] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks, 2017.
- [35] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023.
- [36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [37] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network, 2015.
- [38] Aurelien Geron. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Sebastopol, CA, 2017.
- [39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

- [40] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.
- [41] N. M. Cardoso, G. B. O. Schwarz, L. O. Dias, C. R. Bom, L. Sodré Jr. au2, and C. Mendes de Oliveira. Morphological classification of galaxies in s-plus using an ensemble of convolutional networks, 2021.
- [42] Edward J. Kim and Robert J. Brunner. Star–galaxy classification using deep convolutional neural networks. *Monthly Notices of the Royal Astronomical Society*, 464(4):4463–4475, oct 2016.
- [43] Bernardo M O Fraga, Ulisses Barres de Almeida, Clécio R Bom, Carlos H Brandt, Paolo Giommi, Patrick Schubert, and Márcio P de Albuquerque. Deep learning blazar classification based on multifrequency spectral energy distribution data. *Monthly Notices of the Royal Astronomical Society*, 505(1):1268–1279, may 2021.
- [44] Mohammad Taye. Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions. *Computation*, 11:52, 03 2023.
- [45] Aurélien Géron. *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow*. Alta Books, 2019.
- [46] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ROC Analysis in Pattern Recognition.
- [47] Random decision forest. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1995.
- [48] A. Shandonay, R. Morgan, K. Bechtol, C. R. Bom, B. Nord, A. Garcia, B. Henghes, K. Herner, M. Tabbutt, A. Palmese, L. Santana-Silva, M. Soares-Santos, M. S. S. Gill, and J. Garcí a-Bellido. Expediting DECam multimessenger counterpart searches with convolutional neural networks. *The Astrophysical Journal*, 925(1):44, jan 2022.
- [49] and A. G. A. Brown, A. Vallenari, T. Prusti, J. H. J. de Bruijne, C. Babusiaux, C. A. L. Bailer-Jones, M. Biermann, D. W. Evans, L. Eyer, F. Jansen, C. Jordi, S. A. Klioner, U. Lammers, L. Lindegren, X. Luri, F. Mignard, C. Panem, D. Pourbaix, S. Randich, P. Sartoretti, H. I. Siddiqui, C. Soubiran, F. van Leeuwen, N. A. Walton, F. Arenou, U. Bastian, M. Cropper, R. Drimmel, D. Katz, M. G. Lattanzi, J. Bakker, C. Cacciari, J. Castañeda, L. Chaoul, N. Cheek, F. De Angeli, C. Fabricius, R. Guerra, B. Holl, E. Masana, R. Messineo, N. Mowlavi, K. Nienartowicz, P. Panuzzo, J. Portell, M. Riello, G. M. Seabroke, P. Tanga, F. Thévenin, G. Gracia-Abril, G. Comoretto, M. Garcia-Reinaldos, D. Teyssier, M. Altmann,

R. Andrae, M. Audard, I. Bellas-Velidis, K. Benson, J. Berthier, R. Blomme, P. Burgess, G. Busso, B. Carry, A. Cellino, G. Clementini, M. Clotet, O. Creevey, M. Davidson, J. De Ridder, L. Delchambre, A. Dell’Oro, C. Ducourant, J. Fernández-Hernández, M. Fouesneau, Y. Frémat, L. Galluccio, M. García-Torres, J. González-Núñez, J. J. González-Vidal, E. Gosset, L. P. Guy, J.-L. Halbwachs, N. C. Hambly, D. L. Harrison, J. Hernández, D. Hestroffer, S. T. Hodgkin, A. Hutton, G. Jasni ewicz, A. Jean-Antoine-Piccolo, S. Jordan, A. J. Korn, A. Krone-Martins, A. C. Lanzafame, T. Lebzelter, W. Löfller, M. Manteiga, P. M. Marrese, J. M. Martín-Fleitas, A. Moitinho, A. Mora, K. Muinonen, J. Osinde, E. Pancino, T. Pauwels, J.-M. Petit, A. Recio-Blanco, P. J. Richards, L. Rimoldini, A. C. Robin, L. M. Sarro, C. Siopis, M. Smith, A. Sozzetti, M. Süveges, J. Torra, W. van Reeven, U. Abbas, A. Abreu Aramburu, S. Accart, C. Aerts, G. Altavilla, M. A. Álvarez, R. Alvarez, J. Alves, R. I. Anderson, A. H. Andrei, E. Anglada Varela, E. Antiche, T. Antoja, B. Arcay, T. L. Astraatmadja, N. Bach, S. G. Baker, L. Balaguer-Núñez, P. Balm, C. Barache, C. Barata, D. Barbato, F. Barblan, P. S. Barklem, D. Barrado, M. Barros, M. A. Barstow, S. Bartholomé Muñoz, J.-L. Bassilana, U. Becciani, M. Bellazzini, A. Be rihuete, S. Bertone, L. Bianchi, O. Bienaymé, S. Blanco-Cuaresma, T. Boch, C. Boe che, A. Bombrun, R. Borrachero, D. Bossini, S. Bouquillon, G. Bourda, A. Bragaglia, L. Bramante, M. A. Breddels, A. Bressan, N. Brouillet, T. Brüsemeister, E. Bruga letta, B. Bucciarelli, A. Burlacu, D. Busonero, A. G. Butkevich, R. Buzzi, E. Caffau, R. Cancelliere, G. Cannizzaro, T. Cantat-Gaudin, R. Carballo, T. Carlucci, J. M. Carrasco, L. Casamiquela, M. Castellani, A. Castro-Ginard, P. Charlot, L. Chemin, A. Chiavassa, G. Cocozza, G. Costigan, S. Cowell, F. Crifo, M. Crosta, C. Crowley, J. Cuypers†, C. Dafonte, Y. Damerdji, A. Dapergolas, P. David, M. David, P. de La verny, F. De Luise, R. De March, D. de Martino, R. de Souza, A. de Torres, J. Debos scher, E. del Pozo, M. Delbo, A. Delgado, H. E. Delgado, P. Di Matteo, S. Diakite, C. Diener, E. Distefano, C. Dolding, P. Drazinos, J. Durán, B. Edvardsson, H. Enke, K. Eriksson, P. Esquej, G. Eynard Bontemps, C. Fabre, M. Fabrizio, S. Faigler, A. J. Falcão, M. Farràs Casas, L. Federici, G. Fedorets, P. Fernique, F. Figueras, F. Filippi, K. Findeisen, A. Fonti, E. Fraile, M. Fraser, B. Frézouls, M. Gai, S. Galleti, D. Garabato, F. García-Sedano, A. Garofalo, N. Garralda, A. Gavel, P. Gavras, J. Gerssen, R. Geyer, P. Giacobbe, G. Gilmore, S. Girona, G. Giuffrida, F. Glass, M. Gomes, M. Granvik, A. Gueguen, A. Guerrier, J. Guiraud, R. Gutiérrez-Sánchez, R. Haigron, D. Hatzidimitriou, M. Hauser, M. Haywood, U. Heiter, A. Helmi, J. Heu, T. Hilger, D. Hobbs, W. Hofmann, G. Holland, H. E. Huckle, A. Hypki, V. Icardi, K. Janßen, G. Jevardat de Fombelle, P. G. Jonker, Á. L. Juhász, F. Julbe, A. Karampelas, A. Ke wley, J. Klar, A. Kochoska, R. Kohley, K. Kolenberg, M. Kontizas, E. Kontizas, S. E. Koposov, G. Kordopatis, Z. Kostrzewska-Rutkowska, P. Koubeky, S. Lambert, A. F. Lanza, Y. Lasne, J.-B. Lavigne, Y. Le Fustec, C. Le Poncin-Lafitte, Y. Lebreton,

S. Leccia, N. Leclerc, I. Lecoeur-Taibi, H. Lenhardt, F. Leroux, S. Liao, E. Licata, H. E. P. Lindstrøm, T. A. Lister, E. Livanou, A. Lobel, M. López, S. Managau, R. G. Mann, G. Mantelet, O. Marchal, J. M. Marchant, M. Marconi, S. Marinoni, G. Marschalkó, D. J. Marshall, M. Martino, G. Marton, N. Mary, D. Massari, G. Matijević, T. Mazeh, P. J. McMillan, S. Messina, D. Michalik, N. R. Millar, D. Molina, R. Molinaro, L. Molnár, P. Montegriffo, R. Mor, R. Morbidelli, T. Morel, D. Morris, A. F. Mulone, T. Muraveva, I. Musella, G. Nelemans, L. Nicastro, L. Noval, W. O’Mullane, C. Ordénovic, D. Ordóñez-Blanco, P. Osborne, C. Pagani, I. Pagano, F. Pailler, H. Palacin, L. Palaversa, A. Panahi, M. Pawlak, A. M. Piersimoni, F.-X. Pineau, E. Plachy, G. Plum, E. Poggio, E. Poujoulet, A. Prša, L. Pulone, E. Raćero, S. Ragaini, N. Rambaux, M. Ramos-Lerate, S. Regibo, C. Reylé, F. Riclet, V. Ripepi, A. Riva, A. Rivard, G. Rixon, T. Roegiers, M. Roelens, M. Romero-Gómez, N. Rowell, F. Royer, L. Ruiz-Dern, G. Sadowski, T. Sagristà Sellés, J. Sahlmann, J. Salgado, E. Salguero, N. Sanna, T. Santana-Ros, M. Sarasso, H. Savietto, M. Schultheis, E. Sciacca, M. Segol, J. C. Segovia, D. Ségransan, I-C. Shih, L. Siltala, A. F. Silva, R. L. Smart, K. W. Smith, E. Solano, F. Solitro, R. Sordo, S. Soria Nieto, J. Souchay, A. Spagna, F. Spoto, U. Stampa, I. A. Steele, H. Steidelmüller, C. A. Stephenson, H. Stoev, F. F. Suess, J. Surdej, L. Szabados, E. Szegedi-Elek, D. Tapiador, F. Taris, G. Tauran, M. B. Taylor, R. Teixeira, D. Terrett, P. Teys-sandier, W. Thuillot, A. Titarenko, F. Torra Clotet, C. Turon, A. Ulla, E. Utrilla, S. Uzzi, M. Vaillant, G. Valentini, V. Valette, A. van Elteren, E. Van Hemelryck, M. van Leeuwen, M. Vaschetto, A. Vecchiato, J. Veljanoski, Y. Viala, D. Vicente, S. Vogt, C. von Essen, H. Voss, V. Votruba, S. Voutsinas, G. Walmsley, M. Weiler, O. Wertz, T. Wevers, L. Wyrzykowski, A. Yoldas, M. Žerjal, H. Ziaeepour, J. Zorec, S. Zschocke, S. Zucker, C. Zurbach, and T. Zwitter. *igaia/i* data release 2. *Astronomy & Astrophysics*, 616:A1, aug 2018.

- [50] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [52] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.
- [53] Tom O’Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Kerastuner. <https://github.com/keras-team/keras-tuner>, 2019.
- [54] Timothy Dozat. Incorporating nesterov momentum into adam. 2016.

- [55] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning, 2016.
- [56] Daquan Zhou, Qibin Hou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Rethinking bottleneck structure for efficient mobile network design. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, pages 680–697. Springer, 2020.
- [57] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.
- [58] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.
- [59] C R Bom, A Cortesi, G Lucatelli, L O Dias, P Schubert, G B Oliveira Schwarz, N M Cardoso, E V R Lima, C Mendes de Oliveira, L Sodre, A V Smith Castelli, F Ferrari, G Damke, R Overzier, A Kanaan, T Ribeiro, and W Schoenell. Deep learning assessment of galaxy morphology in s-PLUS data release 1. Monthly Notices of the Royal Astronomical Society, 507(2):1937–1955, jul 2021.
- [60] Ž. Ivezić, S. M. Kahn, J. A. Tyson, B. Abel, E. Acosta, R. Allsman, D. Alonso, Y. AlSayyad, S. F. Anderson, J. Andrew, and et al. LSST: From Science Drivers to Reference Design and Anticipated Data Products. , 873:111, March 2019.