Advanced Analysis Methods for Imaging Atmospheric Cherenkov Telescope Data with SSTCAM and VERITAS

Samuel Timothy Spencer

Exeter College
University of Oxford

A thesis submitted for the degree of Doctor of Philosophy

Michaelmas 2021

Abstract

Imaging Atmospheric Cherenkov Telescope arrays allow us to probe the γ-ray sky from tens of GeV up to hundreds of TeV. They operate by stereoscopically imaging the Cherenkov light generated when an astrophysical γ-ray interacts with Earth's atmosphere. In order to reject charged cosmic ray events, and to reconstruct the direction and energy of the incident γ-ray, machine learning methods are used in combination with parametric descriptions of the detected images. One potential means of improving performance for the next-generation Cherenkov Telescope Array (CTA) is to apply new deep learning methods in place of these parametric techniques. In this thesis, we explore the complexity of deploying deep learning methods, first considering the application of high precision timing data, and then testing such methods' performance on real observations from the current generation VERITAS array. Finally, we explore improvements to the modelling of Night Sky Background observed by Cherenkov instruments, that can be used to both inform the design of the Small Sized Telescope Camera (SSTCAM) for CTA, as well as potentially augment simulated data for deep-learning-based event classification.

Advanced Analysis Methods for Imaging Atmospheric Cherenkov Telescope Data with SSTCAM and VERITAS

Samuel Timothy Spencer

Exeter College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2021

Acknowledgements

Personal

Firstly, I want to thank my supervisor Garret Cotter for getting me through my DPhil, advice writing this thesis and lots of other support over the past four years (including getting me lots of GPUs). Thank you Garret for helping me learn how to be an independent researcher, as well as teaching me how to manage the politics of academia. Similarly, I'd like to thank Gernot Maier for fulfilling a similar role during my time at DESY and since then, working with whom I've learnt a vast amount about γ-ray data analysis.

Secondly, I want to thank my family. Particularly my parents David and Vivien, as well as my brother Jonathan (and his partner Rebecca), for their support during my DPhil (and beforehand). Completing a doctorate is a challenge at the best of times, but without their help during a global pandemic it would have been impossible.

I particularly want to thank Tom Armstrong, Jason Watson and Rich White for their help during my involvement with CHEC (and now SSTCAM). I'd also like to thank the other members of the γ-ray astronomy group during my time in Oxford: Paul Morris, Mario Hörbe, Will Potter, Jamie Davies, Hugh Spackman, Connor Duffy and Jeff Grube for being good travel companions and for the entertaining discussions about γ-ray astronomy. I'm also thankful to Tarek Hassan, Raul Prado and Mireia Nievas Rosillo for being reliable postdocs during my secondment to DESY. I also owe a (life-long) debt of gratitude to (now all professors) Paula Chadwick, Anthony Brown and Sam Nolan for getting me into the γ-ray astronomy field in the first place, by hosting me for multiple summer-studentships during my time as an undergraduate in Durham. Thanks as well to all the people who provided comments on the draft of this thesis, particularly Simon Mason.

I'm grateful to Ashling Gordon and Leanne O'Donnell for saving me from admin nightmares multiple times (as well as support during the darkest days of COVID-19). I'm also grateful to Jonathan Patterson for IT support during the GPU upgrade of the *Glamdring* cluster.

Finally, there's a long list of friends and colleagues who I've met during my time at Oxford Astrophysics (particularly those in the same DPhil year as me), at DESY, and within the CTA consortium, that I don't have space to thank individually here. I'm very grateful for your help and to have known you all nonetheless.

Institutional

All Chapters: I acknowledge an STFC doctoral studentship, as well as additional support from Oxford University Department of Physics following the COVID-19 pandemic. I also thank the developers of *ctapipe* for providing an invaluable new tool for γ-ray data analysis.

Chapter 3: This work used the European Grid Infrastructure. Thanks to Luisa Arrabito and Johan Bregeon for their assistance in running simulations and code on it. This work has gone through internal review by the CTA Consortium, who I thank for allowing me to use CTA simulation models.

Chapters 3 and 4: This work used the *Glamdring* cluster in Oxford; I thank Jonathan Patterson for his work in upgrading and maintaining this facility, as well as setting up the MongoDB database used for the parallelised optimisation in Chapter 4.

Chapters 3 and 5: I gratefully acknowledge financial support from the agencies and organisations listed here: `http://www.cta-observatory.org/consortium_acknowledgments`.

Chapter 4; I thank the VERITAS Consortium for allowing us to use VERITAS simulation models and data. I also thank the computing support group at DESY for providing cluster resources.

Chapter 5: I thank Matthias Buechele and colleagues in H.E.S.S. for the original development work on the *nsb* tool. This work has made use of data from the European Space Agency (ESA) mission *Gaia*, processed by the *Gaia* Data Processing and Analysis Consortium (DPAC). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the *Gaia* Multilateral Agreement.

Statement of Authorship

The material in this thesis has not been submitted for examination at the University of Oxford or at any other institution. All the work within this thesis is my own, along with all figures (unless explicitly stated otherwise). Work from this thesis has been published in the following works (of which I was first author):

1. Samuel Spencer et al., Prospects for the Use of Photosensor Timing Information with Machine Learning Techniques in Background Rejection, PoS(ICRC2019) 798 (2019).

2. Samuel Spencer et al., Deep Learning with Photosensor Timing Information as a Background Rejection Method for the Cherenkov Telescope Array, Astroparticle Physics 129C 102579 (2021).

3. Samuel Spencer et al., Advanced Studies of SSTCAM Night Sky Background, SSTCAM Internal Note (2021).

Both Publications 1 and 2 were subject to full CTA collaboration review, and Publication 2 also went through a peer review process with Astroparticle Physics. Chapters 3 and 5 detail work that was performed in the context of the CHEC (later SSTCAM) collaborations, and also the wider Analysis and Simulations Working Group of CTA. Chapter 4 contains work performed with the co-operation of the VERITAS collaboration.

Thomas Armstrong performed simulations needed for the results in Chapter 3, though the analysis of those simulations was performed by myself. Raul Prado played a similar role for the simulations used in Chapter 4, and Gernot Maier made the changes needed to *Eventdisplay* to make it compatible with the deep learning analysis, although I was responsible for the creation of the rest of the analysis pipeline.

Abstract

Imaging Atmospheric Cherenkov Telescope arrays allow us to probe the γ-ray sky from tens of GeV up to hundreds of TeV. They operate by stereoscopically imaging the Cherenkov light generated when an astrophysical γ-ray interacts with Earth's atmosphere. In order to reject charged cosmic ray events, and to reconstruct the direction and energy of the incident γ-ray, machine learning methods are used in combination with parametric descriptions of the detected images. One potential means of improving performance for the next-generation Cherenkov Telescope Array (CTA) is to apply new deep learning methods in place of these parametric techniques. In this thesis, we explore the complexity of deploying deep learning methods, first considering the application of high precision timing data, and then testing such methods' performance on real observations from the current generation VERITAS array. Finally, we explore improvements to the modelling of Night Sky Background observed by Cherenkov instruments, that can be used to both inform the design of the Small Sized Telescope Camera (SSTCAM) for CTA, as well as potentially augment simulated data for deep-learning-based event classification.

Contents

List of Figures

List of Abbreviations

1D, 2D, 3D . . One, two or three dimensional, referring in this thesis to spatial (or temporal) dimensions in an image.

AI . . . . . Artificial Intelligence

AGN . . . . . Active Galactic Nuclei

ALT . . . . . Altitude

AZ . . . . . Azimuth

ASIC . . . . . Application-Specific Integrated Circuit

ASTRI . . . . Astrofisica con Specchi a Tecnologia Replicante Italian; a CTA SST optical structure prototype.

AUC . . . . . Area Under Curve Score; a metric used in machine learning.

BDT . . . . . Boosted Decision Tree; a machine learning technique for classification and regression.

CHEC . . . . . The Compact High Energy Camera; a prototype camera for the SSTs.

CNN . . . . . Convolutional Neural Network, a form of machine learning algorithm designed to work with images.

CTA . . . . . Cherenkov Telescope Array, the planned successor to the current generation of IACTs.

CTAO . . . . . Cherenkov Telescope Array Observatory, the observatory associated with CTA.

DEC . . . . . Declination

DL . . . . . Deep Learning. Refers to machine learning with a large number of intermediate 'hidden' layers between the input and output layers (or equivalently a high degree of abstraction of learned features).

EAS . . . . . Extensive Air Shower; a particle cascade in the atmosphere.

ENF . . . . . Excess Noise Factor; a physical property of SiPMs.

FACT . . . . . First G-APD Cherenkov Telescope; An IACT on La Palma.

FLI . . . . . Fractional Lunar Illumination; The percentage of the Moon illuminated.

FNR . . . . . False Negative Rate

FPR . . . . . False Positive Rate

FSRQ . . . . . Flat Spectrum Radio Quasar; a class of AGN.

GAN . . . . . Generative Adversarial Network; a deep learning method of performing pseudo-simulation and domain adaptation.

GCT . . . . . Gamma Cherenkov Telescope;a CTA SST optical structure prototype.

GPU . . . . . Graphics Processing Unit

GRB . . . . . Gamma-Ray-Burst

HAWC . . . . High Altitude Water Cherenkov; A water Cherenkov detector in Mexico.

H.E.S.S. . . . . High Energy Stereoscopic System; An IACT array in Namibia.

HV . . . . . High Voltage

IACT . . . . . Imaging Atmospheric Cherenkov Telescope.

IC . . . . . Inverse Compton

ImPACT . . . Image Pixelwise fit for Atmospheric Cherenkov Telescopes; a template fitting method of event reconstruction used by H.E.S.S..

KSP . . . . . Key Science Project; a major CTA scientific investigation.

IRF . . . . . Instrument Reponse Function; curves that quantify the performance of an IACT.

MLP . . . . . Multi-Layer-Perceptron; A type of neural network.

LHASSO . . . Large High Altitude AirShower Observatory; a hybrid water Cherenkov/ IACT detector under construction in China.

LIV . . . . . Lorentz Invariance Violation

LST . . . . . Large Size Telescope; the largest CTA telescope class designed to operate at the lowest energies.

LSTM . . . . . Long Short-Term Memory Network; a machine learning method for handling data series.

MAGIC . . . . Major Atmospheric Gamma Imaging Cherenkov Telescopes; an IACT array on La Palma.

MAPM . . . . Multi-Anode Photomultipliers

ML . . . . . Machine Learning

MST . . . . . The Medium Size Telescope; the intermediate size CTA telescope class.

NaN . . . . . Not a Number

NSB . . . . . Night Sky Background to IACTs

OCT . . . . . Optical Cross-Talk; an effect observed in SiPMs.

OSO 3 . . . . . Orbiting Solar Observer 3

PDE . . . . . Photo-Detection Efficiency; the percentage of photons incident on a PMT that are detected.

pe . . . . . Photo-electrons

PMT . . . . . Photomultiplier Tube; a high-sensitivity light detector.

PSF . . . . . Point Spread Function

RA . . . . . Right Ascension

RF . . . . . Random Forest; a machine learning technique similar to BDTs.

ROC . . . . . Receiver Operator Characteristic Curve; a curve to demonstrate machine learning performance.

SAS-2 . . . . . Small Astronomy Satellite 2

SCT . . . . . The Swartzchild Couder Telescope; an alternative, American lead dual mirror prototype design for the MST.

SiPM . . . . . Silicon Photomultiplier; the type of photomultiplier used by CHEC-S

SST . . . . . Small Size Telescope; the smallest CTA telescope designed to operate at the highest energies.

SSTCAM . . . Small Size Telescope Camera; the camera for the SSTs.

TM . . . . . Target Module

TNR . . . . . True Negative Rate

ToO . . . . . Target of Opportunity

TPR . . . . . True Positive Rate

VERITAS . . Very Energetic Radiation Imaging Telescope Array System; An IACT array in Arizona.

WCS . . . . . World Co-ordinate System; a means of defining the location of an image on the sky.

1 Introduction

Contents

1.1 Astrophysics in the γ-Ray Domain

Astrophysical γ-rays are generally considered to be photons with energies in excess of 100 keV, though there is no strict cutoff against X-rays. A wide variety of astrophysical objects produce γ-rays, including Supernova Remnants (SNR), Active Galactic Nuclei (AGN) and pulsars (to name but a few) [1]. In order to fully understand these objects, we need to capture maximal information about their emission across the entire electromagnetic spectrum, as well as their multi-messenger emission (neutrinos, gravitational waves and cosmic rays). Performing electromagnetic observations of these sources at high photon energies is challenging for a number of reasons [2]. Firstly, γ-rays cannot be focused and as such it is

necessary to use particle physics techniques to reconstruct their point of origin. Secondly, γ-rays do not penetrate the Earth's atmosphere, meaning that in order to detect γ-rays directly the instrument must be in space. Finally, γ-rays are comparatively rare (the brightest astrophysical γ-ray sources have a flux of about $6\,\text{photons}\,\text{m}^{-2}\,\text{year}^{-1}$ above 1 TeV [2]), meaning that in order to detect the highest energy γ-rays one must use an instrument with a very large effective area.

1.2 Direct γ-ray Detection

1.2.1 Early Missions

The problems in observing astrophysical γ-rays associated with absorption in Earth's atmosphere can be solved by observing in space. Following theoretical predictions of cosmic γ-ray emission in the 1940s and 1950s [3], the first space mission to directly observe astrophysical γ-rays was *Explorer* 11. Launched on 17th November 1961 [4], it consisted of two instruments. The first was the Phoswich-Cherenkov counter telescope, which consisted of a sandwich of NaI and CsI scintillator viewed by a single photomultiplier. The second was a lucite Cherenkov counter viewed by two photomultipliers. *Explorer* 11 detected 22 γ-rays from all directions, suggesting the existence of a γ-ray background, but no point sources were seen in the data. The later, *Orbiting Solar Observer 3* (OSO 3) mission (which was similar but with improved energy resolution) was able to separate this diffuse emission into galactic and extragalactic components [5]. It wasn't until the *Small Astronomy Satellite 2* (SAS-2) [6] and *COS-B* [7] missions (the latter of which used an innovative spark chamber system to track electrons and positrons from γ-ray pair production) in the 1970s that persistant γ-ray point sources, such as the pulsar Geminga, were observed.

Concurrently, the *Vela* satellites (the first of which was launched in 1963) were a series of American satellites launched to monitor the partial nuclear test ban treaty. On July 2, 1967 at 14:19, *Vela* 3 and 4 observed a bright flash of γ-rays unlike

any nuclear explosion observed before, seemingly originating from space[1]. Due to the *Vela* results being classified, it wasn't until 1973 that this signal (along with 15 others) was published as evidence of astrophysical Gamma-Ray Bursts (GRBs), the most energetic astrophysical objects ever detected [9]. The first space-based telescope to simultaneously perform GRB monitoring and a survey of persistant γ-ray emitters was the *Compton* γ-ray observatory in the 1990s (billed as one of NASA's 'great observatories' alongside the Hubble Space Telescope) [10]. The technical challenge of building instruments sensitive enough to observe persistent sources as well as agile enough to observe GRBs continues to this day [11].

1.2.2 The Fermi Space Telescope

The current workhorse for the majority of space-based γ-ray observations is the very successful *Fermi* space telescope, which has operated since 2008. *Fermi* is equipped with two instruments, the Large Area Telescope (LAT) which surveys the entire γ-ray sky every three hours, and the Gamma-Ray Burst Monitor (GBM), which is specifically designed to detect GRBs and solar flares. LAT consists of 18 layers of silicon strip detectors designed to promote pair production from incident γ-rays, and detect the resultant charged products, as well as a scintillator calorimeter to measure the resultant e^+/e^- energy (similar in concept to *COS-B*). LAT is able to achieve near perfect background rejection of charged cosmic rays due to the presence of a conducting anti-coincidence shield that surrounds the instrument, such that if a charge signal is detected the instrument will not trigger. GBM consists of two sets of detectors: twelve sodium iodide (NaI) scintillators, and two cylindrical bismuth germanate (BGO) scintillators. Through this it can reconstruct the positions of GRBs anywhere on the sky not occluded by the Earth.

Notable major discoveries by *Fermi* include the detection of the Fermi bubbles, large γ-ray emitting structures extending to around 20° degrees above and below the galactic plane [12], as well as the detection of the short GRB 170817 coincident with

[1]One would hope this alone was evidence enough of astrophysical γ-ray emission, however the USA had previously detonated a series of nuclear weapons in space (at an altitude of 250 miles), beginning with the 1.4 megaton *Starfish Prime* test in 1962 [8].

Figure 1.1: The *Fermi* space telescope on Earth, with its solar arrays folded. LAT is the block at the top of the satellite covered with the silver anti-coincidence shield. GBM consists of the cylindrical structures protruding from the lower half of the telescope. Note the size of the engineers relative to the size of the instrument, this highlights the comparatively small effective area of LAT. Image credit: NASA/Kim Shiflett.

a gravitational wave event (GW 170817) from a binary neutron star merger [13]. However, the energy range of LAT is limited to between 20 MeV and 300 GeV because of the small effective area of the detector. In order to observe higher-energy photons (in the high GeV-TeV energy range) we must detect them indirectly. This can be performed using the Imaging Atmospheric Cherenkov Telescope (IACT) method, pioneered by Jelly and Galbraith in 1953 [14], and by utilising water Cherenkov detectors [15]. Both of these techniques operate by detecting the Extensive Air Shower (EAS) created when a high energy γ-ray interacts with Earth's atmosphere. We discuss the specifics of these detectors in the next section.

1.3 Indirect γ-ray Detection

1.3.1 Historical Instruments

Jelly and Galbraith [14] were the first to create an atmospheric Cherenkov light detector. It consisted of a searchlight mirror, a rubbish bin (painted black on the inside), an amplifier, an oscilloscope and a single photomultiplier. With it, they were able to detect pulses of light corresponding to Cherenkov emission from EAS. Following this initial detection, there began a long period of development of the IACT technique (both in terms of instruments and analysis) for astrophysical γ-ray detection. Early experiments included the 10-meter Whipple telescope in Arizona [16], which is generally considered the first to reliably detect a γ-ray source from the ground (the Crab Nebula, we will return to the specifics of this detection later in this chapter), as well as the High-Energy-Gamma-Ray Astronomy (HEGRA) experiment on La Palma [17] and the CANGAROO [18] and Durham [19] telescopes in Australia. Whilst some of the astrophysical detections claimed using these instruments have since been considered controversial [20], the current era of reliable γ-ray astronomy from the ground would likely not have been possible without the work performed for these experiments. In the rest of this section we will discuss how current generation IACTs work.

1.3.2 The Cherenkov Effect

Both IACTs and water Cherenkov detectors ultimately rely upon the Cherenkov effect. Cherenkov light is the light emitted by a medium when a charged particle passes through it faster than the phase velocity of light. An image qualitatively describing the effect can be seen in Figure 1.2. When a charged particle is at a point P in a dielectric, the atoms surrounding it behave like simple dipoles, with the positive end of the dipole reorienting to point towards the particle if the particle is negatively charged and vice versa. If the particle travels sufficiently slowly, the polarization field surrounding the particle will be completely symmetric, and so when the particle moves on to a new position P', there is no net electric field at

a far distance and so no emission. However, if the particle travels faster than the phase velocity of light, whilst the polarization field will still be symmetric along the azimuth, it will no longer be symmetric along the direction of motion. This means that when the system relaxes as the particle moves on a light pulse will be generated [21]. As such, as the particle travels through the medium, successive brief pulses of light are generated.

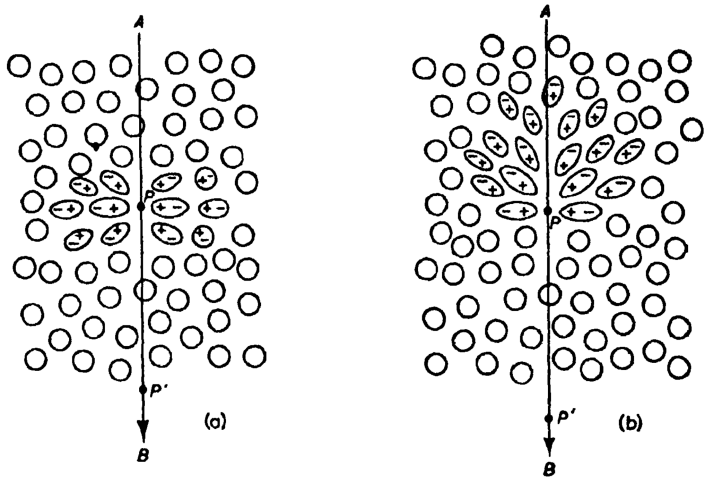


Figure 1.2: The polarization set up in a medium by the passage of a charged particle at low velocity (left, a) and high velocity (right, b). Figure taken from [21].

As can be seen in Figure 1.3, this results in the emitted light being distributed according to the Cherenkov opening angle θ_c

$$\cos\theta_c = \frac{1}{n\beta} \tag{1.1}$$

where n is the refractive index, $\beta{=}v/c$ where v is the velocity of light in the medium and c is the speed of light in a vacuum. The energy dE emitted as a result of the particle traversing the medium per unit length dx is given by the Frank-Tamm formula, derived using the framework of special relativity in 1937 (and winning the authors a Nobel prize in 1958 along with Cherenkov). This holds provided the velocity v as a fraction of the speed of light in a vacuum c is greater than $\frac{1}{n(\omega)}$ (where $n(\omega)$ is the frequency dependent refractive index)

$$\frac{d^2E}{dx\ d\omega} = \frac{q^2}{4\pi}\mu(\omega)\omega\left(1 - \frac{c^2}{v^2n^2(\omega)}\right) \tag{1.2}$$

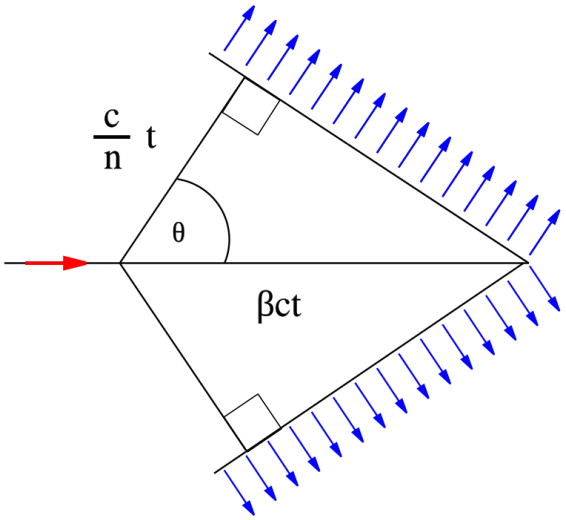


Figure 1.3: The geometry of the Cherenkov effect, assuming no dispersion. Image credit: A. Horvarth.

where $\mu(\omega)$ is the frequency dependent permeability, and q is the charge of the particle [22]. The total energy emitted in Cherenkov radiation is therefore given by the corresponding integral

$$\frac{dE}{dx} = \frac{q^2}{4\pi} \int_{v > \frac{c}{n(\omega)}}^{\infty} \mu(\omega)\omega \left(1 - \frac{c^2}{v^2 n^2(\omega)}\right) d\omega. \tag{1.3}$$

This integral is finite as $\lim_{\omega\to\infty} n(\omega) < 1$ and $\lim_{\omega\to 0} n(\omega) = 1$. Assuming $\mu(v)$ is unity equation 1.2 can also be integrated over frequency [23] to get the number of Cherenkov photons dN_γ emitted per unit length

$$\frac{dN_\gamma}{dx} = 2\pi\alpha \left(1 - \frac{1}{\beta^2 n^2}\right) \left(\frac{1}{\lambda_{min}} - \frac{1}{\lambda_{max}}\right) \tag{1.4}$$

where α is the fine structure constant, and the λ terms represent the minimum and maximum wavelength range of the emission. Taking this band to be $300\,\text{nm} - 600\,\text{nm}$ and $\beta \approx 1$ implies $\frac{dN_\gamma}{dx}$ is around $15\,\text{m}^{-1}$ at an $8\,\text{km}$ altitude in air [23].

1.3.3 IACT Basics

When a photon with an energy of above around $20\,\text{GeV}$ encounters the Earth's atmosphere, it undergoes pair production in the electric field of the nucleus of

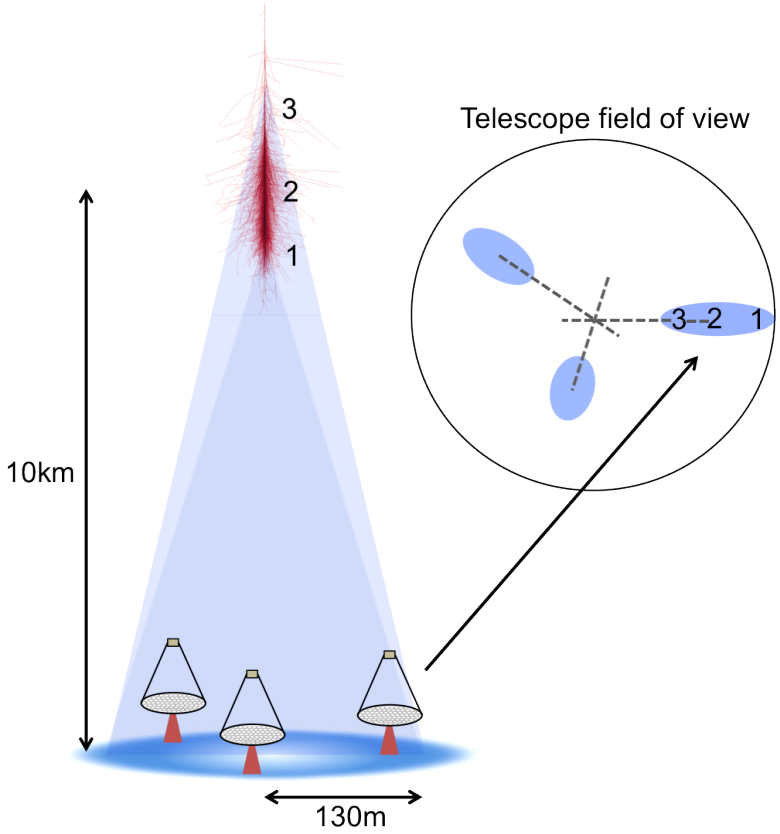


Figure 1.4: A schematic of the IACT technique, taken from [2]. Three telescopes stereoscopically observe a Cherenkov light pool caused by an EAS. The direction of the incident photon is reconstructed by intersecting the primary axes of the elliptical images in the three cameras. On the right, an illustration that the light from the lowest altitude part of the shower is the first to arrive on the ground.

an atmospheric molecule X

$$\gamma + \mathrm{X} \rightarrow \mathrm{X} + \mathrm{e}^{+} + \mathrm{e}^{-}. \tag{1.5}$$

This occurs on average after the γ-ray travels through one radiation length's worth of atmosphere, typically at around a 20 km altitude [24] (for a more detailed study of the first interaction height of γ-rays see [25], it can be estimated using measurements of shower maximum). The resulting electron and positron have very high kinetic energy, and so travel very close to the speed of light in a vacuum and faster than the speed of light in air. As a result, Cherenkov radiation is emitted by the molecules in the vicinity of the particle. The generated particles continue to interact, emitting photons via Brehmsstrahlung radiation, and creating a particle cascade or Extensive Air Shower (EAS). This cascade continues down into the atmosphere

until the radiation and ionisation losses become equal, which is also the point at which the maximum number of particles (N_{Max}) are present in the shower as a function of atmospheric mass traversed (in $\mathrm{g\,cm^{-2}}$). This point is known as the shower maximum (denoted X_{Max}). The altitude at which this occurs is referred to as h_{Max}, after which point the ionisation losses dominate and the number of produced particles significantly reduces. For an incident γ-ray of 10 TeV energy, $X_{Max} \approx 431\,\mathrm{g\,cm^{-2}}$, $h_{Max} = 6.8\,\mathrm{km}$ and $N_{Max} = 1.0 \times 10^4$ [24]. The Cherenkov light produced from this EAS forms a pool on the ground of radius $\sim 120\,\mathrm{m}$ [24], and this can be observed with an optical frequency telescope on the ground at night provided it is equipped with an extremely fast (ns timescale) camera, as the Cherenkov light flash from an EAS is of the order of a few ns.

In order to optimally reconstruct the energy and direction of the incident γ-ray, data from an array of such telescopes is typically combined stereoscopically, a technique pioneered by the HEGRA instrument in the 1990s [17]. This use of stereoscopy improves background rejection power as the chances of directly imaging a muon ring originating from a proton shower are higher, and with multiple images the determination of shower width is more accurate. The optimal spacing for these telescopes is of the same order as the size of the light pool on the ground [24].

Given that the only (not heavily-suppressed) interactions possible in an γ-ray or electron induced air shower are pair production and γ-ray brehmstrahlung, it is possible to construct a simple model of their evolution (the Heitler model) [26]. Given that the radiation length of an electron in the atmosphere is known to be $X_0 = 36.7\,\mathrm{g\,cm^{-2}}$. After n interactions the number of surviving particles is $N = 2^n = \exp\left(\frac{x}{X_0}\right)$, from which it follows the atmospheric grammage traversed is $x = nX_0 \ln 2$. Then one assumes that the typical length over which a new pair production will occur is $N_0 = X_0 \ln 2$ (this typically slightly overestimates the true number of particles [27]). Assuming also that the particle energies are uniformly distributed, the energy per particle is given by $E = E_0/N$ (where E_0 is the initial energy of the originally incident particle. Given that the shower maximum occurs

when $E = E_c^e = 85\,\text{MeV}$, the number of particles at X_{Max} is given by [26]

$$N_{Max} = 2^{n_c} = \frac{E_0}{E_c^e} \tag{1.6}$$

where

$$n_c = \frac{\ln(\frac{E_0}{E_e^c})}{\ln 2}. \tag{1.7}$$

1.4 Current Generation Indirect Detection Instruments

1.4.1 VERITAS

Very Energetic Radiation Imaging Telescope Array System (VERITAS) consists of four 12-meter parabolic, single mirror IACTs at the Fred Lawrence Whipple Observatory in Arizona (the original VERITAS design intended eight IACTs). Construction of VERITAS began in 2003 and was completed in 2007. It is designed to give optimal sensitivity in the 100 GeV to 10 TeV energy band. Recent work performed by VERITAS includes the discovery of a flare in the gamma-ray emission from the Blazar TXS 0506+056, coincident in space and time with a neutrino detection from IceCube [28], and the detection of TeV gamma-rays from the redshift 0.939 Blazar PKS 1441+25, indicating the transparency of the universe to photons of such energies [29]. In Chapter 4, we use real observations from VERITAS to test deep-learning-based event classification as a technique intended for use by the next generation Cherenkov Telescope Array (CTA).

1.4.2 MAGIC

Major Atmospheric Gamma Imaging Cherenkov Telescopes (MAGIC) consists of two 17-meter-diameter IACTs on La Palma, designed to detect γ-rays with energies between 25 GeV and 30 TeV. MAGIC's cameras are unusual as they consist of two separate types of conventional photomultiplier, 386 hexagonal pixels in the centre surrounded by 180 larger pixels at the edge. A major aim of MAGIC was to hunt for Gamma-Ray Bursts (GRBs) from the ground, a goal it achieved when

it detected TeV γ-rays from GRB 190114C in early 2019 [11] (H.E.S.S. CT5 also detected the subsequent afterglow).

1.4.3 H.E.S.S.

The High Energy Stereoscopic System (H.E.S.S.) is an IACT array in the Khomas Highlands of Namibia. H.E.S.S. is the only current generation instrument to consist of multiple classes of IACT, four with a ∼12-meter-diameter mirror (CT1-4) and a fifth (CT5) with a 28-meter-diameter mirror, giving it an operational energy range of 30 GeV to 100 TeV. Construction of the four original 12 m instruments was completed in 2003, with the larger telescope completed in 2012. The H.E.S.S. cameras have been upgraded twice. In 2015 the CT1-4 telescopes were upgraded with improved ventilation and NECTAr-based front-end electronics [30]. Then in 2019 CT5 was upgraded with a new fully-digitized trigger and readout system and higher-efficiency photomultipliers (a variant of the CTA FlashCam camera intended for deployment on the MSTs). Notable recent achievements of H.E.S.S. include resolving the extension of the Crab Nebula [31] and the jet of Centaurus A [32] for the first time (with an IACT array).

1.4.4 FACT

The First G-APD Cherenkov Telescope (FACT) is an autonomous upgrade of a former HEGRA Cherenkov telescope, notable for being the first instrument to attempt the use of Silicon Photomultipliers (SiPMs) in Cherenkov astronomy. Given it is a single IACT and not an array, however, its sensitivity is limited. As such it largely operates as a blazar monitoring instrument, such that other instruments can be notified in the event of a flare.

FACT arguably has one of the most advanced analysis chains of the current IACTs, which is largely based in Python, and this has had a partial influence on CTA's analysis chain. Notably the FACT collaboration used open source, Python-based analysis tools developed using modern version control (with git and Github)

[33]. This was not the case for any of the other analysis chains developed for the other current IACTs which remain largely based on ROOT [34].

1.4.5 HAWC, Tibet AS-γ and LHASSO

The IACT technique is not the only means of indirect γ-ray detection from the ground. At sufficiently high altitude, one can use Water Cherenkov detectors to directly observe the charged particles from an incident shower, and then use arrival time information as a background rejection method. This is the key principle underlying the successful High Altitude Water Cherenkov (HAWC) observatory in Mexico [15], as well the principle behind the Chinese Tibet AS-γ [35] and Large High Altitude Air Shower Observatory (LHASSO) [36] experiments (the currently under construction LHASSO detector will also have small IACTs on site). These complement IACTs with their main advantage being that they are survey instruments with a wide field of view, higher operational energy range and nearly 100% duty cycle (for IACTs this is closer to 10%), but they have comparatively poor angular resolution and background rejection power.

1.5 The Cherenkov Telescope Array and Associated Instruments

1.5.1 Concept

The Cherenkov Telescope Array (CTA) is an ambitious project to build a next-generation VHE γ-ray facility, which aims to improve on the sensitivity of the current generation instruments by roughly an order of magnitude [1]. The CTA consortium, involved in directing CTA Observatory (CTAO) science goals and array design, consists of over 1400 scientists from 31 countries around the globe. Given the large energy range (20 GeV to 300 TeV) that CTA will operate over [1], three classes of IACT are required. These are designated as the Small-, Medium- and Large-Sized Telescopes (SSTs, MSTs and LSTs). The 23-meter-diameter LSTs are designed for high sensitivity observations of γ-rays over the 20 GeV to 150 GeV energy range, increasing the overlap in sensitivity with space-based missions such

as *Fermi* [37]. The SSTs will be the smallest (approximately 4-meter-diameter) but most numerous telescopes, and will be spread out over a large ($\sim 4\,\mathrm{km}^2$) area in order to maximise CTA's effective area for γ-ray energies over 5 TeV. The MSTs will provide unprecedented sensitivity to cosmic γ-ray fluxes in the intermediate energy range. The baseline ('omega' configuration) design of CTA consists of 99 telescopes (4 LSTs, 25 MSTs, 70 SSTs) placed on a southern site at Cerro Paranal in Chile, whereas a smaller array of 19 telescopes (4 LSTs, 15 MSTs, and no SSTs) will be placed on a northern site on Spanish Canary Island of La Palma. By having both a northern and southern site, the CTA Observatory will cover the whole sky [1]. However, as of writing this thesis, the current construction plan for CTA is to first build a so-called 'alpha' configuration array consisting of 4 LSTs and 9 MSTs in the northern site, and 14 MSTs and 37 SSTs in the southern array. Further details regarding the science performance of the SST sub-array and the complete CTA instrument can be found in Maier et. al [38] and Hassan et. al [39].

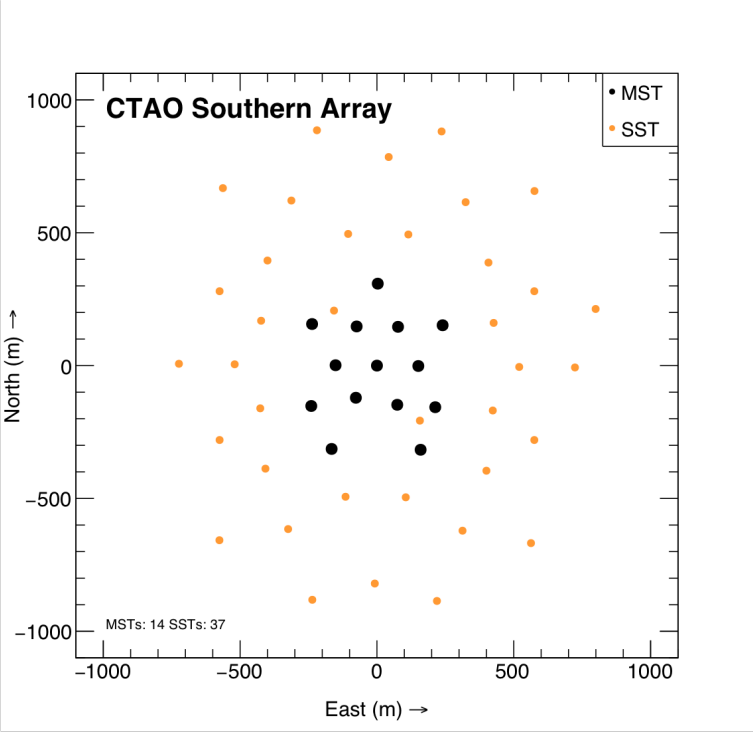


Figure 1.5: The CTA-South alpha array configuration, taken from [40].

1.5.2 CHEC and SSTCAM

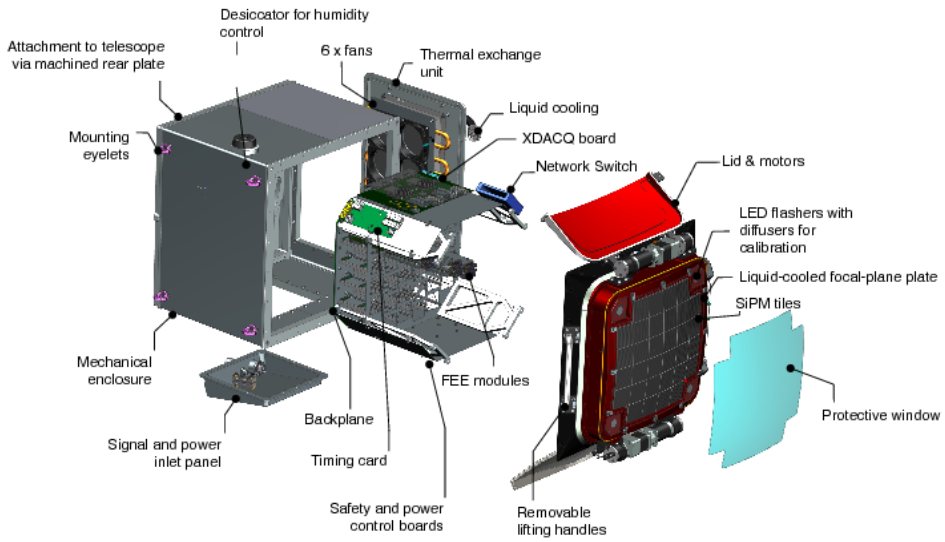


Figure 1.6: The CHEC-S Computer Aided Design (CAD) model with key elements highlighted, taken from [41].

The UK's main material contribution to CTA construction is planned to be construction of SST Cameras (SSTCAMs). These are an upgraded version of the earlier Compact High Energy Camera (CHEC) prototypes, which were designed to work with SST structures that have a dual-mirror (Schwarzschild-Couder) optical design. This design allows for a compact camera (approximately 0.4-meter-diameter) with small scale photosensors (6/7 mm, corresponding to roughly a third of the plate scale of a comparable single-mirror design) and a wide field of view ($\sim 0.2\,^{\circ}$/pixel). Two operational CHEC prototypes have already been built. The first, CHEC-M, has as its light detectors 32 Multi-Anode Photomultipliers (MAPMs) [42]. These consist of a block of 8x8 pixels, the signals from which are digitised at a rate of 1GSa/s [42] by front end electronics based upon the custom TARGET Application-Specific Integrated Circuit (ASIC) [43]. These blocks of front end electronics per

photomultiplier tile are known as a TARGET Module (TM). When the camera is triggered by two or more four pixel blocks (superpixels) exceeding a discriminator threshold, the data is read out of the 'fast chain' of the camera over a configurable window in blocks of 32 samples (roughly corresponding to a nanosecond each). CHEC simultaneously has a slow-signal chain with a longer integration window in order to detect stars for pointing calibration. A refined prototype, CHEC-S, is currently undergoing testing at the Max Planck Institute for Nuclear Physics in Heidelberg. CHEC-S contains a refinement of the TARGET-based electronics of its predecessor, but replaces the MAPMs with SiPMs. These have the advantage of being more physically durable and easier to operate than MAPMs (not requiring a high-voltage supply), as well as having a less notable drop in Photon Detection Efficiency (PDE) over time [44]. Crucially SiPMs can also operate in conditions with a higher Night Sky Background (NSB) compared to MAPMs, allowing for routine observations under partial moonlight with little degradation in performance. CHEC-S also added a curved front window that helps to attenuate NSB whilst transmitting Cherenkov light [45]. A final prototype SSTCAM engineering camera (with a new, easier to fabricate flat window) is currently undergoing construction, and will be the first SST camera delivered to CTAO.

Both CHEC cameras and the SSTCAM engineering camera have self-calibration LED flasher systems, designed to flat-field the camera. The flashers on the CHEC prototypes were attached to the camera corners and reflected off the secondary mirror of the dual-mirror telescope the camera was attached to; for the final SST Camera (SSTCAM) design the system will have an additional flasher which will shine through the secondary mirror [46].

1.5.3 GCT, ASTRI and the SST Harmonization Process

There are two SST dual-mirror prototype SST designs that CHEC cameras can be attached to, the French led *Gamma Cherenkov Telescope* (GCT), and the Italian led *Astrofisica con Specchi a Tecnologia Replicante Italiana* (ASTRI) instrument. Both share the same Schwarzchild-Couder optical design, which allows for a compact SiPM

camera as well as a more uniform Point Spread Function (PSF) across the Field of View (FoV). Following a harmonisation process, the ASTRI telescope structure with a CHEC-type camera was selected as the basis for the final SST design, taking into account lessons learned from all prototypes. As a result, the CHEC collaboration changed its name to the SSTCAM collaboration. But for the results in this thesis (which are primarily concerned with the camera), these two optical structures are largely interchangeable (in Chapter 3 we consider CHEC-S with GCT as our instrument, whereas in Chapter 5 we consider CHEC-S on ASTRI). We also consider CHEC-S to be broadly representative of the final SSTCAM, as the pixel sizes and configurations (the biggest influence upon our analysis) will likely be very similar.

Figure 1.7: The ASTRI-Horn prototype on Sicily. Image Credit: CTA Collaboration.

1.6 Types of IACT Backgrounds

1.6.1 Hadronic Air Showers

In order to perform IACT γ-ray observations, we need to be able to distinguish astrophysical γ-ray signals from background reliably. At most energies, EAS caused

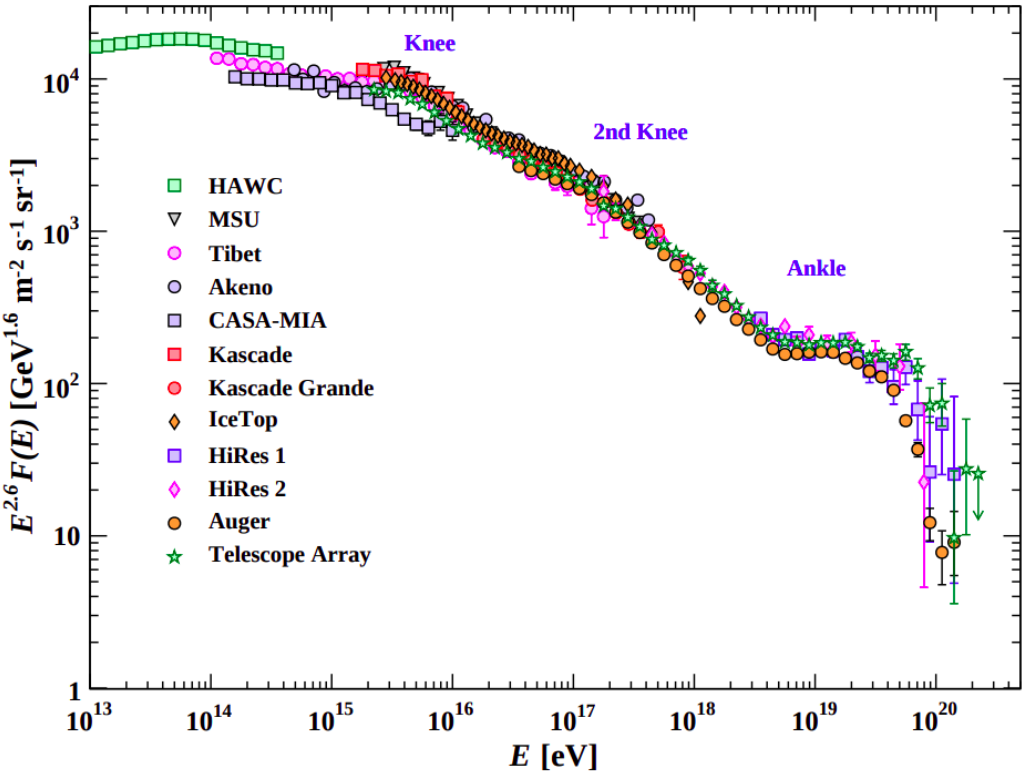


Figure 1.8: The all particle cosmic ray spectrum, taken from [48]. Whilst mainly protons, this spectrum includes both cosmic ray electrons and heavier element cosmic rays. The spectrum is a power law with index -2.7 up to around 3×10^6 GeV, where it steepens to an index of -3.1, a transition feature known as 'the knee'. It then flattens back to an index of -2.7 at 4×10^9 GeV (a feature known as 'the ankle'. The lower energy cosmic rays below the knee are believed to be galactic in origin, most likely a result of acceleration in SNRs. The higher energy cosmic rays above the knee are believed to be the result of cosmic ray acceleration in extragalactic sources such as AGN. The so-called GZK cutoff is theorised to exist at energies around 50 EeV [49], the limit at which a cosmic ray could propagate from another galaxy without interacting with the CMB. For comparison, the Large Hadron Collider operates at a centre of mass energy of 14 TeV for proton-proton collisions, corresponding to an initial EAS proton energy of 1×10^{17} eV.

by incident charged hadrons (most commonly protons) outnumber those caused by photons by a factor of roughly 10,000 [47]. These provide a significant background to IACTs, and are the largest constraint on their sensitivity, as at IACT energies charged cosmic rays (protons, electrons and heavier nuclei) cannot be traced back to their origin due to scattering in the galactic magnetic field.

However, because the quarks contained within protons experience the strong

nuclear force, the interactions hadrons undergo on entering the atmosphere are different to photons. When these charged hadrons collide with atmospheric nuclei, a typical interaction scheme between the two protons [50] is

$$\mathrm{p} + \mathrm{p} \rightarrow \mathrm{N} + \mathrm{N} + n_1(\pi^+ + \pi^-) + n_2\pi^0$$
$$\pi^0 \rightarrow 2\gamma$$
$$\pi^\pm \rightarrow \mu^\pm + \overset{(-)}{\nu}_\mu$$
$$\mu^\pm \rightarrow \mathrm{e}^\pm + \overset{(-)}{\nu}_\mu + \overset{(-)}{\nu}_\mathrm{e}$$

where N represents the resulting fragmented hadrons which are produced along with n_1 charged pions and n_2 neutral pions. The charged pions have a short lifetime of only 26 ns, but the neutral pions' lifetime is even shorter at 5×10^{-17} s because they decay to two photons via the electromagnetic force. Kaon production is also possible in hadronic EAS at higher energies, though the resultant products are the same. Despite their short lifetime, as the charged muons produced in EAS are highly relativistic some can survive to ground level as a result of time dilation effects. The other charged muons decay to electrons, positrons and neutrinos via the weak nuclear force. In a hadronic EAS muons undergo a significant number of inelastic scattering interactions, carrying a larger fraction of the total transverse momentum away from the shower core and resulting in a wider overall EAS for proton showers compared to a γ-ray induced shower [42]. The resultant γ-rays from π_0 decay can themselves undergo pair production creating electromagnetic sub-showers of the hadronic shower core. Similarly to γ-ray EAS, as the charged products from all of these interactions are typically highly relativistic they produce detectable Cherenkov light. But these physical differences in the interactions between γ-ray and hadronic EAS are the basis for the background rejection techniques we'll encounter later in this chapter. Whilst the interaction scheme for hadronic air showers is more complex than for γ-ray or electron induced ones, it is still possible to construct semi-analytic models of their evolution. The most notable of these is the Heitler-Matthew model, details regarding this can be found in [26]. As we'll see later in this chapter, similar hadronic interactions can occur in astrophysical environments.

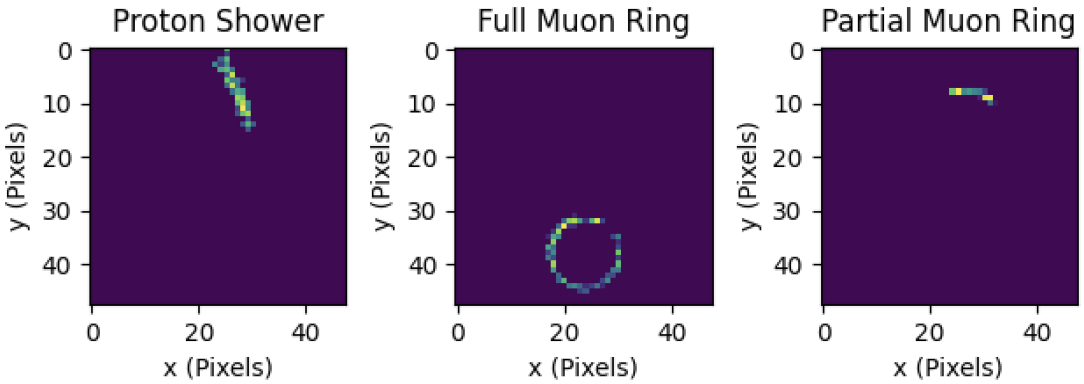


Figure 1.9: A proton shower image, a full muon ring image, and a partial muon ring image. These were generated using *CORSIKA*/*sim_telarray* simulations of CHEC-S on ASTRI. The partial muon ring effect was generated by changing the impact distance from the telescope using the *CORSIKA* CSCAT parameter.

Muons that survive to ground level and that pass directly along the optical axis of the telescope create characteristic Cherenkov 'ring' images in the camera. These muon rings are useful for online calibration as the width of the muon ring image is proportional to the incident particle energy. The observed pixel intensity can then be compared to the expected intensity for a muon with that energy and can thus be used to compute the optical efficiency of the telescope system. Muons that pass off-axis to the telescope are only partially imaged; at sufficiently large impact parameters these partial muon ring images can closely resemble those of γ-ray EAS, creating an additional source of background. An example of such images can be seen in Figure 1.9.

1.6.2 Electron Air Showers

In addition to the charged hadrons previously discussed, electrons can also be a significant source of background at low TeV energies. This is as they are difficult to distinguish from γ-rays and undergo similar interactions. There are two small differences between electron and γ-ray induced air showers. The first is the process by which the primary particles interact on hitting the atmosphere, a primary γ-ray will typically lose all its energy in a single pair production event. A primary electron of the same initial energy can lose energy via production of numerous lower-energy

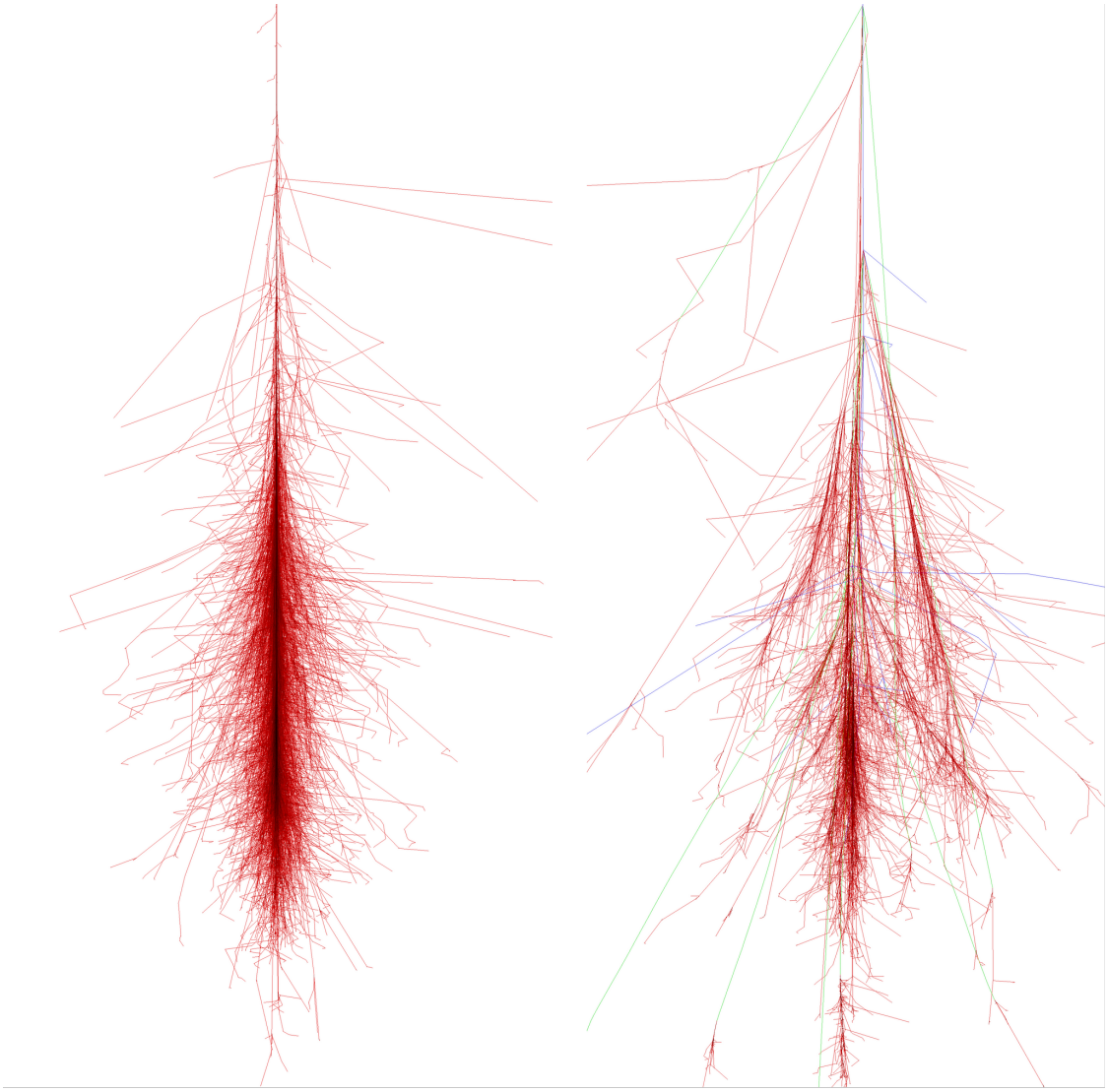


Figure 1.10: XZ plots of *CORSIKA* simulations of particle tracks for 100 GeV photon (left) and proton (right) events. e^+, e^- and γ-rays are shown in red, μ^+ and μ^- in green, and hadrons in blue. Note the wider, less concentrated proton shower which contains muons and hadrons that the γ-ray shower lacks (taken from [51]).

Bremsstrahlung photons which can create electromagnetic sub-showers. This can create a relatively greater number of Cherenkov photons at higher altitudes for electron-induced showers. The other small difference between electron and γ-ray induced showers is the average altitudes at which they first interact [25], which is due to the radiative length of an electron being smaller than the pair production length of a γ-ray. As a result, a primary cosmic-ray electron of the same energy as a primary γ-ray will begin interacting higher in the atmosphere. This results in a higher-altitude shower maximum [52].

1.6.3 Night Sky Background

Not all of the background observed by an IACT can be attributed to EAS from charged particles. Night Sky Background (NSB) is the light detected in IACT camera images that is not Cherenkov light. Modelling NSB is particularly complex, as it consists of photons from a variety of sources (including, but not limited to):

- Atmospheric air glow emission lines, particularly from atomic oxygen, hydroxide and sodium.
- Moonlight.
- Starlight.
- Light from planes and satellites.
- Zodiacal light, along with diffuse galactic light and extragalactic background light (though these contributions are small) [53].
- Light pollution from population centres.

Historically, analytic studies of NSB have been limited, as standard EAS and IACT simulation packages do not attempt to model NSB in a realistic way (for entirely legitimate reasons of computational cost). Most previous work on NSB has focused on direct observations with photomultipliers [54]. We investigate the potential for advanced NSB modelling and how it might aid deep learning analysis in Chapter 5.

1.7 Standard IACT Stages of Event Analysis

Now that we have established the basics of IACT operation, we turn to describing the stages that transform IACT event images into high-level astrophysical data.

1.7.1 Trigger Selection

Trigger selection is the first step in an IACT analysis chain. It is a feature of an IACT array designed to automatically reject events with a low probability of being astrophysical γ-rays in the readouts of the camera. CHEC-S, for example, only reads out the camera photomultipliers if two adjacent superpixels (blocks of four adjacent pixels) passes a comparator check. Being a significant influence on the results in Chapter 4, the VERITAS trigger is slightly more complex. It consists of a single pixel trigger which acts on individual pixels (and includes timing analysis), a camera level trigger which works on the pattern and relative timing of the single pixel triggers, and an array trigger which requires that more than two telescopes are triggered in order such that an event is stored to disk [55]. This array-level trigger in particular significantly reduces the number of false triggers associated with muons passing through the instrument and telescope optics, as it is unlikely a single cosmic ray muon will generate sufficient Cherenkov light to trigger a second VERITAS telescope a few tens of meters away.

1.7.2 Pedestal Subtraction/Calibration

After trigger selection, the next step is low level calibration. Electronic noise present in IACT camera data needs to be removed. For this purpose, measurements of the camera noise are taken when the camera is shut off from the outside environment with the shutter closed (similar to dark framing for optical astronomy). These measured pedestals are then subtracted from the observed live data. Flat-fielding of Cherenkov cameras is also necessary; it is typically performed by firing blue LED flasher units with known intensity profiles at the cameras.

1.7.3 Charge Integration

Once the raw data has been calibrated, the total number of photons collected by the photomultiplier from the EAS must be measured. Charge integration is the process of taking a calibrated PMT trace and extracting the number of photoelectrons. Various schemes exist for performing this operation, CHEC/SSTCAM relies on a cross-correlation template fitting model developed by J. Watson [56].

1.7.4 Tailcut Cleaning

The next step in a conventional IACT analysis is tailcut cleaning. Before conventional IACT event reconstruction and classification, the sensitivity of these tasks to features in images from NSB requires that the images from the Cherenkov cameras are cleaned. The standard method of doing this is tailcut cleaning with two thresholds, whereby a pixel is only included in the analysis if the number of photoelectrons in the pixel exceeds a given threshold, or if it is the neighbour of such a pixel and also has a greater number of photoelectrons than a (typically lower) second threshold [57]. These thresholds require optimisation to achieve a desired event rate, and the implementations of this procedure vary by instrument (see for example [47, 58, 59]). Alternatively methods based upon wavelet image cleaning have been proposed [60], but these are far more rarely used in practice. One of the aims of deep-learning-based image analyses that we explore in Chapters 3 and 4 is to avoid this tailcut cleaning step, as some light from the Cherenkov shower itself is sometimes lost. But previous attempts at performing deep learning analysis without requiring this cleaning step have failed when the event classifier or reconstruction method is exposed to real observations.

1.7.5 Hillas-Parameter-Based Techniques for Event Classification

Hillas Parameters form the basis of the methods for event discrimination in the current generation of IACTs, and were instrumental in obtaining the first reliable gamma-ray source detection of the Crab nebula from the ground [16] (several

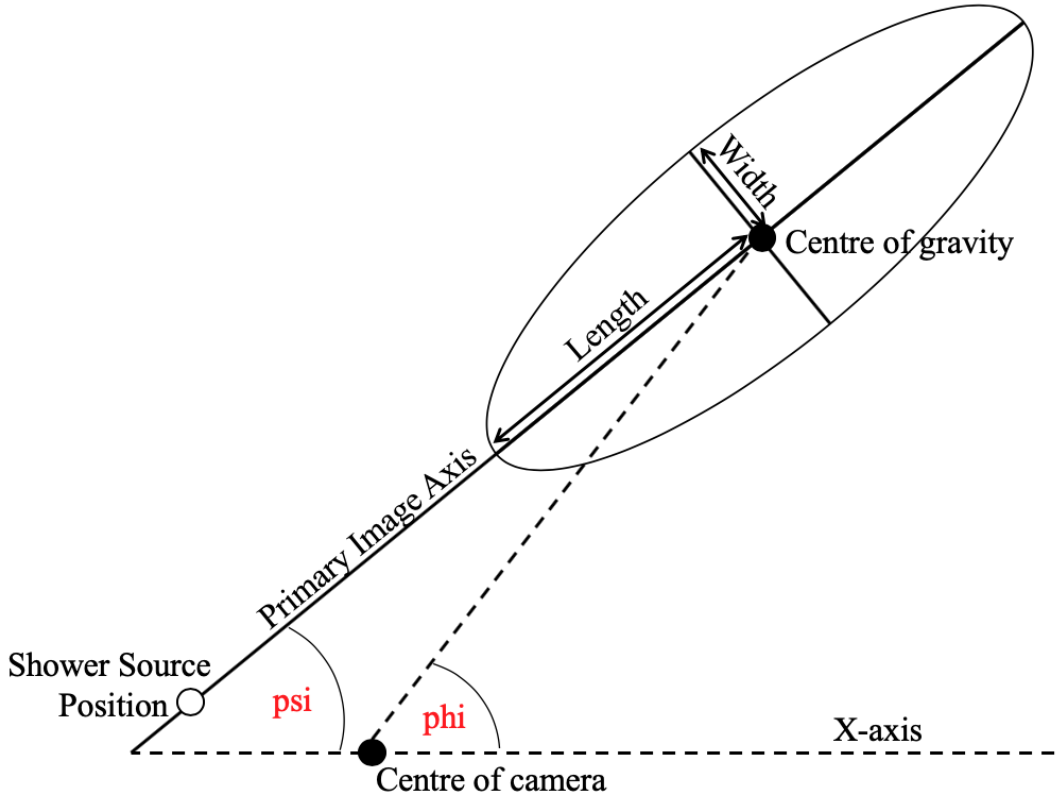


Figure 1.11: The definition of Hillas parameters, taken from [61].

detections of astronomical objects were claimed before this but have since been disproved [20, 62]). They are obtained [42, 63] from the second moments of an IACT camera image (constructed from the total integrated charge for each photomultiplier pixel), defined as

$$\langle x^2 \rangle = \frac{\sum_i I_i x_i^2}{\sum_i I_i} \qquad \langle y^2 \rangle = \frac{\sum_i I_i y_i^2}{\sum_i I_i}$$

where x_i and y_i are the pixel co-ordinates and I_i the associated pixel intensity. From these and the expectation values of x and y one can construct the variance and covariance

$$\sigma_x^2 = \langle x^2 \rangle - \langle x \rangle^2 \qquad \sigma_y^2 = \langle y^2 \rangle - \langle y \rangle^2 \qquad \sigma_{xy} = \langle xy \rangle - \langle x \rangle \langle y \rangle$$

and define the following quantities

$$d = \sigma_x^2 - \sigma_y^2 \qquad a = (d + \sqrt{d^2 + 4\sigma_{xy}^2})/2\sigma_{xy}.$$

The Hillas Parameters relevant for event classification are the width and length of the ellipse which characterize the transverse development of the shower and are defined by

$$W = \sqrt{\frac{\sigma_y^2 + a^2\sigma_x^2 + 2a\sigma_{xy}}{1+a^2}} \qquad L = \sqrt{\frac{\sigma_x^2 + a^2\sigma_y^2 + 2a\sigma_{xy}}{1+a^2}}.$$

In order to take into account information from all of the telescopes in the array, these parameters must be combined into the Mean Reduced Scaled Width and Length (MRSW/MRSL), defined by a sum over all telescopes such that

$$SW = \frac{W - \langle W \rangle}{\sigma_W} \qquad SL = \frac{L - \langle L \rangle}{\sigma_L}$$

$$MRSW = \frac{1}{\sum \omega} \sum SW \cdot \omega \qquad MRSL = \frac{1}{\sum \omega} \sum SL \cdot \omega$$

where σ_W is the spread of the expected width which must be obtained from Monte-Carlo generated lookup tables and $\omega = \langle W \rangle^2 / \sigma_W^2$ is a weighting factor to take into account these tables' accuracy.

Initially Hillas Parameters were simply used to perform data cuts to separate hadronic showers from γ-ray induced showers based on their differing morphology. A combination of trigger selection and simple parameter-based cuts can reduce the signal to background ratio to $\sim$1:1 for a bright source like the Crab Nebula using a current generation instrument like H.E.S.S. [64]. In recent years, more sophisticated decision-tree-based methods such as Boosted Decision Trees (BDTs) or Random Forests (RFs), (taking the MRSL, MRSW, total integrated charge and other parameters derived from Monte-Carlo lookup tables) eventually became the preferred methods for incident particle classification [65]. These are essentially a sophisticated method of performing Hillas Parameter Cuts. For γ-hadron separation, H.E.S.S. [65] and VERITAS [66] typically use BDTs whereas MAGIC normally uses RFs.

An example of a binary tree structure used for IACT event classification with BDTs can be seen in Figure 1.12. In this, events are classified as either Signal (S) or Background (B) by the tree based on whether at each node on the tree the parameters $m_{i,j}$ for a given event are larger than the weights $M^c = (m_j^c, ...)$.

During training, these weights are optimised to produce the desired results for the training data. The ultimate event classification is not performed by one tree, but by an ensemble of trees in order to improve performance (the same is true for RF methods). BDTs iteratively improve tree performance by increasing the weighting for events that were misclassified by the previous tree in all subsequent trees. In the common Adaboost BDT optimisation algorithm, this weighting factor α is given by

$$\alpha = \frac{1 - \text{err}}{\text{err}} \tag{1.8}$$

where err is the fraction of misclassified events in all leaves (endpoints in Figure 1.12) of the decision tree [65]. In RFs, a similar effect as achieved by randomly selecting a subset of input features at each node's splitting point. One of the greatest advantages of decision-tree-based methods is that it is comparatively trivial to determine features in the input data that contributed to a classification; typically for IACT data the most important feature is the MSCW, but at energies below 100 GeV estimation of X_{Max} can be important [65] (X_{Max} for an IACT event can be estimated from the camera images and simple geometric considerations [25]). These feature importances are not as trivial to obtain for the deep learning methods we will encounter in later chapters. BDTs and RFs can also be used to perform directional and energy regression for an event; for further technical details regarding the application of BDTs and RFs to IACT data see [25, 59, 65, 67].

These techniques based on Hillas parameterisation don't take advantage of the full camera image of the EAS, and as such subtle details (such as 'halos' [68] [2]) in the images are not taken into account. This becomes an issue at the sensitivity boundaries that CTA is aiming to considerably improve (particularly in the case of weak or very extended sources), as hadronic and electron induced showers at these energies can closely resemble those generated by γ-rays. This motivates us to investigate new analysis techniques for event discrimination in order to improve the IACT sensitivity, as even small improvements to γ/hadron separation can translate into a significant increase in performance.

[2] Cherenkov light from a charged primary particle.

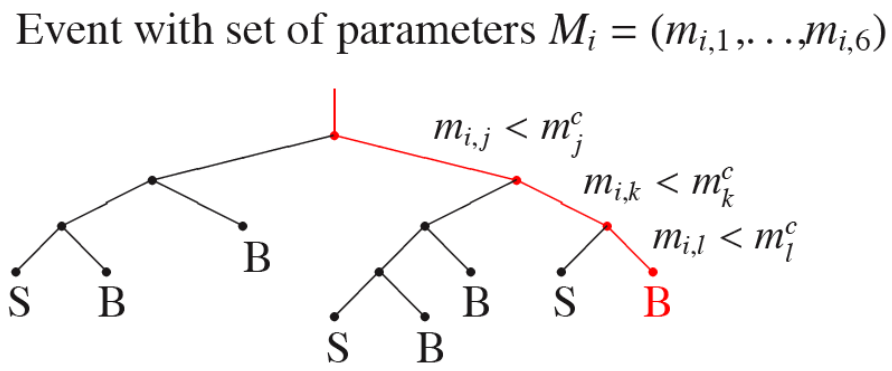


Figure 1.12: An example of a BDT, taken from [65].

1.7.6 Event Selection Cuts and Run Selection

IACT observations are normally conducted in runs, where the IACT array tracks and observes a single source. The duration of these runs varies, but most IACT observatories use between 20 and 30 minutes. This is long enough to obtain a sufficient number of events to accurately model the background, but short enough to ensure that the observation conditions are roughly constant.

Most IACT astronomical analyses begin with Hillas parameter cuts on the data to remove obviously hadronic events (with very high MRSW) and events far from the desired source (so-called theta-squared cuts). It is also common to remove runs or events from an IACT analysis for a variety of reasons. The data from a run may be low quality due to poor weather conditions, hardware or tracking issues or large zenith angle (at which simulation accuracy drops due to the complexity of modelling the atmosphere correctly). Most IACT analyses use multiple runs per source, though the work in Chapter 4 considers only a single VERITAS run as we attempt to match the data more closely with simulations.

1.7.7 ON/OFF Region and Reflected Region Analysis

So far we have focused on Cherenkov camera based background rejection, however this only reduces the signal to background rate (by around a factor of 100) to ∼1:1 for bright sources. As such, there is a need for a higher level background rejection method, similar to aperture photometry in optical astronomy. The simplest method possible is to take one ON region and one OFF region at the same right ascension but differing declination and use this to compute the γ-ray excess. The disadvantage of this is the loss of time on source. An alternative developed by the Whipple observatory is so called 'Wobble Mode', where the telescope wobbles around the source in declination, allowing for more time on source. However, this is complicated by two factors, the comparatively poor angular resolution of IACTs, and the significant extension of some sources such as the SNR RXJ 1713.7-3946. As such, the HEGRA collaboration [17] developed so called reflected-region analysis, where multiple non-overlapping OFF regions (at different distances from the ON region but of the same angular size) are used. This allows for better statistics compared to a singular OFF region and allows for measurement of a potentially non-uniform night sky background across the field. This is the technique we use in Chapter 4. However, such methods can run into difficulties for very extended objects such as Geminga, as the size of the source emission region might exceed the size of the FoV [69]. In such scenarios a 'FoV' background model, which models the radial acceptance of the IACT system, can be used. The technical details of this are beyond the scope of this thesis; for more detail we refer the reader to [64, 69].

1.7.8 Pointing Corrections

Also necessary for reliable IACT operation is pointing calibration, as the size of the telescopes means they can bend slightly [70]. This is typically performed by reconstructing the positions of stars using CCDs attached to the telescope structure, though SSTCAM has a slow signal chain to perform this (we explore the consequences of this in Chapter 5). These pointing corrections allow for positions of

γ-ray sources to be reconstructed to within $2 - 3'$ at 100 TeV; this can be important, especially for sources with an extended morphology [32, 71].

1.7.9 CTA Data Analysis Levels

The proposed CTA analysis pipeline consists of a number of data analysis levels. Data processors exist to transform data at each level into the next. Only some of these levels are relevant for deep learning analyses. The current CTA data structure consists of [56]:

- **r0 data**, which is the uncalibrated raw data generated by the Cherenkov camera.
- **r1 data**, which is online, calibrated camera data, calibrated using a scheme specific to the camera.
- **dl0 data**, which is calibrated waveform data along with event metadata, and is zero suppressed to save bandwidth and storage space.
- **dl1 data**, which consists of charge and peak time information extracted from the calibrated waveforms, as well as Hillas parameters derived from them.
- **dl2 data**, at this stage the telescope multiplicity is dropped and the overall events are characterised (based on their Hillas parameters extracted during the dl1 stage) into shower parameters including energy, direction, and source particle
- **dl3 data**, at this stage the dl2 events are sorted into sets of (for example) γ-ray candidates. This also includes any relevant instrument response data needed for higher level analysis.
- **dl4 data**, which consists of fully processed event lists, ready for high level astronomical analysis.
- **dl5 data**, which consists of high level analysis data products produced from astronomical analysis, such as the CTA galactic plane survey.

The CTA analysis chain will therefore reduce the raw amount of data from the telescopes from Petabytes to data scales tractable with a laptop.

1.8 Alternatives to Hillas-Parameter-Based Techniques

In this section we briefly consider the existing alternatives to Hillas analysis for IACT event classification and reconstruction.

1.8.1 Template Analyses and Model++

Template analyses [68] [72] [73] [74] form the current main alternative to Hillas parameterisation. They rely on the generation of a library of template IACT images. The images from the IACTs are then compared to the templates via a pixel-by-pixel minimisation process, such that the template most resembling the IACT image can be found and thus the shower properties inferred. In particular, the H.E.S.S. Model++ analysis chain uses chi-squared fitting against semi-analytic models of γ-ray showers as a background rejection method [68]. As Model++ is not open-source, it is not currently being considered as an analysis method for CTA, and its efficacy in the SST energy range has also not been tested. At the time of writing there is not a major effort within CTA to recreate it.

1.8.2 ImPACT

The most widely used tool for template analysis is the *Image Pixelwise fit for Atmospheric Cherenkov Telescopes* (ImPACT) code [74]. This works by comparing the shower images to interpolated templates generated using Monte-Carlo simulations. ImPACT was developed for the (previously unprecedentedly large) H.E.S.S. CT5 telescope as Hillas parameter based techniques are known to be unreliable at photon energies lower than 50 GeV. The reconstructed event parameters are the combination of shower parameters for the template which minimizes the total

negative log likelihood $\ln \mathcal{L} = \sum_{pixel,i} -2 \ln P(s_i|\mu, \sigma_p, \sigma_y)$ (maximizes the likelihood) over every pixel in the camera for every telescope in the array, where

$$P(s_i|\mu, \sigma_p, \sigma_y) = \sum_n \frac{\mu^n e^{-\mu}}{n!\sqrt{2\pi(\sigma_p^2 + n\sigma_\gamma^2)}} \exp\left(-\frac{(s-n)^2}{2(\sigma_p^2 + n\sigma_\gamma^2)}\right). \tag{1.9}$$

Here, n is the photoelectron number, σ_p is the pedestal width (the width of the charge distribution under pure noise), σ_γ represents the photomultiplier resolution, s is the number of signal photoelectrons detected and μ is the predicted number of photoelectrons from the template. In practice, ImPACT is only used for directional and energy reconstruction in H.E.S.S. due to concerns about its sensitivity to NSB photons (a problem similar to the real observations problem experienced by Convolutional Neural Network (CNN)-type methods that we will explore in Chapters 2, 3 and 4). The H.E.S.S. ImPACT chain uses a BDT based on Hillas Parameters for γ/hadron separation. The ability of ImPACT to utilise stereoscopic sets of complete IACT images can significantly improve directional reconstruction compared to Hillas-parameter-based methods, especially at low energies [74].

The difficulty in using ImPACT for CTA is the requirement of having a sufficient number of template simulations to cope with every possible type of event with a given energy and incident direction for a large number of telescopes, and the computational cost of searching through those simulations to find the optimal event parameters. That said, there is an open-source implementation of ImPACT available within *ctapipe* [61, 75].

1.9 *ctapipe* and the Prototype CTA Analysis Chain

CTA is an instrument that is currently in the development phase, and this is true both of the instruments and the analysis pipelines that go with them. For much of the writing of this thesis a full, pythonic analysis pipeline for CTA did not exist, and CTA analyses were a complex mixture of modern code and software written for previous instruments. Additionally, a detailed, complex Monte-Carlo validation procedure was underway for the simulations of the prototype cameras.

ctapipe is a python-based tool developed as an open-source prototype (which follows modern code practices for version control and code development) for the low-level analysis of CTA data. It is now recognised as a critical tool for CTA pipeline development, and is used in much of the analysis in this thesis. It provides tools to both handle *CORSIKA*/*sim_telarray* simulations, perform data calibration and charge extraction, and to extract Hillas and muon ring parameters for events. It also contains *astropy* based tools to compute camera pixel positions on sky, needed for the NSB work we present in Chapter 5. *ctapipe* can even be used to analyze data from existing and historical IACT arrays; elements of our VERITAS work in Chapter 4 used reverse engineered code from *ctapipe*.

The current open-source prototypes for high-level analysis (following similar development practice to *ctapipe*) are *gammapy* and *ctools*; of these *gammapy* has been selected as the official high-level science tool for CTA. These allow one to generate source significance maps, lightcurves and spectra using IACT event lists. For simplicity, our *VERITAS* work utilised the older, more stable *Eventdisplay* tools for the work presented in Chapter 4. But in time, high level data from all existing IACTs could be analysed with these new open-source packages.

1.10 Methods and Environments of Astrophysical γ-ray Emission

1.10.1 Inverse Compton Scattering

In this section we briefly discuss how and where astrophysical γ-rays are produced. The majority of astrophysical γ-rays are believed to be produced by Inverse-Compton (IC) scattering. IC scattering describes the process by which low energy photons are scattered by ultra-relativistic electrons, significantly increasing the energy of the photons. The so-called seed photon fields accelerated by the electrons vary; they can be a population of photons produced through synchrotron emission that are then hit by the high energy electrons that emitted them (so-called synchrotron self-Compton emission), can be the result of starlight, or originate from the Cosmic

Microwave Background (CMB). In the rest frame of the electron, the ratio of final photon energy over initial energy is given by

$$\frac{E_f}{E_i} = \gamma(1 + \beta\cos(\theta)) \tag{1.10}$$

where γ is the Lorentz factor of the electrons and θ is the deflection angle. Transferring to the observer frame this works out at around $\frac{E_f}{E_i} \sim \gamma^2$. Using this, combined with considerations of the energy density of the photon field (u_{rad} in the observer frame), it can be shown [76] that the energy loss rate from a relativistic electron travelling at speed β (where $\beta = \frac{v}{c}$) due to IC scattering is

$$\left(\frac{dE}{dt}\right)_{\mathrm{IC}} = \frac{4}{3}\sigma_T c u_{\mathrm{rad}}\left(\beta^2\right). \tag{1.11}$$

This results in the spectral emissivity from IC scattering of an isotropic photon field of single photon frequency ν_0 by a power law distribution of electrons being given by

$$I(\nu)d\nu = \frac{3\sigma_T c v N(\nu_0)}{16\gamma^4\nu_0^2}\left(2c\ln\left(\frac{\nu}{4\gamma^2\nu_0}\right) + \nu + 4\gamma^2\nu_0 - \frac{\nu^2}{2\gamma^2\nu_0}\right)dv \tag{1.12}$$

where $N(\nu_0)$ is the number density of photons, σ_T is the Thompson cross section and ν is the frequency of the γ-rays [77].

Whilst these expressions for IC scattering are valid for most currently observed IACT sources, the SSTs will observe sources emitting γ-rays in the higher-energy Klein-Nishina regime [78] (where $h\nu > m_e c^2$), which we will now explore. In general, for an isotropic (same in all directions) and homogeneous (same at all points) electron energy distribution with a differential energy density $dn = n(\epsilon)d\epsilon$ it's useful to express the energy of the scattered photon (ϵ_1) in terms of the initial electron energy E_i such that $\epsilon_i = \gamma mc^2 E_i$, where m is the electron rest mass. This results in a general expression for the photon spectrum per electron being [77]

$$\frac{dN_{\gamma,\epsilon}}{dtdE_i} = \frac{2\pi r_0 mc^3 n(\epsilon)d\epsilon}{\gamma\epsilon} \times \left[2q\ln q + (1+2q)(1-q) + \frac{(\Gamma_\epsilon q)^2}{2\times(1+\Gamma_\epsilon q)}(1-q)\right] \tag{1.13}$$

where r_0 is the classical electron radius with the value of $r_0 = 2.82 \times 10^{-13}\,\mathrm{cm}^2$, and Γ_ϵ and q are defined by

$$\Gamma_\epsilon = \frac{4\epsilon\gamma}{mc^2} \tag{1.14}$$

and

$$q = \frac{E_i}{\Gamma_\epsilon(1 - E_i)} \tag{1.15}$$

respectively. Due to kinematic constraints, the range of possible values of E_i and correspondingly q are

$$1 \gg \frac{\epsilon}{\gamma mc^2} \leq E_i \leq \frac{\Gamma_\epsilon}{1 + \Gamma_\epsilon} \tag{1.16}$$

$$1 \gg \frac{1}{4\gamma^2} \leq q \leq 1. \tag{1.17}$$

By re-defining the scattered photon energy in terms of its maximum value, and introducing a scaling factor $\hat{E}_i$

$$E_i = \frac{\Gamma_\epsilon \hat{E}_i}{1 + \Gamma_\epsilon} \tag{1.18}$$

we can obtain a photon distribution function for the scattering in the Klein-Nishina limit ($\Gamma_\epsilon \gg 1$), where an electron in the distribution loses the vast majority of its energy in a single interaction with a photon [77] (the case for many very-high-energy γ-ray sources)

$$F(\hat{E}_i, \Gamma_\epsilon)\Big|_{\Gamma_\epsilon \gg 1} = \frac{1}{\ln \Gamma_\epsilon}\left(1 + \frac{(\Gamma_\epsilon q)^2}{2 \times (1 + \Gamma_\epsilon q)}(1 - q)\right) \tag{1.19}$$

For the case that $\hat{E}_i$ is not near 1 and $\Gamma_\epsilon \gg 1$ there is an additional solution

$$F(\hat{E}_i, \Gamma_\epsilon)\Big|_{\hat{E}_i \text{not near } 1, \Gamma_\epsilon \gg 1} = \frac{1}{\ln \Gamma_\epsilon}\left(1 - \hat{E}_i + \frac{1}{1 - \hat{E}_i}\right), \tag{1.20}$$

in both cases $F(\hat{E}_i, \Gamma_\epsilon)$ is normalised such that

$$\int_0^1 F(\hat{E}_i, \Gamma_\epsilon)\, d\hat{E}_i = 1. \tag{1.21}$$

From these, expressions for the spectrum resulting from IC scattering from a power law distribution of electrons in the extreme Klein-Nishina limit similar to Equation

1.12 can be derived, for full details see [77]. From quantum electrodynamics Klein and Nishina [78] also derived the total scattering cross section in this limit σ as a function of the variable $x = (h\nu_i)/(m_e c^2)$

$$\sigma(x) \approx \frac{3\sigma_T}{8x}\left(\ln 2x + \frac{1}{2}\right), \text{ for } x \gg 1 \tag{1.22}$$

compared to the Thomson (low-x) regime, where the equivalent formula is

$$\sigma(x) \approx \sigma_T(1 - 2x + ...), \text{ for } x \ll 1. \tag{1.23}$$

1.10.2 Hadronic Emission

The second method of producing astrophysical γ-rays is through hadronic interactions. As we saw for EAS, proton-proton interactions can produce π^0 particles which decay to a pair of γ-rays. Hadronic emission processes are challenging to discriminate from IC scattering in some sources, particularly SNRs [71]. The recent detection of an astrophysical neutrino (IceCube-170922A) by IceCube that was coincident with a γ-ray flare of the blazar TXS 0506+056 suggests that some γ-rays from AGN are produced by hadronic processes [28].

1.10.3 SNRs, Pulsar Wind Nebulae and Pulsars

Now that we have considered the physical mechanisms by which astrophysical γ-rays are produced, let us consider the environments where this production might occur. Given the innately high energies involved, SNRs are prime sources of galactic γ-ray emission. SNRs are also suspected of being one of the galactic sources of the highest energy cosmic rays (so-called PeVatrons). Most notably the SNR RXJ1713.7-3946 has been detected to be spatially extended by H.E.S.S.; CTA might help to understand the emission mechanisms present in this source [71]. Pulsar Wind Nebulae (PWN) such as the one in the Crab Nebula are another related class of γ-ray sources. These are where a wind from a central pulsar interacts with the surrounding ejecta to produce high-energy emission [79]. PWN are often embedded within a larger SNR (as is the case for the Crab Nebula), but a certain fraction of PWN are so-called TeV Halos. These are highly-evolved (believed to be older than

100 kyr) systems where the pulsar has passed through the ejecta of the associated SNR and is now accelerating electrons in a spatially isolated region [80]. Geminga is a likely candidate for such a system [69]. Magnetospheric IC γ-ray emission from Pulsars themselves is also possible, either from the poles or the so-called 'outer-gap' [81], though so-far such emission has only been observed from the ground with the H.E.S.S. CT5 telescope [82]. This suggests such emission will not be observable with the CTA SSTs as they operate in a much higher energy range, though with the unprecedented sensitivity of CTA serendipitous detections may occur.

1.10.4 γ-ray Binaries

γ-ray Binaries are similar to their X-Ray counterparts, where a compact object (such as a neutron star or black hole) accretes matter from a companion star. γ-ray binaries differ from X-Ray Binaries typically because of the comparatively high mass of the companion star (typically an O or B class giant), as well as their higher-energy emission. Very few of these sources have been detected by IACTs ($\sim$10 at the time of writing), and they are generally fairly poorly understood [1], with it often being unclear if the accreting compact object is a neutron star or a black hole. CTA will likely help us discover more of these objects and increase our understanding of those that have already been detected.

1.10.5 Active Galactic Nuclei

AGN are known sources of GeV and TeV γ-rays, typically these are blazars (where the jet of the AGN directly points towards Earth). These blazars fall into two categories based on how energetic they are, these are known as BL Lacs (for the low energy case) or Flat Spectrum Radio Quasars (FSRQs, for the high energy case). The characteristic double-peaked Spectral Energy Distribution (SED) of blazars is evidence for there being some synchrotron-self-Compton emission from these objects. Flares (unusual periods of bright emission) from AGN are also observed as a class of extragalactic γ-ray transient [see for example 28].

1.10.6 Galactic and Extragalactic Transients

GRBs are typically bright, but comparatively short lived in their γ-ray emission. They are believed to originate from neutron star collisions or core-collapse supernovae; differences in the redshift distributions of short-lived and long-lived GRBs is evidence to support this hypothesis [76]. GRBs have only recently been observed from the ground, partly due to the technical challenge of turning the instruments quickly enough to detect the prompt emission, but also due to the complex factors involved in background rejection and the fact that most GRBs do not have spectra extending into the TeV. This first detection of both prompt emission from a GRB, and the associated lower-energy afterglow from the ground, was by MAGIC and H.E.S.S. in 2019 [11].

The first galactic transient detected by an IACT array was the recurring Novae RS Ophiuchi by H.E.S.S. [H.E.S.S. Collaboration, in prep.]. Emission from Novae is typically a result of a thermonuclear explosion of the surface of an accreting white dwarf, although the source of the TeV photons is potentially unclear. This detection was somewhat unexpected, but suggests CTA might be able to detect a greater population of galactic transient objects.

1.10.7 Diffuse Emission

Both galactic and extragalactic diffuse γ-ray emission are observered by IACTs. In the galactic case it is believed the majority of the emission results from hadronic emission due to collisions of cosmic rays with galactic gas [83]. The extragalactic case is less well understood [83], with the majority of emission likely due to unresolved blazars, radio galaxies and star-forming galaxies, but some additional components may be due to dark matter annihilation or cosmic ray interactions [83]. Dark matter annihilation is a scenario predicted by a number of dark matter models, which suggest dark matter particles are their own antiparticles. Such dark matter particles will annihilate with each other, producing high-energy γ-rays [1].

1.11 The Key Science Projects of CTA

Now that we have discussed the origin and production of γ-rays, we will briefly discuss the observing strategy for the scientific investigations of CTA. The CTA observing time not allocated to open observations or Target of Opportunity (ToO) observations will be dedicated to a number of Key Science Projects (KSPs). In this section we briefly explore the science potential for each of these campaigns.

1.11.1 The Galactic Centre KSP

CTA observations are planned to be made of the central square degree of the Milky Way for 500 hours [1], with a further 300 hours spent observing the region up to 10 degrees in galactic latitude. It is anticipated that this will be one of the first major observing campaigns for CTA due to its implications for other CTA surveys. In particular, the galactic centre is considered one of the best places to search for evidence of dark matter annihilation. These observations of the galactic centre should be sufficient to obtain either a detection or an upper limit for many of the current best dark matter models [1]. Additionally, these observations hope to uncover the nature of the galactic centre γ-ray source, and to probe the nature of very high energy particle acceleration in the galactic centre. This is key to understanding the nature of the Fermi Bubbles, which we discussed earlier in Section 1.1.2.

1.11.2 The Galactic Plane Survey (GPS) and the Cosmic Ray PeVatron KSP

Around 1800 total hours of observing time will be spent performing observations of the galactic plane, with a sensitivity of at least 4.2 mCrab [1] (an improvement of 5-20 times relative to current instruments). It is planned that this survey will lead to the discovery of new galactic sources of γ-rays, including SNRs, γ-ray binaries and PWN. Additionally, it will provide new constraints for models of the galactic diffuse emission. It is anticipated that the maps generated by the GPS will be periodically made public, in order to maximize their multi-wavelength science potential. Additionally, the PeVatron KSP aims to perform dedicated, deep observations of known γ-ray

sources with particularly hard spectra (such as RX J1713.7-3946), and search for diffuse γ-ray emission around known SNRs, in an attempt to understand the origins and acceleration methods of very-high-energy (PeV) cosmic rays [1].

1.11.3 The Large Magellanic Cloud Survey and the Star Forming Systems KSP

Around 600 hours of CTA observation time will be spent on observing the Large Magellanic Cloud (LMC). In particular, the LMC is interesting at TeV energies due to its high rate of star formation; the LMC produces about one tenth of the number of stars as the Milky Way, despite having only 2% of the Milky Way's volume [1]. As such, it contains a disproportionate number of SNRs and PWN. These can be observed more easily in the LMC than in the Milky Way due to the absence of source confusion, interstellar absorption and line of sight crowding at the LMC's high galactic latitude [1]. Other observations of galactic star forming regions (for example Westerlund 1) are also planned as part of the star forming systems KSP, in order to investigate the link between star formation and particle acceleration.

1.11.4 The Extragalactic Survey and the AGN KSP

One of the largest time allocations, the extragalactic survey will cover 25% of the sky down to 6 mCrab sensitivity [1]. It will aim to achieve an unbiased census of γ-ray AGN, observe fast-flaring sources and gamma-ray busts serendipitously, and measure the anisotropies in the electron cosmic ray spectrum.

An AGN KSP is also envisaged to perform long term monitoring of a select set of AGN, in order to understand how the different types of AGN are related and how their particle and γ-ray acceleration mechanisms work [1]. By extension, these observations should also act as a probe of the intergalactic magnetic field and the extragalactic background light. The AGN KSP is also a means of performing tests of fundamental physics; in particular CTA should be able to place constraints on Lorentz Invariance Violation (LIV). LIV is a theory which predicts the speed of light originating from astrophysical γ-ray sources is a mild function of energy

as a result of quantum gravity effects [1]. CTA will also examine the possibility of the production of Axion-like particles (another class of dark matter candidate) through the monitoring of AGN [1].

1.11.5 The Transient KSP

Somewhat intertwined with the extragalactic survey, CTA will have the capability to respond to transient events from both internal and external triggers. These external triggers will be sent by gravitational wave, neutrino, radio, optical, x-ray and other γ-ray telescopes around the world when they detect a transient object. If the trigger is likely to be of scientific value and is within observability constraints, CTA will immediately perform follow-up observations to help constrain the emitting object's nature [1]. Likewise notifications of transient events serendipitously detected by CTA will be sent to partner observatories. Key to CTA's ability to detect transients is CTA's divergent pointing mode, where not all telescopes in the array point at the same patch on the sky. This slightly reduces CTA's spectral and angular resolutions [1], but allows CTA to tile a larger region of sky relative to the (standard) parallel observation mode. This is important if the estimated source position from the observatory that provided the external trigger has a large uncertainty.

1.12 Thesis Structure

The remainder of this thesis is structured into five chapters. In Chapter 2, we introduce deep learning methodologies, before exploring past and concurrent work on their applications for IACTs, including introducing multiple problems with such applications that are currently unsolved. In Chapter 3 we explore the (previously unconsidered) option of utilising photosensor timing information as a basis for a deep learning event classification method, utilising the advanced capability of waveform storage possible with SSTCAM. In Chapter 4, we explore the difficulties in using charge-based deep learning classifiers with existing data from VERITAS, exploring in detail the complication of optimising deep learning classifiers and the complexities arising from not wishing to perform tailcut image cleaning. Chapter 5

concerns a detailed study into the likely NSB conditions that will be experienced by SSTCAM. Whilst NSB represents a significant challenge to deep learning event classification and reconstruction methods, the work within this chapter is largely self-contained. Finally in Chapter 6 we investigate future prospects for analyses in these fields and conclude.

2

Deep learning methods and their applications for IACTs

Contents

Abstract

Deep learning analysis methods based upon CNNs are becoming increasingly widely utilised throughout the physical sciences. In this chapter we explore the basic properties of such methods, before reviewing past and concurrent work on utilising these methods with IACT data. We then go on to explore the known issues with applying these methods to IACT data in depth.

2.1 Introduction

Machine learning has been critical to the operation of the current generation of IACTs, H.E.S.S., MAGIC and VERITAS. As explored in Chapter 1, BDT- and RF-based event classification methods used currently have offered a significant sensitivity improvement over conventional Hillas parameter cuts alone. However, when dealing with a new generation of IACT cameras with improved pixel density and timing precision, these BDT/RF methods have a notable shortcoming. This is that they do not take account of all the image and timing information available to them. This is a very similar situation to IACT astronomy in the early 2000s, as the construction of the current generation of IACTs with larger cameras necessitated the development of BDT/RF and template fitting techniques. As such, the machine learning task force in CTA was set up to investigate whether it is possible to obtain a sensitivity improvement for CTA by using new deep learning techniques. At the same time, deep learning is a very rapidly advancing field; new models and methods of handling data are constantly becoming available (see for example [84, 85]), but these new methods must be adapted and tempered for IACT analysis. In this chapter, we explore the current state of the art in machine learning methods in greater depth, before going on to explore past work on implementing newly available deep learning techniques for IACT data, and some of the problems that have already been found.

New deep learning methods, which are neural network methods with many layers, or equivalently a high degree of feature abstraction, are frequently associated with 'AI'. Despite this, true emergent AI that can interpret universal input data and formulate meaningful responses remains science fiction, with no serious research currently being performed [86]. In practice, most supervised deep learning techniques are simply an advanced form of pattern recognition, with weights being adjusted to provide desired responses from known training data. When exposed to new data outside of their original training data sample, these methods often run into difficulties, an area of computer science research referred to as 'Domain Adaptation' [87]. We will begin to encounter such difficulties in Chapter 4.

2.1.1 Types of Machine Learning

Before addressing substantially the novel elements of this thesis, it is necessary for us to introduce a number of basic concepts and terms. **Supervised Machine Learning** refers to the areas of automated pattern recognition whereby labelled **Training Data** exists such that a configurable function can be optimised to map an input to a desired output. **Unsupervised Machine Learning** refers to the field of machine learning research where the underlying relationship between input and output is not known a-priori, such as in the case of clustering analysis. Whilst this can have some analysis uses in γ-ray astronomy [42], such as source detection, it is not the focus of the work in this thesis. In our supervised learning case the configurable function takes the form of a **Neural Network**, which consists of layers of **Artificial Neurons** that evaluate a set of inputs (with associated configurable weights and biases) against output **Activation Functions**. **Training** describes the process by which the weights in a supervised machine learning algorithm are manipulated such as to minimise a loss function against the labels provided for training data, an **Epoch** is defined as one pass through the entirety of the training dataset during training, typically deep learning methods are trained for many epochs. **Testing** describes the process of determining these trained networks' efficacy against previously unseen data. It is also an option to additionally define **Validation Data** to test against epoch by epoch during training, this data can then not be used as part of the final testing. In practice with the ConvLSTM2D classifiers we present later in the thesis, we found validation data to be noisy and a misrepresentation of final test performance, so in Chapter 3 validation data is not used, but in Chapter 4 it is used. Finally, **Graphics Processing Units (GPUs)** are currently the optimal hardware for performing most deep learning analyses in physics. Whilst a detailed explanation of the algorithmic details of these methods is beyond the scope of this work, we refer the interested reader to [88–90] for further information.

2.2 Basic Machine Learning Concepts and Definitions

2.2.1 Neural Networks

Neural networks are a class of machine learning algorithm designed to mimic the human brain's ability to recognise patterns. Their most fundamental building block is the artificial neuron, which is mathematically described by

$$O = f(\sum_{i=1}^{n} w_i x_i + b_i) = f(W^T \cdot x + b) \tag{2.1}$$

where O is the output from evaluating the 'activation function' f using the n inputs x_i (the ith component of the input vector x) and their associated weights w_i (the ith component of the weight vector W) and b_i (using the same notation) is a bias term [91]. For the rest of this thesis we choose to set b_i to 0 for simplicity. But future studies considering class imbalance [92], or trying to correct for bright stars in the camera field (that we see in Chapter 5), might choose to set this differently. The most basic neural networks, known as Multi-Layer Perceptrons (MLPs), are constructed from fully interconnected layers of these artificial neurons. Fully-connected MLP-like layers are often included towards the end of CNN-type analyses, where they are known as Dense layers. An input vector containing n parameters is fed to the first layer of neurons, whose outputs O are then used as the inputs for additional 'hidden' layers of neurons. These the last of these hidden layers feeds into an output layer which outputs a vector of size m, which in the case of particle classification, is the pseudo-probability of the input vector belonging each of the m possible classes. One then 'trains' the network by minimising a loss function $\mathcal{E}$ through iteratively adjusting the weights w_i. The network can finally be 'tested' by evaluating its classification accuracy against a different set of events.

Whilst it is possible to train these MLPs with the Hillas parameters for a set of IACT images (though they tend to be inferior classifiers to BDT/RF methods due to the fact that they don't prioritise parameters based on separation power [65]), using every pixel value of an image as an input to an MLP tends to be computationally prohibitive.

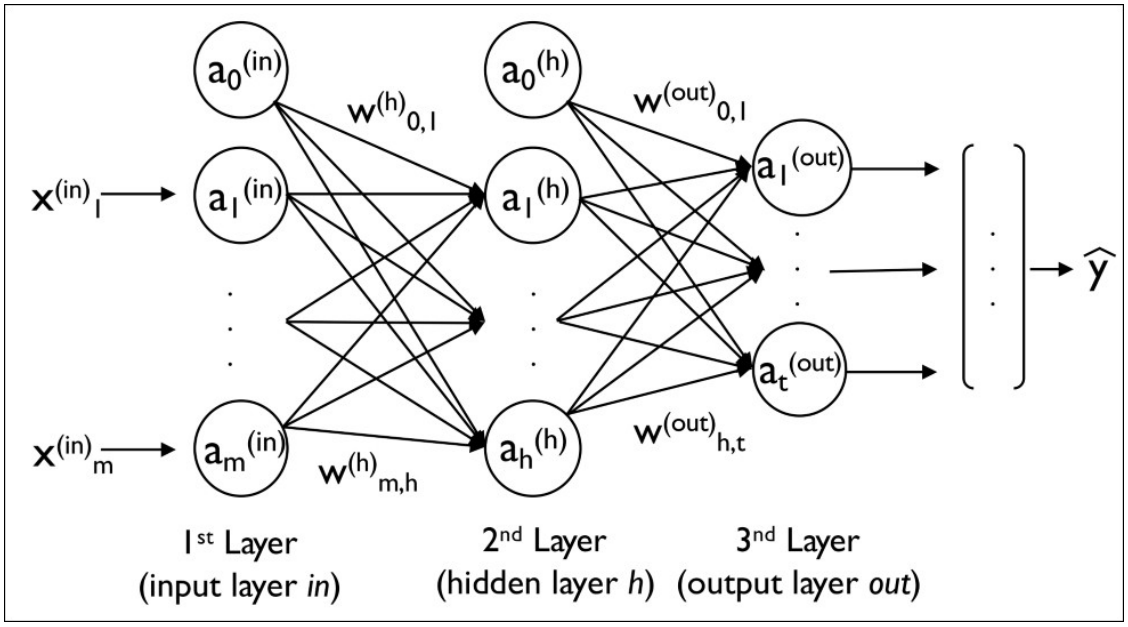


Figure 2.1: An example of a MLP, taken from [93]. In this example, predictions $\hat{y}$ are generated from inputs $x^{(in)}$ using artificial neurons a and weights w.

2.2.2 Loss Functions and Metrics

In order to train a neural network, one needs to define a loss function for the training algorithm to optimise against. For binary, two class classification, the appropriate choice for this is the binary cross entropy (BE) between the actual known classes t_i and the score assigned by the network for an event having that class s_i [94]. This is defined over N events as

$$\mathrm{BE} = -\frac{1}{N}\sum_{i=1}^{N} t_i \ln s_i + (1 - t_i) \ln (1 - s_i). \tag{2.2}$$

Equation 2.2 can be generalised to perform classification for an arbitrary number of classes n, in which case it is known as a **Categorical Cross-Entropy Loss Function (CE)**. This is defined [94] as:

$$\mathrm{CE} = -\sum_{i}^{N} \ln \left(\frac{e^{t_i}}{\sum_j^n e^{s_{ij}}} \right) \tag{2.3}$$

where there are N events, s_{ij} is the CNN score for each of the possible event classes for the event i and t_i is the CNN score for the target (true) class for that event. This equation relies on the CNN input labels being 'one-hot' encoded such that the label for class 0 is represented as $[1, 0, ...]$, class 1 as $[0, 1, ...]$ so on [95]. In

practice, during the training of the ConvLSTM2D networks in Chapters 3 and 4, there is an additional regularization penalty added to the total loss used due to the L2 regularization we use in the first two ConvLSTM2D layers. This regularization is designed to prevent network weights from taking extreme values.

Term	Definition
True Positive (TP)	True Label Positive and Classification Positive
True Negative (TN)	True Label Negative and Classification Negative
False Positive (FP)	True Label Negative and Classified Positive
False Negative (FN)	True Label Positive and Classified Negative

Table 2.1: Definitions of Event Classifications for binary classification, taken from [96].

For the purposes of evaluation of network performance, **Categorical Accuracy** [94] is defined as the percentage of events for which the *Argmax* of the one-hot-encoded predicted label is the same as the *Argmax* for the one-hot-encoded true label (the definition of binary accuracy is similar for 2-class classification [94]). Given the definitions for binary classification in Table 2.1, the **True Positive Rate (TPR)** and **False Negative Rate (FNR)** are defined by [96]:

$$TPR = \sum_{Events} \frac{TP}{TP + FN} = 1 - FNR \tag{2.4}$$

and the **True Negative Rate (TNR)** and **False Positive Rate (FPR)** by

$$TNR = \sum_{Events} \frac{TN}{TN + FP} = 1 - FPR. \tag{2.5}$$

For multi-class (> 2 class) classification, such as in Chapter 3, we use a one-versus-all approach where for each individual class the classifications are treated in a binary way (such as γ-ray or not-γ-ray) [96, 97].

2.2.3 Universal Approximation Theory

Universal functional theory states that a neural network can be made to replicate the behaviour of any function, provided sufficient training data exists, and that the network has sufficient capacity (ideally infinite depth, width and neuron/filter count). In practice, this is limited by finite training data and the capacity of a neural network being finite. Even then, training extremely deep networks presents issues,

regardless of the use of dropout, batch normalization and weight regularization (discussed later in Subsection 2.2.2) [98]. Ultimately for IACT event classification, it is a complex function that we wish CNN-based event classifiers to approximate, mapping a set of input images to an output event score.

2.2.4 Activation Functions

The summed inputs to an artificial neuron are evaluated against activation functions. This is crucial to neural networks being able to solve highly non-linear problems such as the EXOR problem [99]. In particular, the final layer activation function in a neural network determines the form of the neural network output. The precise mathematical form of the activation function used by the neuron is a choice for the user; the earliest neural networks used simple Heaviside Step functions, but it is common to use Linear Rectifier (ReLU), tanh (σ_T) or sigmoid (σ_S) functions as they are useful for analysing nonlinear relationships [91, 94].

The Linear Rectifier Function (ReLu(x)) as a function of input variable x as a function of threshold variable y (typically 0, but configurable) is defined by

$$\mathrm{ReLu}(x) = \begin{cases} 0 & \text{, if } x \leq y \\ \mathrm{x} & \text{, otherwise} \end{cases} \tag{2.6}$$

[94]. The Sigmoid function ($\sigma(x)$), appropriate as the final layer activation function for binary classification. It takes the form

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{2.7}$$

The Softmax function is a generalization of the Sigmoid function, and is appropriate as the final layer activation function for multi-class (>2 class) classification. For a vector input ($\mathbf{x}$) with ith components x_i and a given number of classes K it is defined as

$$\sigma(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}. \tag{2.8}$$

Neural networks can also be used as a method for regression, by having training data with continuous labels and modifying the activation function of the final layer

in the network to provide a score on a continuous space. This can be used for IACTs for directional and energy reconstruction, however this thesis doesn't approach this task (further information can be found for example in [100] and [101]).

2.2.5 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of neural network developed specifically to handle image data, inspired by the mechanics of the visual cortex in the brain. Although the concept of a CNN dates back to 1980 (the so called neocognitron [102]), the first modern use of this taking advantage of GPU power was in a paper by Ciregan et al. in 2012 [103]. CNNs take advantage of two properties of images, the translational invariance of useful data in the image (i.e. in Figure 2.2 it doesn't matter where in the image the car is located for the image to be classified as containing a car) and the relative sparsity of useful information in the image (i.e. the network does not need a million pixels to detect the presence of a tyre). The CNN achieves this in a 'convolutional layer' by only feeding a few neighbouring pixels (in a so-called 'receptive field') into each neuron, which are arranged in a 2D grid. The weights of these neurons are then shared throughout the convolutional layer in order to achieve the effect of translational invariance. Through pooling, downsampling and flattening, these neurons feed into MLP-like layers at the end of the network to perform a classification. CNNs are a promising technology for CTA analysis, as the entirety of the information contained in an IACT image can be used. Another key advantage such CNN-based methods is the speed at which they can perform classifications of complete images (approximately 1kHz for mono-telescope analysis using a NVidia 1080Ti GPU), which is a major factor to consider given the high expected array trigger rate of CTA (approximately 10 kHz for a full CTA array [56], although this estimate requires updating for the 'Alpha' array configuration).

2.2.6 Long Short-Term Memory Networks and ConvLSTMs

Long Short-Term Memory networks (LSTMs) are are an established form of neural networks for time series analysis (which are more generally known as Recurrent

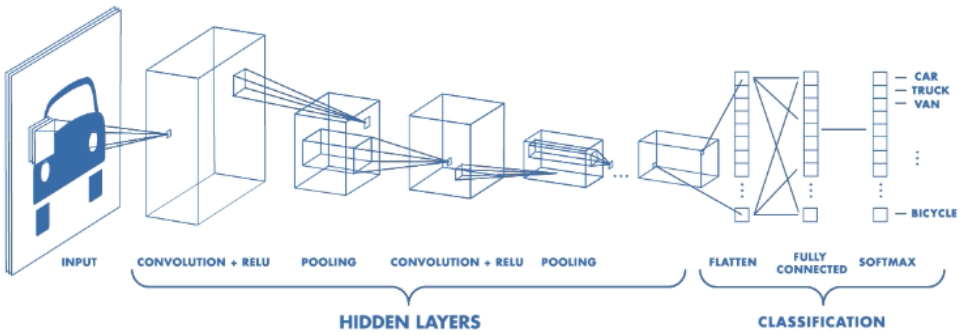


Figure 2.2: An example of a CNN, taken from [104]. Images are fed into convolutional layers, the outputs of which are flattened and fed into a MLP-like network prior to the output layer. Increasing the number of layers (depth) of the network increases the abstraction of the features recognised.

Neural Networks or RNNs). They have been heavily used in industry for classification tasks which require the ability to handle input vectors of varying length and having varying time intervals between useful information. A classic example of such a task is speech recognition. LSTMs are mathematically similar to MLPs, except that the neurons are arranged such that there are loops in the network architecture, and the individual cells within the network contain vectors c to act as a memory as well as a function to 'forget' elements from that memory f. They also contain an input function $i(x)$ and an output function o that produces an output vector O_t at a timestep t. The equations used in the initial paper on LSTMs define a forward pass through a cell [105] as

$$\begin{aligned} f_t &= \sigma_s(W_f x_t + O_f h_{t-1} + b_f) \\ i_t &= \sigma_s(W_i x_t + O_i h_{t-1} + b_i) \\ o_t &= \sigma_s(W_o x_t + O_o h_{t-1} + b_o) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_t(W_c x_t + U_c O_{t-1} + b_c) \\ O_t &= o_t \circ \sigma_t(c_t) \end{aligned} \tag{2.9}$$

where $\circ$ is an element wise product, U is an additional weight matrix, and finally b are bias vectors which must be trained. It is also possible to combine the properties of LSTMs and CNNs together to form ConvLSTMs, the 2D versions of which are

ideal for classifying time sequences of images from IACTs. These ConvLSTM2Ds are similar in structure to conventional LSTMs, but their internal matrix operations are replaced by convolutional operations. A more detailed comparison between

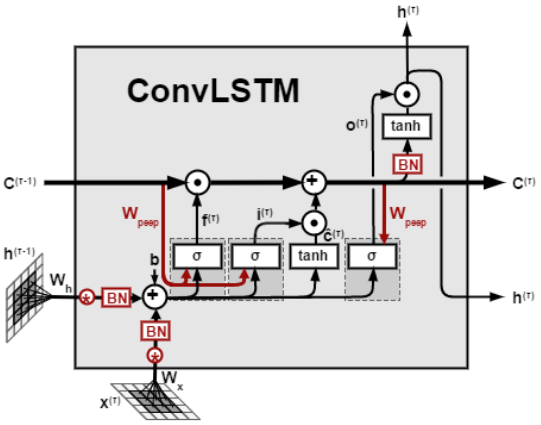


Figure 2.3: An illustration of the architecture of a ConvLSTM cell, taken from [106].

our ConvLSTM approach and the existing CRNN approach from Shilon et. al [107] can be found in Chapter 3.

LSTMs have now largely been supplanted by methods based on *Attention* (a form of trainable weighted sum applied using fully connected dense layers in parallel) for most common time series analyses in industry [108]. The *CTLearn* [101] CTA deep learning event reconstruction framework now offers a hybrid CNN/attention scheme, although the benefit of this for IACT analysis (compared to fully-optimised conventional CRNN methods) is currently unclear.

2.2.7 Hyperparameters

All machine learning algorithms (more complex than linear regression) have some number of hyperparameters, user-configurable options that define their architecture and the process of their optimisation. As examples, for a BDT or RF, this can be the number of trees in the classifier ensemble or the rate of 'pruning' (the removal of branches from the trees), and for neural networks these can be the number of artificial neurons/convolutional filters per layer. Deep learning analyses innately require a greater number of hyperparameters than their conventional counterparts

[109]. Setting the values for these hyperparameters has typically been something of a dark art, although automated methods for determining hyperparameters have been developed [109]. We explore this further in Chapters 3 and 4.

2.2.8 Wweight Optimisation Algorithms

The choice of weight optimiser used during training is also an example of a hyperparameter. Various algorithms exist to optimise the weights in a neural network during training. In Chapter 3 we use the Adadelta algorithm to optimise our ConvLSTM2Ds as this was the *Keras* recommended choice [110], in Chapter 4 we found the Adam optimiser [111] performed marginally better. Adadelta updates the weights in the network w at time steps t using an adaptive learning rate technique through the following formulae

$$w_t = w_{t-1} + \Delta w \tag{2.10}$$

$$\Delta w = -\frac{\mathrm{RMS}[\Delta w]_{t-1}}{\mathrm{RMS}[g_t]} \cdot g_t \tag{2.11}$$

where RMS represents taking the Root Mean Squared value of a parameter, and g_t is the gradient of the parameters (equal to $\frac{\partial f(w)}{\partial w}$ where $f(w)$ is the objective function) [110]. Adam, on the other hand, uses an adaptive learning rate derived from estimates of the first and second moments of the gradients. Adam also uses an exponential moving average of the gradients to scale the learning rate compared to the simpler implementation in Adadelta. The hyperparameters β_1 and β_2 control the decay rates of these exponential moving averages through the formulae

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{2.12}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{2.13}$$

where m_t and v_t are the first and second order estimates of the moments of the gradients. As these are initialised to zero at the start of training, this can lead to biasing the moment estimates around 0, this can be countered by using bias-corrected estimates ($\hat{m}_t$ and $\hat{v}_t$) of the moments

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{2.14}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{2.15}$$

the weights are then optimised using the formula

$$w_{t+1} = w_t - \frac{\eta \hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \tag{2.16}$$

where η is the step size and ϵ is a small term added to prevent division by zero [111]. Crucial to the performance of these algorithms is the concept of back-propagation, which in effect fine tunes the weights in a neural network based on the loss value produced during the last epoch of training. Through this, the neurons responsible for the greatest increase in the loss function can be identified and re-weighted (and because this task has a high degree of parallelism GPUs are a good means of performing it). Using this information decreases the loss rate in the next epoch and allows for better generalisation of the network to unseen data. Virtually all neural network optimisation algorithms use this technique [88].

2.2.9 Over- and Under-training

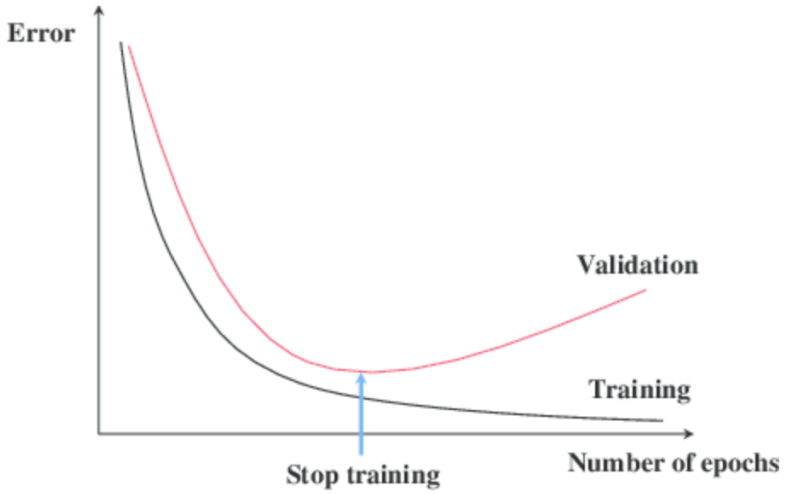


Figure 2.4: An illustration of the concept of over and under training, where error here can be considered equivalent to loss, taken from [112]. In practice we found the use of validation data with ConvLSTMs to be unreliable and highly noisy.

Neural networks can be both over- and under- optimised during the training process, a feature illustrated in Figure 2.4. One option to prevent this is to define an early stopping condition, whereby training is abandoned and the weights in

a network revert to those from a previous epoch, though this depends on both reliable validation performance and the ability to define a suitable metric by which to stop training. We encounter such issues in Chapter 3.

2.2.10 Dropout, Regularization and Batch Normalization

Dropout refers to a procedure during training of randomly setting weights in the network to zero, at a configurable rate. This is an extremely common practice in deep learning and is designed to prevent overtraining of networks. Typically, the dropped out weights are frozen after training, but applying dropout during testing can be used as a method of pseudo-simulating Bayesian posteriors [113, 114].

Regularization is a similar technique whereby additional bias terms are introduced to neural network loss functions to penalize overly-large weights, and thus also prevent overtraining. In this work, we use L2 regularization which is a quantity ($L2$) defined by

$$L2 = \lambda \sum_i w_i^2, \tag{2.17}$$

where λ is a configurable hyperparameter and w_i are the network weights, is added to the loss function [94]. In the *Keras* [94] package, dropout and regularization can be applied to the CNN elements of a network (kernel dropout and regularization) or the recurrent elements of a network (recurrent dropout and regularization) or both; the rate and λ terms of both being configurable hyperparameters.

Batch normalisation is a technique for easing training of deep neural networks, using a re-parameterisation method that reduces the complexity of adjusting the weights in multiple layers [88]. It works by scaling the output of a batch normalisation layer by normalising the input activation functions of a batch during training to a Gaussian distribution centred on zero with a standard deviation of one. Specifically, for a batch normalisation layer, *Keras* returns during training

$$\text{Output}_i = \gamma \times \frac{(\text{Batch}_i - \text{Batch Mean}_i)}{\sqrt{\text{Batch Variance}_i + \epsilon}} + \beta \tag{2.18}$$

where the batch mean and variance are those calculated for a given epoch i, ϵ is a small constant hyperparameter set during network design. γ and β are both learned by the network during training; they describe a scaling factor and offset factor relative to a Gaussian distribution.

Batch normalisation has both a regularising effect, preventing individual weights from becoming too large, as well as ensuring that the next layer in the network does not have to deal with weight changes that are too large, which reduces the overall training time required [115]. At inference (testing) time, the layer normalizes its output using a moving average of the mean and variance of the batches it has seen during training [94], this can causes issues if the batches used during training are not a representative distribution of examples of the overall training data [115]. These moving averages in *Keras* are defined iteratively epoch by epoch using the formulae

$$\begin{aligned} \text{Moving Mean}_i = \text{Moving Mean}_{i-1} \times \text{Momentum} + \\ \text{Batch Mean}_i \times (1 - \text{Momentum}) \end{aligned} \tag{2.19}$$

$$\begin{aligned} \text{Moving Variance}_i = \text{Moving Variance}_{i-1} \times \text{Momentum} + \\ \text{Batch Variance}_i \times (1 - \text{Momentum}) \end{aligned} \tag{2.20}$$

where momentum is a configurable hyperparameter, which in effect dampens the effect of oscillations in the moving mean and variance.

2.3 Past and Concurrent Work on CNNs With IACT data

In this section, we review recent relevant work concerning the use of these new machine learning methods for IACTs.

2.3.1 Muon Tagging and Muon-Hunters

The first published work to consider the potential for CNN-type analysis with IACTs was Feng et.al in 2016 [116]. The authors consider the use of CNNs for

muon ring classification (selecting events where a muon has passed directly down the optical path of an IACT and has therefore produced a characteristic 'ring' image) as well as muon ring radius and size regression with VERITAS. This is needed for online calibration, as by measuring the width and detected Cherenkov light intensity of such events, one can infer the efficiency of the IACT's optics. VERITAS has also investigated the use of unsupervised 'deep embedded clustering', in combination with cluster refinements from citizen scientists, for muon tagging through the Muon Hunters 2.0 project [117].

Another recent proposal for muon analysis and γ/hadron separation is to use an approach where classifications from citizen scientists are used in combination with Bayesian deep learning to refine classifications (so called 'active-learning') [118]. Whilst active-learning is an extremely interesting area of computer science research, and an excellent opportunity to engage the public, the task of γ/hadron separation for CTA is less well suited for active-learning-type analyses for a number of reasons. Firstly, the likely trigger rates ($\sim 10\,\mathrm{kHz}$) [119] of CTA, which results in large numbers of images, are probably beyond the reasonable capacity of an active-learning pipeline. This is as the number of citizen scientist volunteers over time is usually approximately constant. Secondly, current active-learning analyses can potentially have biases inherited from the intuition of volunteers. An example of this is the redshift dependence on galaxy morphology classification in [113], whereby contrast between low and high surface brightness features in galaxies increasing at high redshift leads to galaxy mis-classifications. A similar effect could occur in IACT data for low-energy events with few triggered pixels. Thirdly, γ/hadron separation for IACTs is a subtle task, relying on small differences between images, and has little aesthetic reward for the citizen scientists. That said, competitive elements of citizen scientist platforms could help make the classification appealing. Finally, as these methods ultimately rely upon CNNs, they are vulnerable to the same run-by-run experimental variations as supervised deep learning techniques. We discuss these issues later Chapter 4. As such, in our opinion, supervised machine learning methods remain the most promising for IACT event classification for CTA.

Further potential options for muon tagging with CHEC-S and ASTRI simulated data using CNNs have also been investigated by R. Clark [120], particularly concerning the tagging of partial muon rings against proton EAS background. Whilst he found that CNNs can identify partial muon rings (despite the complexity of generating a simulated dataset suitable for this analysis), it appears as if using supervised CNNs for online muon tagging will suffer many of the same issues surrounding CNN use for IACT background rejection that we explore further in Chapter 4.

2.3.2 Work of Shilon et al. and Parsons et al.

The application of deep learning event classification and reconstruction methods to data from H.E.S.S. is explored in Shilon et al. [107] and Parsons et al. [121]. In the Shilon et al. work [107], the authors feed images from the CT1-4 telescopes into convolutional layers that then feed into an LSTM Network. This technique, named as the Convolutional Recurrent Neural Network (CRNN) method by the authors, treats the set of four charge images from the four IACTs as a time series. This is performed by feeding CNN layers per telescope into an LSTM. The authors of the Shilon et al. work justify this network architecture using the assumption that that the images with the largest size parameter (most total charge) are those closest to the shower core, and that this ordering compensates for the lack of available temporal information in their case [107].

Using this technique, the Shilon et al. authors observed performance in background rejection superior to the current BDT/RF paradigm with simulations, and were able to perform a significant detection of PKS 2155-304 with real observations. When applied to this data, the CRNN technique a $50.8\,\sigma$ Li and Ma significance [122], relative to a BDT which achieved $46.1\,\sigma$. But PKS 2155-304 is extremely bright, and so this data is possibly γ-dominated. It is not currently clear how well these CNN-based event classification methods will work with lower-luminosity astronomical sources, where background dominates. This is due to the

high level of sensitivity to small differences between real observations and simulations. We explore such issues related to real observations further in Chapter 4.

Whilst the CRNN architecture allowed for the first CNN-based detection of an astrophysical source, the assumption of Shilon et al. that the ordering of the images in the LSTM is significant has not necessarily held up to subsequent scrutiny. Other charge image orderings [123] have been seen to provide equivalent or superior performance. For the purposes of our work in later chapters, we therefore treat the LSTM-type features in similar networks as a computational tool. In particular, this tool is useful for avoiding the need to stack images from multiple telescopes by summing them into single images before applying CNN analyses. Image stacking was the standard deep-learning analysis method previous to the Shilon et al. work [124], but the Shilon et al. authors found it doesn't perform well in the case of events triggering multiple IACTs [107, 125]. This is likely because features in the stacked images are lost when many telescopes are triggered. The Shilon et al. paper also considers directional reconstruction, however the directional and γ/hadron separation elements of the paper are applied to differing flaring states of PKS 2155-304.

Parsons et al. [121] attempts to hybridise CNN-type methods with parametric inputs such as impact parameters for background rejection, and achieves similar results to the Shilon et al. paper. But in time, we would hope CNN-type classifiers would be capable of learning image features related to (for example) impact distance from the images themselves. Also note that the Parsons et al. work doesn't fully document the training or optimisation of the neural networks presented.

Both these papers share a reliance on tailcut cleaning of images prior to deep learning analysis. But the ultimate aim of deep-learning-based event classification methods on photomultiplier charge data is to improve IACT sensitivity by extracting subtle features in IACT images. Tailcut cleaning makes analysis of real observations simpler (as the total NSB in the image is reduced), but can also remove some of these features in the images we aim to exploit. To date, this has been a necessary compromise in order to perform analysis of real observations (and is common in

many deep learning papers in astronomy, see for example [126]), but ideally this image cleaning would not be needed. Performing tailcut image cleaning before directional and energy reconstruction is more tolerable, given that there is less need to examine small features around the edge of the shower image. We explore the limitations of performing charge-based deep learning event classification, without tailcut image cleaning, in Chapter 4.

2.3.3 Full Event Reconstruction and Instrument Response Functions

For most of the previous decade, a Python-based method of generating effective areas, angular resolution plots and differential flux sensitivities (examples of Instrument Response Functions (IRFs)) for CTA based on *ctapipe* and the associated prototype pipelines did not exist. At the time of writing this is now only partially the case, and prototype methods of IRF generation have yet to undergo a full Monte-Carlo validation effort for the final SSTCAM. Much of the current work of the CTA task force on machine learning now focuses on generating these IRFs with deep learning event classifiers, directional reconstructors and energy reconstructors [101]. Given the real observations issues explored Chapter 4, IRFs generated using these pipelines might have difficult-to-quantify systematic errors associated with the discrepancies between real and simulated data, particularly for effective areas. Such IRFs might also have issues arising from problems with gammaness cut optimisation and gammaness values (which arise as a result of a single CNN-type classifier being used rather than an ensemble) [113]. We explore these issues in depth in Chapters 3, 4 and 6.

2.3.4 Multi-Task Learning

One option for deep-learning-based IACT analysis is to design a CNN-based architecture to perform the tasks of event classification, directional reconstruction and energy reconstruction simultaneously, work currently being done by the *Gammalearn* project [100] (currently only for single telescope analysis). The motivation for this

is that there are physical correlations between the event impact distance and apparent energy. Whilst a very interesting area of investigation, attempting to perform multiple IACT analysis tasks simultaneously makes it difficult in practice to discern where any improvement in performance originates from. That said, the clear sensitivity of CNNs to event direction we find in Chapter 3 suggests there may be a significant benefit to performing event classification and directional reconstruction simultaneously.

2.4 Known Issues with CNN-Type Analysis Techniques

In the following subsections we discuss the various known issues associated with the use of CNNs for CTA analysis. It is helpful to define specific problems when exploring new analysis methods as it helps in the process of defining goals and limitations. This is particularly true in the case of CTA, as it consists of a new set of instruments with new procedures and analysis requirements.

2.4.1 The Camera Geometry Problem

CNNs built for analysis of 2D arrays are primarily designed to process rectangular images. This creates an issue for using CNNs with CTA data as many of the camera prototypes have focal planes which are either hexagonal or have regions without photodetectors (like CHEC, which doesn't have photodetectors in its corners). In the case of hexagonal cameras, there are two options to avoid the risk of geometric bias. The first is to resample or oversample the hexagonal image in order to make it square, this is the approach taken in a paper by Holch et.al. [127] and used in Chapter 4 for our VERITAS data. The second option is to write a custom CNN code that can handle hexagonal images based on computing convolutions based on nearest neighbour information, which may also be applicable to CHEC/SSTCAM (in the analysis in Chapter 3 we just crop images to the square central region of the camera). The disadvantage of this custom CNN approach is that it limits the potential for more sophisticated network architectures, and at present it has

not been extended to stereoscopic analysis. Graph-based Chebyshev networks [84] might provide an alternative solution to this problem in the long term, as they only require that an adjacency matrix can be defined for the camera plane, as opposed to having to perform transformations to place the IACT images on a rectilinear grid.

2.4.2 The Multiple Telescope Problem

CTA will operate stereoscopically, and combining images from multiple telescopes (separated by varying distances on the ground and with differing impact distances) to use with CNNs is a problem unique to IACTs. The simplest method of solving this problem is to simply add the pixel values from the various telescopes together and feed this single image to a CNN [124]. The problem with this is that it doesn't take account of the telescopes' positions on the ground, which is an issue for performing reliable event classification and accurate directional and energy reconstruction. It also doesn't scale well to large arrays; in high-multiplicity events where many telescopes are triggered, this stacking results in a loss of detail related to shower morphology, which can also cause issues with event classification. The Shilon et al. authors encountered this problem [107], but stacking of images for stereoscopic analysis is still undergone in some deep-learning projects.

Over the course of the investigations included in Chapters 3 and 4, it has become apparent that there will likely always be some degree of tension between the most advanced deep learning analyses possible and stereoscopic IACT analysis. Single image analysis is both a more common task and is easier to perform. But stereoscopic analysis for IACTs always achieves superior flux sensitivity and event angular resolution against single image analysis for any analysis method, as the pixel density and effective area of a single IACT is limited by technological constraints. As such, if deep learning analysis cannot eventually reliably work on images from multiple IACTs, its worth is significantly reduced in the context of an advanced analysis method for CTA. Following the construction of the CTA array beyond the prototype phase it is unlikely that observations in mono-analysis mode (single telescope analysis) will be performed routinely for these reasons.

As of the time of writing, no CTA deep learning analysis study has considered the prospect of deep learning event classification with multiple classes of CTA telescope observing the same event. This may be one of the must crucial avenues of investigation given the flexibility deep learning offers in considering partially truncated shower images. This image truncation results in a fall-off in flux sensitivity at high energies in Hillas-parameter-based analysis, as Hillas parameter extractors are either unreliable or fail if a significant amount of shower Cherenkov light falls outside the camera image. However, investigating this comes with significant computational cost and simulation complexity.

2.4.3 The Hyperparameter Problem

CNN-type methods require a large number of hyperparameters, which define the network architecture and training process. These aren't always trivial for a human to interpret. The optimal combinations of these for IACT data may be a function of instrument observational energy, plate scale and observation conditions, and automated optimisation strategies for these are highly computationally costly. Additionally, hyperparameter selection can have an effect on training convergence, which makes comparison of differing models complex. We explore these issues in depth in Chapter 4.

2.4.4 The Real Observation Problem

Whilst one of their greatest strengths, the fact that CNNs are sensitive to minute morphological features in their training datasets can also present significant issues. The most troublesome problem arises from the fact that we need a labeled input dataset in order to train our networks. In certain cases (such as transient searches [128]) this dataset can be produced from real observations by hand, or through the involvement of citizen scientists. For the purpose of CTA image analysis this is not possible. As a result, we are forced to train our neural networks on data simulated using Monte-Carlo simulations for which we know the 'true' event parameters, and then use these trained networks to analyse real observations. The slightest

discrepancy between the real and simulated data (for example from night sky background, optical efficiency losses due to the ageing of the telescopes or imperfect calibration) has the potential to affect the resulting analysis. One particular source of concern for the CTA SSTs in particular is the uncertainty in modelling hadronic processes at high energies, notably in the CTA-standard simulation *Quark Gluon String with JETs (QGSJet)* package (which is used within *CORSIKA*). This is due to a lack of forward momentum data from the LHC, but significant work to improve these models is underway [129].

One potential solution to this problem has been proposed by Erdmann et.al. [130]. Their suggestion is the use of Generative Adversarial Networks (GANs), which are essentially two CNNs playing off each other in a zero sum game [88], to perform an intelligent, machine learning based correction to simulated data such that it resembles real observations more closely for the purpose of training a CNN. We briefly further discuss the potential applications of GANs in Chapters 3 and 6.

2.4.5 The Class Imbalance Problem

Not every γ-ray source on the sky is equally bright, and so real test data will typically be imbalanced (i.e. not have a 50:50 γ-to-proton ratio). In certain IACT use cases, such as obtaining dark matter annihilation constraints from observations of (believed to be dark-matter-dominated) dwarf spheroidal galaxies [131], the observational data will be proton dominated. In particular, this is complicated by trigger selection. As a result, there is a need for a future detailed investigation into the effects of using more realistic, imbalanced training data with CNN-type IACT background rejection methods, as such imbalanced datasets require specialised deep learning treatments [92].

2.4.6 The GPU Cost and Availability Problem

Without access to high-performance GPUs, deep learning analysis would be prohibitively slow as an analysis method for CTA at both training and inference times. Whilst extremely powerful computational tools, GPUs are very expensive

[132], in very short supply [133], power hungry [134] and thus costly to run [134]. These practical considerations must be taken into account when considering deep learning as a potential general analysis method for CTA. In the long term, and broadly speaking for the majority of analyses, deep learning analysis must have a reasonable prospect of passing a cost/benefit analysis whereby if deep learning offers a 10% improvement in sensitivity, it must be less than 10% more computationally and financially costly compared to conventional analysis. At the time of writing, this is not the case. However, there could be specific analysis cases where such costly analysis methods could be justified if it opens the potential for new physics (such as looking for dark matter annihilation constraints through observing dwarf spheroidal galaxies [131]).

Given these issues of availability and cost, there is a risk of deep-learning-based research in the physical sciences becoming a 'walled garden', whereby only the richest or most cited researchers have access to sufficient compute power to perform state-of-the-art deep learning analyses [135]. Ideally, GPU time should be allocated based on scientific merit, and to their credit the UK's Science and Technology Facilities Council (STFC) are now providing such a peer reviewed resource through their eInfrastructure for Research and Innovation for STFC (IRIS) scheme. This is now being utilised by CTA.

3

Use of Timing Information In Combination With Deep Learning as an Event Classifier for SSTCAM

Contents

Abstract

New deep learning techniques present promising new analysis methods for Imaging Atmospheric Cherenkov Telescopes (IACTs) such as CTA. In particular, the use of Convolutional Neural Networks (CNNs) could provide a direct event classification method that uses the entire information contained within the Cherenkov shower image, bypassing the need to Hillas parameterise the image and allowing fast processing of the data.

Existing work in this field has utilised images of the integrated charge from IACT camera photomultipliers, however the majority of current and upcoming generation IACT cameras have the capacity to read out the entire photosensor waveform following a trigger. As the arrival times of Cherenkov photons from EAS at the camera plane are dependent upon the altitude of their emission and the impact distance from the telescope, these waveforms contain information potentially useful for IACT event classification.

In this simulation study, we investigate the potential for using these camera pixel waveforms with new deep learning techniques as a background rejection method, against both proton and electron induced EAS. We find that a means of utilising their information is to create a set of seven additional 2D pixel maps of waveform parameters, to be fed into the machine learning algorithm along with the integrated charge image. Methods based upon timing information appear to out-perform similar charge based methods for γ/hadron separation, but there appears to be little prospect of performing electron event classification, as all separation power against electrons is based on event direction. We also review existing studies of event classification using a combination of deep learning and timing information in other astroparticle physics experiments.

3.1 Introduction

In this chapter we present the first application of ConvLSTM2D network techniques in the high energy range probed by the SSTs. This work is also the first known attempt to determine if new deep learning techniques and new prototype instruments have any potential to aid in discriminating against cosmic-ray electrons using a combination of complete images and photosensor waveform data, suggested as a possibility by Nieto et. al in [136]. Previous (largely unsuccessful) attempts at the highly challenging task of γ-electron event separation for IACTs have relied upon indirect estimation of first interaction height from IACT images using either shower image shape or combined shower energy and maximum estimation [25].

We split the remainder of this chapter into seven sections. In Section 3.2, we discuss recent advances in machine learning techniques in astroparticle physics, particularly concerning the use of timing information in combination with deep learning techniques as a background rejection method. In Section 3.3 we detail the simulated datasets we use, and in Section 3.4 we describe the analysis we perform. In Section 3.5 we present our results for our ConvLSTM classifiers, in Section 3.6 we perform an investigation into the features learned by CNN-type classifiers, in Section 3.7 we compare our results to RF classifiers, and we conclude in Section 3.8.

3.2 Timing Information as an Event Classification Method in Other Experiments

3.2.1 IACTs

The use of EAS timing information for event classification in IACT data has was considered as early as 1987 [62]. More recently, it has been used as an image cleaning method on MAGIC (whereby pixels associated with Cherenkov light are partly selected based upon photon arrival time) [59], prior to their RF analysis. Similar studies to this have also been performed with VERITAS [137]. MAGIC also use parametric timing data such as a time gradient (how fast the arrival time changes along the major IACT image axis) and the RMS arrival time of photons to all

pixels that survive an image cleaning process as part of their RF-based background rejection method [67]. In that work [67], the MAGIC collaboration also demonstrate the dependency of timing structure upon the impact parameter of the shower, also information useful for background rejection. The attraction of deep learning in our use case is the ability to use this timing information for background rejection directly, in addition to the other pixel-wise data (such as evidence of electromagnetic shower sub-structure in hadronic showers), without the need for parameterisation. An illustration of this information can be seen in Figure 3.3. Whilst this is the first work to attempt such a combination of timing information and deep learning for IACTs, this is not the case for other air shower experiments, and so in the remainder of this section we investigate the approaches taken by other groups.

3.2.2 Cosmic Ray Detection

Erdmann et al. [138] investigated the use of deep learning to perform energy and angular event reconstruction of air showers for a model cosmic ray experiment consisting of a hexagonal array of 81 ground based water Cherenkov detectors (similar to a subsection of the Auger cosmic ray experiment) using a custom simulation code. In this study waveforms are fed in from their array into a CNN, which is concatenated with 2D feature maps of the first particle arrival time and integrated charge for the entire detector grid approximately half way through their network architecture. These feature maps are then treated with separable convolutions, as it is not expected that the correlations between the feature maps will be coupled to spatial correlations within the maps themselves. The use of this information achieves an energy resolution of 5% in the EeV energy range, compared to 8% without the additional concatenated feature maps. We do not attempt a similar approach here (as has recently been attempted as an expansion to the Shilon et al. work by [121]) as it becomes even more difficult to interpret the behaviour of the neural network when parameterised data (such as impact parameters) is mixed with image data. In a follow-up paper with the same lead author [130], the authors investigate their analysis robustness to real data effects by creating air

shower simulation datasets with different divisions of energy between the muonic and electromagnetic components of the showers. They successfully train a 1D Generative Adversarial Network (GAN, which is essentially two neural networks engaged in a zero-sum-game) in order to attempt to refine their 1D waveform traces from one simulation set such that it resembles the second set of simulations more closely. This is an attempt to handle the problem that minute differences between training and test sets in machine learning analyses can cause discrepancies; however they find that the differences between their simulated datasets are insufficiently large to affect their quality of the energy reconstruction. 2D image data is substantially more complex and it is unclear how well a GAN would perform in our case. A study has also been performed on data from the Tunka-Rex experiment [139] on 1D timing information regarding the radio emission from > 100 PeV air showers which provided improved background suppression, however this was only performed on single traces from a single antenna and not the complete array.

3.2.3 IceCube

In [140], the IceCube collaboration perform reconstruction of muon neutrino events using deep learning. Each of the 5160 in-ice Digital Optical Modules (DOMs, a photomuliplier with associated electronics enclosed in a sealed unit) records a waveform when Cherenkov light emitted from the particles generated by interactions in the ice reaches the photomultiplier in the DOM. In that work, the authors parameterise the registered waveforms using 7 variables (sum of all charges, number of pulses, time of first pulse, time of last pulse, the average time of a pulse, the standard deviation of pulse times and the highest pulse charge) which are then inserted into a 3-dimensional neural network. On simulated events, they are able to obtain an improvement in energy resolution relative to a more conventional reconstruction method. This approach to parameterizing waveforms partially inspired our work, as use of full CHEC waveforms with e.g. 3D CNNs is likely to be prohibitively computationally expensive (our initial experiments with CHEC-S data also demonstrated that sparsity of waveform data was also an issue), especially

given the fact that CTA will be an array consisting of multiple telescopes and multiple telescope classes. Choma et al. [141] also investigate peak waveform timing information as an input to their novel graph-network based architecture (which takes graph objects with edges and vertices rather than arrays as inputs), which they use to classify IceCube events. However, the particular network architecture used in the Choma et al. work is currently not suitable for use on images from IACTs.

3.3 Datasets

All current IACT cameras have the ability to record some measure of timing information. Many of the camera prototypes under development for CTA (and many other current-generation cameras) have the ability to read out the entirety of the waveforms from their photosensors, so we are motivated to explore how this complete waveform information could be used for event classification. This waveform information can be combined with new deep learning techniques in order to take maximal advantage of the information available about the EAS. In an attempt to keep our exploration of this challenge tractable, we restrict ourselves to differential consideration of event classification alone. This is only one particular element of a complete IACT analysis necessary to perform observations of real sources [64, 122].

To investigate this task, we require large, labelled training datasets. In our case, this was Monte Carlo simulations of γ-ray, hadronic and electron induced EAS. To generate the simulated datasets, we used the *CORSIKA* package [142] to simulate the particle physics of the EAS and simulate the Cherenkov light they produce. We then passed this data to the *sim_telarray* package [143] to simulate the telescope optics and camera electronics using the CTA Prod3b array model [144]. It should be noted that these simulation models (along with the associated pipeline) are under continuous refinement. Parameters describing the *CORSIKA* simulations are shown in Table 3.1. For protons, only around a third of the energy of the incident particle goes into producing the electromagnetic component which ultimately produces Cherenkov light [42]. As such, the hadronic showers that are a background to our IACT telescopes have higher primary energies than the γ-rays with the same

total Cherenkov signal, and this is reflected in our simulations. We simulated 4 GCTs [145] equipped with CHEC-S cameras [1], although the conclusions of this work are relevant to other instruments with similar capabilities (such as CHEC-S on ASTRI), and this is a reasonable substitute for the final SSTCAM attached to a dual-mirror structure. The telescopes are arranged in a cross-configuration (+) on the Cerro Paranal site separated from the centre of the array by ±80 meters, and are simulated in the standard parallel pointing mode. This array configuration was chosen both to resemble the one in [107] and also to produce data that could be processed with finite GPU resources. Particularly in the case of the SSTs, larger simulated arrays will eventually be required for detailed studies into the use of CNNs with VHE events that trigger readout from many cameras.

Despite a growing amount of literature and work on the use of CNN-based classifiers for IACT background rejection [107, 146, 147], there is currently little consensus or detailed study of the types of datasets required to train such networks (especially in the case of application to real data). Even the pioneering Shilon et al. work used pre-existing simulations to avoid the significant computational cost of investigating this [107]. CNNs are potentially sensitive to different features in image data compared to template fitting methods, and the fact that they are capable of handling complete images rather than parameterised data means that the approaches used in most current analyses cannot be assumed to be optimal with CNN-based classifiers. As a result, we investigate two scenarios. In the first, we attempt to differentiate between simulated γ-rays from a point source against a diffuse, simulated background of proton and electron events. For the purposes of this work we define point source simulations as those with a *CORSIKA* opening angle of 0°, and diffuse simulations as those with a *CORSIKA* opening angle larger than 0°. Using a simulation approach like this would require generating new simulations and performing network training for every run (∼30 minutes of observing time). But it is our opinion that this is the most promising approach for applications of CNN-based classifiers to real observations in the medium term. This is as discrepancies between

[1] In this work, we simulate a 96 ns camera readout window with 96 samples.

training simulations and real observations are a significant issue for such sensitive techniques, that we later explore in Chapter 4. We believe that a run-wise approach, which is now being used by H.E.S.S. [148] might possibly help in observing real sources. We explore this further in Section 3.8. Additionally, we postulate that the high sensitivity of CNN-type methods to pixel wise features may allow for improved background rejection power (and ultimately improved source significance) using a combination of directional and morphological pixel wise information relative to more conventional directional and shape Hillas parameter cuts. However, such a run wise approach comes with significant caveats. For example, in using this approach blind surveys would be prohibitively computationally expensive, there would be substantial challenges related to source confusion in the galactic plane, and it would not be possible to observe extended sources (or fields of view with multiple sources). Realistically, CNN-type background rejection methods will require further study in the simplest analysis case of strong, isolated point sources using real data, before considering more complex analyses. In the second scenario, we attempt to discriminate between diffuse γ-ray events, diffuse proton events and diffuse electron events, this is similar to the simulation set-up in the Shilon et. al work [107]. This indicates the classification power based upon shower morphology alone and also allows us to test the sensitivity of the methods in detecting the direction of the shower by comparison to the point source results.

Parameter	**Point Source γ**	**Diffuse γ**	**Diffuse p**	**Diffuse e**
No. Training Events Point Source Run	360995	-	361135	361135
No. Testing Events Point Source Run	239128	-	239227	239227
No. Training Events Diffuse Run	-	365763	365903	365903
No. Testing Events Diffuse Run	-	240453	240552	239894
Energy Range (TeV)	0.3-330	0.3-330	1-600	0.3-330
Opening Angle	0°	0°-10°	0°-10°	0°-10°
Spectral Index	-2	-2	-2	-2
Zenith Angle	20°	20°	20°	20°
Azimuth Angle	0°	0°	0°	0°
Cherenkov light waveband (nm)	240-700	240-700	240-700	240-700

Table 3.1: Parameters used in our two datasets. In the point source run, the point source γ-rays are mixed in equal ratios with the diffuse proton and diffuse electron events. In the diffuse run, all three event classes are diffuse.

3.3.1 Data Processing

The data from the event files generated by *CORSIKA/sim_telarray* was then fed into a python script to process them into calibrated waveforms in units of photoelectrons (p.e.) using the prototype *ctapipe* software package [61, 75]. This process includes pedestal subtraction and smoothing with a Hanning filter [149] of length 8 (effectively performing all the operations necessary to transform the data from CTA dl0 to dl1, bar the final charge extraction). Our analysis code also extracts the waveform parameters (e.g. the FWHM), and makes use of the *numba* [150] just-in-time compilation package such that this parameter extraction can be performed at high speed, making the task of extracting these parameters for a large number of events feasible. The γ-ray, proton and electron data is then randomly mixed together in approximately equal quantities and then saved in the HDF5 [151] file format for convenience and long term storage. The use of roughly the same number of events from each class, whilst the current standard practice with these CNN-type methods [107], is a simplification to ease network training. This is necessary as training with imbalanced datasets requires specialised deep learning techniques [92]. Ideally one would match the ratio of source to background events in the training data to the real data, but this varies as a function of source. The data processing, along with the *CORSIKA/sim_telarray* simulations was performed on the European Grid Infrastructure (EGI) using the CTA-DiRAC middleware [152]. Minimal code to recreate the results in this work can be found, along with trained *Keras* models, at `https://www.github.com/STSpencer/wavelearn_release`.

3.3.2 Complexities of running code on the EGI

There were a number of issues associated with running our *ctapipe*-based dl0 data handler on the grid. In particular, the RAM available on a single EGI node is currently limited to 4GB. This can be an issue when attempting to perform analysis and calibration of large datasets. As such, we reduced the precision of the stored floats from 64 to 16 bits, this substantially reduced the RAM overhead of the code and did not have an impact on the resulting deep learning analysis (likely because

NVidia 1080Ti and 2080Ti GPUs only support 16-bit floats [134]). Additionally, to handle the multiple outputs from *sim_telarray*, we used a randomized cipher system to merge together .simtel.gz files from γ-ray, proton and electron runs into deep learning compatible HDF5 data. This allowed for us to handle the large number of incomplete or failed runs on CTA-DiRAC (which was common at the time, it's now much improved).

3.4 Methods

3.4.1 Memory Limitations

As GPUs have a finite amount of **Video Random Access Memory (VRAM)** with which to perform tensor operations, events are loaded in to the GPU during training in batches smaller than the size of the total dataset. **Batch Size** is the number of events (each with a single label) in a single batch, this is limited by the finite VRAM of the GPU used. The batch size needs to be smaller if the degree of model complexity is higher and the number of trainable parameters is larger. Using a smaller batch size increases the training time for a given number of events as there is a time overhead associated with the transferring of data from RAM to GPU VRAM.

3.4.2 Analysis Pipeline

In order to perform our analysis, we used the *Keras* [94] deep learning Python package with the *Tensorflow_gpu* backend [153]. We also used the *scikit-learn* [97] and *matplotlib* [154] packages to analyse and plot the results. The analysis pipeline we developed to perform this is illustrated in Figure 3.4. Similarly to [107], we use a mixture of convolutional layers and recurrence (specifically ConvLSTM2D layers, algorithmic details of which can be found in [155]) to robustly handle data from multiple telescopes. We compared four different techniques (Methods A-D) for classifying events on a like for like basis with these networks. These are summarised in Table 3.2 and detailed in the next three paragraphs.

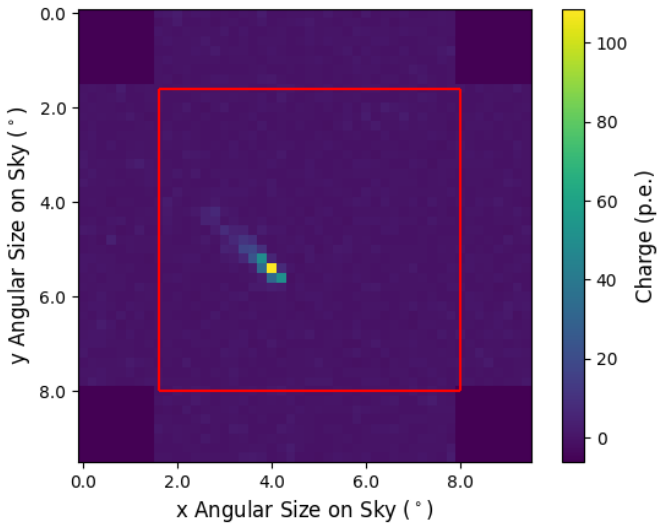


Figure 3.1: A charge image from a simulated γ-ray event from the point source run dataset as described in section 3.3. The region that survives the geometry cut is highlighted in red, the holes in the camera geometry to accommodate the flasher calibration system can be seen in the corners.

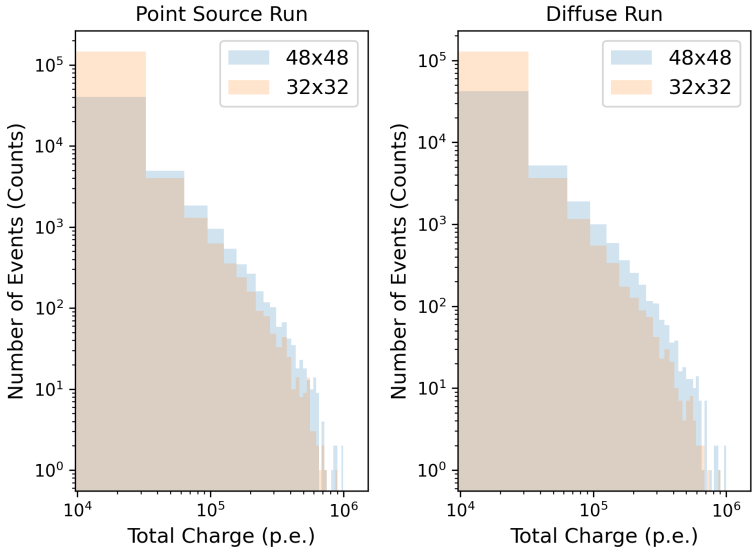


Figure 3.2: Total integrated charge from all four telescopes over the two datasets used, as described in section 3.3. Comparisons between the complete camera (48x48) and the cropped central region (32x32) are shown. Most of the charge losses appear to be Night Sky Background photons as the two distributions are smooth and close together. The same logarithmic binning is used for both the cropped and uncropped regions.

Method	Pixel Maps Used	Ordering	Parameters
A	Timing	Median Peak Time	293,673
B	Charge, Timing, Mean Amplitude, Peak Amplitude FWHM, RMS, RT, FT	Median Peak Time	301,233
C	Charge	Size Parameter	293,673
D	Charge	Median Peak Time	293,673

Table 3.2: Summary of methods used. The ordering of input channels into the ConvLSTM2D networks is as presented in this table. The number of trainable parameters is also shown, highlighting the effect of using a different number of channels in method B.

3.4.3 CHEC Camera Geometry

Part of the challenge in the Shilon et al. work [107] was developing an unbiased method to handle Cherenkov camera images that are from a non-square (typically hexagonal) pixel grid [127]. CHEC has largely square pixel geometry, bar for holes in its corners to accommodate the flasher calibration system. In our work, in order to be conservative and minimise the risk of potential geometric bias in which event classifications might be affected by proximity to these corners (which could be the case if they were instead filled by constant values), we simply crop images to the central region of the camera. This can be seen in Figure 3.1. This is unlikely to have a serious effect on the signal from the EAS as our simulation set-up (described in Section 3.3) produces images where the shower is close to the centre of the camera (as shown in Figure 3.2). We disregard the small amounts of 'dead space' between CHEC photomultiplier blocks for the same reason. More advanced tools are currently under development to cope with these issues [147, 156].

3.4.4 Experimental Methods

In this subsection, we detail the four analysis methods we explore. For method A, we use the available waveform information to generate a 2D timing pixel map across the camera plane. We then select the pixels that have a large waveform amplitude, and set the other pixels in the pixel map to zero, in order to highlight the position of the shower in the image. The location of the peaks themselves is found using a wavelet convolution of width 8 and the *scipy signal.find_peaks_cwt* function [157] [158].

These four timing pixel maps (one for each telescope) are then fed into the network without anything else, in the order of increasing median photon arrival time.

In method B, we feed in the integrated charge image (without pixel selection) as the first channel to the ConvLSTM2D. We then feed in the timing pixel map created using the same technique as method A as the second channel, and similarly cut 2D pixel maps of six other parameters from the smoothed waveform: the mean amplitude, the peak amplitude, the root mean square value (RMS), the full width at half maximum (FWHM, calculated using a B-spline interpolation [157]), the pulse rise time (RT) and the pulse fall time (FT) relative to the peak of the waveform. Both the RT and FT are calculated using a change-in-gradient method. This results in four images each with eight channels being fed into the ConvLSTM2D; examples of these pixel maps are shown in Figure 3.3. These channels are treated as extra dimensions in the input tensor, as Red, Green and Blue channels would be for a colour image.

Method C uses only charge images, ordered such that those with the largest total charge appear first in the LSTM sequence (as in [107]). It should however be noted that we do not perform an exact replication of the method in [107]. This is due to a number of factors. Firstly, we consider different instruments with differing camera designs and operational energy ranges. Secondly, we use a marginally different ConvLSTM2D network architecture (due to the CRNN code from [107] being proprietary), this differs from the CRNN architecture in that recurrent (LSTM-like) features are included from input layer until the final layer of the network (though the basic principle of merging convolutional and recurrent features is the same). We also use differing hyperparameters, partially due to this different architecture, and partly because the differing plate scales and operational energy ranges of the H.E.S.S.-1 cameras and CHEC means that there is no guarantee that the ideal hyperparameter choice will be the same for both instruments. This hyperparameter choice can have a significant effect in CNN-based analysis [109]. Finally, in order to test the efficacy of CNN-based background rejection against electrons, we include these (realistically) in the analysis whereas [107] did not. This would cause differences with the training

of any machine learning classifier (particularly in the diffuse case) as there are two nearly identical data samples with differing class assignments.

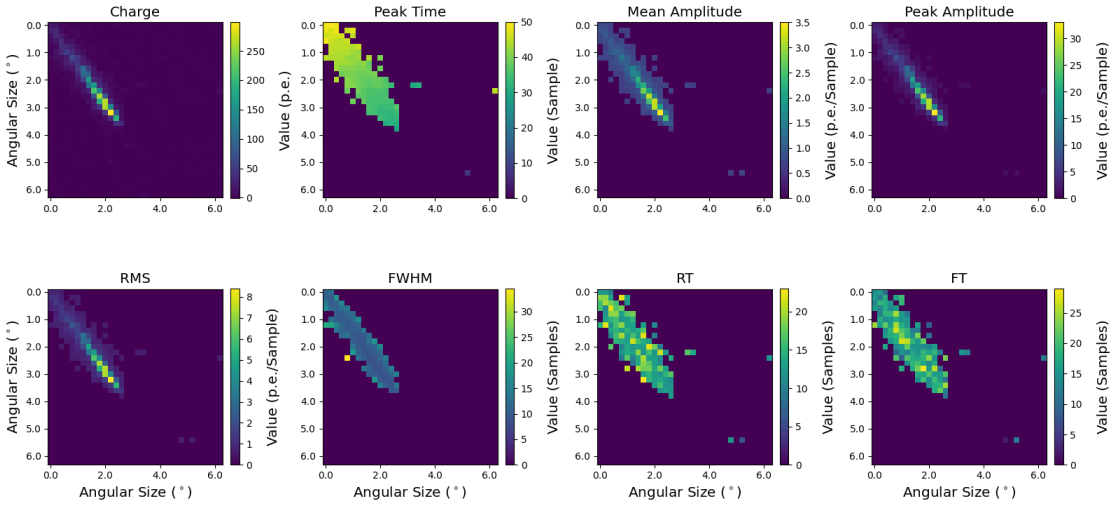


Figure 3.3: As an illustration of the different pixel maps used with Method A, this figure shows pixel maps used (cut to the central region of the camera) from a 56 TeV γ-ray event in the point source dataset for one telescope. Note that these pixel maps are normalised prior to analysis, and that 1 sample represents approximately 1ns.

In method D, we extract the median photon peak time from the waveforms and use this as a time proxy to order the images in place of the sum of the photomultiplier charges (as used in method C). In this case the only images actually fed to the network are those of integrated charge. As such the only difference between methods C and D is the order in which the charge images for an event are fed into the ConvLSTM2D, such that we can test what effect this has. Therefore methods A, C and D all result in 4 images each with 1 channel (i.e. grayscale) being fed into the ConvLSTM2D network.

3.4.5 Hyperparameter Selection

The datasets used, along with all training hyperparameters (see Table 3.3), were the same across all the four methods during the initial training. As this is intended as a differential study, we did not perform hyperparameter optimisation. Whilst there are Bayesian methods such as hyperopt [109] for performing hyperparameter optimisation with CNN-based classifiers, it would not be trivial to cross-optimise

the networks fairly across the four methods described here and it would also involve significant computational cost. Optimising the networks for say method B might disadvantage Method A unfairly. The least biased option is to use default hyperparameter options across the four methods. But to apply these methods to real observations, hyperparameter optimisation would be required, as we discover in the later work in Chapter 4. The ConvLSTM2D network architecture was also identical across the four methods, with the exception of the number of channels in the input vector (1-8). This architecture is shown in Figure 3.5.

Hyperparameter	**Value**
Batch Size	50
Training Epochs	30
Training Optimiser	Adadelta [110]
Adadelta Learning Rate	1.0
Adadelta Decay Factor	0.95
Output Layer Activation Function	Softmax

Table 3.3: Training hyperparameters used; these were the same throughout all the networks during the initial training for both the point source and diffuse cases. The diffuse ES runs also used these hyperparameters.

3.4.6 Training

In order to prevent over-fitting, the ConvLSTM2D layers are followed by dropout and batch normalisation layers, and recurrent L2 regularisation [94] is employed on the first two ConvLSTM2D layers. At the end of the network, global average pooling is used to connect to a final dense output layer of size 3 with a softmax activation function. The network is trained for 30 epochs with a batch size of 50 events per step, where the images and pixel maps have amplitudes normalised to the range [0,1]. There is a question over the efficacy of normalisation of IACT images with CNN-based methods as it can potentially prevent generalization, but in this case we observed superior overall training performance when the images were normalised.

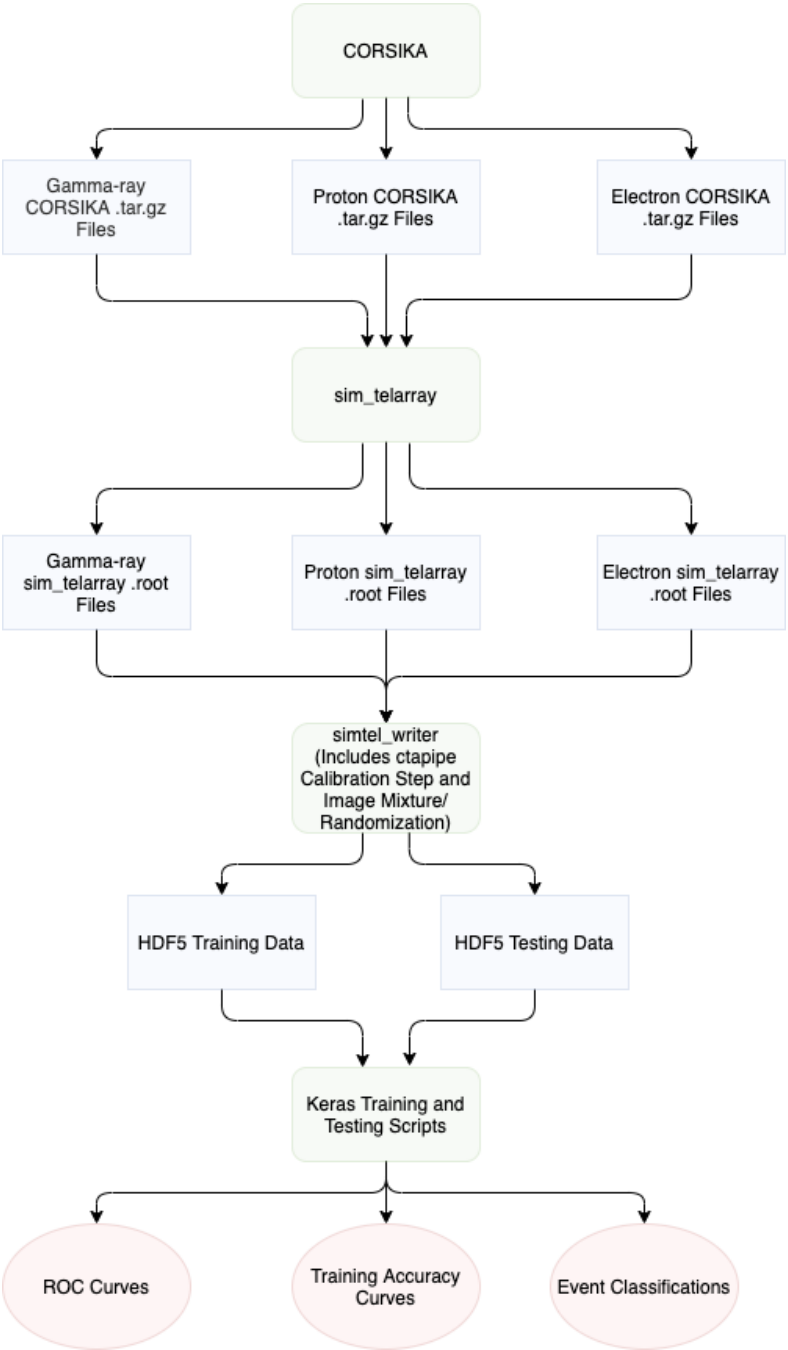


Figure 3.4: The analysis pipeline used to generate the results in this paper. Ellipses denote analysis products, cornered rectangles denote stored files and rounded rectangles denote code elements.

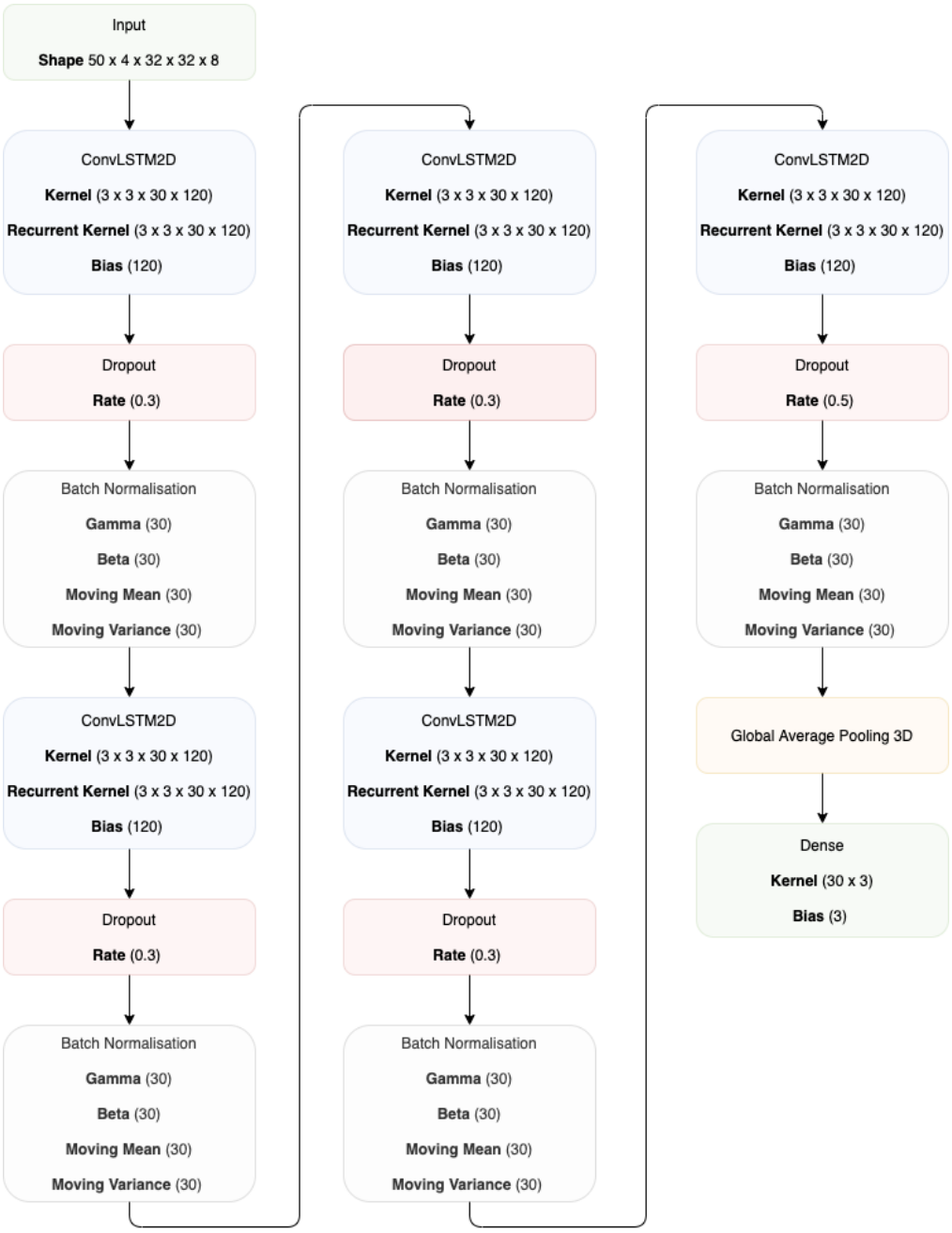


Figure 3.5: Example of our CNN-based ConvLSTM2D architecture. The only difference between the various methods A-D was the shape of the input vector, as some methods used more pixel maps than others.

Method	Point Source Run Classification Rate (Hz)	Diffuse Run Classification Rate (Hz)
A	308	296
B	240	231
C	339	320
D	308	336

Table 3.4: Classification rates (the number of classifications per second) based on testing against ∼100,000 events for each method using 4 telescopes. Note that due to the LSTM features in our network this is substantially slower than classifications can be performed for Mono-telescope analysis (∼1 kHz). This is also dependent on the GPU used (a NVidia 1080Ti in this case), and the data loader has not been fully optimised for speed. The test system had 32GB of Random Access Memory and operated at a clock speed of 1866 MHz.

3.5 Results

3.5.1 Initial Results

The initial training took approximately 3 days on a NVidia 1080Ti GPU for each ConvLSTM2D network, meaning the training time for the results presented in Figures 3.6, 3.7 and 3.8 was around a month. Each network for each method (A through D) and each of the two datasets (point source and diffuse) was trained from scratch independently, so each sub-figure in Figure 3.7 represents the test results for a uniquely trained classifier (the results from which are replotted in Figure 3.8). We do not mix classifiers trained on different datasets, so classifiers trained on point source data are tested on (previously unseen) point source data, and not diffuse data. As such, and given the fact that there is no preferential direction in any of diffuse data *CORSIKA/sim_telarray* events (as seen in Table 3.1), the diffuse data results demonstrate classification power based solely upon shower morphology. Training accuracies as a function of epoch for this initial training are shown in Figure 3.6. This training was a significant computational task, and represents the limit of what can be performed with a single GPU. It should also be noted that in this particular use case, the 'compute capacity' (NVidia's measure of a GPU's specifications) of the GPU is an important attribute due to the presence of RNN features in our network. Additionally, having a large amount of VRAM allowed for the processing of large datasets.

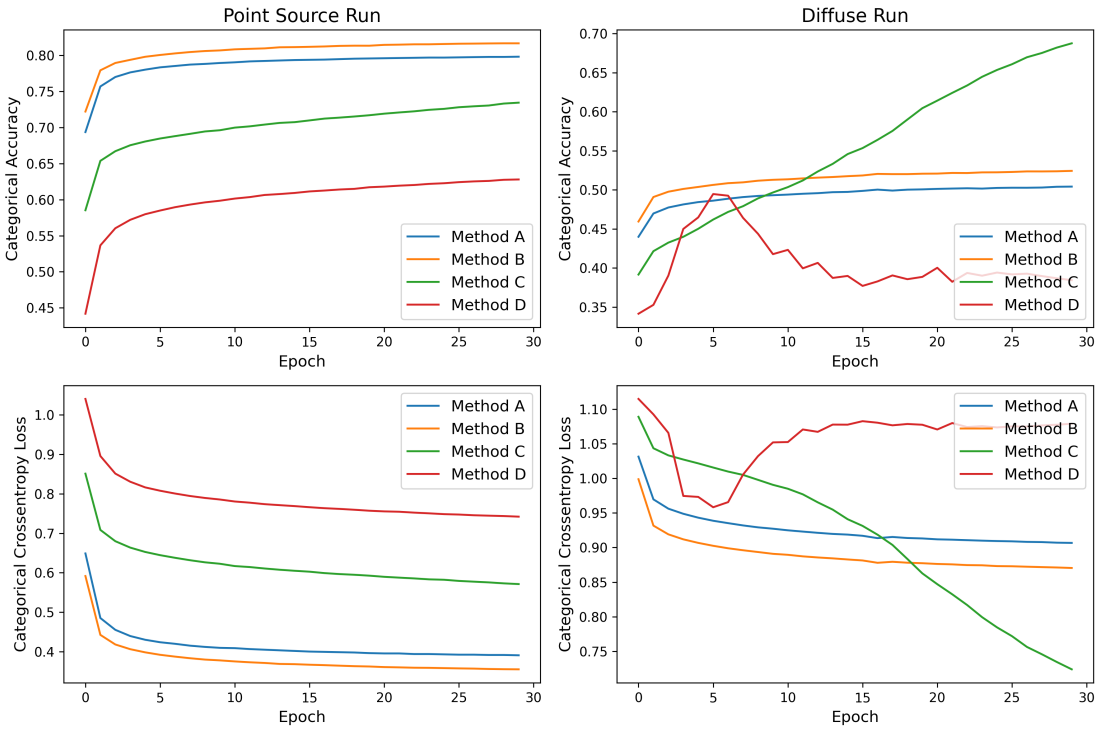


Figure 3.6: Initial training losses and accuracies for the four methods computed using the training dataset. Reasonable convergence is achieved for most methods given finite computational resources, but methods C and D in the diffuse case show some evidence of overtraining given their poor performance on the test data. This overtraining is evident here by either inversion or runaway acceleration in the loss function.

3.5.2 Difficulty in Identifying Classification Features

Determining the source of classification power with ConvLSTM2D classifiers is non-trivial. Whilst gradient and perturbation based attribution techniques have been developed for standard single-image CNN classifiers (which can produce heat maps showing pixels relevant for classification) [159], open source packages for this do not currently extend support for such methods to the multiple image ConvLSTM2D case. As such, for stereoscopic analysis we are forced to interpret performance of these classifiers on different datasets in order to infer the information being used for classification. In our case, the results for the point source run and methods A and B achieve high performance metrics, but it appears as if this is as the classifier is largely using the directional information about the shower to perform classification, rather than morphological features in the images. This can be seen in Figures 3.7

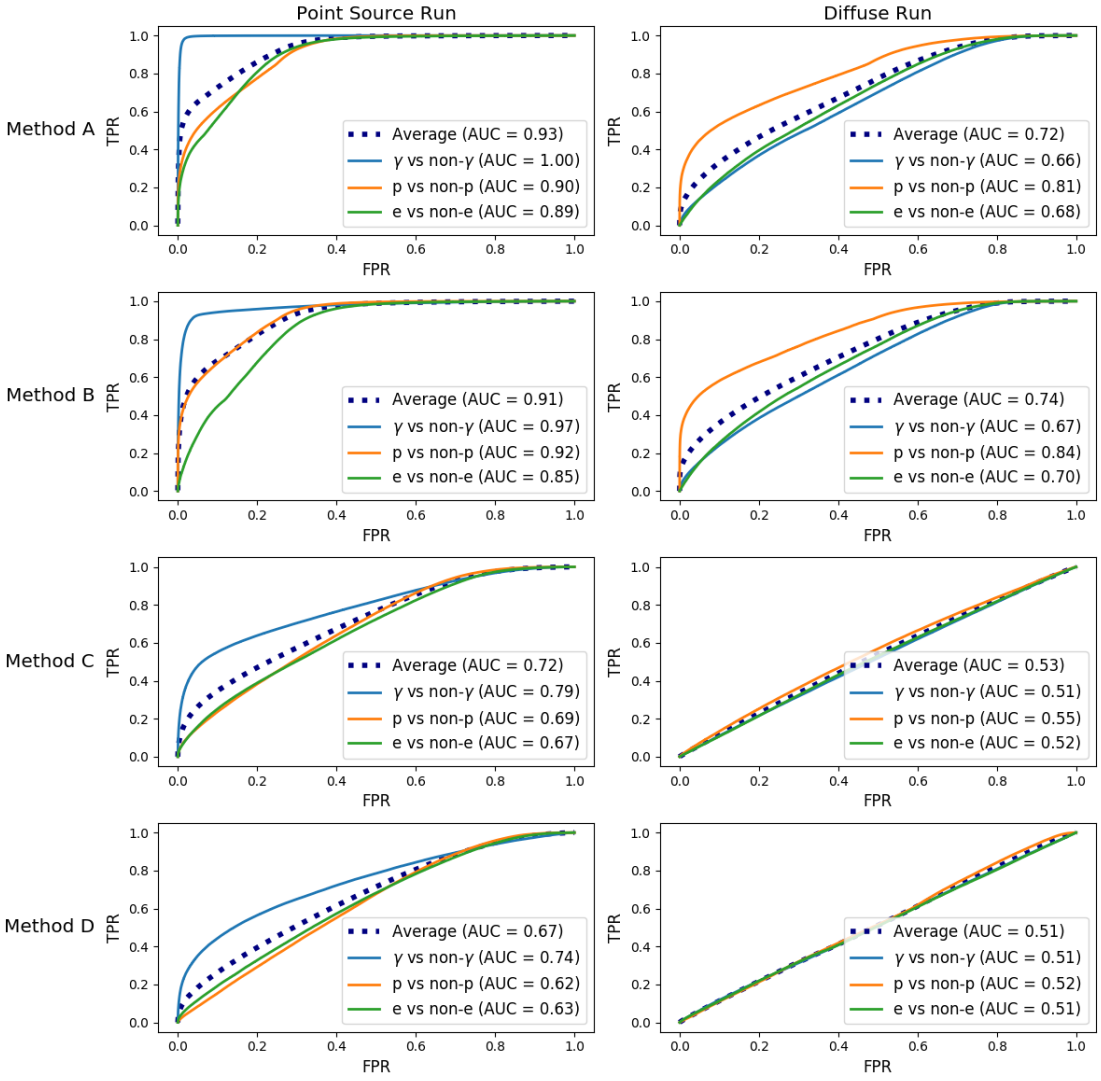


Figure 3.7: Receiver Operator Characteristic (ROC) curves for the four methods described showing the True Positive Rate (TPR) against the False Positive Rate (FPR). The Area Under Curve (AUC) performance metrics (the integral under these curves) are also shown. Deep learning analyses typically consider one versus all scenarios when calculating ROC curves [97], which is what we display here. This means we consider if an event is classified a γ-ray or not a γ-ray when calculating the FPR and TPR in the multi-class scenario. Macro-averages across the three classes are also shown. It should be noted that the AUC results here and in Figure 3.8 are only quoted to 2 decimal places and there are $\sim$700,000 testing events. As such, even in the optimal point source case, there are a number of misclassified events (i.e. $\sim$5500 γ-hadron misclassifications and $\sim$7000 γ-electron misclassifications for method A).

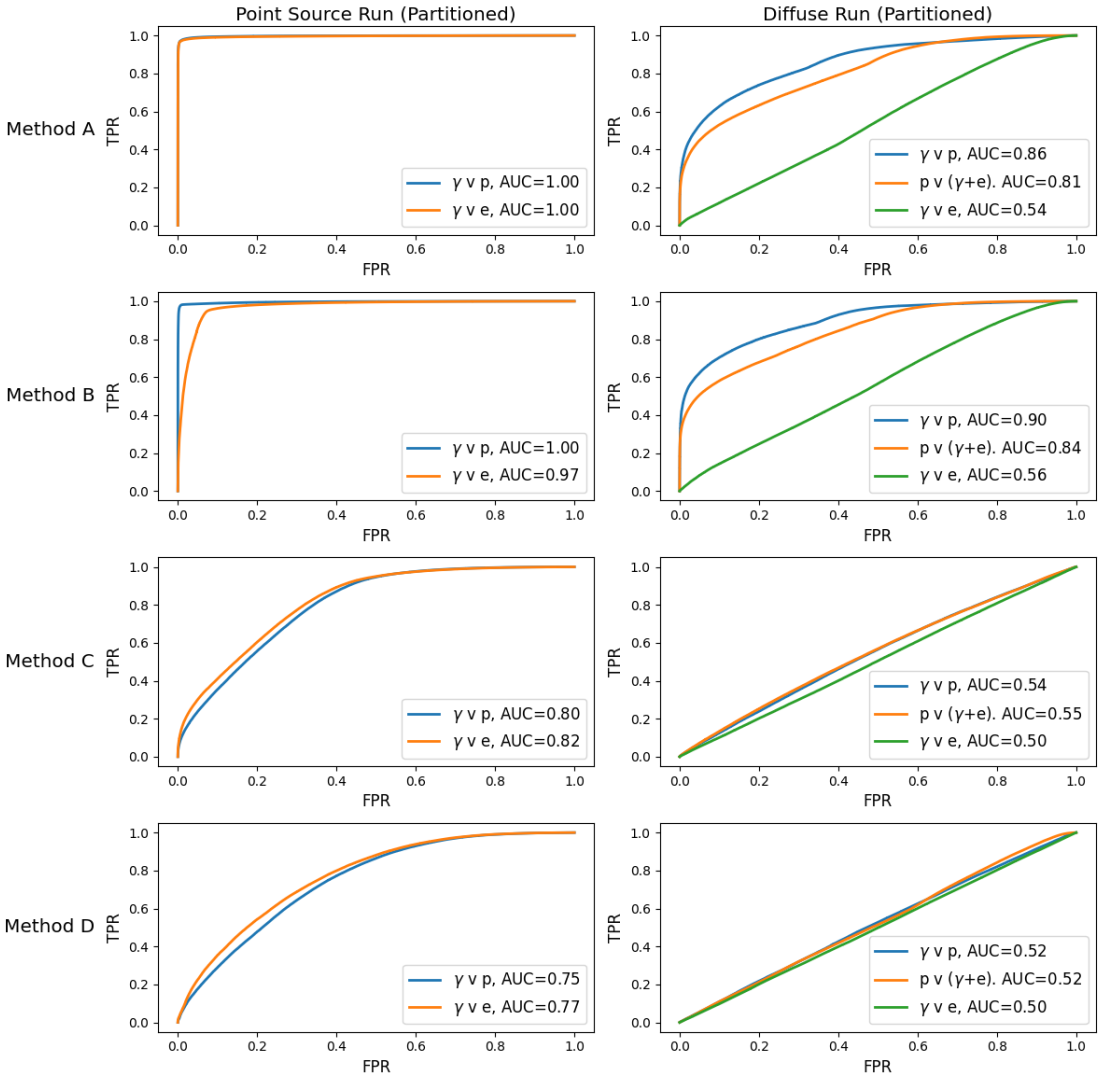


Figure 3.8: The same results from Figure 3.7, this time shown in an astrophysical context. In this case we remove the other particle classifications from the analysis and consider only test FPR/TPR between various particles. In the point source run case, we consider the ROC curves for classification between γ-rays and protons and between γ-rays and electrons. This demonstrates the combined classification power of both morphological and directional information. The equivalent curves for the diffuse case highlight solely the morphological classification power. The proton versus (γ+e) curve demonstrates the difficulty of distinguishing between γ-rays and electrons.

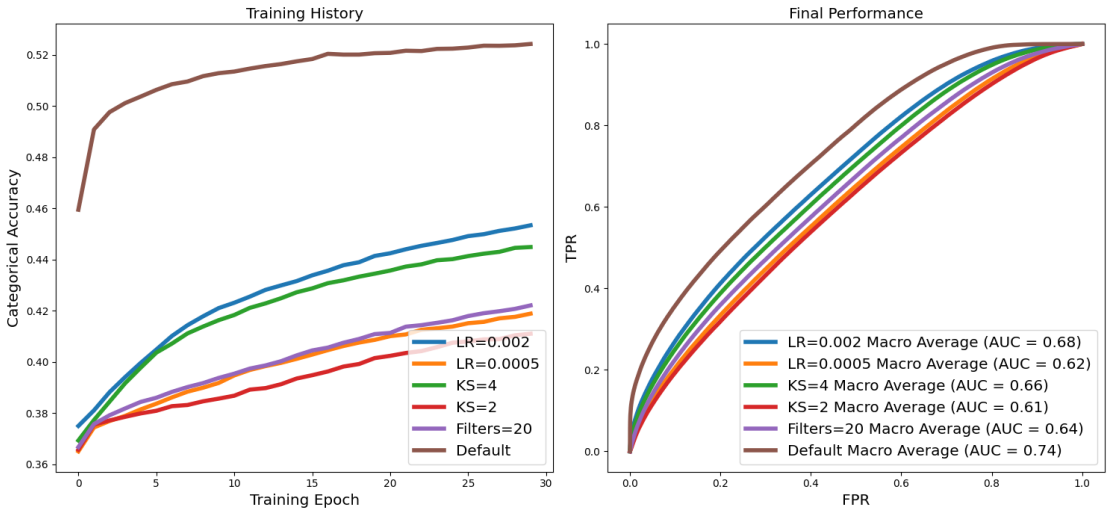


Figure 3.9: This plot shows the effect of modifying the hyperparameters for training Method B on the diffuse data. Left: The categorical accuracy as a function of epoch for the modified hyperparameter training runs. Right: The associated test ROC curves.

and 3.8, as the classifier differentiates between γ-rays and other classes easily in the point source case for methods A and B, but not between the diffuse protons and electrons. But it is noteworthy that the ConvLSTM2D network performs this task so well and that methods A and B perform this in a superior way to methods C and D. This bodes well for future studies into directional reconstruction using CNN-based methods, as well as demonstrating the merit for multi-task deep learning studies that attempt to perform the tasks of event classification and directional reconstruction simultaneously (such as [160]). This knowledge of the strong directional sensitivity of CNN-type methods can be used to inform the design of future CNN-based analysis pipelines for CTA.

In the diffuse case (where all three event classes come from an even spread of directions), methods C and D overtrain and cannot distinguish between any of the event classes. The timing based methods A and B appear able to perform good event classification based on shower morphology in Figure 3.8 between the (combined) γ-ray and electron showers versus protons (that could be optimised further), but they cannot distinguish between the γ-ray and electron induced showers. The classification rates for the four different methods are shown in Table 3.4, despite using many more pixel maps method B classifies at $\sim 70\%$ the rate of the other

methods. Despite including more information than method A, method B doesn't appear to offer any significant improvement. This is likely as the timing information is likely to be the key component in the classification, though the marginally increased number of trainable parameters (as seen in Table 3.2) for method B might disadvantage it slightly on a differential test basis.

3.5.3 Effects of Hyperparameter Selection

In order to test the effect of varying hyperparameters, we ran additional training runs for Method B on the diffuse data. These runs were identical to the previous training runs for method B bar a single change each from the 'default' hyperparameters shown in Table 3.3 and Figure 3.5. One run each had doubled and halved the Learning Rate (LR) from the default Adadelta value of 0.001, one run had all ConvLSTM2D kernel filter sizes (KS) set to 2x2, one with all filter sizes set to 4x4 and a final run with the number of filters per layer set to 20 rather than 30 (this value could not be increased due to VRAM constraints). The training accuracy curves (all showing reasonable convergence) and test results from this are shown in Figure 3.9. The 'default' hyperparameter setup outperforms the alternatives, however this is to be expected as changing the *Adadelta* optimiser parameters is not recommended by the developers of *Keras* [94], and the 3x3 filter size option also provided the good results in the original ConvLSTM2D paper [155]. Whilst this gives us some idea of the spread in performance due to hyperparameter selection, this is not a substitute for an exhaustive hyperparameter space search, the optimal combinations of which are often not human-interpretable. The uncertainty associated with individual event classifications is also not trivial to obtain with CNN-type methods [113] [114], and this is likely to be a necessary area of further development for these analysis techniques.

3.5.4 Convergence and the Effect of Early Stopping

In order to investigate the influence of overtraining of methods C and D in the initial diffuse runs (as seen in Figure 3.6) with the original hyperparameter configuration

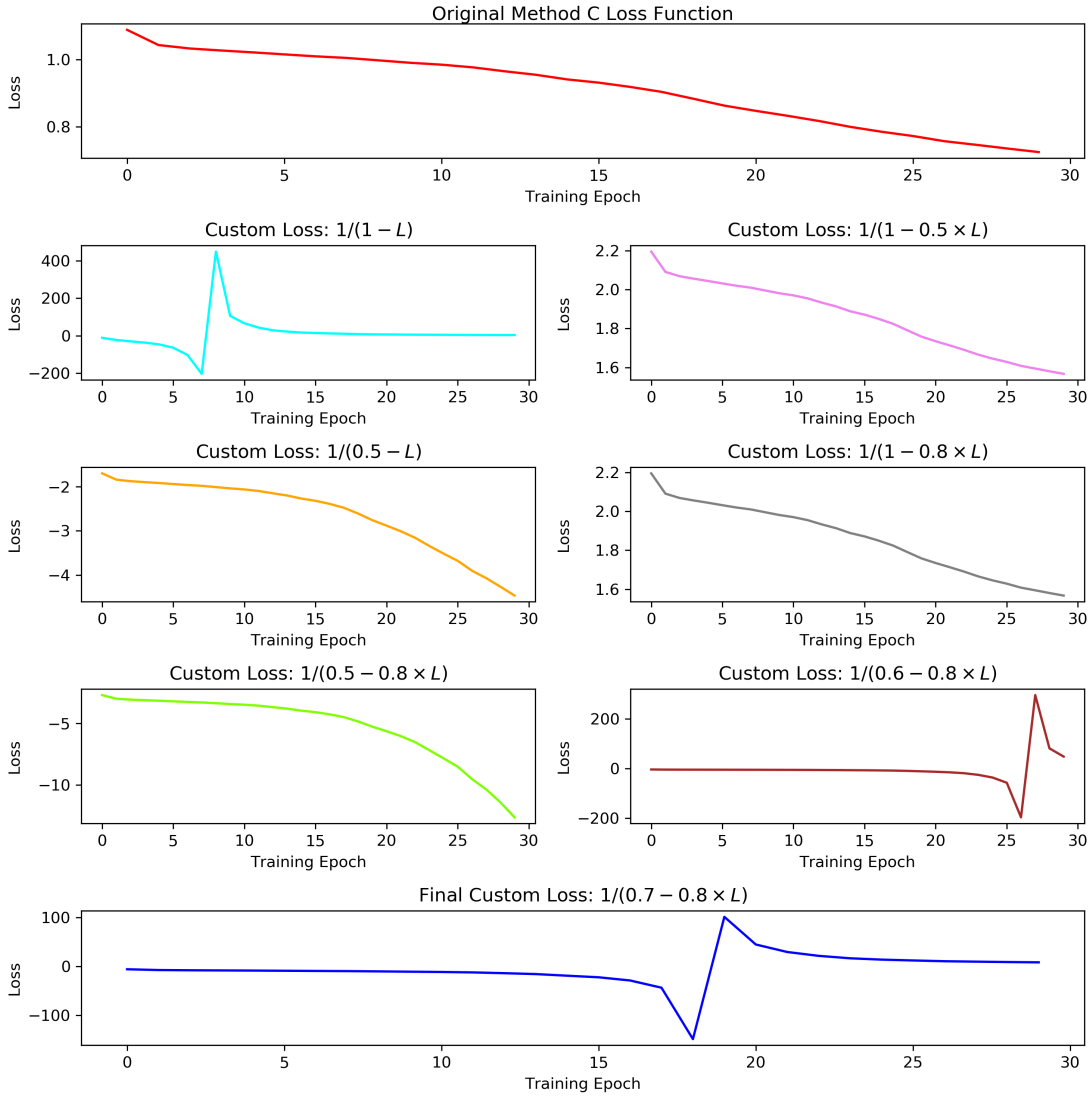


Figure 3.10: An illustration of the search for a metric with a turning point at an appropriate point to prevent method C overtraining. The selected final metric is very finely tuned to our earlier results for method C and is not suitable for general use.

from Table 3.3 and Figure 3.5, we ran additional training and testing runs on this diffuse data using this 'default' configuration along with two early stopping criteria (that we reference by the abbreviation ES). Early stopping in *Keras* allows for the stopping of training when there is an inversion in a chosen metric function, and for the restoration of the earlier optimal model weights obtained during training. In the case of preventing the overtraining scenario we observed for method D (where there's a clear inversion in the loss at epoch 6), the choice of this early stopping metric is the total loss function used during training. However, choosing such a

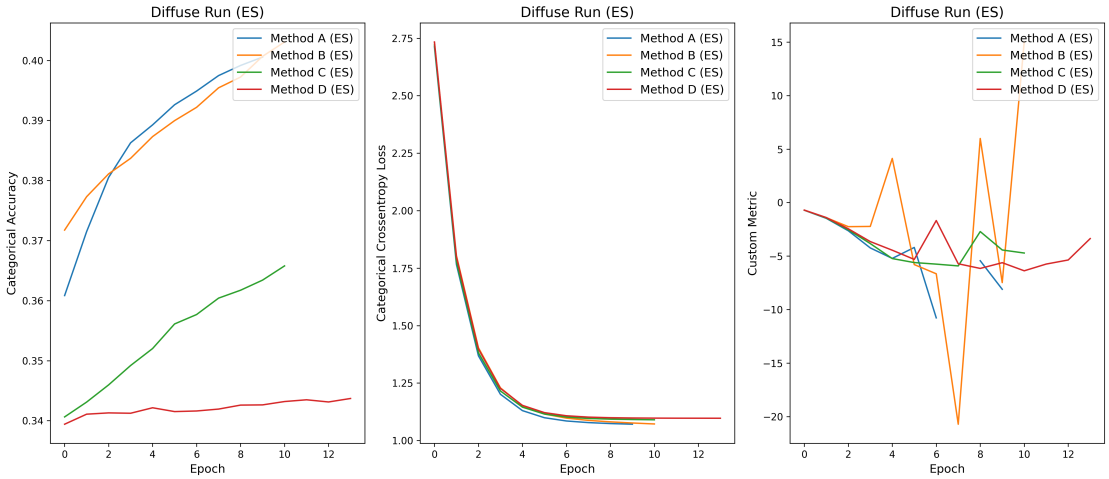


Figure 3.11: Training total loss, categorical accuracy, and custom metric (M) values for the four ES runs, all on the diffuse data. The custom metric curve for method A diverges towards an infinite value due to finite numerical precision on epoch 7.

metric to prevent the overtraining scenario we observed for Method C whereby there is undesired, runaway acceleration in the training data loss is more complex. We searched by hand for a metric that had a turning point (that will trigger the *Keras* early stopping) around the epoch that method C appeared to start overtraining in the original training runs. An illustration of this process is shown in Figure 3.10; the selected metric has the form

$$M = \frac{1}{0.7 - 0.8 \times L} \tag{3.1}$$

where L is the total loss shown in for example Figure 3.6. This L includes both the categorical cross-entropy loss and the regularization penalties which we explicitly sum over the first two ConvLSTM2D layers. For both metrics we choose a *patience* parameter of 3, representing the number of epochs to tolerate the inversion of the metrics.

The training curves and results from these ES runs can be seen in Figures 3.11 and 3.12. During the ES training for Method A, the early stopping was triggered on epoch 10, for Method B and C on epoch 11, and Method D on epoch 14. We still observe superior performance (and some ability to perform γ/hadron separation using morphological information) for the timing based methods A and B, despite this early stopping procedure (which was human-engineered for Method C). As such,

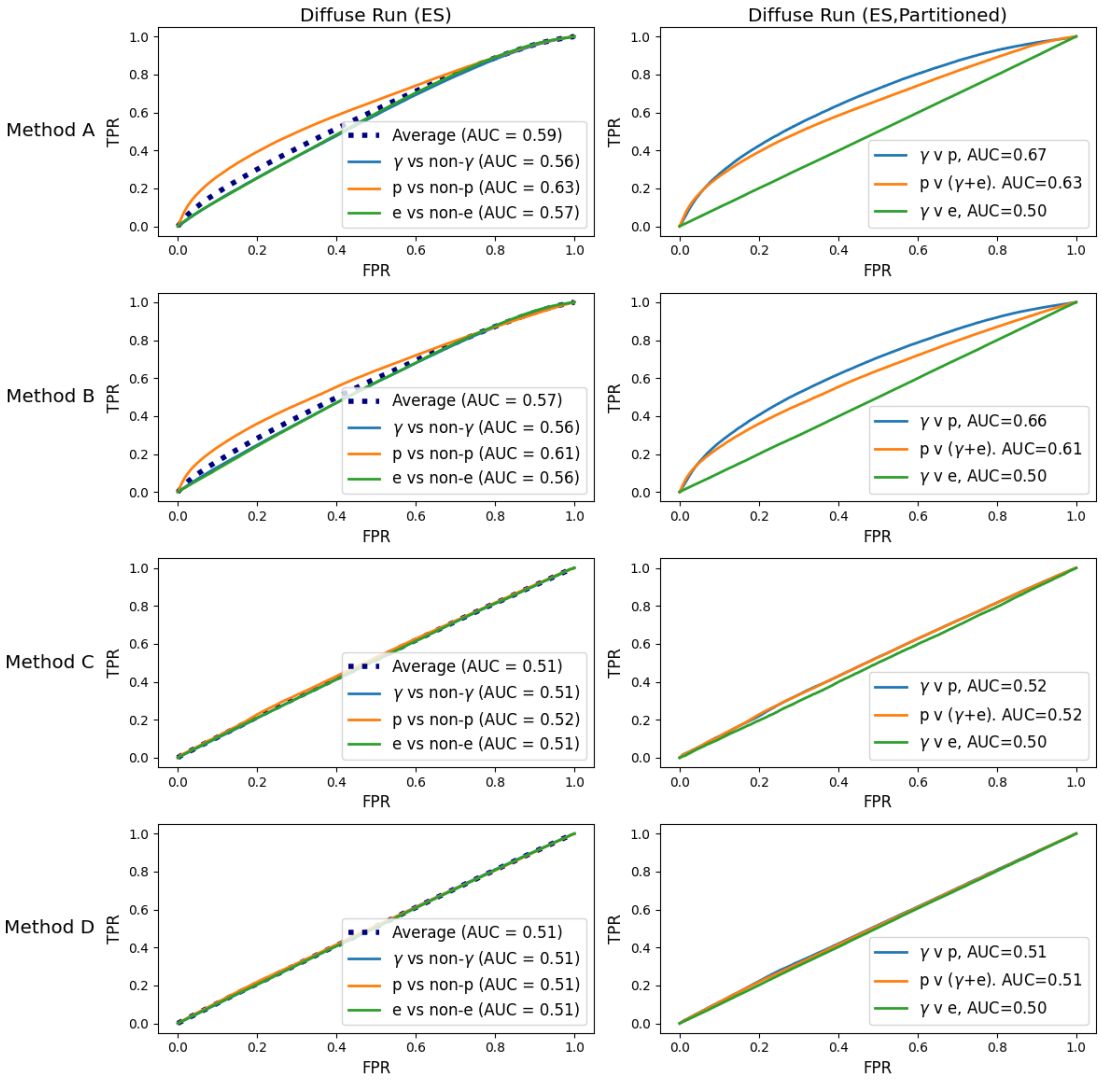


Figure 3.12: This figure shows the final results from the ES runs, which are only trained and tested on diffuse data. Left: The results presented in the standard machine learning one versus all approach as in Figure 3.7. Right: The same ES results shown in the astrophysical context of Figure 3.8.

we believe the superior performance of methods A and B in the diffuse case cannot be solely attributed to a bias from the non-convergence of the initial training for methods C and D, and is a result of the charge-only dataset used for methods C and D rather than an effect due to the design or optimisation of the classifier network.

The charge-only methods [107] (methods C and D) perform poorly in our analysis in both the point source and diffuse cases (regardless of whether early stopping is used or not); this isn't to say that such charge-only methods could not

function well given additional hyperparameter optimisation and image cleaning. We seek simply to demonstrate the effectiveness of our timing techniques on a differential basis, which is confirmed by our results. As such, the absolute ROC, AUC, loss and accuracy values here should not be considered to be optimal for any of the four methods. It should also be noted that the realistic presence of electron events in the training datasets likely makes training the networks in the diffuse case significantly harder than straightforward γ-hadron separation given their similarity to γ-ray events, and this is likely part of the reason these two methods perform poorly in the diffuse case. Part of our reasoning behind these conclusions is that all four methods are affected to roughly the same extent (~ 0.2 AUC) by the change between (non-ES) diffuse and point source runs. Small variations in the accuracy values between the different methods could be statistical or due to hyperparameter configuration; averaging the results from multiple trials with different random seeds might help to confirm this, however this would be highly computationally intensive.

3.6 Investigation into Identified Features

As mentioned earlier, whilst methods of feature identification do not exist for stereoscopic analysis, they do exist for single image classifier CNNs. Performing such analysis on single IACT images is worthwhile as it can demonstrate the capability of convolutional layers to extract physically meaningful features in IACT images. As such, we trained a single CNN classifier on single telescope timing pixel maps from the diffuse dataset (i.e. Method B), the architecture of which can be seen in Figure 3.13. As in the stereoscopic case, the images are cropped to the central region of the camera and are normalized, but the electrons are removed to ease training. In order to perform the feature importance analysis, this network had a different architecture to the ConvLSTM2Ds used earlier in this chapter. Additionally a cut on the size of the timing pixel maps was necessary to account for the fact that the simulation configuration necessary for multiple telescope analysis can produce a number of events with a low overall charge for given individual camera. Unfortunately this generated a greater than expected number of partially

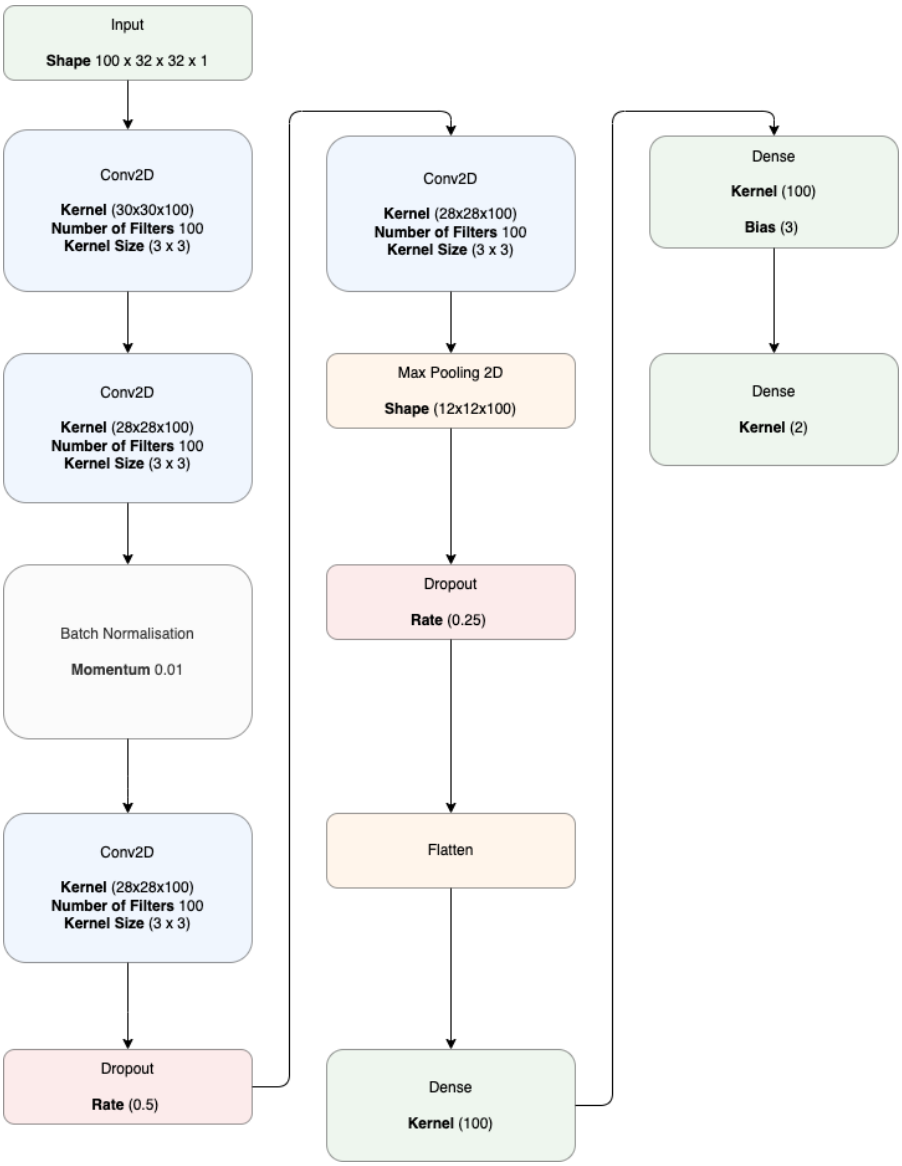


Figure 3.13: The architecture of the single image CNN used for our LRP investigation. The architecture differed from that used for stereoscopic analysis by necessity, both to achieve reasonable test accuracy and for compatibility with *deepexplain*.

truncated events, which is not the case more widely in the dataset (as can be seen in Figure 3.2). There are two primary established techniques for performing this salient feature identification [159]; the first is to iteratively mask out features in input images and observe the change in CNN predictions (so-called occlusion methods). The second is to perform a weighted sum backwards through the network and identify the pixels that contributed most to a classification (so-called gradient based methods). We focus on the latter as it allows for pixels which counted against

the classification to also be identified. Specifically in our case using the Layerwise Relevance Propagation (LRP) technique [161]. LRP algorithms compute a sum backwards through a trained network, starting at the output layer it determines a 'relevance' $r_i^{(l)}$ to every ith neuron in each layer l until it reaches the input layer. In this output layer, the prediction score is defined as

$$r_i^{\text{Output Layer}} = \begin{cases} S_i, \text{ if the unit } i \text{ is the target unit of interest} \\ 0, \text{ otherwise} \end{cases} \tag{3.2}$$

the LRP algorithm then redistributes S_i as it moves backwards through the network until the input layer is reached by performing the following sum for each layer

$$r_i^{(l)} = \sum_j \frac{z_{ji}}{\sum_{i'}(z_{ji'} + b_j) + \epsilon \cdot \text{sign}(\sum_{i'}(z_{ji'} + b_j))} r_j^{(l+1)} \tag{3.3}$$

where $z_{ji} = w_{ji}^{(l+1,l)} x_i^{(l)}$, w_{ji} is the weight associated with the jith connection in the neuron, x_i is an element of the input vector to a neuron, b_j is a bias term (that is not used in this particular work) and ϵ is a small term added to improve numerical stability. The ultimate values we want to compute are the relevances for the input layer $r_i^{(1)}$. For further details of the LRP technique see [161]; we used the *deepexplain* package to perform this calculation.

The results from this investigation are show in Figure 3.14, the classifier achieved a 70% test accuracy (likely a result of the training dataset not being designed for mono-instrument analysis). It appears that the position of the shower in the image correlates well with the relevances calculated, and the correct features associated with hadronic interactions are being identified. Additionally, the pixels with the greatest positive or negative relevance appear to be located near the edge of the shower, which is promising as it indicates that the classifier is learning that proton showers are both wider than γ-rays and can have electromagnetic shower substructure.

3.7 Comparison to RFs

Performing a fair comparison between existing Hillas-parameter-based techniques and CNN-based methods is an involved problem, especially as CTA reference

analyses have yet to be established. An initial comparison can be found in the Shilon et. al work [107], several problems remain unsolved. These include: how to robustly perform such comparisons include the quantification of the computational cost, the management of the sensitivity versus robustness trade-off, and the correct way to co-optimise the classifiers. Whilst BDTs/RFs may be trivial to optimise, the same cannot be said for CNN-type methods, and this ultimately this might limit their practicability.

In this section we provide a crude comparison of our deep learning methods to RF classifiers trained on the same data. Whilst a detailed comparison is difficult for the reasons described above, it is particularly important for us to demonstrate that attempting to train a parametric method on this extremely challenging dataset is not trivial. A 'Pythonic' tool for generating lookup tables suitable for generating the weight terms in the formula for mean reduced scaled width and lengths does not currently exist, so we simply use averaged length and width of the images (however in this special case of a very symmetric array looking at high azimuth it is unlikely to have a major effect).

To implement this, we reverse engineered the Hillas parameter extractor from *ctapipe* to extract Hillas parameters from the diffuse and point source training datasets used with the ConvLSTM2Ds. We then used the *scikit-learn* python package to train a RF conditioned on the existing training events, and tested them against the same test events. In order to test the effect of including electrons in the analysis, we also compare the accuracies of RFs trained and tested without these events (this was computationally prohibitive to investigate with the ConvLSTMs). The RF classifier used the *scikit-learn* default hyperparameters, with 100 estimators and using the 'Gini impurity' to measure the quality of the split [97].

Figure 3.15 clearly shows the (expected) inferior performances of the RFs on this simulated data, as well as clearly demonstrating that the presence of electrons in the data has a significant effect on the average AUC value. However given the issues with applying CNN-type methods to real data explored in the next chapter, it is likely these RFs would still outperform the ConvLSTM on real observations.

3.8 Conclusions

We have investigated four methods of applying new deep learning methods to the task of classifying events for IACTs, finding that the use of timing information is an viable method of event discrimination against simulated hadronic air showers. Whilst not a full flux sensitivity analysis, our results demonstrate the value in including timing information in future CTA deep learning analyses. This became possible thanks to the significant technical innovation that has gone into the design of CHEC and other similar modern Cherenkov cameras. Our work also demonstrates that there is little prospect of CNN-based methods being used to filter out electron induced air showers, and so in future IACT deep learning studies (in the SST energy range) electrons should be treated as an irreducible background. That said, future IACT instruments (such as the CTA-affiliated Schwarzschild-Couder Telescope with it's 12000 pixel camera [162]) may provide even more detailed images of EAS that might make electron event rejection feasible if used in combination with these techniques.

Another area of uncertainty related to this research is whether the representation of stereoscopic timing information in *sim_telarray* is sufficiently accurate to be used for these purposes. In the absence of multiple SSTs existing on site this is difficult to currently test, however we speculate that it might be possible to overcome such errors by simply adding offsets to the timing pixel maps based on telescope pointing and shower direction.

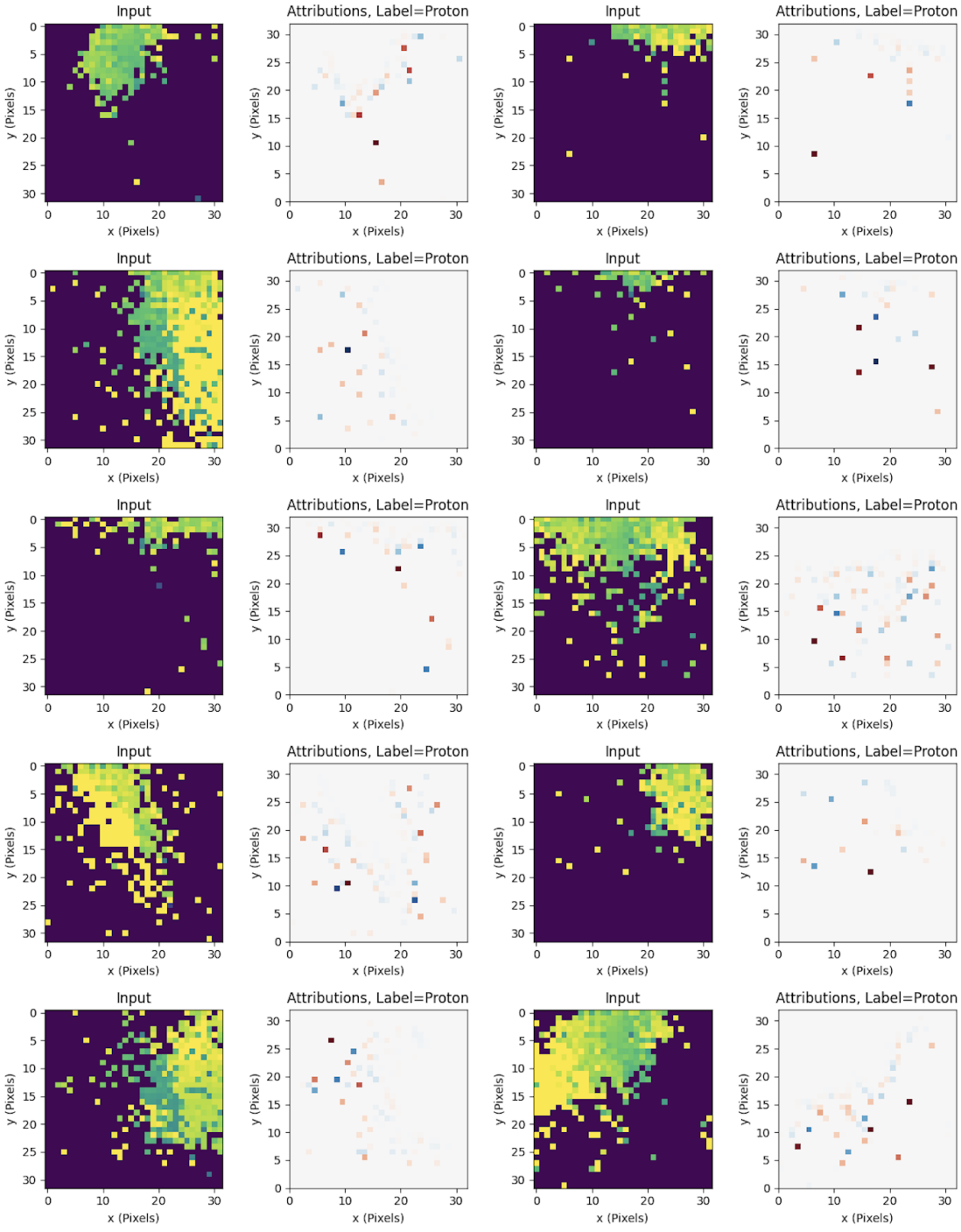


Figure 3.14: The results of our LRP investigations. Test images fed to the network are shown along with the feature maps associated with them, where deeper shades of red indicate the pixels that contributed the most to the classification as a gamma ray induced shower or hadronic shower and deeper shades of blue pixels that counted against the classification. The intensity cut preferentially selects proton events at the highest energies.

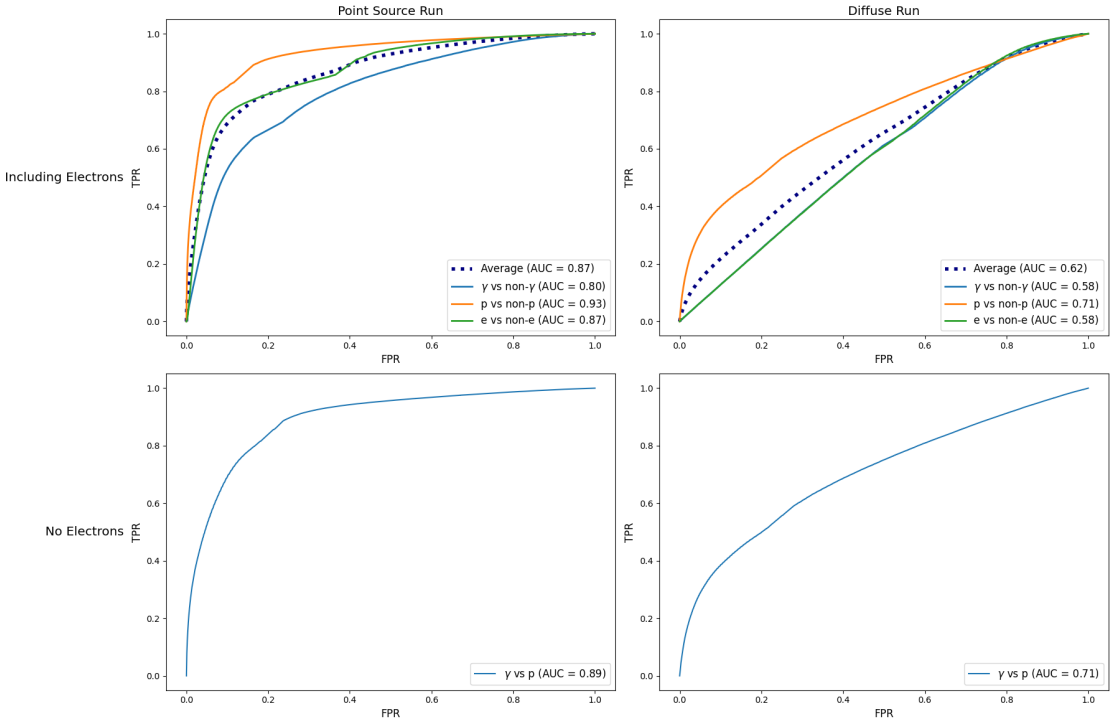


Figure 3.15: Point source and Diffuse RF test ROC curves with AUC values shown, showing the effect of inclusion of electrons.

4

Challenges of Applying Deep Learning Event Classification Methods to Real Observations with VERITAS

Contents

Abstract

It is becoming increasingly recognised that artefacts in real IACT observations have the potential to seriously disrupt the efficacy of deep-learning-based event classifiers. This issue has so far been relatively poorly understood, and these artefacts are not modelled in simulations. In this chapter, we attempt to investigate the difficulties in performing observations with real observations from VERITAS when a deep learning event classifier is used. In contrast with previous efforts with H.E.S.S. data, we do not use tailcut image cleaning with charge data. We also explore the limitations of using a custom simulation approach as a means of mitigating real observations issues, as well as the utilisation of Bayesian optimisation (which for the first time we attempt to use against real IACT observations). After developing a pipeline for performing deep learning analysis with VERITAS data, we present the first detection of the Crab Nebula using a deep learning event classifier with a stereoscopic IACT array. We conclude that with current computational power and techniques, the complexity and cost of optimising deep learning event classifiers in this manner is significant. Whilst these methods are still promising, this limits the current applicability of these methods to current generation instruments or data from CTA.

4.1 Introduction

In this chapter, we explore potential issues with the application of CNN-based classifiers that have been trained with simulations against real low-level data from VERITAS. In contrast with Chapter 3 we cannot use timing information with VERITAS. This is as VERITAS has a ~ 7 ns timing jitter in the peak time results (large given that the typical duration of a Cherenkov shower is approximately 10 ns), as well as a 1.2 ns resolution limit from the array-level trigger [55]. Therefore we concern ourselves with methods using integrated charge-only images, similar to more conventional IACT analysis. Given the results in Chapter 3, where we found that there was no potential for separating electron-induced events from γ-ray events, we do not include electrons in this analysis.

Our ConvLSTM algorithms have thus far been trained entirely with simulated data. There are potentially serious issues to be considered when applying these techniques with real observations: variations in night sky background levels, disabled pixels or front end electronics, bright stars in the field of view, cloud cover and the reduction of telescope optical efficiency with age could all create biases in the CNN response. Not all of these effects are trivial to simulate. The aim of this chapter is to determine the extent to which these artefacts influence the ability of deep learning methods to detect astrophysical sources in real observations, and determine what, if any, strategies to combat this might be feasible. This problem of training deep learning models with simulated data is not unique to IACTs; training computer vision models with simulations is a problem present in many other research areas such as autonomous driving [163] and facial recognition [164].

We investigate two novel approaches. The first is the application of custom Monte Carlo simulations as training data, and the second the use of Bayesian optimisation to determine model hyperparameters.

4.2 A VERITAS Deep Learning Analysis Chain

Unlike for CTA, where the complete pipeline for taking IACT event images and creating astronomical data products is still under development, VERITAS has a very stable data pipeline which has been under development since 2004. This stability is part of our motivation for utilising VERITAS for this study. As the earliest studies of deep learning analyses for IACTs only date back to 2016 [116], the VERITAS pipeline was not designed for this purpose. Developing an analysis chain capable of taking real and simulated VERITAS data and producing detections using deep learning analysis was therefore a complex task. This section details the stages to this analysis, and how we developed a 'bootstrap' counterpart deep learning analysis chain. We will begin by discussing the steps of performing standard IACT analysis using the *Eventdisplay* package [1].

4.2.1 *Eventdisplay*

Eventdisplay (primarily written by G. Maier and J. Holder) [66] forms the basis of one of the two analysis chains used for VERITAS. It has also been used to generate IRFs for CTA. Using the ROOT software framework [34], the *Eventdisplay* pipeline can be used to generate astronomical spectra, lightcurves and skymaps from IACT data. Its main steps are:

- The Eventdisplay Step: This concerns low-level calibration of the VERITAS camera images (such as pedestal subtraction and flat-fielding) along with generation of Hillas parameters. This step can also be used to produce Data Storage Tape (DST) files containing calibrated event images, which are not normally stored during standard analysis, but which are required for performing deep learning analysis. This computationally intensive step is not generally stored to disk for reasons of efficiency.

[1] Throughout this chapter, we adopt some conventions standard in VERITAS. The 'gammaness' quantity can interchangeably be referred to as an 'isGamma score', similarly we swap ROC curves for 'Signal Efficiency' curves, which are equivalent bar the x axis being inverted.

- The Mean Scaled Width (MSCW) step: This takes the image-wise Hillas parameterisations of the images from the cameras and combines them to generate the mean scaled width and length parameters for a given event. BDT event classification is also performed at this step.

- The Anasum Step: This takes information from the MSCW step and uses it to generate the required data for the creation of Skymaps and Spectra (in particular the directional separation of events into ON and OFF regions).

- The Plotting Step: This contains two ROOT classes (VPlotAnasumHistograms and VEnergySpectrum) to plot astronomical sky significance maps and spectra.

Further details of specific elements of the analysis chain we implemented are detailed in the next two subsections.

4.2.2 Disabled Pixels

VERITAS's camera consists of standard (non-silicon) high-efficiency [165] Hamamatsu R10560-100-20 MOD vacuum photomultipliers. They have a quantum efficiency greater than 32%. It is unsafe to saturate these routinely, and such pixels have safeties built into them to shut the pixel down if the current in the PMT is too high (though this depends on the gain in the PMT). Particularly in the case of bright stars, this results in blocks of up to four pixels (due to the VERITAS Point Spread Function) that are classified as disabled and so don't record data in a given event window. Pixels may also be absent due to faulty electronics. Such disabled pixels are one of the prime culprits of discrepancies between real and simulated data, so we designed an interpolation strategy for the real observations that replaces the zeros in the image arrays for these pixels and interpolates based on pixel intensity values up to six pixels away in the image prior to the mapping stage (discussed in the next Subsection 4.4.2).

4.2.3 Image Mapping

The VERITAS cameras consist of 499 photomultipliers as shown in Figure 4.1 [165]. As the camera is not square, an appropriate geometric transformation must be applied. To do this, we use the ImageMapping classes from the CTA dl1-data-handler project [147]. The analysis pipeline we have developed (see Section 2.9) writes, and can process, VERITAS data transformed in 8 different ways: axial addressing, oversampling, rebinning, nearest interpolation, bilinear interpolation, bicubic interpolation and image shifting. However, internal results from the CTA consortium have shown that the effect of the difference between these mapping methods is negligible on simulations [166]. So we primarily concern ourselves with oversampling for the rest of this chapter; a future study could look into this issue in further detail with this pipeline. Oversampling works by dividing the charge from a hexagonal pixel into $n \times n$ (typically $n = 2$) smaller square pixels; more detailed explanations of these methods can be found in [127, 167].

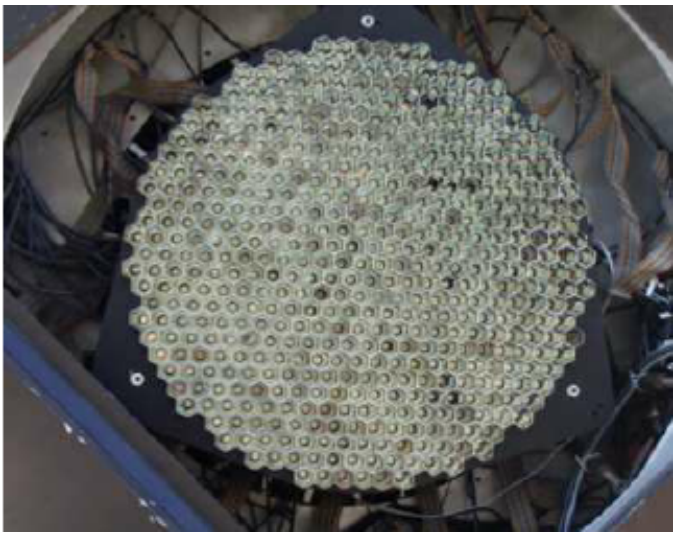


Figure 4.1: A VERITAS camera, taken from [165]. Note the use of Winston cones to cover the entire focal plane, these maximise the effective light collection area of the conventional photomultipliers used [168].

4.3 Simulations

The first issue that we contend with in this section is choice of training dataset; which also encompasses choices regarding image cleaning and data pre-selection cuts. The training data should resemble the real observations as closely as possible, but there is a computational trade off between using run-wise simulations (which match the particular set of observations as closely as possible) and completely diffuse training data. All previous attempts at using CNN-based methods have relied upon using tailcut-cleaned images [107] for both the simulated training data and the real test data. We consider this to be counter-productive for background rejection [2], despite the fact this procedure simplifies the real observations problem (as it removes the need to accurately model an entire image with for example 499 pixels versus ∼8). Our reasoning for considering tailcut cleaning unhelpful is that the purpose of investigating CNNs for γ/hadron separation is to utilise their sensitivity to minute features in the air shower images. These small features, such as hadronic halos [68] (Cherenkov light from the incident primary charged hadron) or subtle electromagnetic substructure would be removed by tailcut cleaning. As such, whilst using it may provide comparable results to BDTs on bright sources such as PKS 2155-304 [107], using tailcut cleaned charge images with CNNs for background rejection is not ideal.

Whilst there is a large amount of literature available on the problem of domain adaptation [169], the field remains in its relative infancy. Given the almost negligible performance improvement using conventional simulations with a CNN-type approach that utilise image cleaning seen in [107, 121], we speculate that one potential source of error with these CNN-type methods is discrepancies between these conventional simulation approach and the observational data. We speculate that one potential means of bridging this gap will be a partial run wise simulation approach in order to minimise these difference between training and test data. This is a new paradigm in γ-ray astronomy [148], pioneered by the H.E.S.S. collaboration, and entails

[2] This may not be the case for directional reconstruction studies, in which CNNs appear very promising

generating a unique simulation dataset for every run (28 minutes for H.E.S.S., 20 or 30 minutes for VERITAS) of data collection. The notable achievements of this strategy include the detection of the extension of Centaurus A by an IACT array [32]. The VERITAS simulation chain is limited in its ability to reproduce as many run wise effects as H.E.S.S., though we aim to reproduce this approach as much as possible. These simulations were generated using the *CORSIKA* and *CARE* [170] simulation packages. *CARE* fulfils the same instrument simulation role in VERITAS as *sim_telarray* does in CTA and H.E.S.S..

4.3.1 Magnetic Field Effects

In the following subsections we will discuss specific elements of the simulations we utilised. Because of the Lorentz force, and the comparatively strong geomagnetic field at the VERITAS site [171], VERITAS shower images are broadened in differing directions as a function of azimuth. This magnetic field has an overall magnitude of $47\,\mu$T, but the effect of this is is dependent upon the angle between the shower and magnetic field, so only a component of this has an effect on the shower morphology. To account for this, we match the altitude and azimuth of the simulations to the mean run values in Table 4.1, with a 5 degree opening angle for both the simulated γ-rays and protons.

4.3.2 Energy Thresholds

As with all new techniques, the energy thresholds for *CORSIKA* simulations that should be used when attempting to apply deep learning methods to real observations must be determined iteratively. Set the energy threshold too low, and the classifier will struggle to converge on training data (particularly for proton showers which produce relatively little Cherenkov light at low energies). But set it too high, and you might lose an improvement in sensitivity at low energies that CNN-type methods might bring. We concluded that using a *CORSIKA* energy range of $100\,\text{GeV} - 200\,\text{TeV}$ for the γ-ray simulations, and $300\,\text{GeV} - 600\,\text{TeV}$ for the proton simulations was appropriate as a first step, but further investigations to lower this

energy threshold could be performed (this was after achieving poor results with a lower energy threshold of 30 GeV for both protons and γ-rays). One of the issues with comparing CNNs with Hillas-parameter based methods is that for low energy events or events which only have a small number of triggered pixels in certain telescopes Hillas parameter extraction will fail. This is not the case for CNN-type methods which will attempt to classify sets of images from an event regardless of the number of triggered pixels in each telescope.

4.3.3 Energy Spectrum and Event Ratios

Throughout the work in Chapter 3, and through our initial investigations into real observations, we have been using simulations with power law spectra and equal ratios of events (albeit over differing energy ranges). It may be argued that this source model is overly simplistic for our purposes. Firstly, not every γ-ray source on the sky is equally bright, so the ratio of γ-ray to proton events will not be the same for every observation. Secondly, the spectra both of astrophysical gamma-ray sources and of the cosmic ray background is not a straight power law. This results in an energy dependence in the ratio of gamma-ray to proton events. Thirdly, there is a dependence on low-level trigger selection cuts, if data is only stored for events that produce a certain amount of Cherenkov light then this affects the ratio of stored γ-rays to stored protons. However, a counterargument to this is that with fewer events at a given energy the ConvLSTM will train less well; and there is a trade off to be had between training efficiency and resembling real observations closely. As such, in this chapter we attempt to balance these effects, and use the VERITAS standard -1.5 power law spectral index for the training data. Given the ratio of events after basic event pre-selection is approximately 1:1 for the Crab Nebula, we use this as our event ratio in our training data.

4.4 Gammaness Cut Optimisation

Gammaness (Γ) is the score given to a given event representing how closely the event resembles a γ-ray. In the *Eventdisplay* framework this is a float in the range

[-0.5,0.5]. Typically one discards events from the analysis which have a Γ below a particular value, which has to be optimised for the source flux in question.

Our 2DConvLSTM produces a vector ($\mathbf{x}$) containing two values (one for each event class, $\mathbf{x}_1$ for γ-rays $\mathbf{x}_0$ for protons). The values in this vector span the range [0,1] and must sum to 1. From this, we can construct a Γ score (equivalent to the *Eventdisplay* Gammaness parameter) in the range [-0.5,0.5] through the formula:

if $argmax(\mathbf{x})$=0 **then**

$\Gamma = 0.5 - \mathbf{x}_0$

else

$\Gamma = \mathbf{x}_1 - 0.5$

end if

There is a question as to how meaningful this particular score is, and the role to which model uncertainty might play. We discuss the possibility for future Bayesian Neural Network analysis in Chapter 6.

For conventional BDT analysis methods, the choice of Γ cutoff is optimised to maximise the signal-to-noise ratio for a source. However, this algorithm is known to be difficult to translate to deep learning methods, details of this issue can be found in [107]. For the purposes of our work in this chapter we simply pick a Γ value of 0.2.

4.5 The Crab Nebula

The Crab Nebula was the first astrophysical γ-ray source reliably detected from the ground, in 1989 by the Whipple Observatory [63]. It is the remnant of a massive star that exploded in 1054AD; this explosion was recorded by Chinese observers as a 'guest' star [36]. The first Whipple detection was due to a mixture of advances in Cherenkov camera technology and the advent of early Hillas analysis for background rejection. [172]. Earlier 'detections' largely relied upon pulsar folding methods that have since been found to be incorrect [62], partly as they were limited by available camera technology at the time. As a result, the Crab Nebula is one of the most extensively studied γ-ray objects, and has recently been found to be extended at IACT energies [173]. In 2019, it was found by the Tibet AS-γ experiment that the

Crab Nebula produced some of the highest energy photons ever detected, resulting in a 5σ detection of the Nebula at over 100 TeV [35]. 1.1 PeV photons recently detected by LHASSO are also evidence of the Crab being a galactic source of PeV cosmic rays (making the Crab a so-called PeVatron) [36]. These reasons make the Crab Nebula our first analysis target, though there are a number of complications associated with this (particularly the greater number of bright stars in the camera field relative to an extragalactic source).

4.6 Methods

4.6.1 Our Analysis Framework

Our framework to bootstrap the *Eventdisplay* analysis chain consists of three main steps:

- HDF5 file dumpers to convert both simulated and real VERITAS images in .root DST files into a format suitable for deep learning. This includes image mapping and mixing of proton/γ-ray simulations as appropriate. To obtain data from the ROOT DSTs we use the *root_numpy* [174] package.

- *Keras* scripts to perform training and testing of the ConvLSTMs, and write test predictions for a given event to file.

- Scripts to take a .root file generated using the normal analysis chain and add an additional tree containing the ConvLSTM predictions in order to make spectra and skymaps.

The analysis structure for performing this is shown in Figure 4.4, and the associated code can be found at `https://github.com/STSpencer/Getout`. Modifications to the *Eventdisplay* code were also necessary in order to handle both the new deep learning results and the use of run-wise simulations. A significant complicating factor was matching exactly the same number of predictions with the number of events in the data .root files. This mandated testing events individually with a batch size of

1, which slowed the deep learning analysis. A future sufficiently flexible end to end CTA pipeline would avoid this step; we discuss the potential for this in Chapter 6.

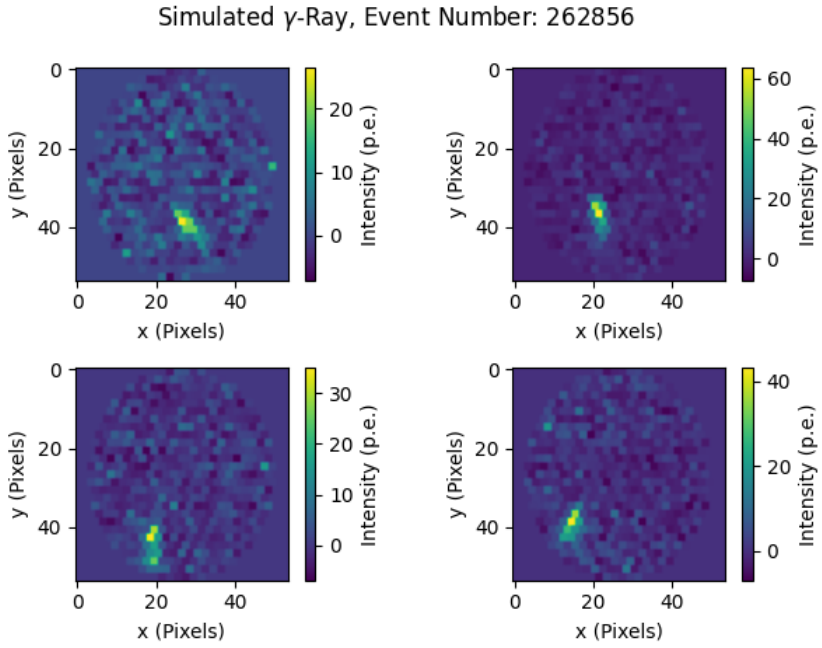


Figure 4.2: A simulated VERITAS MC γ-ray imaged using the oversampling technique. This event is shown as input to the ConvLSTM2D, without image cleaning applied.

4.6.2 Effective Areas

Effective areas represent the collection area of the IACT array as a function of energy (E), and are necessary to generate astrophysical spectra. Our pipeline can generate effective areas, and this gives us an idea of how well our event classifier performs as a function of energy. They are derived from a Γ cut value and Monte Carlo (MC) simulations of γ-rays through the formula

$$A_{eff}(E,\theta,\phi) = A_0 \left(\frac{\text{Number of } \gamma\text{-rays that pass event selection } (E,\theta,\phi)}{\text{Total number of } \gamma\text{-rays } (E,\theta,\phi)} \right) \quad (4.1)$$

which includes a dependence on zenith (θ) and azimuth (ϕ) angles and a normalisation constant A_0.

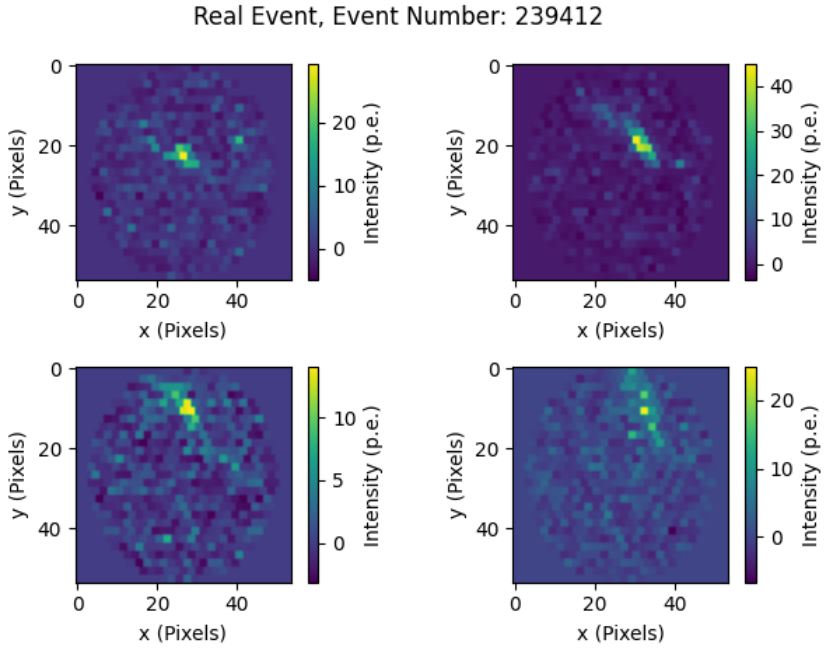


Figure 4.3: For comparison with Figure 4.2, a real VERITAS event from run 64080, for which there is no ground truth available. Note that the detector electronics is simulated whereas realistic NSB is not.

4.6.3 Issues with the Standard VERITAS Analysis Approach and Deep Learning

There were two main complications to performing deep learning analysis with *Eventdisplay*. Firstly, VERITAS does not normally use proton air shower simulations, and instead normally operates using BDTs trained on simulated gamma-rays with real proton air shower data as background. This is for reasons of speed, and because of the difficulty of simulating hadronic EAS. This approach is detailed in [171]. In our case for this purpose we utilised observations of the Tycho supernova remnant (not expected to contain a significant number of γ-rays) with loose Hillas parameter cuts. Early in this investigation we found that this was not going to be a viable approach for CNN-type background rejection methods as the ConvLSTM2D classifier was sufficiently sensitive to be able to determine with 100 percent efficiency the differences between simulated gammas and real proton air showers in training data. When applied to real test data this classifier was no better than a random number generator. As such, we resorted to using proton simulations as training

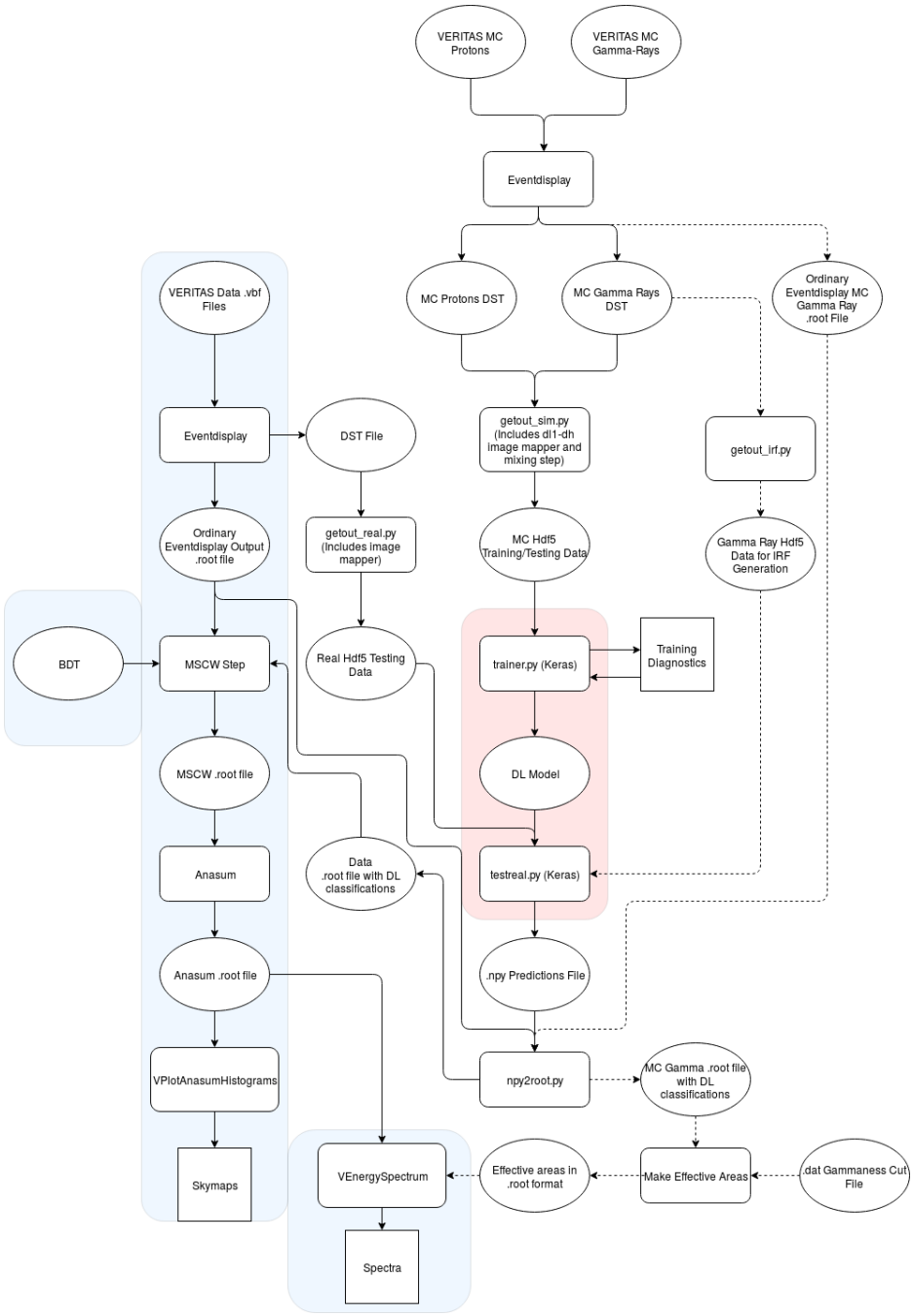


Figure 4.4: The analysis pipeline we have created in order to perform CNN-based event classification using the VERITAS array. Rectangles represent analysis scripts whereas ellipses represent stored data and squares output products. In the shaded blue region is the standard Eventdisplay pipelines, and on the right is the CNN classification pipeline. The shaded red region indicates code that was run on our group's Nvidia 1080Ti GPU in Oxford, for which there are associated transfer steps in order to relay this information to the computing cluster at DESY. The dashed lines show the pipeline for creation of Effective Areas. The only step not currently implemented is the generation of spectra.

data, in contrast to standard VERITAS operating procedure. Pre-trained BDTs trained with a different, more standard training sample (with simulated γ-rays and real protons) are supplied with *Eventdisplay*.

Secondly, the VERITAS analysis pipeline is not sufficiently flexible to support the generation of spectra from effective areas produced using our custom simulation approach. Though these effective areas themselves give us some idea of the classifier performance as a function of energy, albeit only on simulated data.

4.6.4 Analysis Cuts

One of the known issues with deep learning research for IACTs is that one needs to be wary of the effect of event pre-selection cuts. These can make it appear as if deep learning methods are performing roughly as well as conventional BDT/RF analysis. However, in such instances all the effective classification power comes from the cuts selected, and the deep learning classifier is acting as a random number generator due to the previously discussed discrepancies with simulated data. Performing harsh pre-selection cuts on training data (using either Hillas parameters or total intensity of images) will improve the AUC values obtained on simulations, but restricting ourselves to such obvious cases of γ-hadron separation again defeats the purpose of using CNN-based methods. If the only events we consider are events a BDT would easily be able to classify, then there is no potential benefit of using a CNN given their significant complexity and high computational cost. As such, we use minimal dataset cuts throughout this work. We perform no cuts on training data, and only minimal cuts on the MSCW and MSCL (to remove obviously hadronic events) as well as a necessary Theta-Squared cut of 0.008 deg^2 that are a default part of the *Eventdisplay* analysis chain on real observations. The cuts used are described in listing A.1 in Appendix A.

4.7 Initial Results

4.7.1 Run Summaries

We began our investigations by selecting which data to replicate. Run 64080 was selected for our investigation due to its high degree of prior investigation and stability, and clear (category A) weather conditions. The observing conditions for this run are summarised in Table 4.1.

Run Parameter	Value
Run Type	Observing
Weather	A
Run Start	2012-10-13 10:52:08
Observation Mode	Wobble
Offset (RA, DEC)	(0,0.5)
Participating Telescopes	T1 T2 T3 T4
Number of Events	543481
Elapsed Time L3 (seconds)	1201.8
Live Time L3 (seconds)	1013.2 (84.32%)
Mean Trigger Rate (Hz)	536.3
Mean Elevation (deg)	78.86
Mean RA, DEC (deg)	(83.64,22.52)

Table 4.1: Observing summary for run 64080

4.7.2 Effect of Custom Simulations

Figures 4.5, 4.6, 4.7, 4.8 and 4.9 show the initial results from the use of this pipeline and the custom simulations alone. The neural network was not completely optimised on simulations, scoring an 83% test accuracy and a 0.91 AUC (good results given the lack of event selection in the simulation test data). But the initial results on the real Crab data were poor, perhaps this was to be expected given the weak cuts and lack of image cleaning used in comparison to the Shilon et al. work [107]. In particular, it should also be noted that most of the background rejection in these results is due to a cut on the θ^2 Hillas parameter necessary to perform a reflected region analysis. Our initial detection was a $\sim 7.3\,\sigma$ detection of the Crab Nebula (which

includes very soft cuts on the MSCW and MSCL); full details are presented in Table 4.2. However, using the cut on the θ^2, MSCW and MSCL with a random number generator for Γ scores yields a $\sim 7.4\,\sigma$ detection. Evidence for the need for a more thorough analysis of the uncertainty in the model predictions was demonstrated by the significance not changing if the cut on Γ is changed from 0.2 to 0.4.

Run	**N**$_{on}$ (events)	**N**$_{off}$ (events)	**Significance** (σ)	**Signal Rate** (γ/min)	**Error on Signal Rate** (γ/min)	**Background Rate** (events/min)
64080	175	89.17	7.3	4.285	0.688	4.451

Table 4.2: Anasum output for custom simulations alone run.

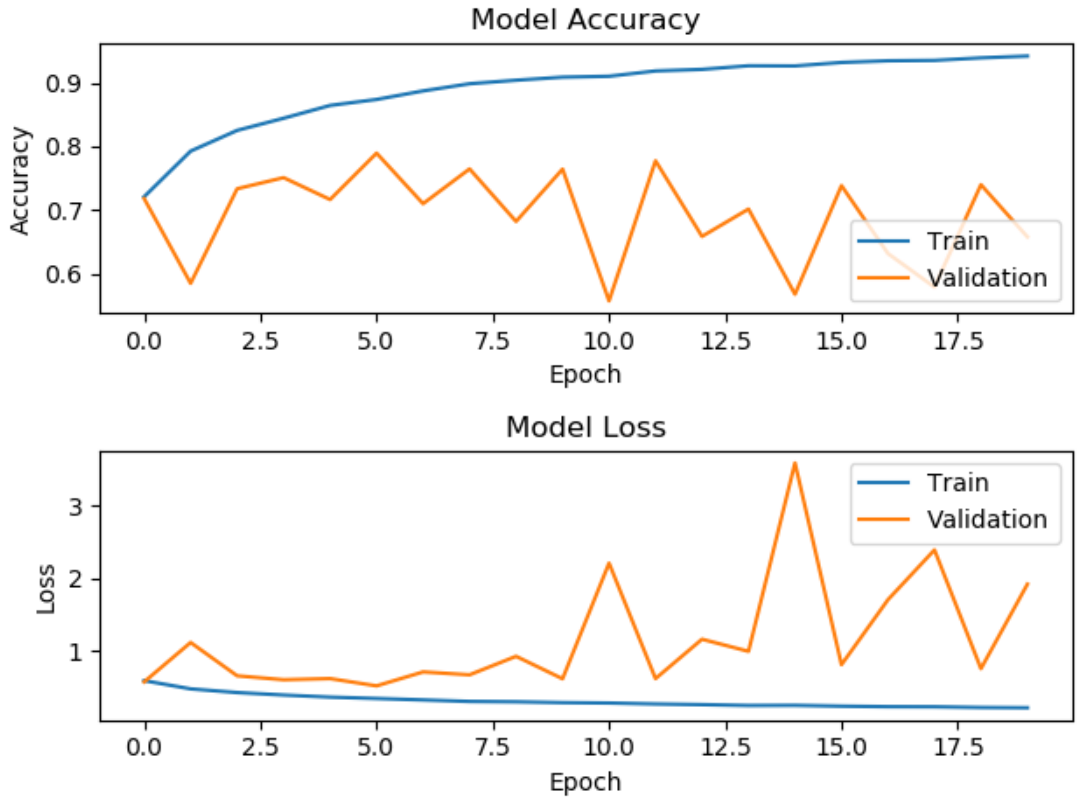


Figure 4.5: The training curve for the custom simulation alone run.

Given these results, it appears as though custom simulations (as they are achievable with VERITAS) on their own are not enough to bridge the domain gap between real IACT data and simulations. The poor, noisy performance on validation data throughout the training process is a common feature we observed in using these ConvLSTM2D methods, and is part of the reason we did not use

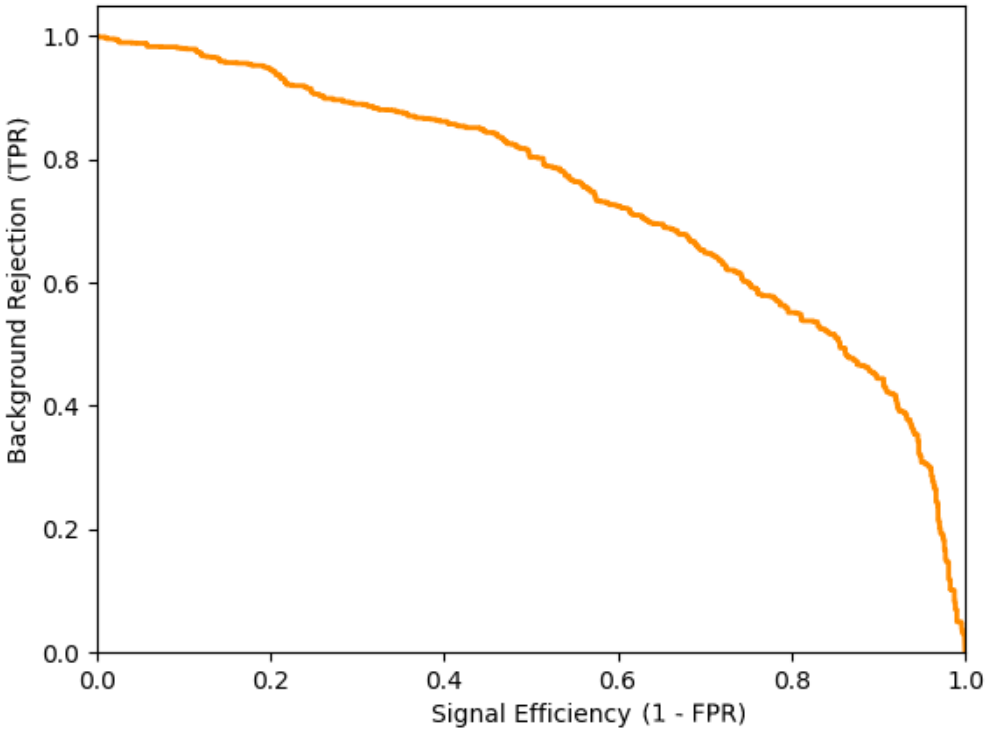


Figure 4.6: The test signal efficiency curve for the custom simulation alone run.

such data in Chapter 3. We will find that hyperparameter selection is an important element of this, but we could also be observing a result of the small batch sizes needed to fit such models onto GPUs with finite VRAM.

4.7.3 Hillas Width Analysis

There is a legitimate question of how closely the proton simulations described earlier in this chapter match the real observations and the γ-ray simulations. If the proton simulations have major discrepancies against the real observations, and if the proton simulations are not reasonably similar to the γ-ray simulations, this may cause issues with the classifier network. As a control, we retrofit the *ctapipe* Hillas extractor to extract Hillas widths from our VERITAS images for the four telescopes (CT1-4), to act on both our simulated data and the real event images (after disabled pixel interpolation).

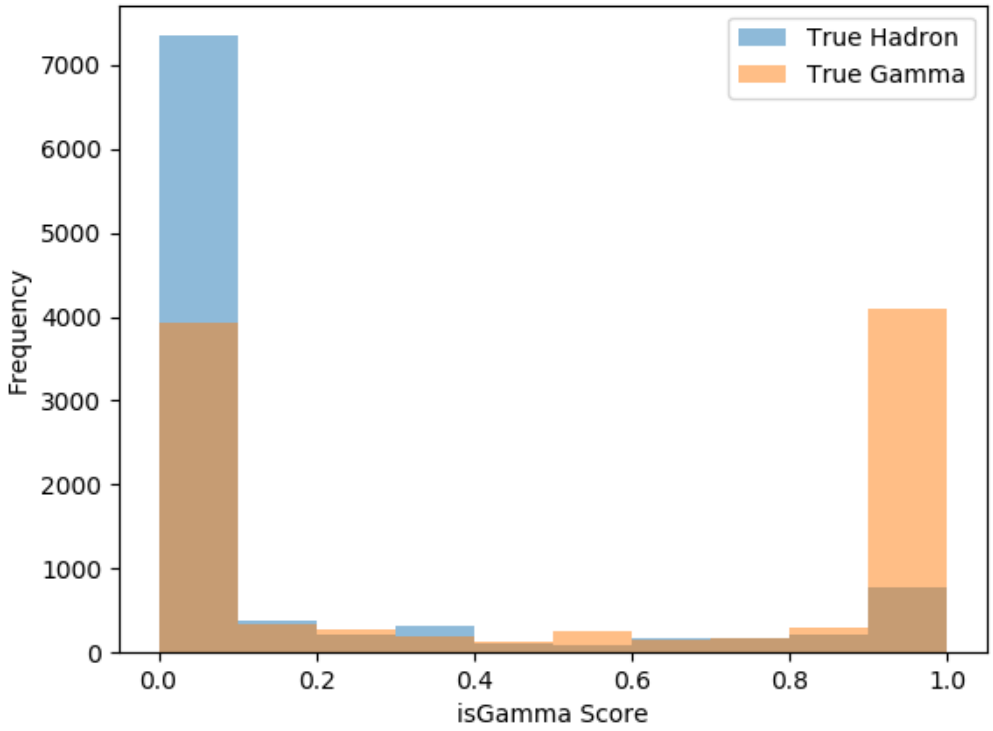


Figure 4.7: The test gammaness histogram for the custom simulation alone run. The model shows poor convergence on validation data.

The results from this investigation are shown in Figures 4.10, 4.11 and 4.12, and we show these results averaged by telescope and normalised such that the distributions integrate to unity in Figure 4.13. All three width distributions are of similar shape with close peak values, suggesting our simulation choice was sound. There are more events for the real data in Figure 4.12 here due to a *ctapipe* feature that cuts off the analysis if the Hillas parameter extraction fails for the first telescope. The reason for the per telescope discrepancies in distributions in the real observations in Figure 4.12 is because the real data contains both proton and γ-ray events; a similar per-telescope discrepancy can be seen in the simulated γ-rays in Figure 4.10. The total number of training events for the ConvLSTM analysis and the real observations are unaffected by this.

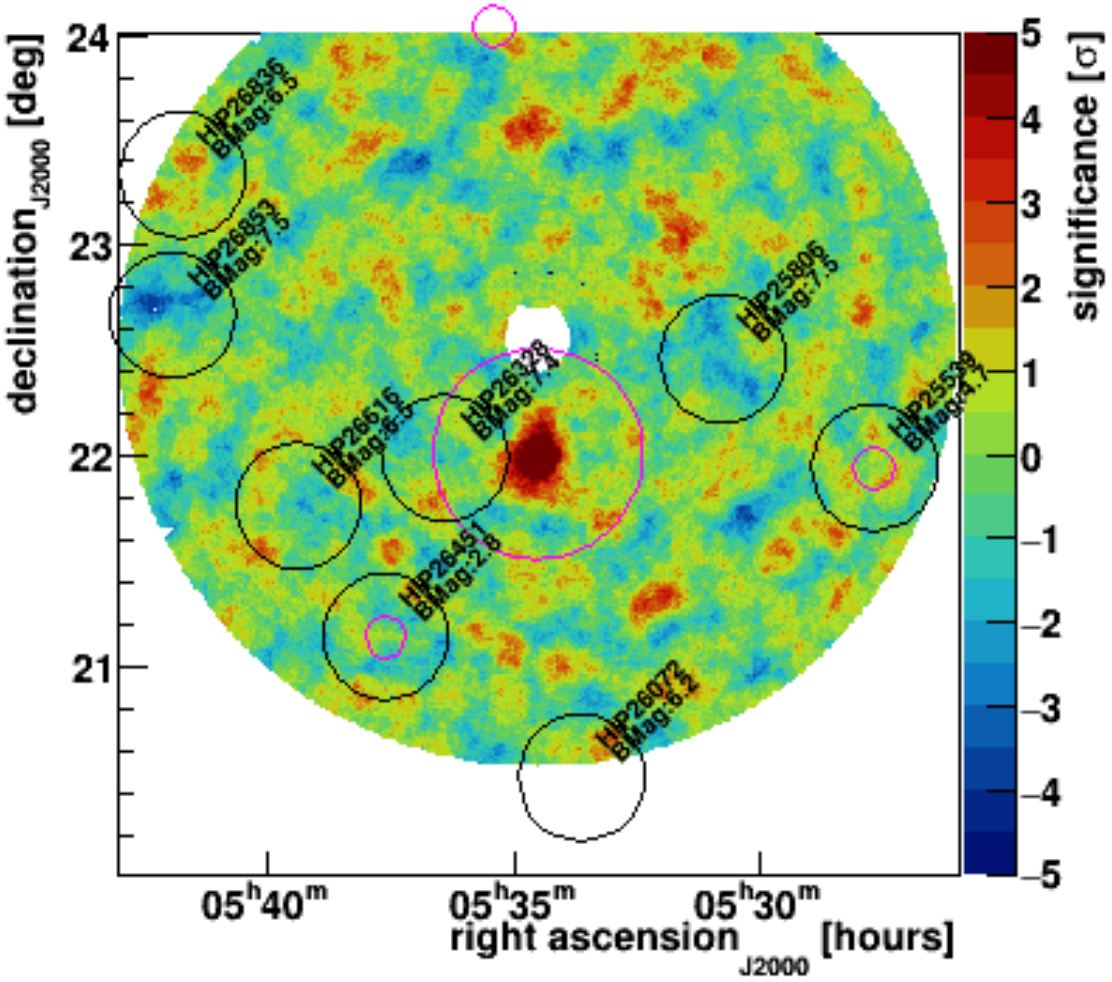


Figure 4.8: A γ-ray image of the Crab Nebula produced using the initial configuration, custom simulations and our analysis pipeline and our ConvLSTM background rejection technique along with a reflected region analysis. Regions containing bright stars recorded by the HIPPARCOS survey are also shown. Source and excluded regions are shown in purple.

4.7.4 Bayesian Optimisation

Hyperparameter selection can have a significant effect on deep learning performance, so we began to consider hyperparameter optimisation, in the hope that better performance on the simulated training data might translate into increased significance on-sky. This is not guaranteed to be the case; it is entirely possible that over-optimising the classifier performance on simulated data might cause the classifier to prioritise artefacts in the simulated data not present in the real

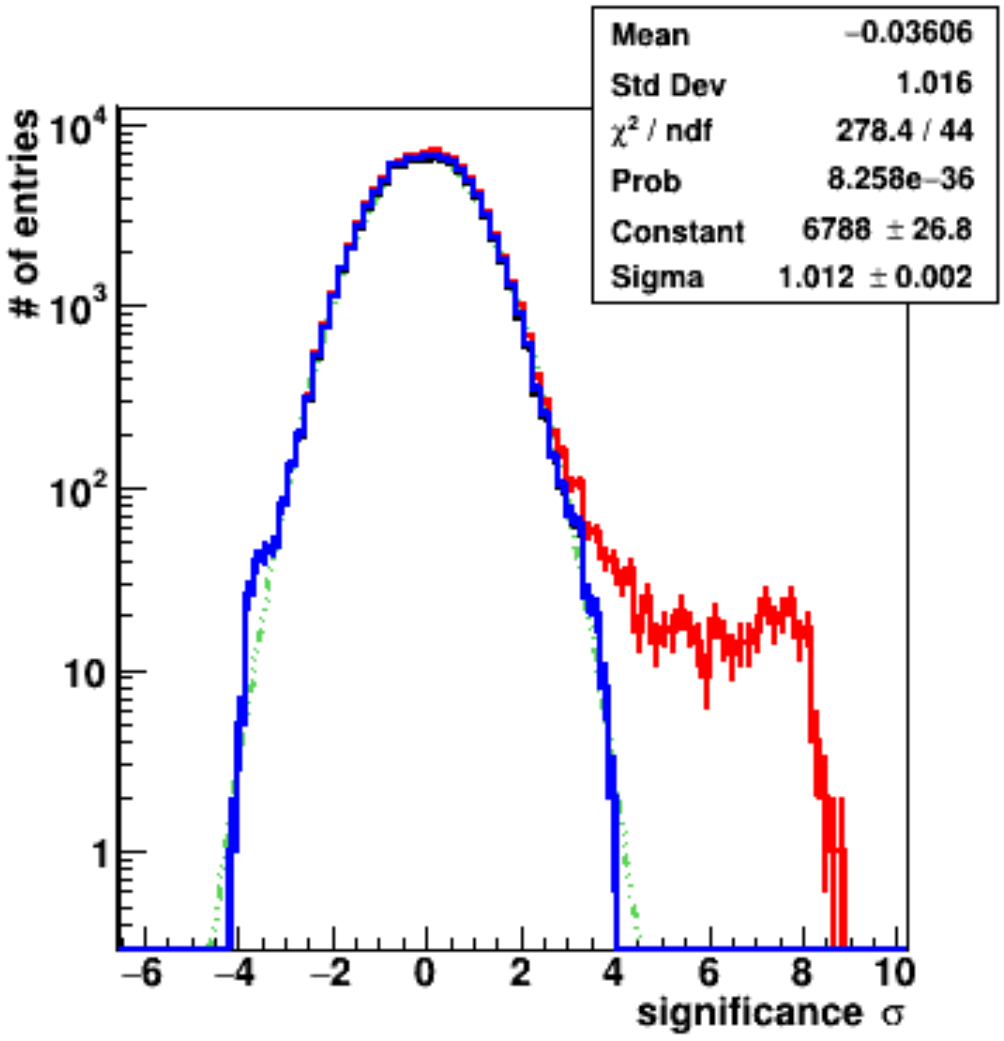


Figure 4.9: The 1D significance distribution including the source region (Red) and excluding it (Blue) and a Gaussian fit to the cosmic ray background (Green). Significances in IACT γ-ray astronomy are defined by [122].

observations. Hyperparameter optimisation is broadly defined as determining x^*, the set of hyperparameters that maximize the objective (minimise the loss) function $(f(x))$ of a machine learning algorithm on data x

$$x^* = \arg\min_{x \in \chi} f(x) \tag{4.2}$$

where x is in a space χ.

One of the most common approaches to hyperparameter optimisation is Bayesian optimisation, and in particular the Tree Parzen Estimator (TPE) approach [175] [176]. Alternatives include evolutionary (designed to mimic genetic mutation) and

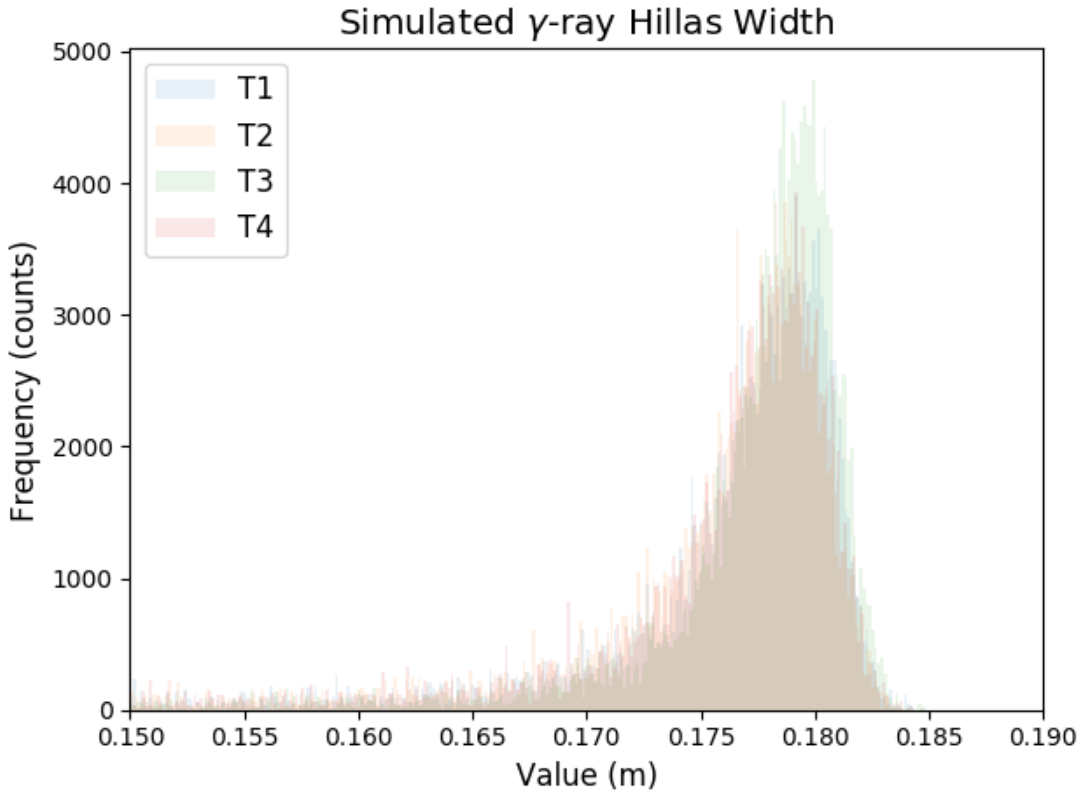


Figure 4.10: Hillas Width Distribution for the Simulated γ-rays for the four telescopes.

gradient-based approaches, we briefly consider the potential for these methods in Chapter 6. One of the key advantages of TPE analysis, unlike other Bayesian optimisation approaches (such as Gaussian processes), is that it is not necessary to specify a Bayesian prior in advance of the analysis. This is well suited to deep learning analysis as the effect of the hyperparameters of a CNN are not always directly human-interpretable. Additionally, one wants the results of prior trials to inform the future searches (formally referred to as sequential model-based optimisation), which is not possible with random and grid search techniques [176]. Leveraging this information, TPE optimisers have been shown to be substantially more effective than random searches on standard computer science datasets [175]. The advantage of using a tree-structure optimisation strategy is that network structure is inherently hierarchical; there's no reason to optimise the number of neurons in a given layer if one of the other optimised quantities is whether or not to include that layer [175]. Where y is the value of the objective function for a given set of hyperparameters x^*,

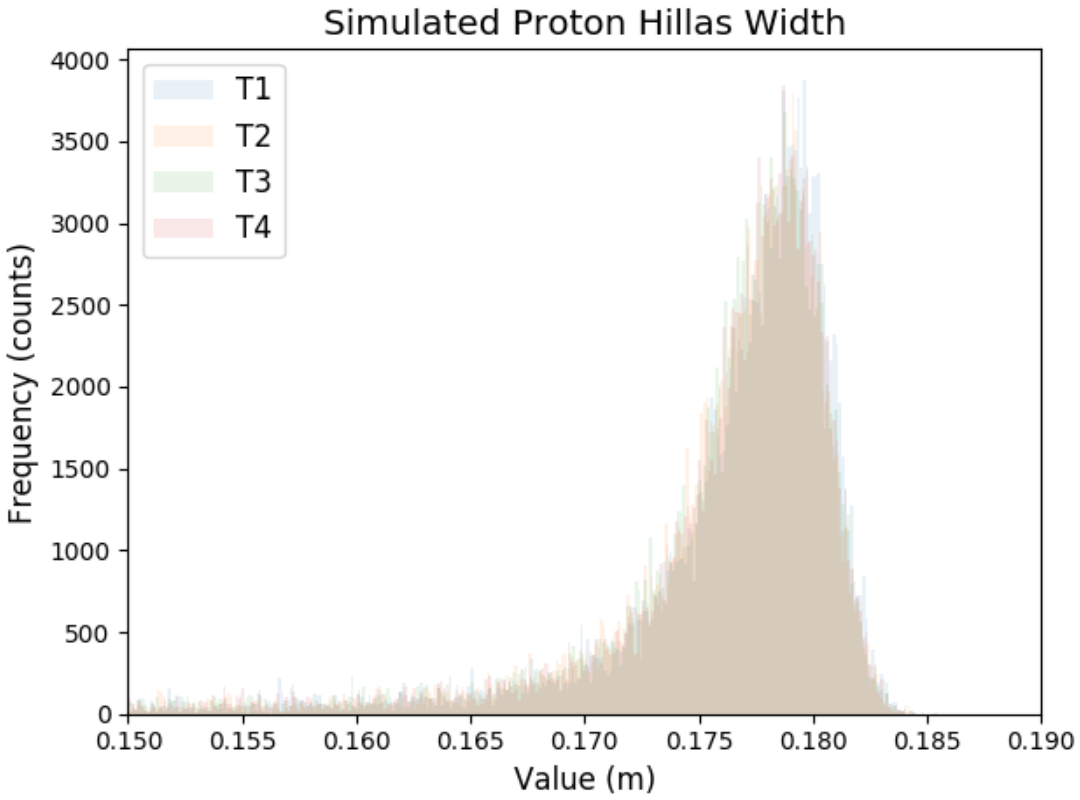


Figure 4.11: Hillas Width Distribution for the Simulated Protons.

the expected improvement EI (which we aim to maximize at each step) is given by

$$\mathrm{EI}_{y^*}(x^*) = \int_{-\infty}^{y^*} (y^* - y) p(y|x^*) dy \tag{4.3}$$

where y^* is a threshold value of the objective function (below which $p(y|x^*)$ is zero). TPE models then construct surrogate distributions $p(x^*|y)$, $p(x^*)$ and $p(y)$ for the objective function indirectly using Bayes' rule

$$p(y|x^*) = \frac{p(x^*|y) \times p(y)}{p(x^*)} \tag{4.4}$$

where $p(x^*|y)$ is the probability of the hyperparameters selected given the score on the objective function [176]. In turn, $p(x^*|y)$ is modelled as two separate distributions by TPE methods

$$p(x^*|y) = \begin{cases} l(x^*) & \text{, if } y < y^* \\ g(x^*) & \text{, if } y \geq y^* \end{cases} \tag{4.5}$$

where y^* is the threshold value of the objective function, $l(x^*)$ is the density function formed by using the set of i observations $x^{*(i)}$ such that the corresponding loss

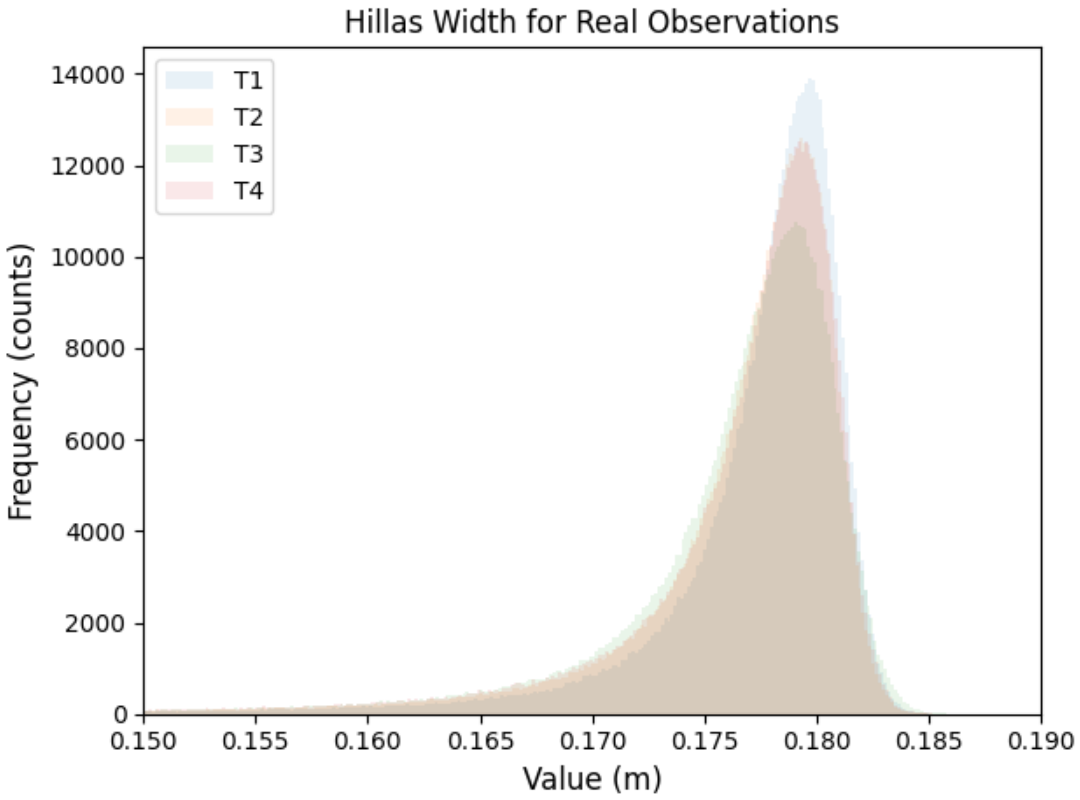


Figure 4.12: Hillas Width Distribution for the Real Events (Run 64080).

$f(x^{*(i)})$ was less that y^* and $g(x^*)$ is the density function formed by using the remaining observations. y^* is chosen by the algorithm to be one of the observed y values γ such that $p(y < y^*) = \gamma$ [175]. Given that $\gamma = p(y < y^*)$ and $p(x^*) = \int_{\mathbb{R}} p(x^*|y)\, p(y) dy = \gamma\, l(x^*) + (1 - \gamma\, g(x^*))$ by construction, and using Bayes' theorem and substitution, the EI can then be written as

$$\mathrm{EI}_{y^*}(x^*) = \frac{\gamma y^* l(x^*) - l(x^*) \int_{-\infty}^{y^*} p(y) dy}{\gamma l(x^*) + (1-\gamma) g(x^*)} \propto \left(\gamma + \frac{g(x^*)}{l(x^*)}(1-\gamma)\right)^{-1}. \tag{4.6}$$

The TPE approach then maximises the EI by drawing values from the $l(x^*)$ distribution, and maximizing the ratio $l(x^*)/g(x^*)$, which in turn maximises the EI [176]. In practice, the fact that Equation 4.4 is not a perfect representation of $p(y|x^*)$ (because of the way $p(x^*|y)$ is modelled) may mean that the hyperparameter choice offers no improvement in the objective. Therefore $l(x^*)$ and $g(x^*)$ need to be updated per iteration of optimisation; the use of a distribution rather than a single value for $l(x^*)$ helps to balance the trade-off between exploration and exploitation, as selecting

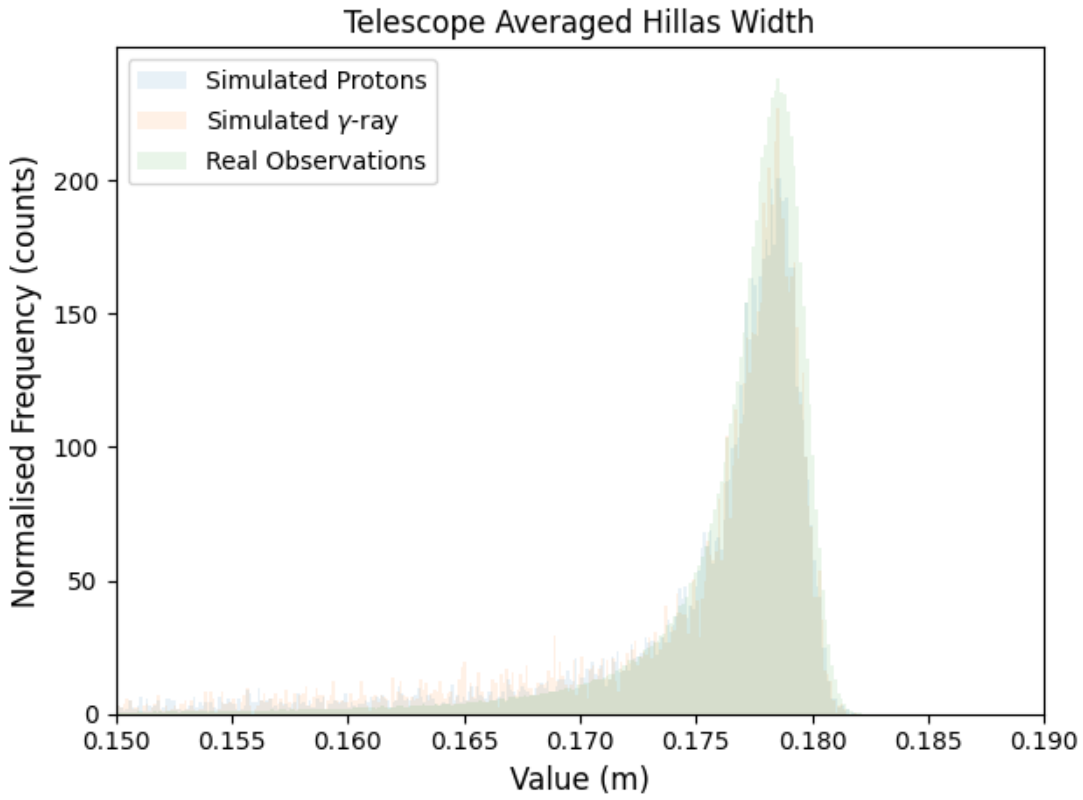


Figure 4.13: For ease of comparison, the telescope-averaged Hillas width distribution for the simulated proton, γ-ray and real events from Run 64080, normalised such that the distributions integrate to unity. The real observations appears to be marginally wider than the simulations, this is likely a limit of the hadronic interaction modelling in CORSIKA.

hyperparameters in a region too far from the explored hyperparameter space is effectively penalised. But in practical terms, TPE optimisation can reasonably be thought of as performing a modified version of gradient descent over hyperparameter values, with the loss function as the quantity being optimised.

4.7.5 Initial Optimisation Efforts

To investigate the potential of Bayesian optimisation, we utilized the *Hyperas* [177] *Keras* wrapper around *Hyperopt* [109], one of the most commonly-used Python packages for performing TPE analysis. However, given that this approach requires (at least partially) training a classifier for every point in hyperparameter space explored, this process is orders of magnitude more computationally costly than simply training a single deep learning classifier with a single hyperparameter

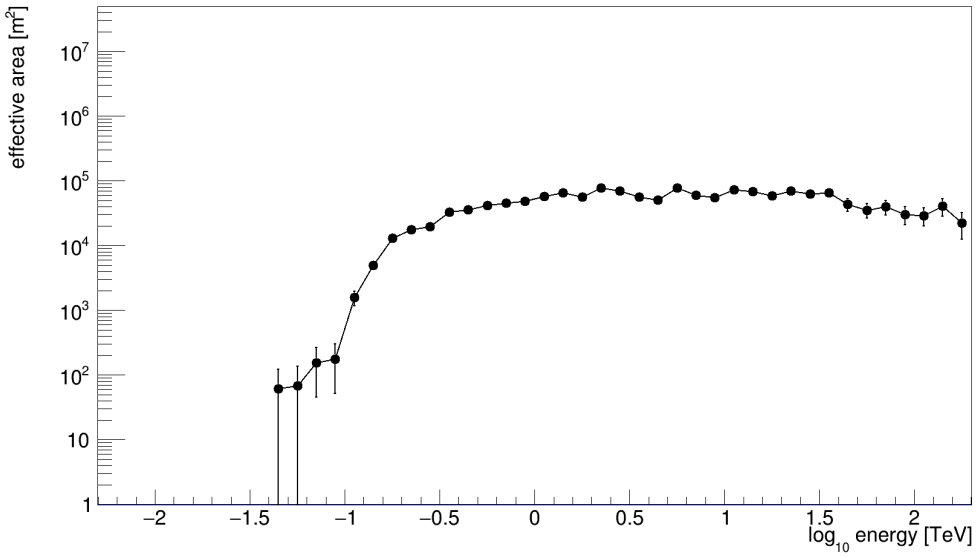


Figure 4.14: Effective area of VERITAS as a function of energy, made using our analysis pipeline and the opt4 configuration. This shows performance broadly similar to BDTs on simulations, see for comparison [178]. However, it does not show a substantial improvement. This could possibly be an effect of the custom simulation approach we use, or as an effect of non-uniform posterior distributions on the Γ scores from the ConvLSTM. The parameters for this are as follows: Zenith angle 10°, wobble offset=0.5°, γ-ray spectral index=-1.5, noise level=250Hz.

configuration. It was also not immediately clear what the best approach to take with these methods given realistically finite compute power and time. There is a trade-off between the number of epochs to train a model for each point in hyperparameter space, the amount of data used for training and the number and range of hyperparameter space points probed. The optimal configuration for this was not known at the starting point of this research. But in practice, VRAM constraints limit the maximum possible range of hyperparameter values probed. Extending this range too far, for example in terms of number of layers, will eventually exhaust finite VRAM and cause crashes. This attempt was the first time TPE optimisation has been attempted with a stereoscopic IACT array [166].

The initial optimisation efforts revolved around training a ConvLSTM2D classifier for a single epoch on a comparatively small number of events (∼19574 simulated events) on a single GPU, using performance on a similarly small amount of validation data as a metric. The hyperparameter space for this optimisation is shown in Table

4.3, where we use hyperopt convention to define the hyperparameter space. This means Choice() represents optimising a discrete choice between a finite number of options (this can include binary selection such as the inclusion of layers, as well as selection between two dimensional objects like filter sizes) and Uniform() means the value can take any float value in a given range. The aim was to find the best 'starting position' for the training. The network configuration generated following the initial hyperopt run was then trained for 100 epochs with the complete training dataset (235174 events in total). This resulted in the 'opt4' configuration presented here, yielding performance superior to random number generation on the real observations. This detection represents the first time the Crab nebula has been detected with an IACT array using a stereoscopic deep-learning-based event classification method, and was the first deep-learning-based detection of an astrophysical γ-ray source not to rely upon image cleaning. These results also represent the first time that Bayesian optimisation has been applied to stereoscopic IACT data.

Block	Layer	Parameter	Potential Values	Optimal Value Diffuse Source
Block 1	ConvLSTM2D	Number of Filters	Choice(10,20,30,40)	10
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5))	(4,4)
		Kernel L2 Regularizer Rate	Uniform(0,1)	0.0797
		Dropout Rate	Uniform(0,1)	0.0361
		Recurrent Dropout Rate	Uniform(0,1)	0.382
Block 2	ConvLSTM2D	Number of Filters	Choice(10,20,30,40)	10
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5))	(2,2)
		Kernel L2 Regularizer Rate	Uniform(0,1)	0.576
		Dropout Rate	Uniform(0,1)	0.371
		Recurrent Dropout Rate	Uniform(0,1)	0.510
Block 3	ConvLSTM2D	Number of Filters	Choice(10,20,30,40)	40
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5)	(2,2)
		Dropout Rate	Uniform(0,1)	0.0
		Recurrent Dropout Rate	Uniform(0,1)	0.1
Block 4	.	Include Block	Choice(Yes,No)	Yes
	ConvLSTM2D	Number of Filters	Choice(10,20,30,40)	30
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5))	(5,5)
		Dropout Rate	Uniform(0,1)	0.5056
Block 5	GlobalAveragePooling3D	-	-	-
Block 7	Dense	Number of Units	Choice(10,50,100,200)	100
Block 8	Dense	-	-	-

Table 4.3: Summary of hyperparameter space and selected configuration for the initial single core optimisation that produced the opt4 configuration.

Run	$\mathbf{N}_{on}$ (events)	$\mathbf{N}_{off}$ (events)	**Significance** (σ)	**Signal Rate** (γ/min)	**Error on Signal Rate** (γ/min)	**Background Rate** (events/min)
64080	573	280.17	13.9	14.617	1.243	13.985

Table 4.4: Anasum output for the opt4 run, without applying a strenuous multiplicity cut.

This significance of this detection is a mild function of multiplicity, the significance can reach 16.1 σ if a telescope multiplicity cut of 4 is imposed, this is a result of the signal-to-noise ratio being higher for such events. The standard VERITAS analysis, reliant upon BDTs, achieves a 22 σ detection on the same data without this strong multiplicity cut, although it is trained with a different dataset so the comparison is imperfect[3]. Curiously the validation data training

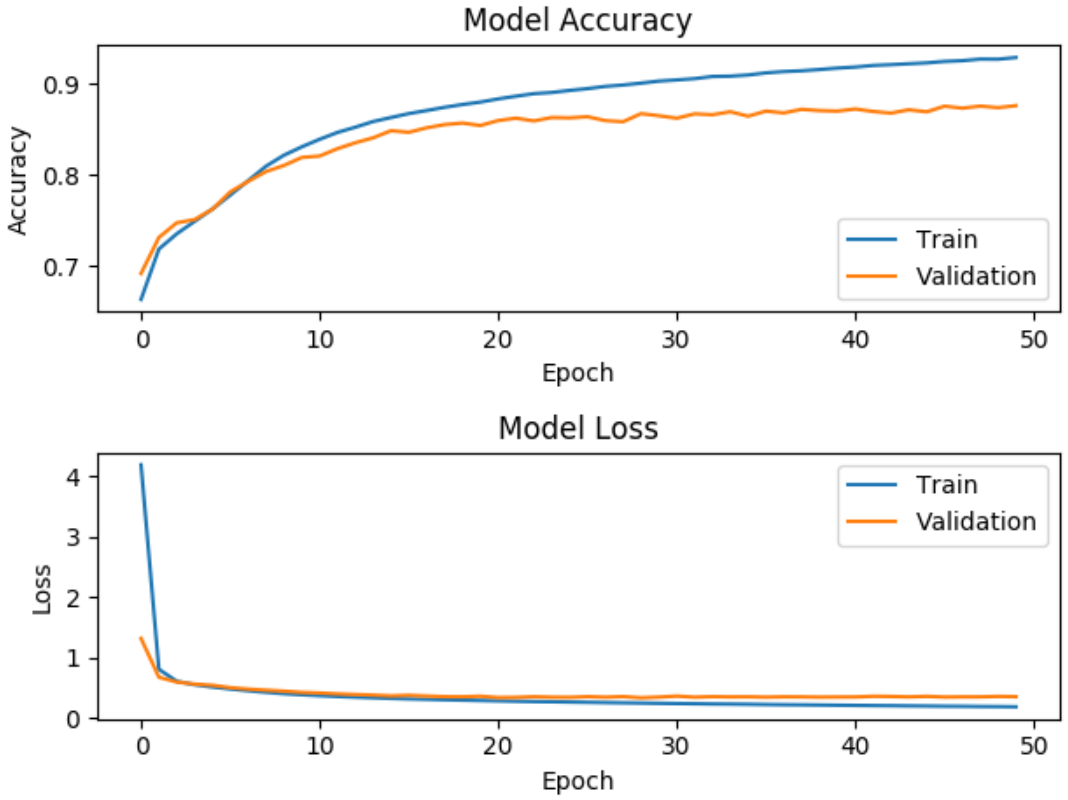


Figure 4.15: The training curve for the opt4 run.

curve appears to converge in this one specific instance, but it is unclear if this relates to the ConvLSTM2D classifier learning meaningful features in the simulated images or if this is simply a coincidence.

[3]Details of this analysis can be found at `https://veritas.sao.arizona.edu/wiki/index.php/Eventdisplay_Manual:_event_display`; but note that this is a VERITAS internal site only accessible from VERITAS member institutes.

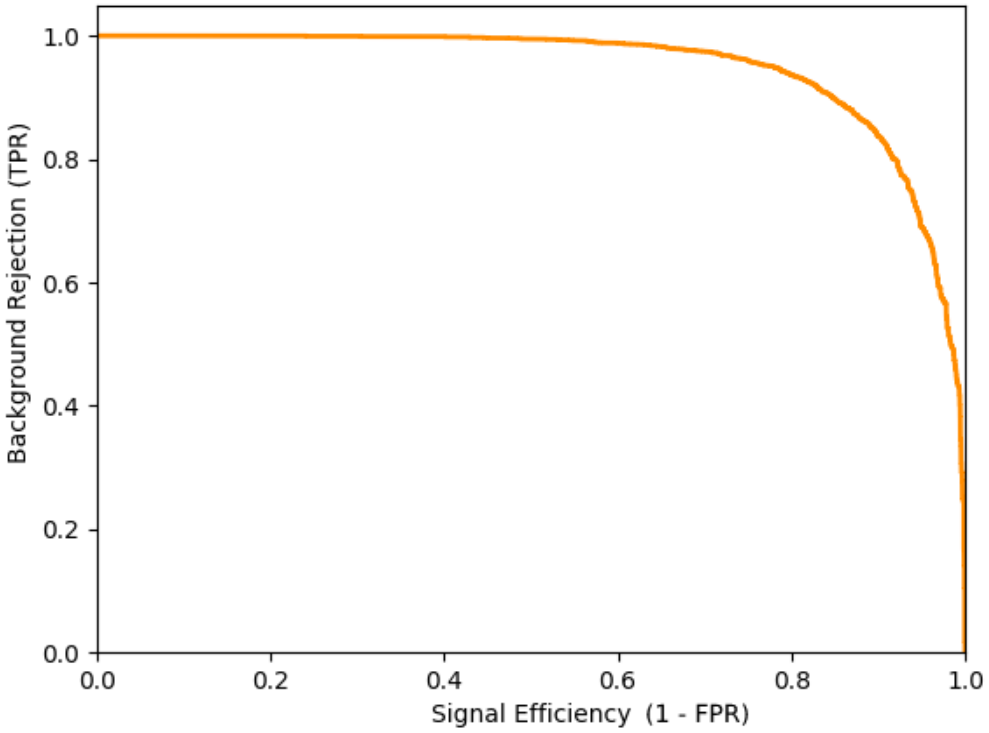


Figure 4.16: The test signal efficiency curve for the opt4 run.

4.8 Parallelised Optimisation Attempts

4.8.1 Optimisation Configurations and Results

Following our initial success with Bayesian optimisation, we ran extensive further investigations with improved computational power in order to test the performance limit of this new technique. The next three figures and three tables show three attempts to perform Bayesian optimisation using *hyperas* in parallel using the recently upgraded *glamdring* cluster. Four of the machines had two NVidia 2080Tis attached, the final had only one. The machines with multiple GPUs made use of Tensorflow's *MirroredStrategy* approach, so the optimisation worker effectively saw a single GPU with double the VRAM of a 2080Ti. We'll refer to these three optimisation attempts as attempt A, attempt B and attempt C. After each optimisation attempt the best performing model was trained with the complete dataset for more epochs, similar to the earlier strategy on the single GPU. In

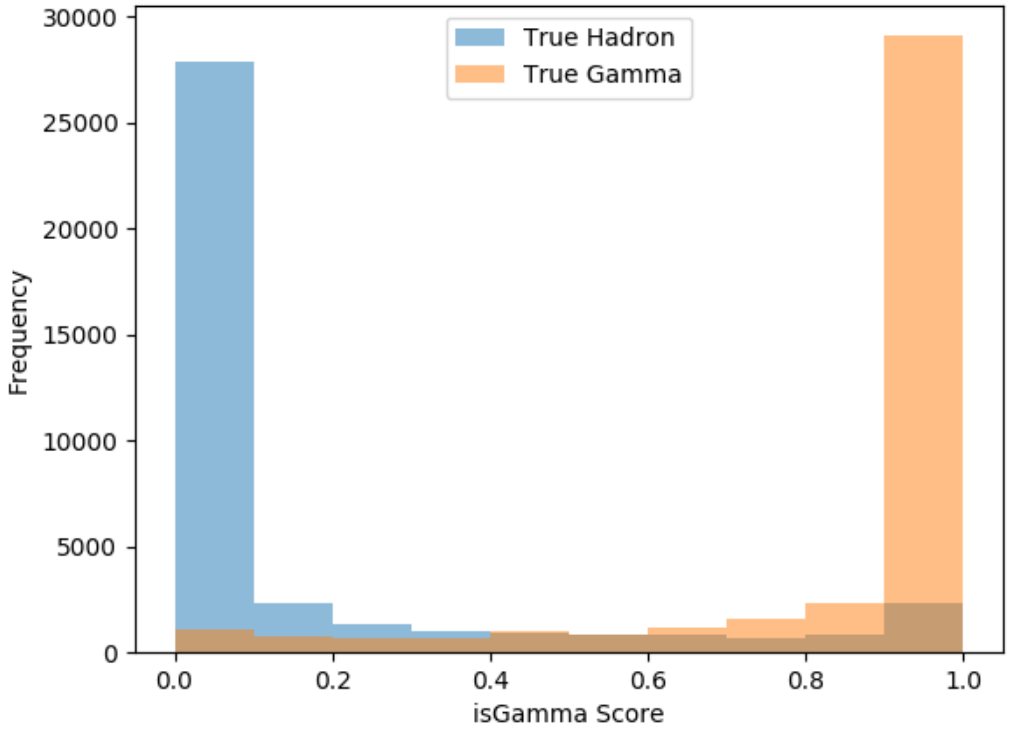


Figure 4.17: The test gammaness histogram for the opt4 run.

all attempts, a unique configuration database was created, with each point in hyperparameter space being allocated a unique working directory. To operate, the optimiser consisted of a head process working on the *glamdring* login node to determine hyperparameter point selection, and up to 5 worker processes on the *glamdring* GPU nodes. These worker processes logged their progress for each hyperparameter space point investigated in a MongoDB database [179], that was accessible to both the worker and head processes. Progress was checkpointed such that there were no information losses if a worker process failed. The classifier trained on a certain subset of training data for a certain number of epochs for each hyperparameter configuration, and the accuracy on test data after this training was used as the metric. Each of these test scores is shown as a blue cross in Figures 4.20, 4.21 and 4.22, running averages and moving averages (over the maximum 4 parallel systems, each with 2 GPUs) are also shown. As individual iterations of the optimisation have some degree of randomness in order to explore the space, the only

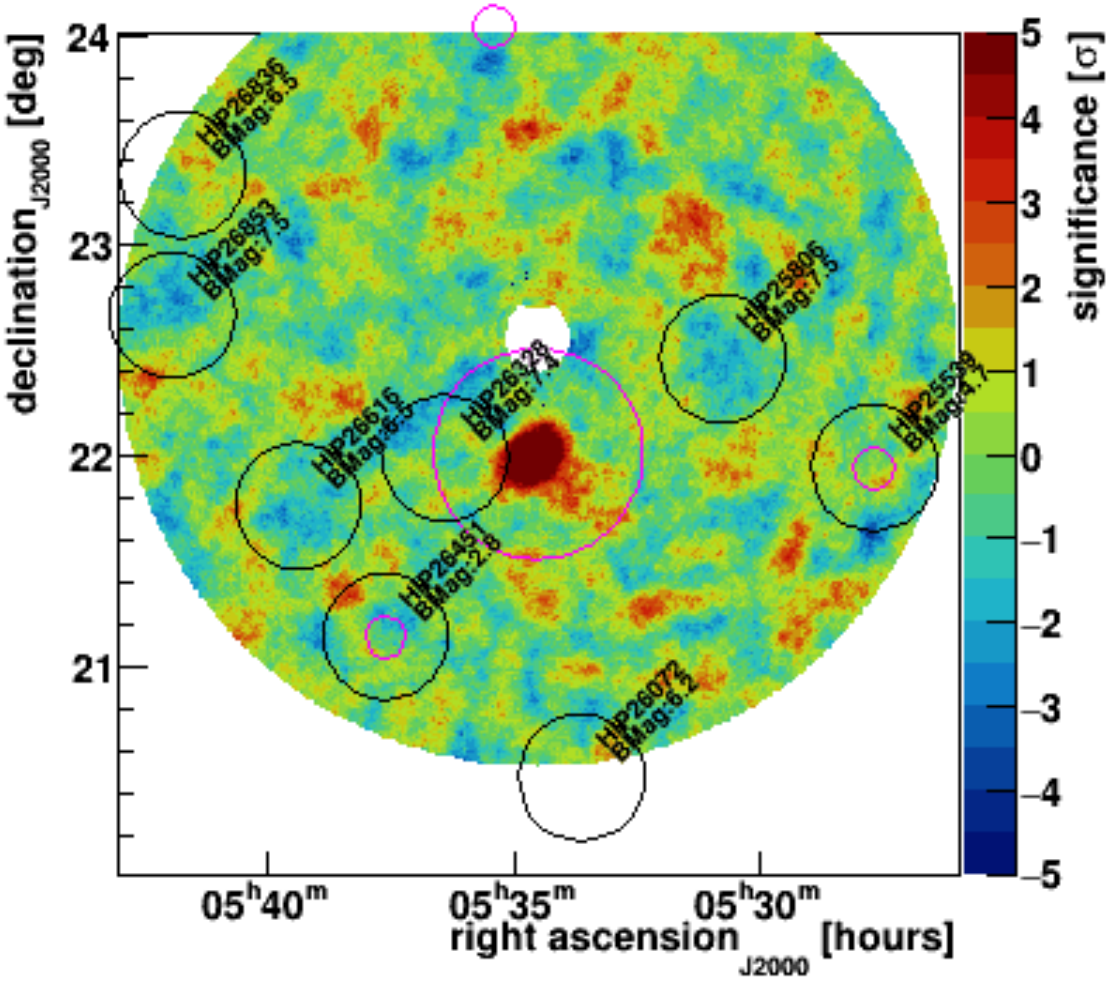


Figure 4.18: The sky significance for the 'opt4' configuration on run 64080.

evidence of the process converging is contained within in these averages. Attempt A had a similar hyperparameter space to the initial single core optimisation run that produced the opt4 configuration, albeit using a greater number of training epochs per hyperparameter point. This hyperparameter space was increased for the subsequent attempts B and C. It should be noted that in particular optimisation attempt C, our final attempt to generate a configuration superior to the opt4 result, was extremely computationally intensive (representing the results of around 5800 GPU hours of compute time).

Final simulated data training curves and signal efficiency curves for the best model configuration for optimisation attempt C, trained with the complete dataset

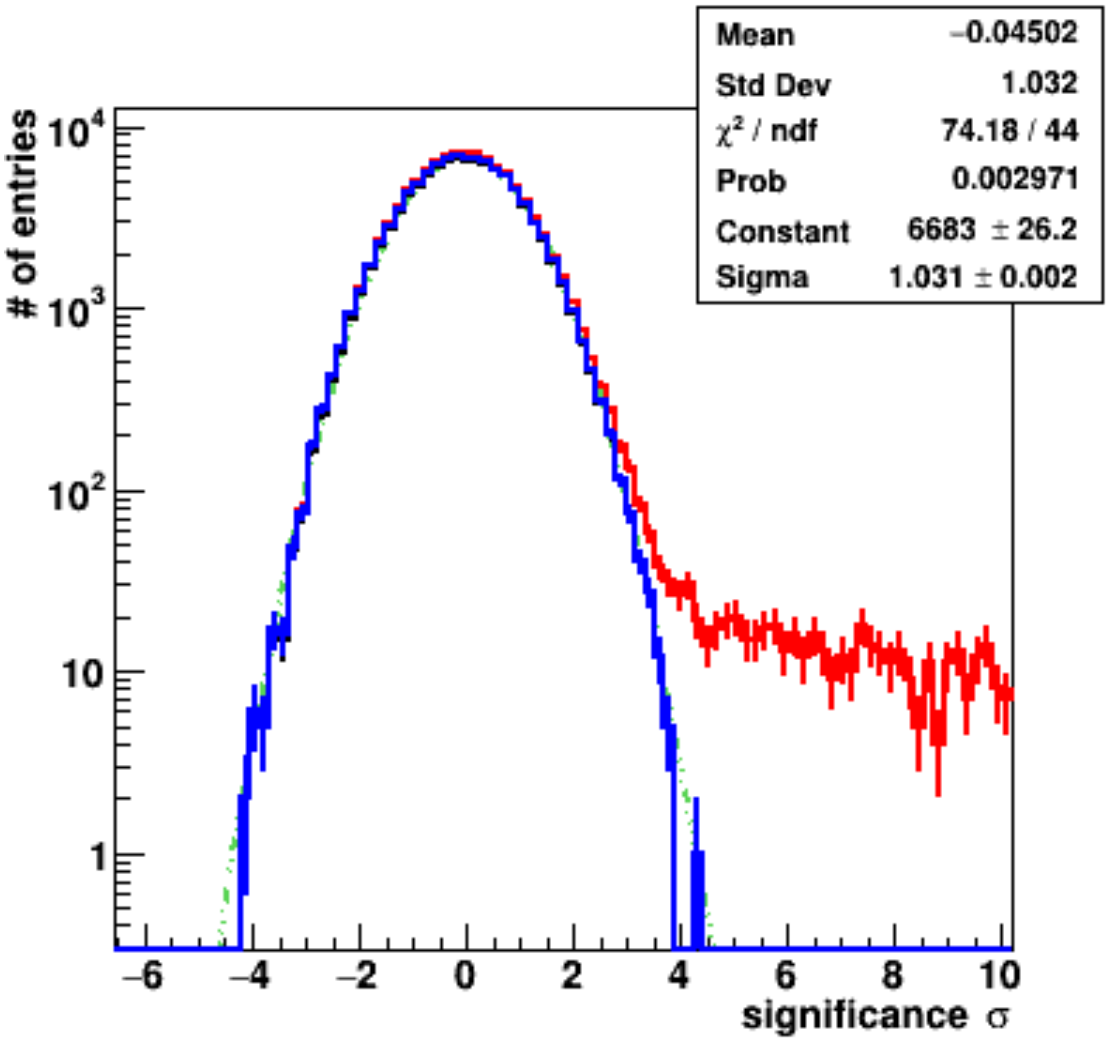


Figure 4.19: The 1D significance for the 'opt4' configuration on run 64080.

are shown in Figures 4.23 and Figure 4.24 for completeness. Whilst none of these attempts generated a model superior to the opt4 configuration, this final optimisation run C is of interest as the running average shows clear evidence of convergence (though there is some evidence of this for methods A and B too). However, there is still significant spread in the distribution of network validation results at late times for Method C; this is a result of the way TPE methods work (the evidence of convergence is seen in the running average). This optimisation convergence is not often seen with TPE methods, and this is an established issue with them [180]. In particular, these results are the first runs of Bayesian optimisation with IACT data to ever see evidence of convergence. This convergence is likely a result of

Attempt	A	B	C
No. Training Events	39148	19574	117574
No. Test Events	39200	19600	39200
Epochs Training per Hyperparameter Point	5	1	25
No. Hyperparameter Points Sampled	2000	1000	500
Best Test Accuracy (%)	77.36	75.9	86.4

Table 4.5: Summary of parallelised optimisation runs and the amount of data used for each. Values presented here represent the number of training events, test events, and test performance for the training and evaluation during the Bayesian optimisation process. The best performing models from each optimisation run were then trained and tested again with the complete dataset for more epochs before being applied to the real observations.

Block	Layer	Parameter	Potential Values	Optimal Value Attempt A	Optimal Value Attempt B
Block 1	ConvLSTM2D	Number of Filters	Choice(10,20,30,40)	20	10
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5))	(4,4)	(4,4)
		Kernel L2 Regularizer Rate	Uniform(0,1)	0.402	0.404
		Dropout Rate	Uniform(0,1)	0.462	0.536
		Recurrent Dropout Rate	Uniform(0,1)	0.552	0.353
Block 2	ConvLSTM2D	Number of Filters	Choice(10,20,30,40)	10	20
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5))	(2,2)	(4,4)
		Kernel L2 Regularizer Rate	Uniform(0,1)	0.234	0.796
		Dropout Rate	Uniform(0,1)	0.774	0.563
		Recurrent Dropout Rate	Uniform(0,1)	0.000	0.116
Block 3	ConvLSTM2D	Number of Filters	Choice(10,20,30,40)	30	30
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5)	(2,2)	(4,4)
		Dropout Rate	Uniform(0,1)	0.444	0.105
Block 4	.	Include Block	Choice(Yes,No)	Yes	Yes
	ConvLSTM2D	Number of Filters	Choice(10,20,30,40)	40	30
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5))	(5,5)	(5,5)
		Dropout Rate	Uniform(0,1)	0.720	0.652
Block 5	GlobalAveragePooling3D	-	-	-	-
Block 7	Dense	Number of Units	Choice(10,50,100,200)	10	100
Block 8	Dense	-	-	-	

Table 4.6: Summary of hyperparameter space and selected configuration for the initial parallelised optimisation runs, attempts A and B.

the large number of epochs, the large amount of data used, and the long period the optimisers were left to run. This combined with the fact we did not achieve performance superior to the opt4 run, suggests that the opt4 configuration represents the current limit of what is technically possible with deep learning event classifiers, with this configuration of simulations and cuts. Given that we observed improved performance on-sky when the classifier achieved optimal performance on simulated training data, if there is a point at which the deep learning classifier becomes 'over-familiar' with the simulated data to the extent that performance with real observations is reduced, we have yet to reach it.

Block	Layer	Parameter	Potential Values	Optimal Value Attempt C
Block 1	ConvLSTM2D	Number of Filters	Choice(10,20,30,40,50,60)	10
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5),(6,6),(7,7))	(4,4)
		Kernel L2 Regularizer Rate	Uniform(0,1)	0.366
		Dropout Rate	Uniform(0,1)	0.414
		Recurrent Dropout Rate	Uniform(0,1)	0.107
Block 2	ConvLSTM2D	Number of Filters	Choice(10,20,30,40,50,60)	10
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5),(6,6),(7,7))	(2,2)
		Kernel L2 Regularizer Rate	Uniform(0,1)	0.167
		Dropout Rate	Uniform(0,1)	0.863
		Recurrent Dropout Rate	Uniform(0,1)	0.321
Block 3	ConvLSTM2D	Number of Filters	Choice(10,20,30,40,50,60)	40
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5),(6,6),(7,7))	(2,2)
		Kernel L2 Regularizer Rate	Uniform(0,1)	0.219
		Dropout Rate	Uniform(0,1)	0.287
		Recurrent Dropout Rate	Uniform(0,1)	0.697
Block 4	ConvLSTM2D	Number of Filters	Choice(10,20,30,40,50,60)	30
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5),(6,6),(7,7))	(3,3)
		Dropout Rate	Uniform(0,1)	0.483
Block 5		Include Block	Choice(Yes, No)	Yes
	ConvLSTM2D	Number of Filters	Choice(10,20,30,40,50,60)	30
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5),(6,6),(7,7))	(3,3)
		Dropout Rate	Uniform(0,1)	0.020
Block 6		Include Block	Choice(Yes, No)	Yes
	ConvLSTM2D	Number of Filters	Choice(10,20,30,40,50,60)	30
		Kernel Size	Choice((2,2),(3, 3),(4,4),(5,5),(6,6),(7,7))	(3,3)
		Dropout Rate	Uniform(0,1)	0.553
Block 5	GlobalAveragePooling3D	-	-	-
Block 6		Include Block	Choice(Yes, No)	Yes
	Dense	Number of Units	Choice(10,50,100,200)	10
		Dropout Rate	Uniform(0,1)	0.190
Block 7		Include Block	Choice(Yes, No)	Yes
	Dense	Number of Units	Choice(10,50,100,200)	200
		Dropout Rate	Uniform(0,1)	0.743
Block 8		Include Block	Choice(Yes, No)	Yes
	Dense	Number of Units	Choice(10,50,100,200)	200
		Dropout Rate	Uniform(0,1)	0.107
Block 9	Dense	-	-	

Table 4.7: Summary of hyperparameter space and selected configuration for the final parallelised optimisation run, attempt C.

4.9 Conclusion

In this chapter, we have demonstrated a deep learning pipeline capable of use with the VERITAS telescopes, and showed that a combination of ConvLSTM2D classifiers, custom simulations and Bayesian optimisation can yield a definite detection of the Crab Nebula. This also is the first detection with an IACT array using such deep learning methods that did not require conventional tailcut cleaning. However, there are significant issues with this approach that limit its current efficacy for CTA that require further detailed study. In particular, it appears that the modelling of NSB used by current generation IACTs is possibly too simplistic for use with deep learning event classification methods for CTA. It is also clear that there is a very significant hyperparameter dependence problem with ConvLSTM2D and similar classifiers which limits their use in practice. It should also be noted that the combination of optimisation and custom simulations we used is likely to be

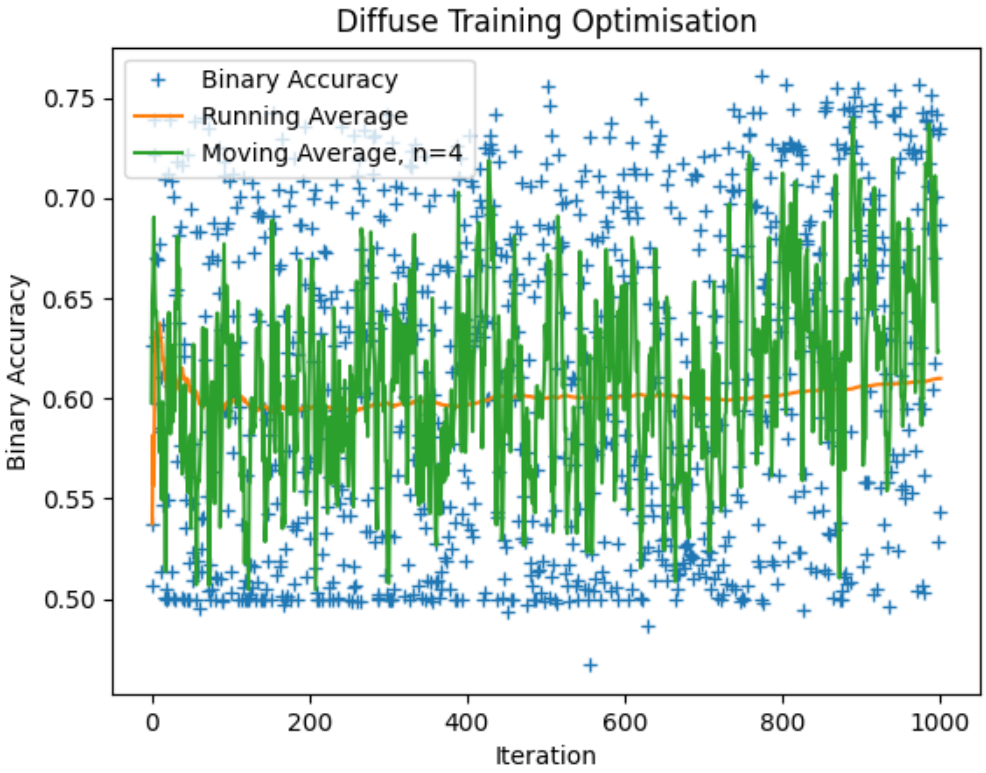


Figure 4.20: Final test accuracies after training optimisation for longer Bayesian optimisation run attempt A.

computationally prohibitive in a real observing scenario. Our achieved aim was to determine the maximal possible potential for these deep learning methods, and we performed this work operating under the hope that at some point new transfer learning, optimisation and simulation approaches might be able to bridge the gap in computing power needed. Given these results, it seems likely that in order to fully leverage the potential for deep-learning-based analyses we must either look to new methods of interpretation of IACT data, such as leveraging high precision timing information as presented in Chapter 3, or to new deep learning techniques beyond conventional CNNs. That said, it also appears that we have discovered a weak correlation between accuracy on simulated test data and performance on real observations. The pipeline we developed for these purposes could be used in future work for VERITAS should new mitigation strategies become apparent.

Ultimately, as none of the longer, parallelised optimisation runs produced a

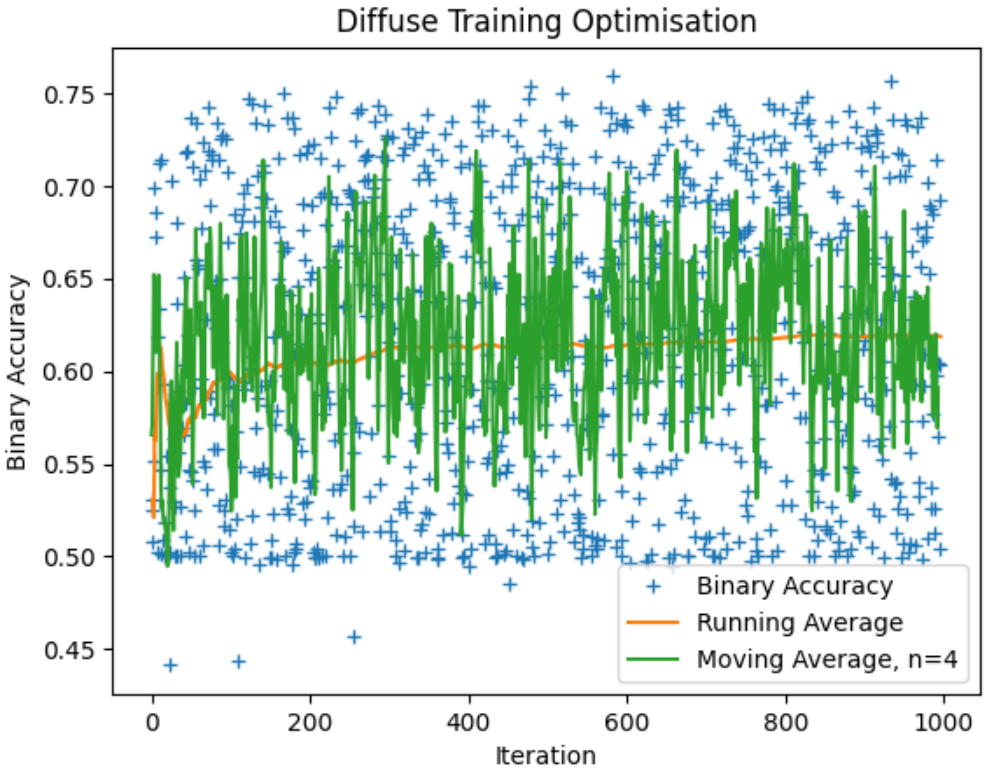


Figure 4.21: Final test accuracies after training optimisation for longer Bayesian optimisation run attempt B.

model that outperformed the opt4 configuration after final training and testing on the whole simulated dataset, combined with the convergence appearing in attempt C's moving average optimisation curve, suggests that we have hit the simulation accuracy limit (87%) for this selection of cuts and this simulation data. So whilst Bayesian optimisation is a powerful tool that can allow for real detections to be made in IACT data, it cannot be claimed to be consistently reliable.

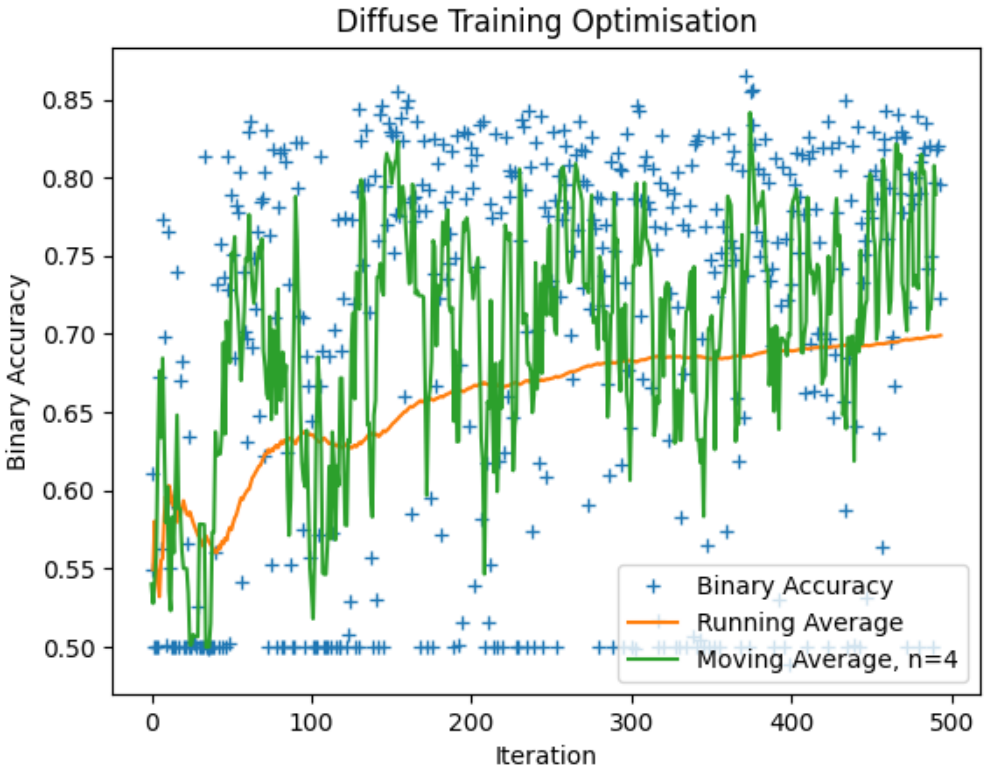


Figure 4.22: Final test accuracies after training optimisation for Bayesian optimisation attempt C.

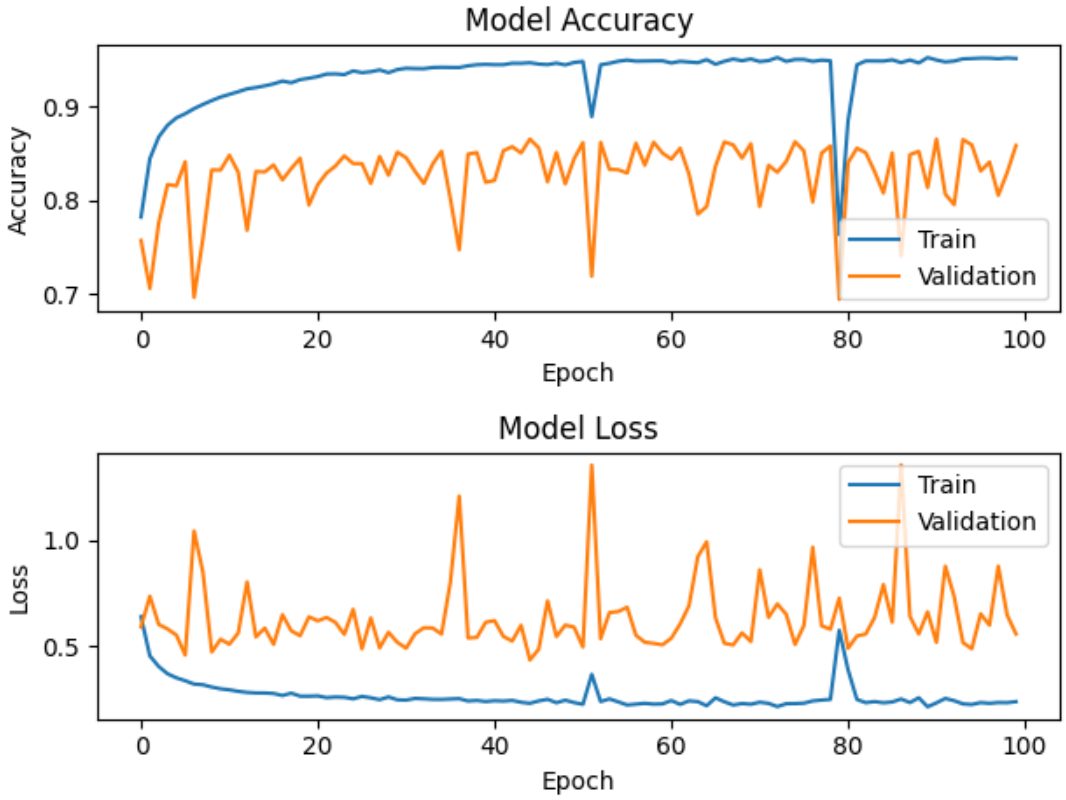


Figure 4.23: Training accuracy curve for the best performing model from optimisation attempt C, trained for 100 epochs with the complete dataset.

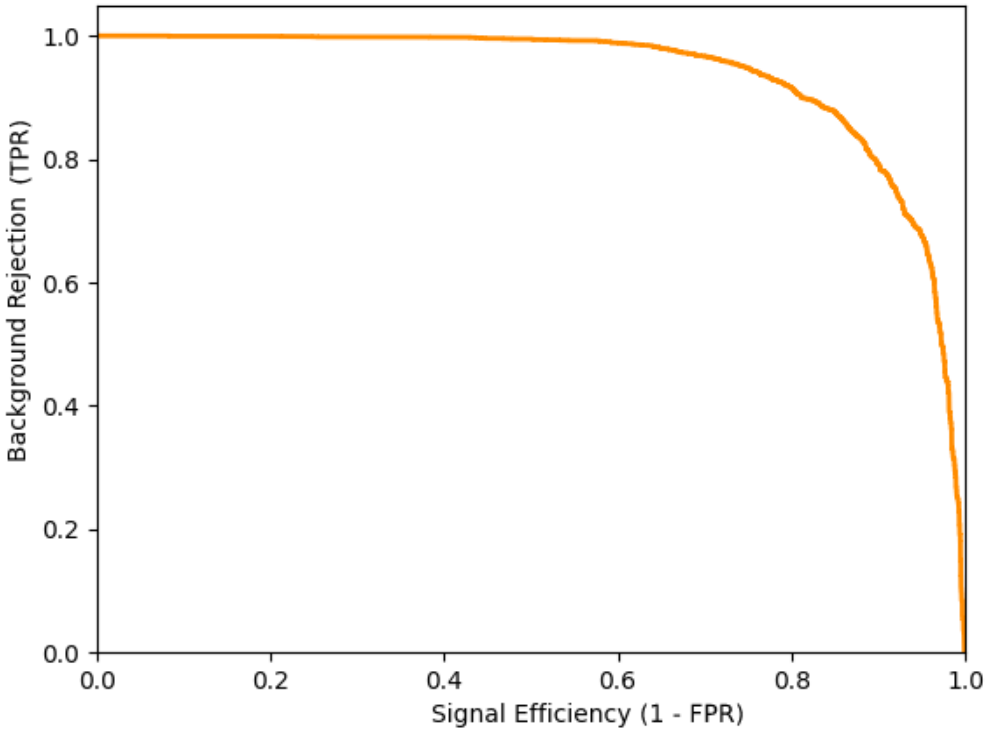


Figure 4.24: Test signal efficiency curve for the best performing model from optimisation attempt C, trained for 100 epochs with the complete dataset.

5

Advanced Study of SSTCAM NSB

Contents

Abstract

Night Sky Background (NSB) is a complex phenomenon, consisting of all light detected by imaging atmospheric Cherenkov telescopes not attributable to Cherenkov light emission. Understanding the effect of NSB on cameras for the next-generation Cherenkov Telescope Array (CTA) is important, as it affects the astrophysical systematic errors on observations, the ability of the telescopes to operate under partial moonlight conditions and the thermal control of the cameras. The capacity to observe under partial moonlight conditions is crucial for the CTA transient science programme, as it substantially increases the potential observing time. Using tools initially developed for H.E.S.S. (in combination with the prototype CTA analysis package ctapipe) we will present predictions for the NSB present in images taken by the Small Sized Telescope Camera (SSTCAM), showing that SSTCAM will likely be able to meet the associated CTA requirements. Additionally, we calculate the potential observing time gain by operating under high NSB conditions.

5.1 Introduction

5.1.1 Scope & Purpose of this Chapter

As we've seen in Chapters 3 and 4, it is possible that the current modelling of NSB in both the simulation packages *CARE* (used by VERITAS and in Chapter 4) and *sim_telarray* (used by CTA and H.E.S.S. and in Chapter 3) is insufficient to allow for reliable deep-learning-based event classification. In this chapter, we estimate the expected NSB photon rates of SSTCAM under various observing conditions. This modelling will aid SSTCAM design, as well as begin a possible route to improvement in our training data that might one day make deep learning analyses feasible. We perform this by using and expanding upon the capabilities of the *nsb* tool [181], already validated and used by H.E.S.S.. For the purposes of this chapter, capitalised NSB refers to the physical phenomenon of Night Sky Background, where as italic lower case *nsb* refers to the Python package used to simulate it. Full results from this study, along with sandbox scripts to operate *nsb* can be found on the `https://github.com/sstcam/sstcam_NSB` Github page.

5.1.2 Context

The ability for SSTCAM to operate reliably under partial moonlight and other high NSB observing conditions is critical to the transient and multi-messenger/multi-wavelength science goals of CTA. This is as operating under these conditions significantly increases the possible observing time, and so this ability has been designated as a CTA system requirement. The increase in potential science return was judged to outweigh the associated increase in data rate and person-power required. The most notable justification for this is the detection of prompt TeV emission from GRB 190114C (the first such detection from the ground) under partial moonlight conditions (6 times nominal NSB) with MAGIC [11].

Studies of partial moonlight and other high NSB observations can inform temperature management protocols and SiPM selection for SSTCAM. They can also inform estimations of astrophysical systematic errors, particularly on flux

normalization, and the energy threshold [182] (both seen in a paper by MAGIC [182]). This is a result of the sensitivity to light of an SiPM being a function of its illumination [183]. The majority of previous studies concerning NSB for SiPM-based Cherenkov cameras have primarily focused on NSB's effect on single SiPM pixels [183] [184] [185] [186]; the work presented in this chapter is the first detailed study of an entire SiPM-based camera's worth of NSB, allowing us to evaluate both the spatial and temporal properties of NSB across the camera plane.

Light illuminating a SiPM leads to a current flowing through it, so SiPMs are typically operated with a bias resistor connected in series to prevent the SiPMs from being damaged by drawing too much current from the power supply. Current flow through this bias resistor leads to a drop in voltage V_{Drop} across the bias stage, which in turn reduces the current through the SiPM. This bias resistor therefore functions as part of a negative feedback loop, affecting the overvoltage across the SiPM defined as $\Delta V = V_{PS} - V_{BD}$. Here, V_{BD} is the breakdown voltage; this is the minimum (reverse) bias voltage under which electron avalanche production is possible in a SiPM. V_{BD} is a function of the current flowing through the SiPM, and so V_{BD} is a function of V_{Drop}. Finally, V_{PS} is the voltage supplied by the power supply [183]. As ΔV affects most of the measurable properties of the SiPM, such as gain and Photo Detection Efficiency (PDE), it also affects the photomultiplier calibration. SSTCAM will counter this calibration shift with a LED flasher system, which is designed to calibrate the camera in near-real-time. Therefore the rate of change of NSB in the camera affects the number and rate of LED flashes required to achieve a desired near-real-time calibration accuracy. We consider the necessary rate of LED flashes to correct for the effects of NSB later in this chapter.

5.1.3 Model

The standard *CORSIKA/sim_telarray* simulation package [187], used for CTA Monte-Carlo simulation productions, supports NSB maps as an input parameter. It also has some limited functionality to replicate illumination at an infinite distance from the camera (i.e. stars). For reasons of computational efficiency, it does not

calculate where in the camera and how bright those stars might be, nor does it realistically simulate the effect of moonlight. So we choose to model NSB using the *nsb* package, which assumes that NSB comes from two primary sources. The first NSB source is starlight using *Gaia* data, and the second is sky brightness taking account of moonlight and a local extinction coefficient following the semi-analytic model of Krisciunas and Schaeffer [188]. *nsb* makes use of the standard Hierarchical Equal Area isoLatitude Pixelation of a sphere (healpix) pacakge to produce skymaps where each pixel covers the same surface area as every other pixel on the sky [189], using which the *hess_basic* model is expressed as

$$f(\rho, A, B, C) = 10^{A} \times (1.06 + \cos^2(\rho)) + 10^{(B-\rho/40)} + C \times 10^7 \times \rho^2, \tag{5.1}$$

$$I_M(\alpha_M) = 10^{-0.4\times(3.84+0.026\times|\alpha_M|+4\times10^{-9}\times\alpha_M^4))}, \tag{5.2}$$

$$X(Z) = \begin{cases} (1 - 0.96\sin^2 Z)^{-0.5} & \text{, if } Z \leq \pi/2 \\ (1 - 0.96 \times 1)^{-0.5} & \text{, otherwise} \end{cases}, \tag{5.3}$$

$$\begin{aligned} B_{Moon}(\rho, Z, Z_M, k, \alpha_M, A, B, C) = f(\rho, A, B, C) \times I_M(\alpha_M) \\ \times 10^{-0.4kX(Z_M)} \\ \times [1 - 10^{-0.4kX(Z)}] \end{aligned} \tag{5.4}$$

$$B_{Sky}(Z, \phi, B_0, k) = \begin{cases} B_0 \times X(Z) \times 10^{-0.4k(X(Z)-1)} & \text{, if } Z \leq \pi/2 \\ 0 & \text{, otherwise} \end{cases}, \tag{5.5}$$

$$B_{Total} = B_{Moon} + B_{Sky} + B_{Gaia} \tag{5.6}$$

where B_{Total} is the total brightness for a given pixel on the sky, B_{Moon} is the surface brightness from moonlight (compared to the same region of sky without the moon present), B_{Sky} is the intrinsic surface brightness of the sky, B_{Gaia} is the brightness of stars in the healpix pixel [189], Z is the Zenith angle, ϕ is the azimuth (with

subscript $_M$ referring to Moon parameters rather than source parameters), I_m is the illuminance of the Moon in footcandels, $X(Z)$ represents the optical pathlength along the line of sight in units of air masses, B_0 is a constant representing the brightness of the sky at the zenith and $f(\rho, A, B, C)$ is the scattering function as a function of lunar great circle separation angle ρ. In this form $f(\rho, A, B, C)$ takes into account both Rayleigh scattering from atmospheric gases and Mie scattering from atmospheric aerosols (larger solid particles suspended in the atmosphere). Rayleigh and Mie scattering occur at different altitudes and over different distributions of scattering particles in the atmosphere and so require individual (empirically determined) functional forms; A and B are the fitted coefficients to quantify the relative contributions of Rayleigh and Mie scattering respectively. A and B were introduced in the Krisciunas and Schaeffer [188] work, but the additional C parameter in Equation 5.1 was added by the creators of the *hess_basic* model to account for the relative brightness of the sky to the stars from *Gaia*. The factors of $-0.4k$ that appear in the exponents of Equations 5.2 to 5.5 are a result of a simplification of an integral over a volume extinction coefficient, where k is the measured extinction coefficient along the line of sight (see [188] for further details). The remainder of the exponent in Equation 5.2 is an empirical fit [188]; all the fitted values used are presented in Table 5.1. B_{Total} (along with all the other native outputs from the *hess_basic nsb* model) are expressed here in nanoLamberts (nLb), where a brightness in nanoLamberts B relates to magnitudes per square arc second in the V band (V) through

$$\frac{B}{\text{nanoLamberts}} = 34.08 \exp(20.7233 - 0.92104 \times V) \tag{5.7}$$

[188].

The stellar data in *nsb* is obtained using *dr1* stellar magnitude data from the *Gaia* space mission [190] [191]. Understanding the behaviour of the SiPMs under the illumination of a particularly bright star is important for the future operation plan of SSTCAM. Current practice of most Cherenkov cameras is turning the High Voltage (HV) power supply of blocks of pixels containing such bright stars off. SSTCAM

will not always have to do this as saturating SiPMs is more tolerable compared to conventional photomultipliers. If we choose to alter bias voltage for such pixels, it may be adjusted pre-emptively or as a reaction; this is further complicated by the need to use such stars for the slow-signal pointing calibration simultaneously.

Given lack of data from an SSTCAM prototype on the Paranal site it is currently not possible to fit the relative coefficients of the stellar and semi-analytic components of the model. It should also be noted that the *nsb* package does not explicitly take into account certain other known sources of NSB from both the telescope structure, the environment and astronomical sources. In particular we neglect contributions from:

- Zodiacal light, along with galactic and extragalactic background, though some of this sky brightness might be accounted for in the fitted parameters of the Krisciunas et al. model.

- Light from population centres (particularly from mines near ∼25 km from the Cerro Paranal site [192]). This might affect calibration measurements taken at the horizon, but it is practically impossible to simulate this at the Paranal site. Only exploration of all-sky camera data from the Paranal site would allow you to take account of this.

- Stray reflections from the ground, as well a moonlight reflecting from the secondary mirror and associated support structure. Some of these are unique problems with dual mirror designs, but historical investigations related to the optimisation of the colour of telescope structures for Cherenkov light observations suggest these effects will be small [193]. Internal reflections from SSTCAM's front window might also cause 'ghost' star images to appear, though this window will likely be coated to help reduce NSB.

- Light from satellites. The *Starlink* satellite constellation and others like it is of particular concern, with the satellites reaching $m_G = 2.4$ during the deployment phase, though this decreases to $m_G = 6.6$ afterwards [194].

Although the ~ 100 ns integration time in the fast chain might be too small for them to have an effect, they could still cause issues in the slow-signal chain [195]). This might affect the pointing accuracy if mistaken by source extractors for bright stars[1].

- Air glow emission lines in the atmosphere (see Figure 5.1 [54]). A rough calculation using the spectrum from [54] shows that air glow emission lines contribute to a constant NSB value of around ~ 17 mag/pixel in the SSTCAM wavelength range at the zenith, rising to 12.7 mag/pixel at the horizon (utilising the standard [54] measurements from La Palma). This is primarily from the OI 5577Å and Na D 5890Å lines, although at least some of this sky brightness will be incorporated into the Krisciunas et al. model.

- The potential effect of the eight magnitude 6.8 laser guide stars used by the European Extremely Large Telescope (E-ELT). This issue could be potentially mitigated through co-ordinated scheduling [196], unless CTA-South and the E-ELT are both searching the same region of sky for a ToO observation (such as those following a gravitational wave event).

- Lightning, which could cause rapid changes in camera illumination, however the CTA weather monitoring system might be able to prevent this from having too large an effect on camera operation.

All of these effects should, however, be much smaller than the differences between astronomical dark time and full moonlight.

5.2 Methods

5.2.1 Recommended Parameter Values

Figure 5.2 shows our workflow for generating NSB maps. Many free *nsb* parameters can be set automatically using automatic config generator functions we developed,

[1] It is potentially possible to simulate this, see `https://www.howmanystarlinkswillfillyoursky.com/`.

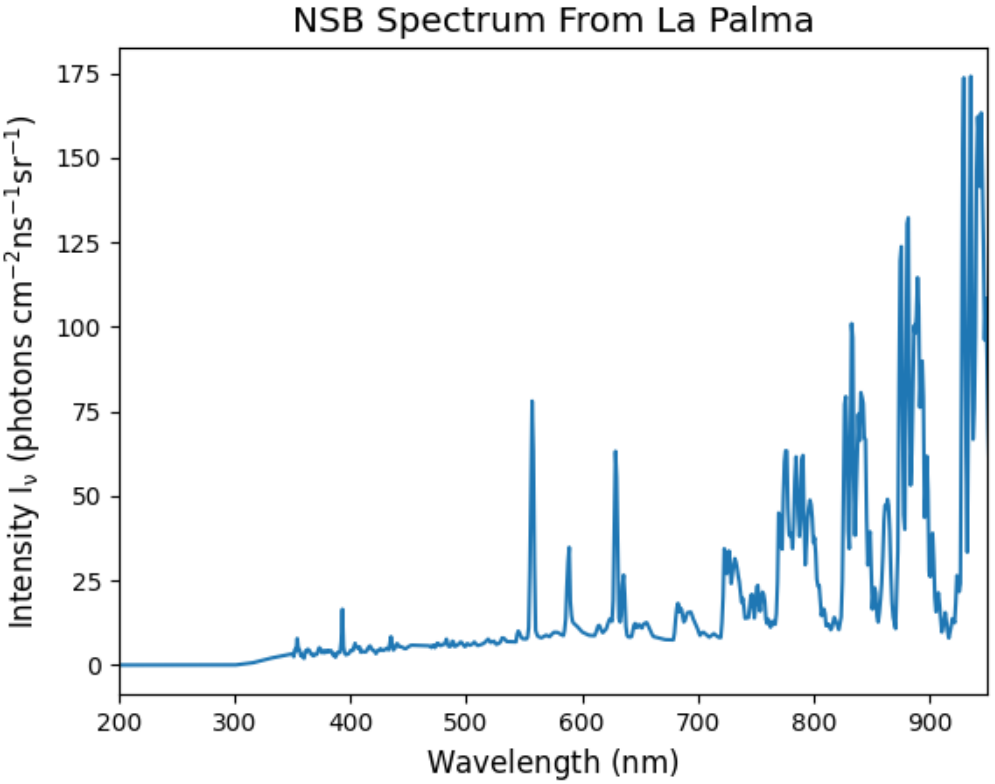


Figure 5.1: Spectrum of NSB over La Palma on a moonless night, data taken from [54] and used in the *sim_telarray* NSB rate calculations (but not *nsb*). Note the prominent airglow emission lines at longer wavelengths. Bar the two major emission lines in the V band ($\sim 500 - 700\,\mathrm{nm}$), this shows that our approximation of a uniform NSB spectrum in the sensitivity region of SSTCAM is reasonable.

but some parameters still need to be set manually. Part of the work for this project was determining what to set these free parameters to for simulations of SSTCAM. After investigating, the *nsb* run parameters we recommend for SSTCAM are in Table 5.1. The form of the Nyquist-Shannon sampling theorem relevant for image analysis in astronomy (to quote from [197]) is

> Aliasing error is something like a beat phenomenon between the sampling frequency and higher frequencies in the information spectrum, and as such is a form of systematic error at the grid points themselves of the digitized image ... the light signal must be sampled at least twice as frequently, in traversal of the continuous image, as the cresting in the finest sinusoid in its Fourier decomposition, if this aliasing error is to be avoided.

As a result, in order to properly sample the *nsb* maps for our pixel extraction, we

Parameter	Value for SSTCAM	Unit	Comments
Observation Latitude	-70.317876	°	Location of Paranal Prod5 SST-1
	(14.974609)	(°)	(Alternative Value for ASTRI site)
Observation Longitude	-24.681546	°	Location of Paranal Prod5 SST-1
	(37.693267)	(°)	(Alternative Value for ASTRI site)
Observation Altitude	2161.25	m	Location of Paranal Prod5 SST-1
	(1750)	(m)	(Alternative Value for ASTRI site)
No. Pixels for All Sky Maps	2000	Pixels	
Healpix Level	11	-	Healpix Cell Needs to be <1 Pixel on Sky
Gaia Catalog Mag Limit	15	mag	
Moon Above Horizon	0	° Altitude	Moonlight Observations Limit
Sun Below Horizon	-18	° Altitude	Daylight Threshold
Source Above Horizon	10	° Altitude	Source Observing Threshold
FoV	10	° Diameter	Affects WCS Projection
No. Pixels for FoV Map	10000	Pixels	Needs to be <1 Pixel on Sky
Sky Pixel Radius	0.19	°	Pixel Size on Sky
Degree of Gaussian Blurring	3	Pixels	
Pixel Shape	Rectangular	-	Pixel Shape on Sky
A	5.1276	-	*hess_basic* Model Parameter
B	5.9596	-	*hess_basic* Model Parameter
C	0	-	*hess_basic* Model Parameter
B_0	52	-	*hess_basic* Model Parameter
k	0.479	-	*hess_basic* Model Parameter
Mirror Area	7.3	m^2	Needed for Hz Conversion
PDE	40	%	Needed for Hz Conversion
Telescope Transmission	0.85	-	Needed for Hz Conversion
Timing Resolution	15	Minutes	Needed for Timespan and Observation Time Gain Plots

Table 5.1: Parameters recommended for use with *nsb* and associated scripts in *sstcam_nsb*. Note that a small amount of Gaussian blurring is necessary to correct for an apparent bug in *nsb*.

ensure that there is more than one *nsb* FoV map pixel, and more than one healpix pixel within in the diameter of every SSTCAM pixel on sky. The parameter selection needed to fulfill these conditions significantly slows *nsb*, but time can be saved by using the same parameter configuration consistently as *Gaia* catalogues and healpix maps are generated whenever the configuration changes. An *nsb* simulation run with these parameters takes approximately 6 hours on a single CPU core to generate an All-Sky Map, an FoV map and a corresponding FITS file (which is typically 763 Mb large using the recommended parameters). This is somewhat longer than the amount of time for which such a map prediction remains accurate, which is around 20 minutes of observing time. It should also be noted that *nsb* is comparatively RAM heavy, needing around 9 GB for these conditions. These RAM constraints rule out performing such simulations on the European Grid Infrastructure using CTA-DiRAC as typical grid nodes tend to only have 4 GB of RAM. Obtaining per-pixel estimates from these FoV maps is much faster (taking only a few minutes).

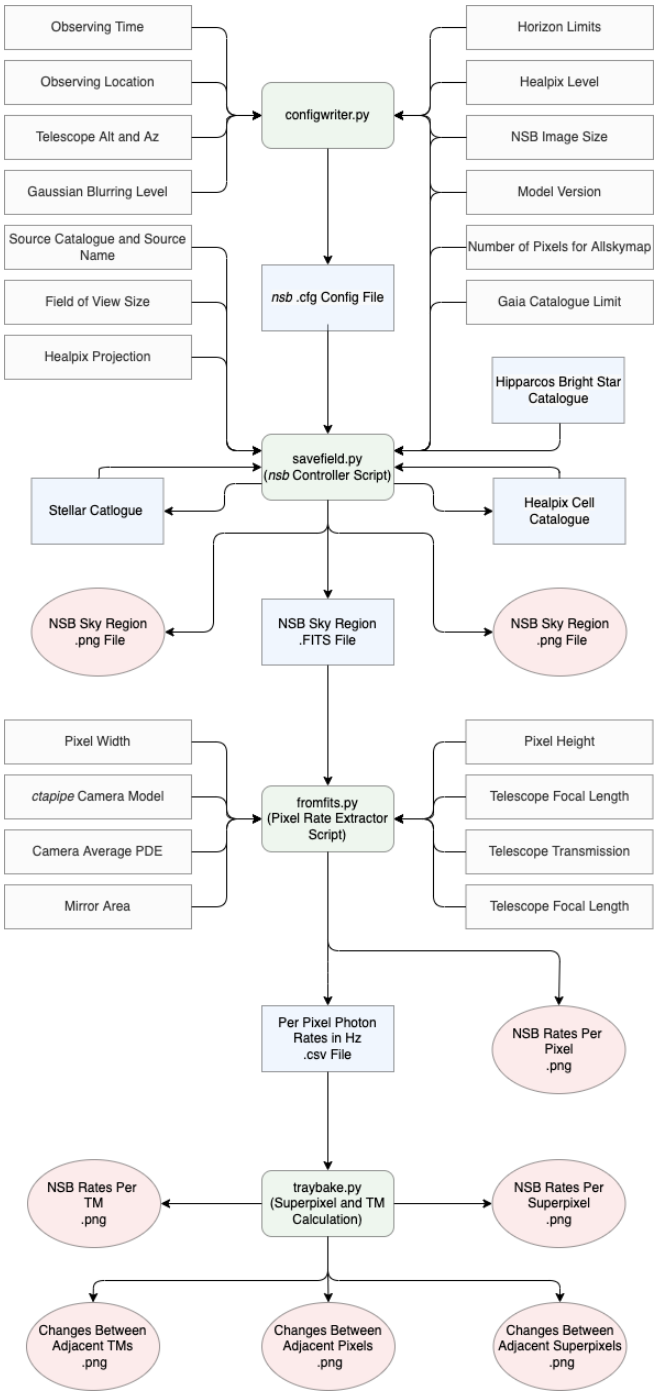


Figure 5.2: The workflow we developed for generating our NSB maps. Following the same design language used in earlier chapters, grey rectangles represent parameter inputs, blue rectangles intermediate data products, red ellipses final outputs and scripts in green rounded rectangles. Some parameters have to be input in multiple places due to the design of *nsb*. The stellar catalogue and healpix cell catalogue are generated the first time that *nsb* runs with a particular set of parameters, but these are then saved and reused, cutting down on the time needed to generate subsequent NSB maps.

WCS Header Name	Value for SSTCAM	Comments
CTYPE1	'RA—TAN'	X-Axis Projection
CUNIT1	'deg'	X-Axis Unit
CDELT1	0.001	X-Axis Increment
CRPIX1	5000	X-Axis Reference Point
CRVAL1	Determined by Pointing	X-Axis Reference Value
NAXIS1	10000	No. Pixels X-Axis
CTYPE2	'DEC–TAN'	Y-Axis Projection
CUNIT2	'deg'	Y-Axis Unit
CDELT2	0.001	Y-Axis Increment
CRPIX2	5000	Y-Axis Reference
CRVAL2	Determined by Pointing	Y-Axis Reference Value
NAXIS2	10000	No. Pixels Y-Axis
CROTA1	0	Field Rotation
CROTA2	0	Field Rotation
RADESYS	'ICRS'	Co-Ordinate System

Table 5.2: WCS dictionary headers and parameters recommended for SSTCAM, assuming the configuration in Table 5.1. Note the CTYPEn projection used affects the necessary size of the FoV map generated by *nsb*. The rotation parameters CROTA1 and CROTA2 are degenerate in their effect, and are not trivially set.

5.2.2 World Co-ordinate System Description

To perform aperture photometry on a fits file using co-ordinate data from *ctapipe* [75], one needs to define a World Co-ordinate System (WCS) for the FITS output from *nsb*. This is not automatically written by *nsb* upon the FoV map being created, but it can be provided by the user manually in an *astropy* WCS dictionary. The requisite WCS coefficient values can be determined by dividing the FoV set in *nsb* by the number of pixels in the FoV map, and setting the centre of the FITS image to the RA/DEC to which *nsb* has been aimed. These are listed in Table 5.2, assuming that the parameters in Table 5.1 are used. It should be noted that the CTYPE parameters affect the size of the FoV that needs to be used with *nsb*, tangential projections (TAN) are preferable as they result in only a 10 degree FoV being required to reliably perform that aperture photometry, which reduces the computational cost of running *nsb*, an alternative HPX healpix projection requires 12 degrees to be simulated to prevent Not a Number (NaN) values being produced at the camera edge by the pixel extractor script (which we discuss in Subsection 5.5.2).

5.2.3 Aperture Photometry

To accurately determine the NSB rates in individual pixels, one must perform aperture photometry on the FoV maps generated by *nsb*. This can be performed using the pixel position information from *ctapipe* [75] for a given observation time, location, altitude and azimuth. The pixel extractor script then uses the *photutils* *astropy*-affiliated [198] photometry package [199] to integrate the FoV map around the pixel positions, assuming they have a rectangular geometry on-sky and a known pixel diameter (0.19°). In effect, this numerically performs the integral

$$B_{Pixel} = \int B_{Total} \cdot dA \tag{5.8}$$

where B_{Pixel} is the brightness of a camera pixel and dA is a rectangular surface element on the sky, so it should be noted that B_{Pixel} is therefore not a linear function of B_{Moon}, B_{Sky} or B_{Gaia}. Validation was performed to ensure that the rotation of the camera images generated were consistent with the *ctapipe* Engineering Camera Frame.

This produces pixel maps in units of nLb/pixel, but in practice we need to convert this to Hz/pixel (which is the directly observable quantity). To do this, we assume the spectrum from *nsb* to be uniform and perform the following calculation. Assuming that the results from *nsb* cover the *Gaia* Blue Photometer (BP) wavelength range ($330 - 680\,\mathrm{nm}$ [191]), and that the *nsb* results are spread evenly across the band, this implies

$$\mathrm{B(photons/(ns\ sr\ nm\ m^2))} = \frac{\mathrm{B(Lamberts)}}{(10^4/\pi \times \mathrm{E} \times (680 - 330) \times 10^9)} \tag{5.9}$$

where $\mathrm{E} = \mathrm{hc}/(505\,\mathrm{nm})$ ($505\,nm$ being the centre of the BP wavelength range), we then multiply this by the range $300 - 550\,\mathrm{nm}$ (SSTCAM's wavelength range), multiply by an assumed 40% PDE, and then multiply by the solid angle subtended by a pixel ($8 \times 10^{-6}\,\mathrm{sr}$), the mirror area ($7.3\,\mathrm{m}^2$) and the telescope transmission (0.85) to get an NSB rate in Hz. A similar result can be obtained through scaling the spectrum from Benn and Ellison [54] (seen in Figure 5.1) and using the existing *sstcam-simulation* package [200]. The presence of strong emission lines in that

spectrum results makes that method less appropriate, as it is unlikely that light from stars or moonlight will follow such a strongly peaked spectrum. The results from this analysis are written to an *astropy* table whereby they can be further analysed at will. It should be noted that the workflow developed for this work could be reasonably trivially used by all CTA cameras, provided these basic parameters are known.

Throughout this work we assume a CTA CHEC-S pixel geometry is roughly indicative of the final SSTCAM, though this will need to be updated once a full Monte Carlo model of the engineering camera has been created and a decision on the SiPM choice for SSTCAM has been made. The value most likely to affect the results presented here is the angular extent of the pixels on the sky. We also use the 'SST-ASTRI' optical structure as defined by *ctapipe*, though the effect of this upon the analysis is minimal, as even the effective focal length of the telescope optics is set manually to 2.15191 m (this sets the plate scale of the camera and therefore determines the pixel size on-sky). Another likely source of complication that is not modelled here (as there's no trivial means of modelling it without resorting to full ray-tracing), is that for a 2 mirror Schwartzchild-Couder optical design telescope (with which SSTCAM was designed to operate) moonlight can reflect off of the secondary mirror in different ways compared to a single mirror design. The ongoing debate about the related issue of what colour to paint IACTs in order to minimise reflections from the telescope structure suggests that this effect will likely be small [193].

5.2.4 Pointing Validation

To verify the pointing accuracy of the *nsb* fields, we ran an observation of Polaris through the *nsb* chain (from the ASTRI site on Sicily). Polaris is expected to be present in the NSB model as it is included a dedicated list of bright *Hipparcos* stars that is loaded by *nsb* for every run (as the *Gaia dr1* catalog only goes up to $m_G = 3.2$ [191]). We then verified the position of Polaris in the frame using the 'V/50' Vizier catalogue [201]. This is somewhat complicated by the fact that the WCS rotation angle keywords CROTA1 and CROTA2 are free parameters (partly because the *ctapipe* skycoord.rotation and skycoord.roll fields are presently

unfilled in the latest version of *ctapipe*), but this investigation (and other similar tests we performed) give us confidence that at least the source pointed at will be within the FoV of the camera. However, this complication means that off-centre sources may by rotated in the resultant pixel maps. This is largely unimportant for our purposes as our main goal for this study was to obtain statistical estimates of NSB under various conditions and not to, for example, use *nsb* for pointing calibration studies (for which it was not designed).

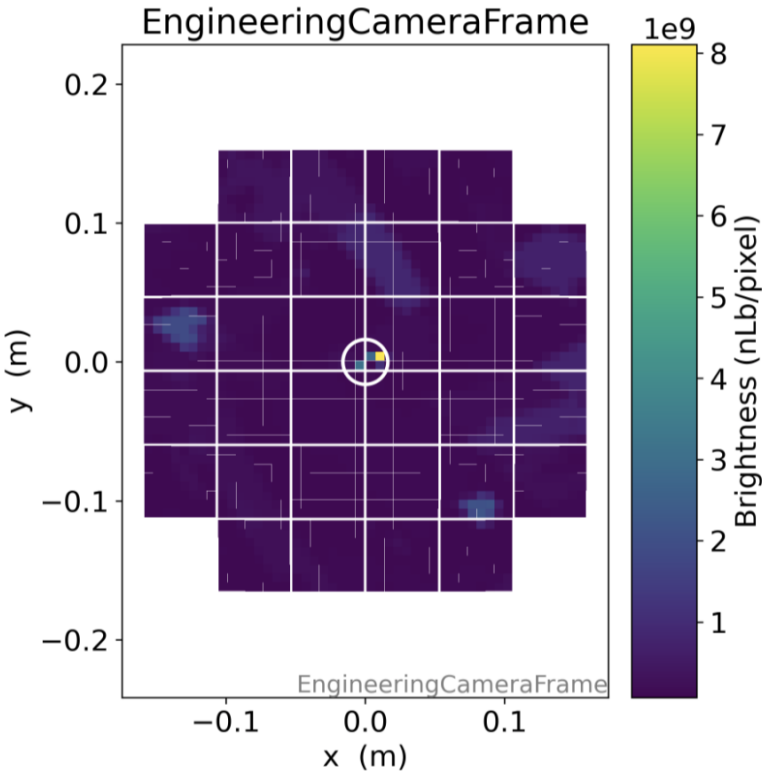


Figure 5.3: Per pixel NSB values for Polaris (position circled in the Engineering Camera Frame, using *ctapipe* and *astroquery*), note that these runs used a lower than normal magnitude limit level (8) for speed, which causes the artefacts in the image. The expected position of Polaris is circled (the associated radius is simply to highlight the position). The presence of a TM gap at the dead centre of the camera likely results in the light from Polaris being slightly off-centre.

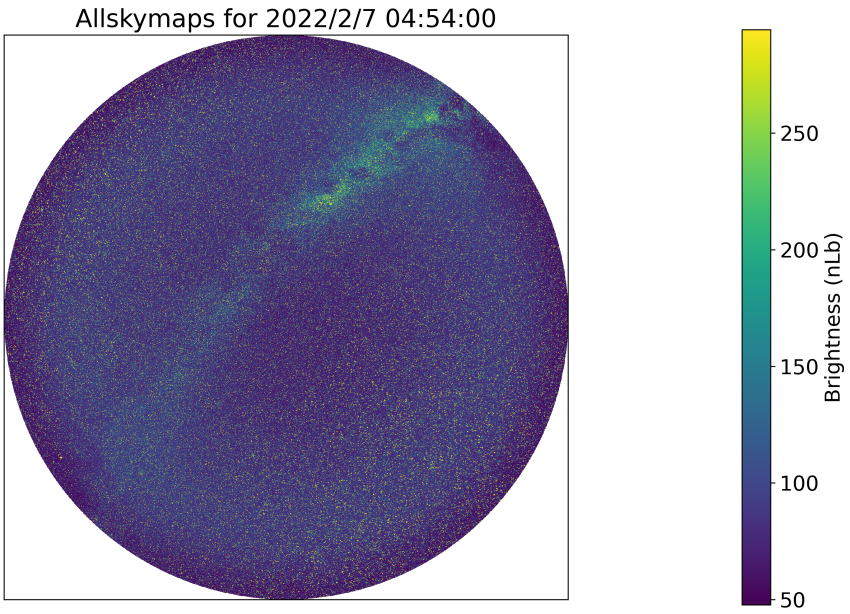


Figure 5.4: An map of the entire sky at the time of the Eta Carinae observations under astronomical dark time, generated using the *hess_basic* model in *nsb* and our selected parameters. The galactic plane is clearly visible.

5.3 Results and Discussion

5.3.1 Eta Carinae Observations Under Varying Conditions

As a reasonable worst-case scenario, we consider four observing scenarios of the colliding wind binary Eta Carinae. Eta Carinae has recently been the subject of a major paper by H.E.S.S. [202], and is considered to be a particularly difficult γ-ray source to observe with IACTs given the high stellar density in the region and the high number of UV photons produced (which causes false triggers, though this is not taken into account by *nsb*). Firstly we consider a 'dark' field, at the same altitude as Eta Carinae but differing azimuth, for comparison. This appears to be a reasonable choice of dark field more generally, as there is only a single star with a V-band magnitude brighter than 3 (Zeta Centauri, $m_V = 2.55$), though through a search of NASA's HEASARC catalogue there are 8 4FGL *Fermi* LAT point sources in the dark field, which may have spectra that extend into the SST energy range (although

this is unlikely as this is a high-galactic-latitude field meaning the 4FGL sources are likely blazars at high redshift). Secondly we consider observing Eta Carinae during astronomical dark time (at the same observing time as the dark field). Thirdly, we observe the Eta Carinae region with half moonlight present. This half moonlit scenario is designed to partially replicate the conditions for CTA requirement B-SST-1680, in this scenario the moon is above the horizon and has 0.53 Fractional Lunar Illumination (FLI). Finally we observe the region under full moonlight. The full moon scenario represents the worst possible observing conditions for Eta Carinae (and by extension the worst observing conditions possible), with the moon both at 1.0 FLI and well above the horizon. The significant all-sky illumination caused by this full moon, which completely dominates over the background stars, can be seen in Figure 5.5 (for comparison with the dark time all-sky plot in Figure 5.4).

In addition to pixel-wise analysis, we can also bin mean NSB values for these observations per superpixel and TM. This is important for understanding the triggering process, which relies on charge values per superpixel rather than per pixel, and for thermal control of the TM electronics (as each TM is thermally bonded to a chiller plate we need one temperature measurement per TM). These results can be seen in Figures 5.6, 5.7 and 5.8. Because the camera trigger rate is reliant upon two adjacent superpixels exceeding a configurable threshold (as discussed in Chapter 1), the camera readout rates will be significantly lower than the NSB photon rates presented here for individual pixels. Here is a rough calculation that demonstrates this effect [46]:

- Let us assume the nominal NSB photon rate hitting an individual pixel is 40 MHz (the currently assumed SSTCAM rate [46]).
- There are approximately 10,000 SiPM μcells per pixel [56], which means that the mean NSB photon rate per μcell is ~ 4 kHz.
- If SSTCAM is equipped with a digital trigger that requires approximately 10 μcells to fire within the FWHM of our SiPM pulse length (10.5 ns), then the

Observation	Observing Time	ALT (°)	AZ (°)	RA (°)	DEC (°)
Dark 'Empty' Field	2022-02-07T04:54:0	33.1	126.7	162.0	-43.0
Eta Carinae No Moonlight	2022-02-07T04:54:0	33.1	146.7	161.3	-59.7
Eta Carinae Half Moonlight	20-02-21T01:10:0	12.5	150.8	161.3	-59.7
Eta Carinae Full Moonlight	2022-05-16T07:00:00	13.1	209.6	161.3	-59.7

Table 5.3: Observation parameters for the four Eta Carinae Runs. The observing Altitude (ALT) and Azimuth (AZ) are presented, along with the simulated source Right Ascension (RA) and Declination (DEC). The moonlit Eta Carinae runs are at a low altitude that an IACT would not normally observe at, but since the *nsb* model does not contain a full atmospheric model this is inconsequential. Times are in UTC.

accidental trigger rate A_{10} for 10 of our μcells out of a total of $10,000\,\mu\text{cells} \times 4$ (a single superpixel) is given by [203]

$$A_{10} = \frac{(4 \times 10,000)!}{10!(4 \times 10,000 - 10)!}(4\,\text{kHz})^{10}(10.5\,\text{ns})^{10-1} \approx 5\,\text{kHz per superpixel.} \tag{5.10}$$

- Then as the backplane co-incidence trigger requires 2 neighbouring superpixels to trigger within 8 ns of each other, and there are 1910 combinations of our $2048/4 = 512$ superpixels that form neighbours, then the camera trigger rate C_T is

$$C_T = 1910 \times (5\,\text{kHz})^2 \times 8\,\text{ns} \approx 380\,\text{Hz.} \tag{5.11}$$

It should be noticed that this rough calculation is very sensitive to the superpixel trigger threshold and the selected TARGET ASIC co-incidence window.

The results from our investigation are encouraging. For the dark field we compute a photon rate of 43 MHz, a calculation performed by K. Bernlohr using *sim_telarray* by scaling the Benn and Ellison spectrum [54] computes 41.9 MHz. Assuming Poisson error these results are in agreement. Even for the worst case scenario the average photon rate is within SSTCAM observing requirements, though the combination of moonlight and a bright stars means pixels greater with an NSB rate of more than a few GHz will need to be disabled (or have their HV ramped down) and removed from Cherenkov analysis in order to manage SiPM heating. In the case of brighter stars illuminating a pixel at around a GHz it is likely such pixels will only be removed from the trigger analysis.

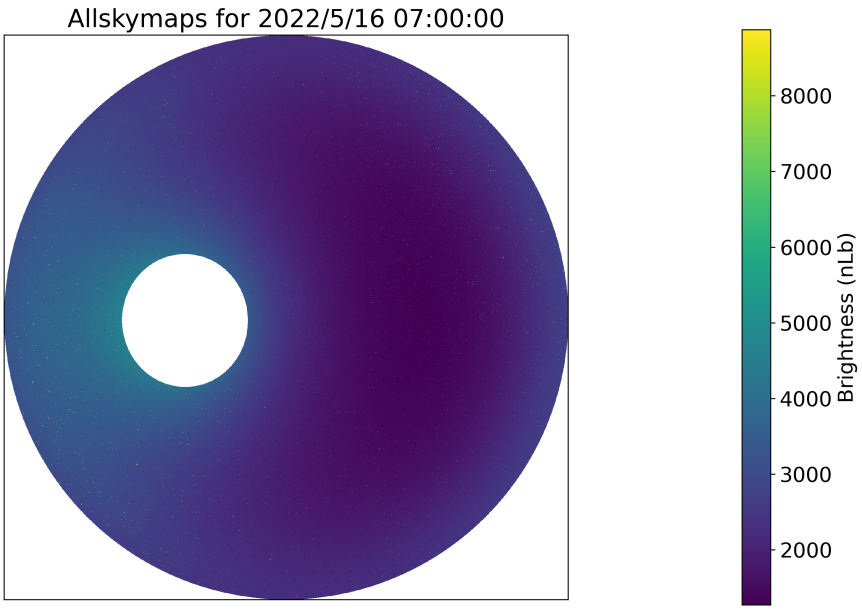
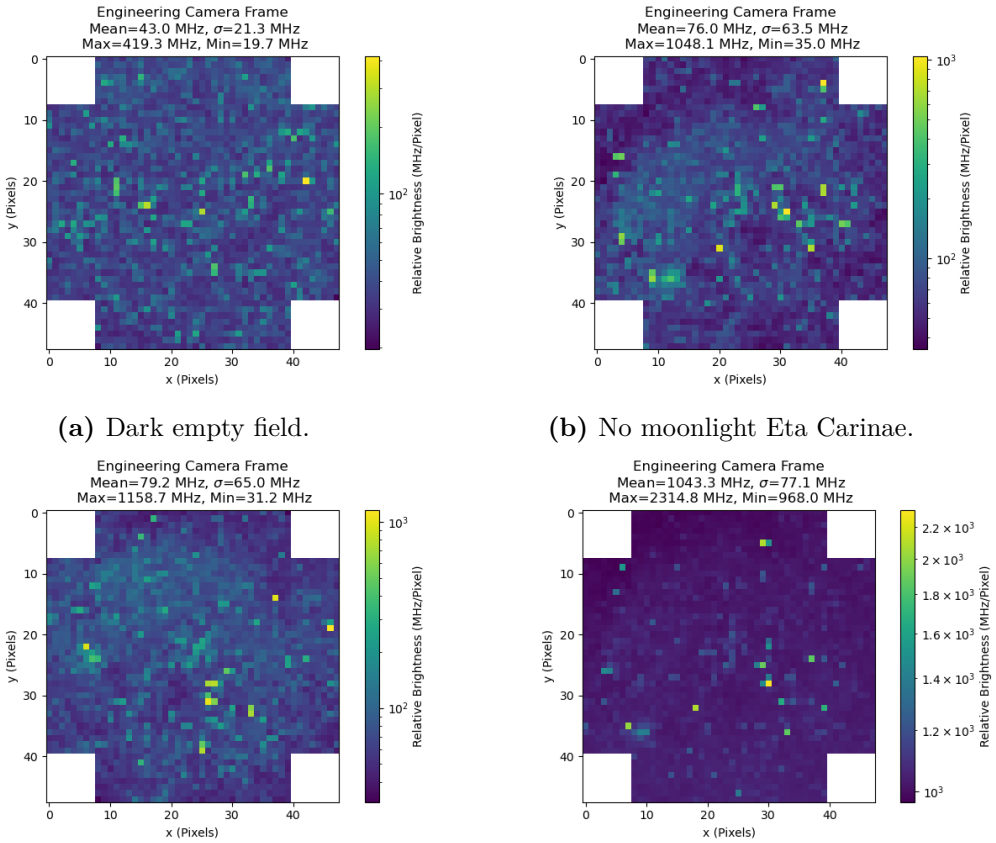


Figure 5.5: An all-sky plot for the full moonlight run, highlighting the presence and brightness of the moon for this observation. *nsb* performs an angular cut around the region of the moon when performing the all-sky-map calculation (resulting in the hole in this map).

5.3.2 Changes Between Adjacent Units for Eta Carinae

We also re-analysed the data for the purposes of determining the maximum change between horizontally and vertically adjacent pixels, superpixels and TMs. This is necessary to predict the influence of stars on the camera trigger (for individual pixels), on bright NSB regions on the superpixel trigger logic, and for understanding the thermal load on the cooling plate bonded to each TM. This analysis was performed by devising a function that, for every pixel in a 2D array, obtains the maximum change in mean NSB rate between itself and its adjacent neighbours (if they exist), then assigns that value to the pixel. This calculation neglects changes in the PSF across the focal plane in the camera. The results from this investigation are shown in Figures 5.9, 5.10 and 5.11.

In the case of individual pixels, it appears the pixels adjacent to those with high NSB from stars are largely isolated from other potential sources of NSB, however



(a) Dark empty field.

(b) No moonlight Eta Carinae.

(c) Half moonlight Eta Carinae.

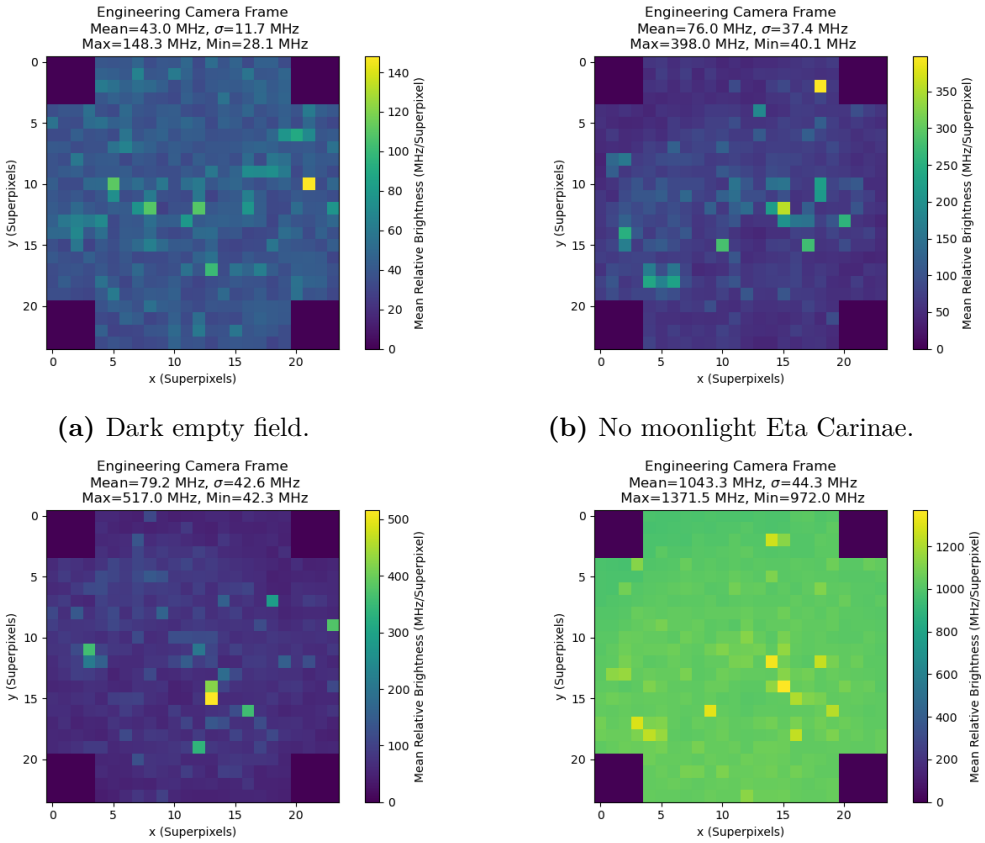
(d) Full moonlight Eta Carinae.

Figure 5.6: Per pixel NSB values for Eta Carinae under the four observing scenarios described.

once these results are binned and averaged by superpixel and TM that one starts to frequently observe the effect of multiple stars overlapping. The increased changes near the camera's uninstrumented corners are an analysis artefact stemming from the lack of pixel, TM and superpixels in these regions. However, it appears clear that pixels, superpixels and TMs need to be robust to a single pixel being at illuminated at a rate of 1 GHz whilst its neighbours are illuminated at a few hundred MHz.

5.3.3 Change in Eta Carinae NSB Over Time

To determine the maximum possible change in NSB through the camera over a run, we ran *nsb* to simulate an observation of the instant exactly a half hour before each of the observations in Table 5.3 (tracking the source). This choice was based on the fact that the source sets after the full moonlight run, and also removes potential



(a) Dark empty field.

(b) No moonlight Eta Carinae.

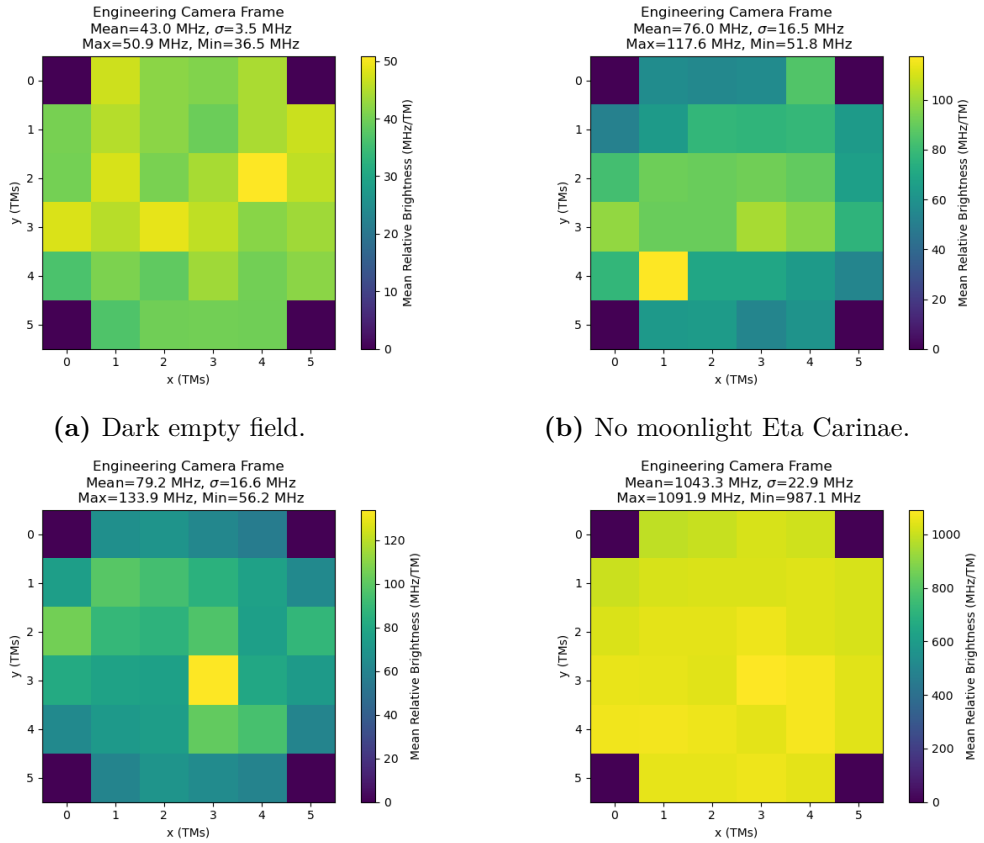
(c) Half moonlight Eta Carinae.

(d) Full moonlight Eta Carinae.

Figure 5.7: Mean NSB values per superpixel for Eta Carinae under the four observing scenarios described.

issues related to normalisation of pixel intensities in nLb over time that we discuss in the next subsection. Whilst these results in Figures 5.12, 5.13 and 5.14 show that the brightness of an individual pixel can change on the order of a GHz per 30 minutes, changes in the mean illumination of the camera are much slower, being of the order of 2 MHz/hour. This is with the exception of the fully moonlit scenario, where the mean change over half an hour is dominated by a bright star.

A procedure to deal with the moon entering the FoV (potentially by accident) has not yet been established for SSTCAM (one option is to pre-emptively drop the gain for the entire camera). However, attempting to run NSB for an observation whereby the moon is directly in the field crashes *nsb*. As such, the change in the full moonlit Eta Carinae field represents the greatest shift in average NSB rate that can be simulated (30 MHz/Minute/pixel) with *nsb*.



(a) Dark empty field.

(b) No moonlight Eta Carinae.

(c) Half moonlight Eta Carinae.

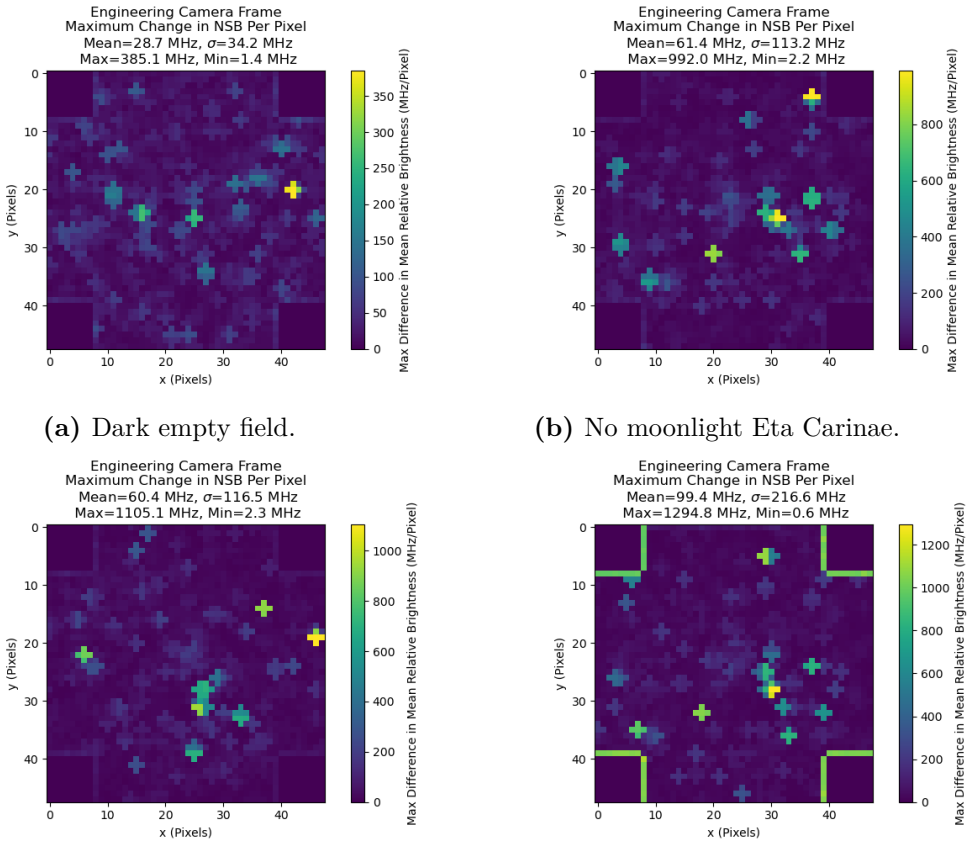
(d) Full moonlight Eta Carinae.

Figure 5.8: Mean NSB values per TM for Eta Carinae under the four observing scenarios described.

5.3.4 Eta Carinae NSB Per Pixel

nsb has a tool to determine the limit of brightness in a single pixel over a year. However, said tool only provides results in nLb at a particular infinitesimal point, which is something of a meaningless quantity. As such, for running this tool for SSTCAM with NSB, we assume our non-moonlight Eta Carinae NSB run from Section 5.5.3 represents a typical observation, and scale the mean of the year long timespan plot to the mean pixel value for that run. In the case of the maximum NSB variation over a single night, we instead normalise the maximum of the NSB rate to the maximum of the year long plot (to which the single night's observation corresponds).

Figure 5.15 clearly shows that whilst there is modulation in the apparent brightness of the region given lunar phase and separation, there is also a very strong



(a) Dark empty field.

(b) No moonlight Eta Carinae.

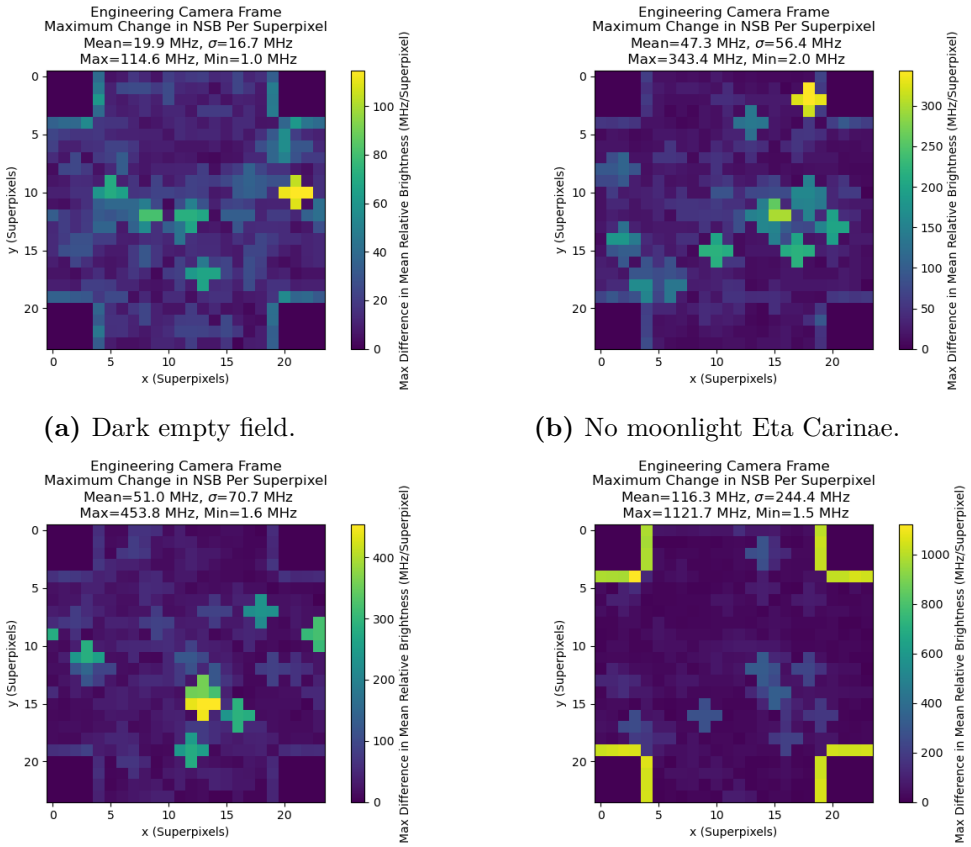
(c) Half moonlight Eta Carinae.

(d) Full moonlight Eta Carinae.

Figure 5.9: Maximum differences between adjacent pixels in NSB values for Eta Carinae under the four observing scenarios described.

dependence on moon altitude, with the brightest times being when the moon is both high on the sky, near 1.0 FLI, and close to the source. Throughout these results and those in Subsection 5.5.3 it appears as though lunar illumination, separation and altitude will be the dominant cause of both high NSB and rapid changes in it.

There is some inconsistency between this result and the result for the entire camera on the same night with a 30 minute difference, which shows an overall drop in NSB over the same time window, although both show extreme variation. This could be the result of a minor bug in *nsb*, or it could be a results of the infinitesimally small region of the sky corresponding to Eta Carinae brightening whilst the larger FoV is darkening. However, this suggests that even on the most extreme sources changes in NSB rate per pixel should be within the capacity of the LED flasher system to correct.



(a) Dark empty field.

(b) No moonlight Eta Carinae.

(c) Half moonlight Eta Carinae.

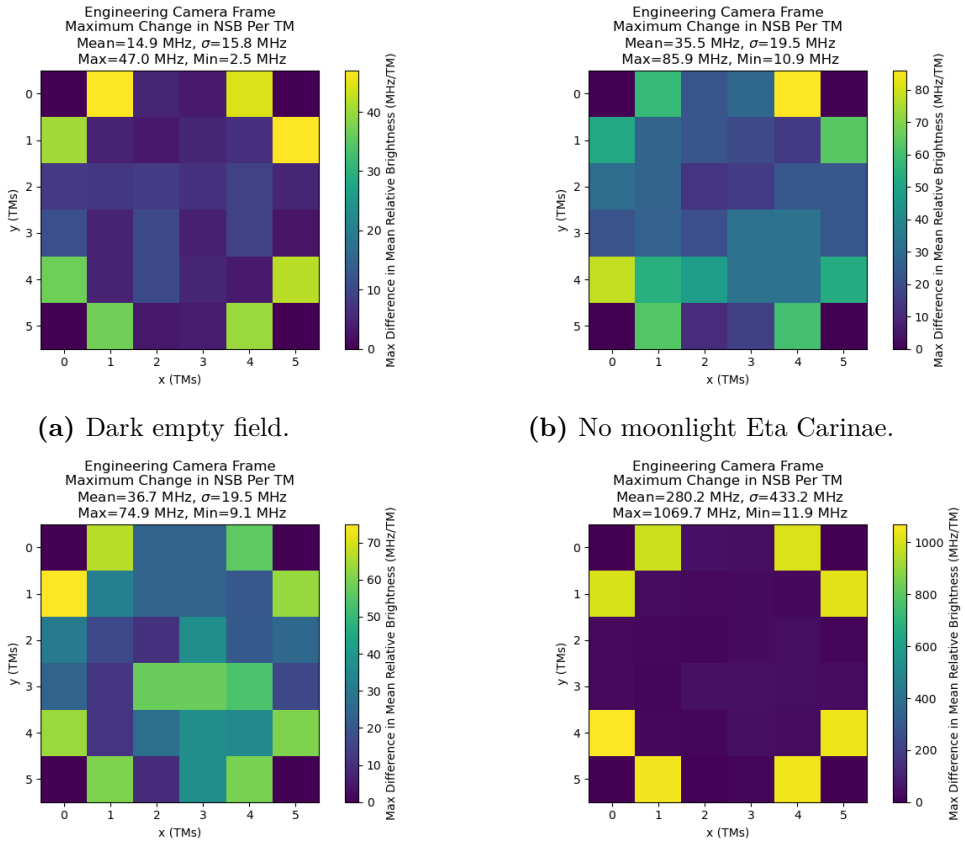
(d) Full moonlight Eta Carinae.

Figure 5.10: Difference in mean NSB values between adjacent superpixels for Eta Carinae under the four observing scenarios described.

5.3.5 Observing Time Gains

Within *nsb*, there is similarly a function to provide plots of the observation time gained as a function of NSB threshold, but this again is in nLb at an infinitesimal point. As such, we treat our Eta Carinae dark frame as nominal, and create a timespan plot for that night (shown in Figure 5.17 and observing location. The total possible observing time is obtained by summing all the elements in the brightness array where the brightness is less than a given threshold , and then multiplying this by the time resolution (as the time resolution for this brightness array is a pre-set parameter).

We then divide the observation time gain values by a the nominal NSB value obtained from this measurement (50 nLb), and perform an observing time gain calculation for the Vela Pulsar (Figure 5.18) and the blazar Markarian 421 (Figure



(a) Dark empty field.

(b) No moonlight Eta Carinae.

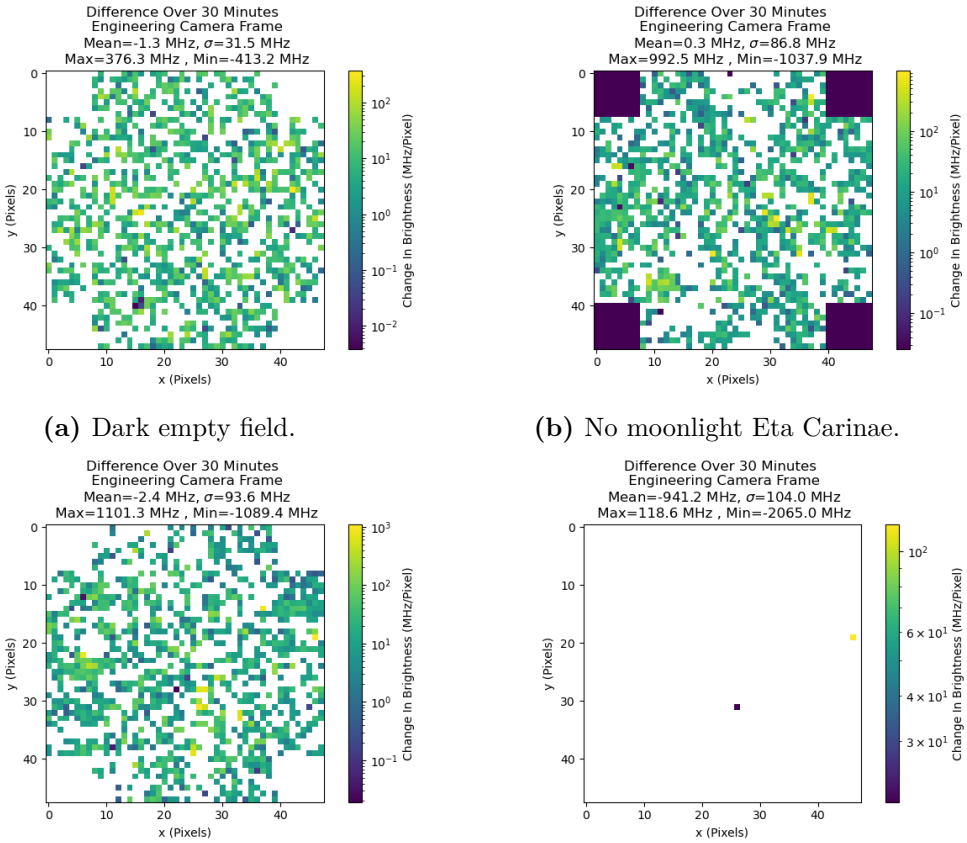
(c) Half moonlight Eta Carinae.

(d) Full moonlight Eta Carinae.

Figure 5.11: Difference in mean NSB values between adjacent TMs for Eta Carinae under the four observing scenarios described.

5.19, in order to consider differences from galactic versus extragalactic sources) from the Paranal site, considering a year at the Paranal site starting at 2022-01-27. The associated CTA requirement is also shown in these figures.

Whilst the potential observation time gains for the Vela Pulsar are significant, the potential benefits of operating at a higher NSB rate are more muted for Markarian 421. This is partly due to the location of the Paranal site, and as such Markarian 421 is typically lower on the sky, but it is also likely that the fact that Markarian 421 is away from the galactic plane (with fewer stars in the field) contributes to this. This therefore bodes well for potential SSTCAM galactic transient science, such as observations of the recurrent nova RS Ophiuchi (recently detected by H.E.S.S. in ATEL 14857).



(a) Dark empty field.

(b) No moonlight Eta Carinae.

(c) Half moonlight Eta Carinae.

(d) Full moonlight Eta Carinae.

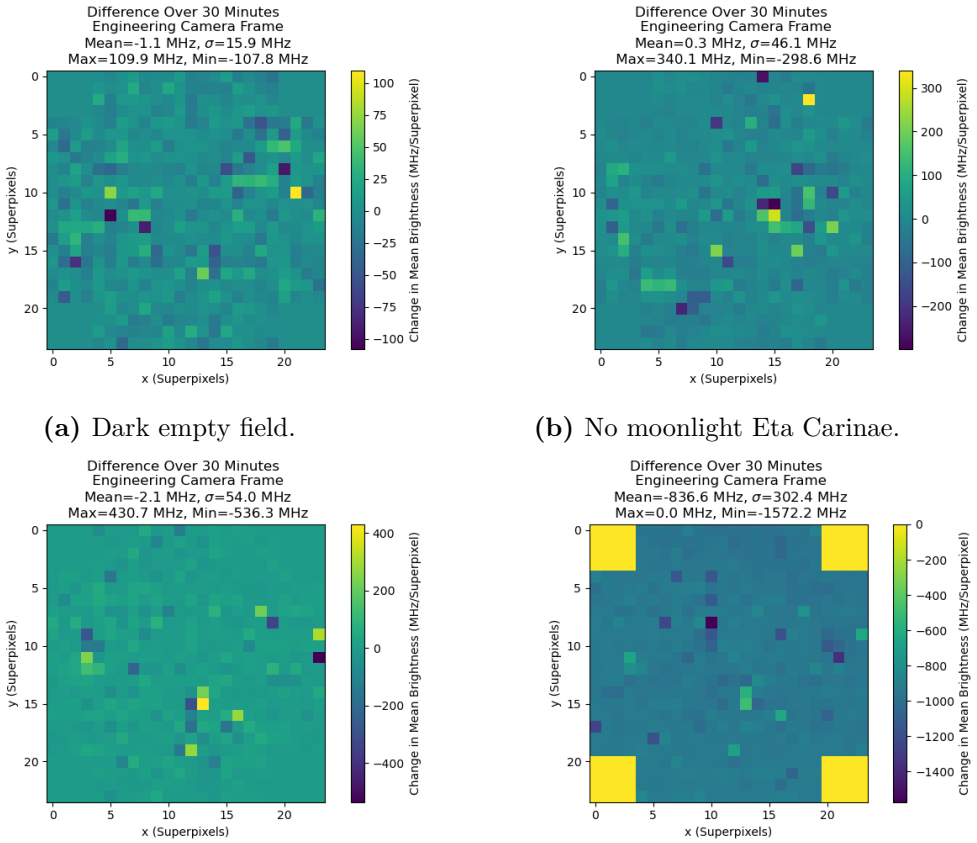
Figure 5.12: Changes in Eta Carinae NSB values per pixel over 30 minutes.

5.3.6 Exact Matching of Half-Moonlight Requirement

Whilst a reasonable approximation given the complexity of observing a particular source, the half-moonlight Eta Carinae runs are not an exact match to the CTA SST Requirement B-SST-1680, which states

> *SSTs must be capable of γ-ray observations with uniform night sky background illumination levels up to at least 4.3 photons $ns^{-1}sr^{-1}cm^{-2}$ in the wavelength range $300 - 650\,nm$ with the Moonlight Reference Spectrum.*

Additionally, this requirement mandates that the moon must be at 0.5 FLI and is at an altitude of 45 degrees and the telescope is pointed at the zenith. To compare to our half-moonlit Eta Carinae run, we ran *nsb* runs with exactly these parameters for UTC time 2022-03-24T07:32:00.000, the results from this investigation are shown in Figure 5.20.



(a) Dark empty field.

(b) No moonlight Eta Carinae.

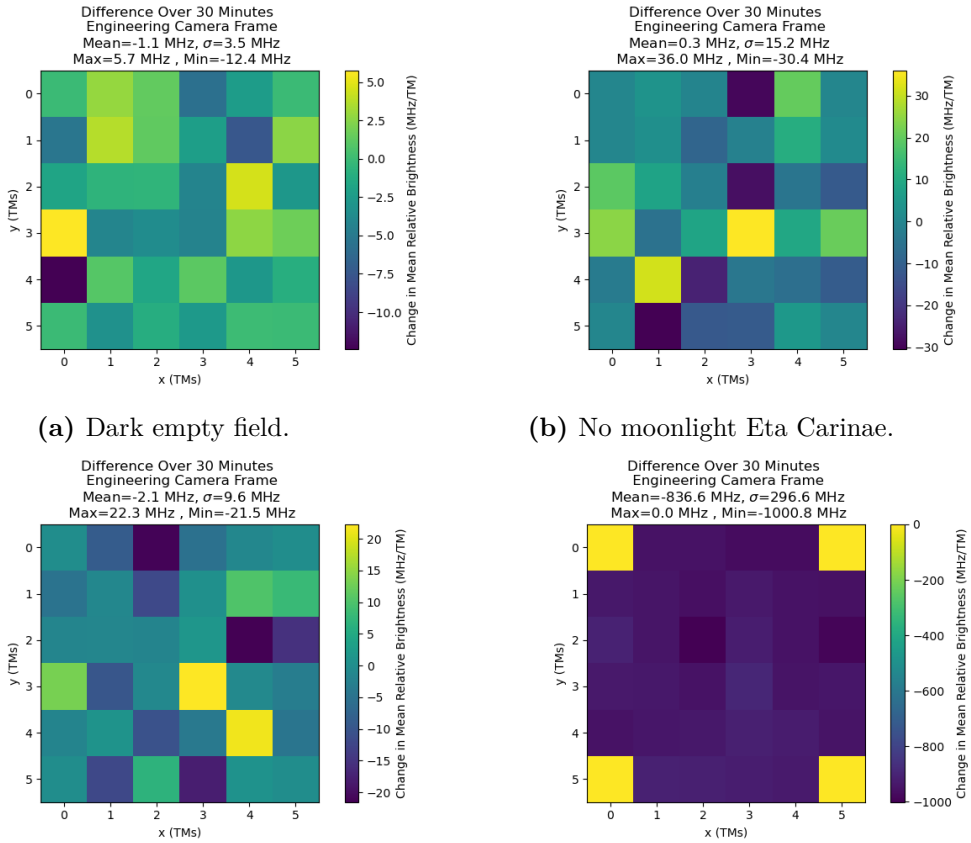
(c) Half moonlight Eta Carinae.

(d) Full moonlight Eta Carinae.

Figure 5.13: Changes in Eta Carinae NSB values per superpixel over 30 minutes.

The results for these half-moonlight observations demonstrate that if SSTCAM performance requirements up to a 160 MHz NSB rate can be met, the overall camera will function well under most realistic observing scenarios. The question becomes if individual pixels will be able to handle such NSB rates without an overwhelming gain×PDE×Optical Cross-Talk (OCT) drop. Whilst in general we want OCT (a phenomenon caused by infrared secondary photons generated during the electron avalanche escaping their SiPM channels) to be as low as possible, as all three of these quantities are proportional to ΔV; if we reduce the overvoltage to minimise the OCT we will degrade the camera's performance.

These half-moonlit results are slightly more pessimistic than the half-moonlit Eta-Carinae runs, due to the difficulty of observing Eta Carinae at the requirement altitude from the southern site, but they are still within a factor of around 2 of the same values. They are also largely consistent with values generated using



(a) Dark empty field.

(b) No moonlight Eta Carinae.

(c) Half moonlight Eta Carinae.

(d) Full moonlight Eta Carinae.

Figure 5.14: Changes in Eta Carinae NSB values per TM over 30 minutes.

sim_telarray; for the simulation production Prod3b *sim_telarray* predicts between 160 and 200MHz (depending on SiPM selection). This *sim_telarray* prediction is based on scaling the Benn and Ellison spectrum under the same conditions. In contrast, the existing half-moonlight investigation for the SST-1M project [185] estimated a value of 670 MHz for the same conditions. Other than human error a possible source of this discrepancy is the physical size of the photomultiplier pixels, which for SSTCAM is 6/7 mm (0.19° on sky) as opposed to around 2.32 cm (0.24° on sky) for Digicam (the SST-1M prototype camera). Other differences between the photomultipliers used (such as PDE) might also affect the Digicam results.

5.3.7 Required Flasher Timespans

The complexity of operating a fast-signal chain for Cherenkov astronomy simultaneously with a slow-signal chain for pointing validation is a problem unique to

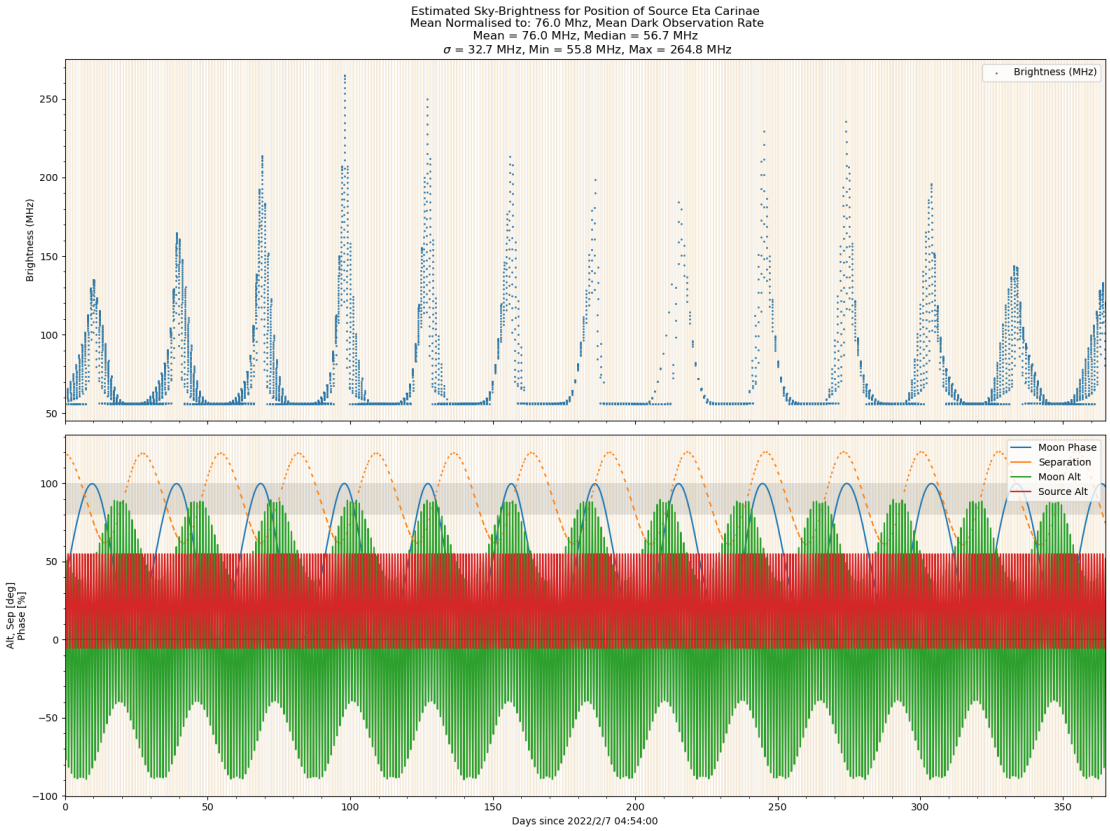


Figure 5.15: Pixel brightness in Hz for Eta Carinae over a year, normalised to a mean observing rate of 76Hz.

SSTCAM, and implementing this requires that the effect of bright stars upon the telescope calibration is well understood. In the next three subsections, we will attempt to better understand the effects of such stars upon SSTCAM calibration.

The angular field rotation of a star R can be described as a function of Latitude (Lat), Altitude (Alt) and Azimuth (Az) using the formula

$$R(\mathrm{Lat}, \mathrm{Alt}, \mathrm{Az}) = K \cos(\mathrm{Lat}) \frac{\cos(\mathrm{Az})}{\cos(\mathrm{Alt})} \tag{5.12}$$

where the constant K is 15.04 degrees/hour. Using this formula, it is possible to extract the number of required flasher calibration pulses to reach a mean flasher error level. If we assume a mean illumination level from the flasher of 50 photoelectrons (pe), and that the Excess Noise Factor (ENF), a property of the SiPMs used that describes the physical limit for the best possible SiPM charge resolution [56], is 1.4 (the physically un-reachable ideal limit at which the charge resolution is limited

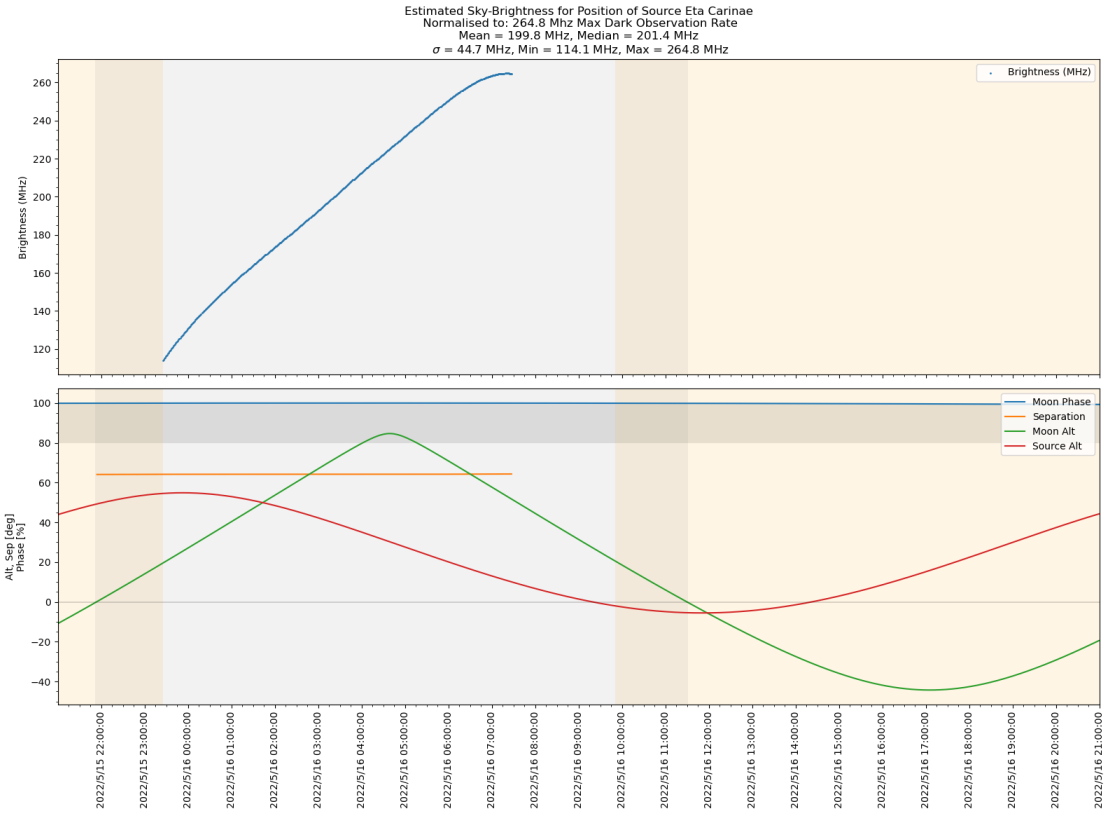


Figure 5.16: Pixel brightness in Hz for Eta Carinae for the brightest observing night in 2022, normalised to a 264.8 Hz max observation rate (as determined from Figure 5.15).

by Poisson fluctuations would be 1). Then we need 40 s (i.e. 400 pulses) to reach a 1% error on the mean flasher level per pixel. This is the needed flasher calibration level for SSTCAM to be able to meet the intensity resolution requirements of CTA for high-amplitude signals, details of this intensity resolution requirement can be found in [204]. The limits on stellar rotation rate occur at the point at which stars cause a gain change in a SiPM pixel that happens more rapidly than can be calibrated for by injecting flashes at a fixed rate.

Figure 5.21 shows the areas on the sky where a 1% flasher error can be achieved given this model of angular field rotation. The sky positions (assuming the Paranal site) of bright stars (brighter than 4 mag) from the *Hipparcos* catalogue are calculated using the *Skyfield* package and are also plotted.

This clearly causes a slight error in calibration along the North-South axis, but this is tolerable given the rate at which stars move across the sky. By doubling

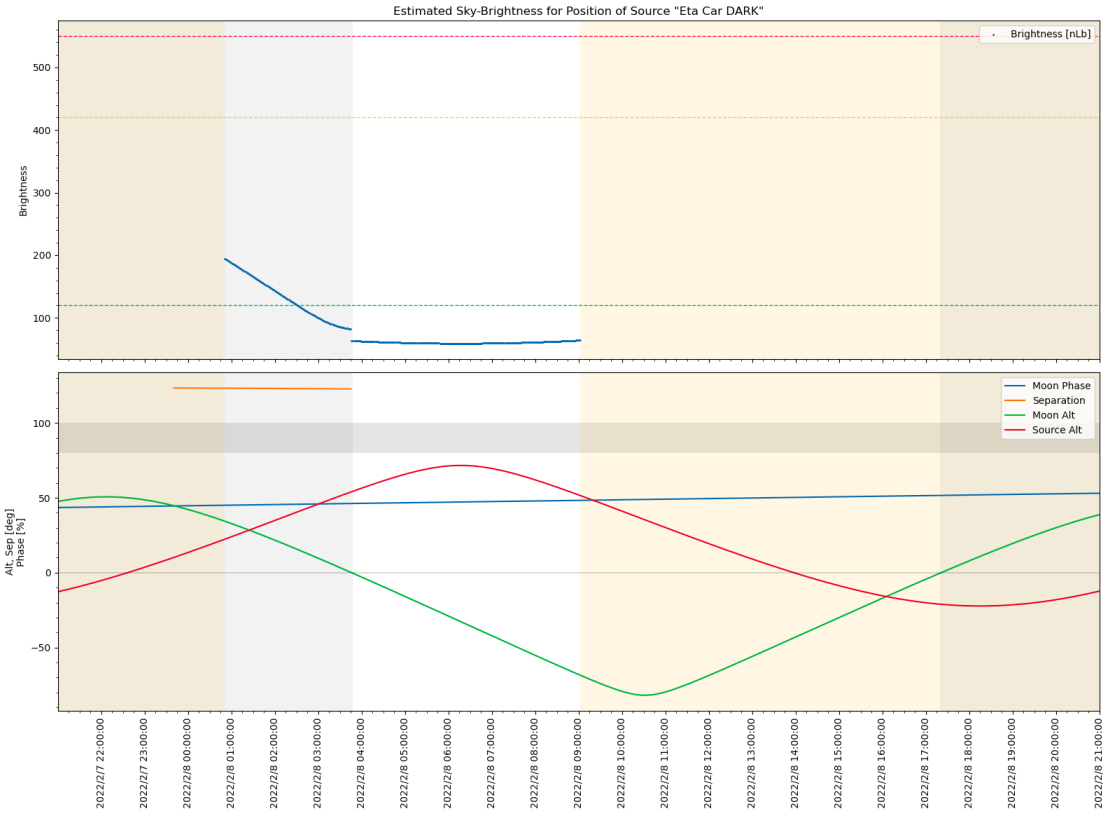


Figure 5.17: NSB timespan plot for the dark field used to calibrate the observing time gain calculation. The CTA SST operation requirement (up to 18x nominal NSB) is also shown.

the error budget and reducing the between the flashing procedure one can achieve much greater calibrated sky coverage, as seen in Figure 5.22, with only a handful of bright stars falling outside the calibrated range. These results suggest that the current flasher calibration plan for SSTCAM of using the flasher to generate 100-200 photons with nanosecond precision [2] is likely to be acceptable.

5.3.8 Star Tracking Investigations

To understand the short-term effect of stars moving through the field, we ran *nsb* fields at 3 minute intervals, pointing at the zenith under half-moonlit conditions, and observed both the absolute pixel NSB values and the overall change since the first field. Note that this is also a worst-case scenario observing the fastest

[2]Further details of this (early) calibration plan can be found on the internal SST-CAM nextcloud pages at `https://pcloud.mpi-hd.mpg.de/index.php/apps/files/?dir=/SSTCamera/03_CameraCalls/CameraCallMaterial&openfile=305207`.

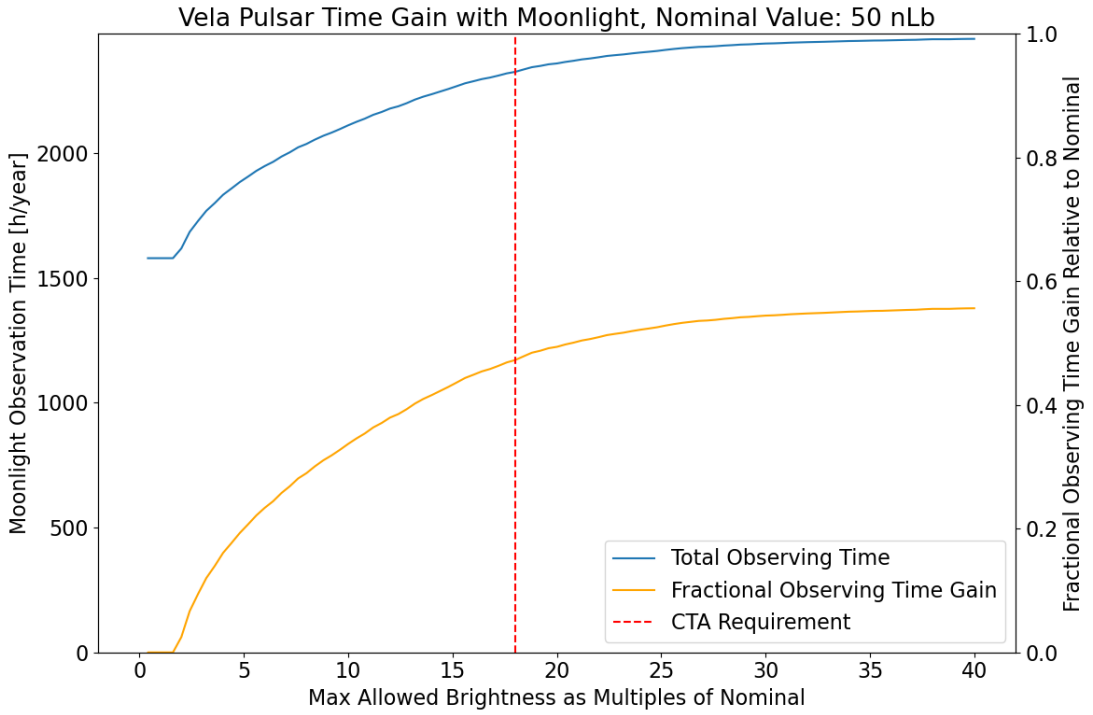


Figure 5.18: Observing time gain for the Vela Pulsar as a function of nominal NSB.

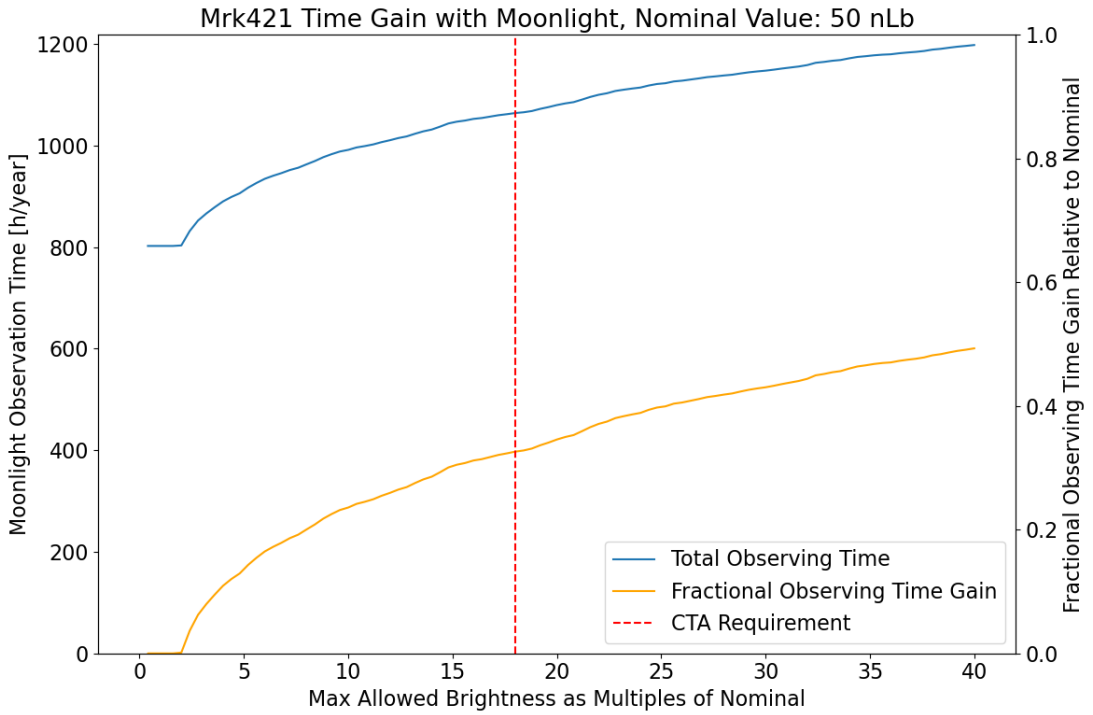
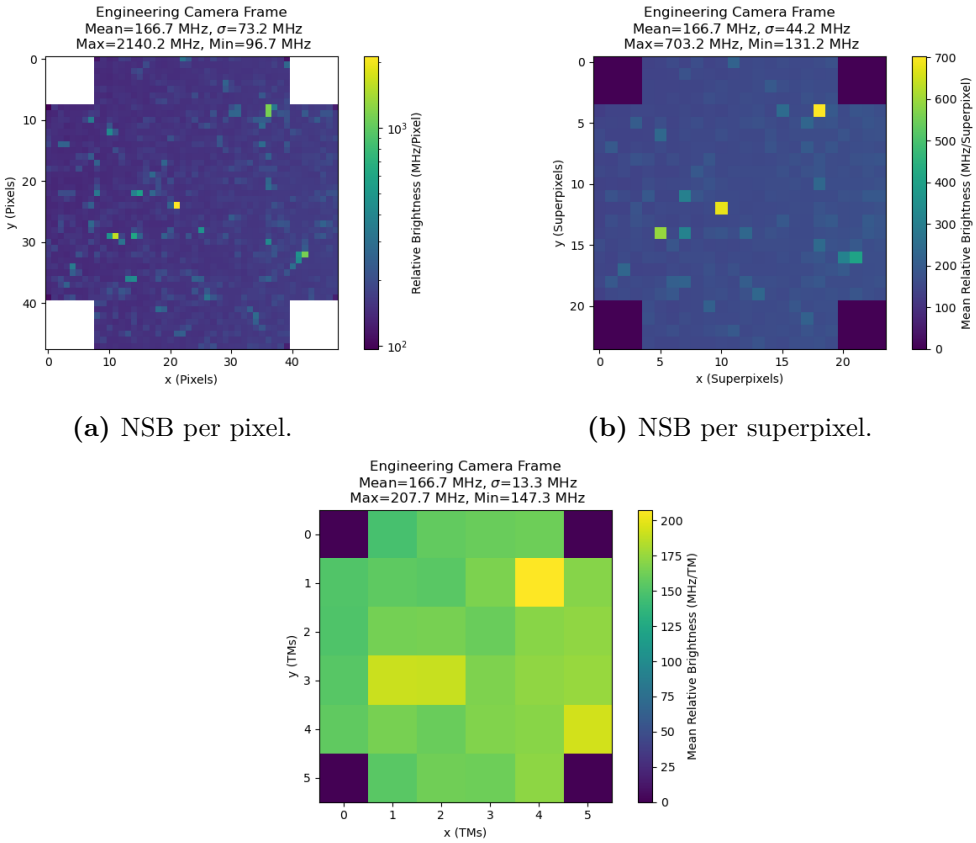


Figure 5.19: Observing time gain for Markarian 421 as a function of nominal NSB.



(a) NSB per pixel.

(b) NSB per superpixel.

(c) NSB per TM.

Figure 5.20: Results for half-moonlit zenith observations.

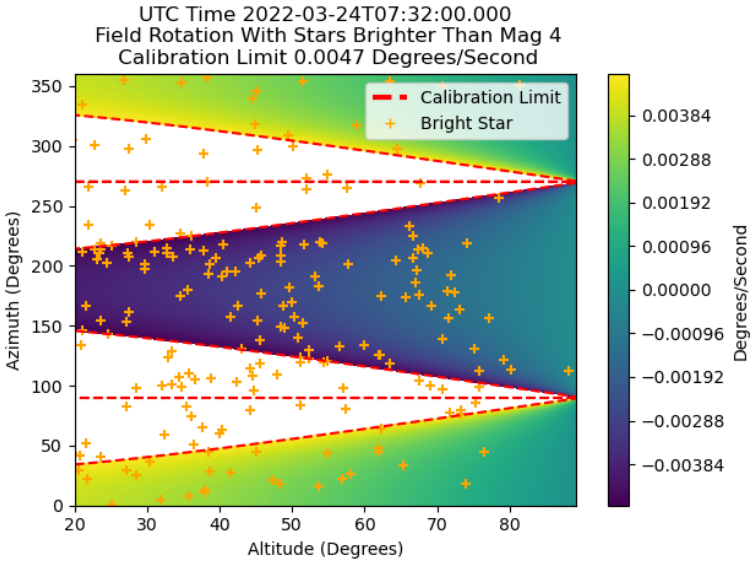


Figure 5.21: Field rotation with flasher calibration limit, assuming 40s flasher calibration intervals.

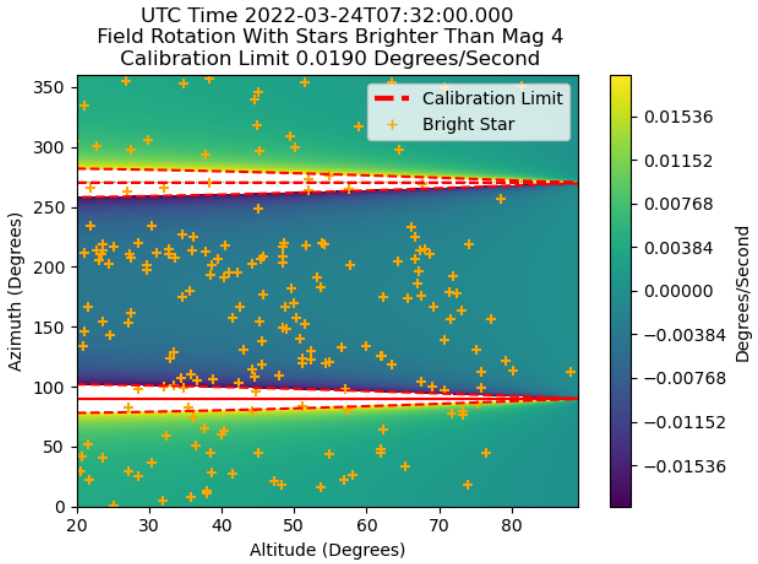


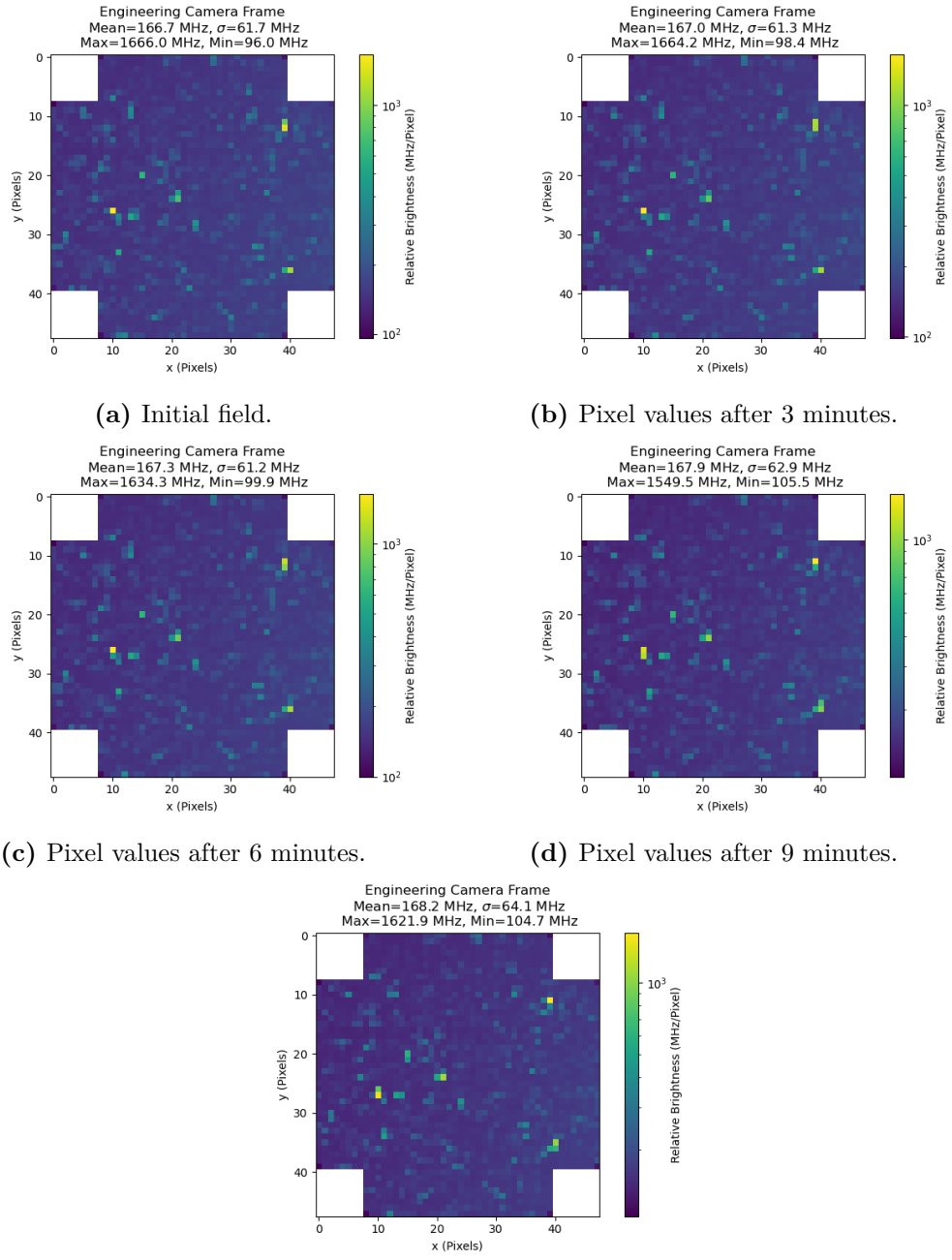
Figure 5.22: Field rotation with flasher calibration limit, assuming 10s flasher calibration intervals.

possible changes in stellar position, since R in Equation 5.12 is largest for a given observing latitude at the zenith.

Despite the PSF not being particularly well modelled, it appears that bright stars move to adjacent pixels over a roughly 15 minute timespan, with light being partially 'split' between multiple pixels and the associated ratio changing over order 3 minutes.

5.3.9 Results for Observations of a Magnitude 0 star

For upcoming comparisons with other ray-tracing simulation methods, we investigated the brightness of Rigel (m=0.12) under astronomical dark time (UTC time 2021-12-08T04:48:00.000). This shows the light from a magnitude 0 star corresponds to an NSB rate of around 900 MHz in a single pixel (lower than previously expected). The movement of the star through multiple pixels in the camera plane could affect this value as seen in the previous subsection.



(a) Initial field.

(b) Pixel values after 3 minutes.

(c) Pixel values after 6 minutes.

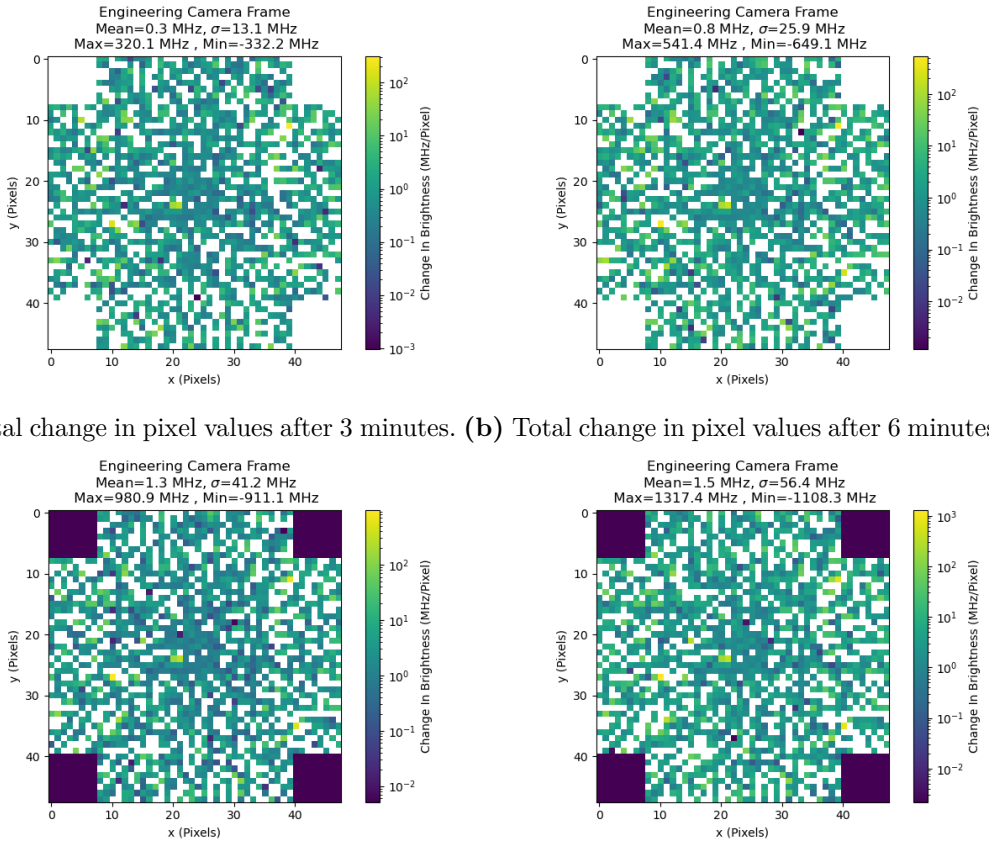
(d) Pixel values after 9 minutes.

(e) Pixel values after 12 minutes.

Figure 5.23: Pixel NSB Values in Hz, pointing at the zenith under 'half-moonlit' conditions with high resolution timing, note in particular the changes to the three brightest pixels in the images.

5.4 Conclusions

In this chapter, we have examined the potential effects of NSB for SSTCAM. We have presented detailed simulations of the background to Eta Carinae under four



(a) Total change in pixel values after 3 minutes. **(b)** Total change in pixel values after 6 minutes.

(c) Total change in pixel values after 9 minutes. **(d)** Total change in pixel values after 12 minutes.

Figure 5.24: Overall change in pixel NSB values in Hz for the high time resolution star tracking investigation.

observing scenarios, calculated the effect on individual pixels observing the same field over time, calculated the potential observing time to be gained for two sources by operating at a high NSB level, verified the half moonlight CTA requirement, investigated on short timescales the movement of stars through the camera and considered the ability of the LED flasher system to compensate for NSB from stars.

Whilst the *nsb* package has proven capable of generating useful results for SSTCAM, it currently does not meet CTA software requirements for test coverage or code quality (and development by H.E.S.S. on the package has stalled). There would be significant advantages to recreating its functionality in the future with a *ctapipe*-affiliated project (following modern code development principles) for all CTA instruments. Notably *nsb*'s speed could likely be improved by using Just-In-Time

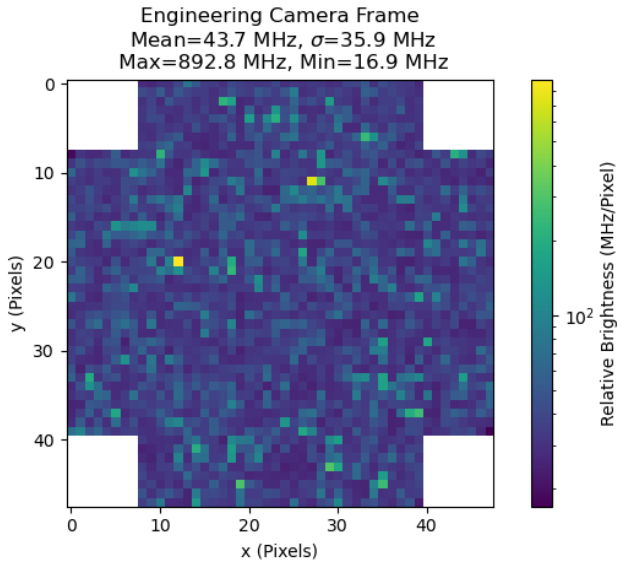


Figure 5.25: Rigel observed with *nsb* under astronomical dark time conditions.

compilation and a storage medium for healpix data that was quicker to access and parse than a text file (as is currently the case).

It should be noted that the results presented here are a significantly simplified analysis. In particular, we do not consider the effects of OCT, focal plane curvature or non-uniformity of the PSF across the camera plane. Similarly, there is a potential for 'ghost' stars to appear in the field of view as a result of reflections of starlight from the SiPMs or window. We also neglect any wavelength dependence of PDE or telescope transmission. That said, our models are significantly more realistic than those available natively in *sim_telarray*, are broadly as expected, and appear to be largely consistent with values obtained from dedicated full Monte Carlo simulations.

6 Conclusions

Contents

6.1 Outlook

Throughout this thesis, we have explored the complex task of performing deep-learning based event classification for stereoscopic IACT arrays, as well as exploring the background present in IACT images due to the night sky. The results presented include a number of interesting discoveries which we will now summarise.

In Chapter 3, we have seen that high-precision timing information can offer a clear advantage over conventional charge images when dealing with deep-learning-based γ-hadron separation for SSTCAM on simulations. However, the full sensitivity

analysis for this method will not be possible until a full CTA analysis chain is in place. Given the potential significant performance boost over integrated charge images, it is possible that the timing methods we demonstrate might perform better with real data compared to charge data analysis, but we will need to wait until there is an SST prototype on the Paranal site in order to verify this. In Chapter 4, we have seen the complexity involved in applying deep-learning-based event classification methods to IACT charge data from VERITAS. Our most notable discovery is that it is indeed possible to perform real detections in IACT data of real sources, without resorting to image cleaning or strenuous cuts. In particular, we have found that Bayesian optimisation can offer significant improvements in classifier performance on both simulated and real observations, although it is currently highly computationally intensive. Despite the fact that it appears we hit the limit of what is currently possible with the training data and optimiser we used, future advances in model optimisation might help make deep learning analyses competitive with the current state-of-the-art techniques. Part of the difficulty of applying deep learning techniques to VERITAS data was the rigidity of the analysis chain. The current development plan for *ctapipe* is to provide a plugin framework for alternative event reconstruction and classification methods, such a plugin framework might be able to overcome the issues with using real observations as test data that we encountered. Finally in Chapter 5, we have demonstrated one area of improvement in NSB simulation (over *sim_telarray*) that might aid deep learning classifiers in the future, and provided NSB rate estimates useful for the design of SSTCAM. This work appears to demonstrate that SSTCAM will meet most of the associated NSB requirements, even under some extreme observing conditions. That said, some small fraction of pixels may still need to be disabled and removed from the Cherenkov analysis if they fall under the light from a bright star whilst observing during bright moonlight.

From the results in this thesis, it appears clear that there is still much work to be done in regards to determining the ideal training datasets (in terms of simulation setup, image cleaning and event selection cuts) to be used for deep-learning-based

event classification, let alone full event reconstruction (where this is performed simultaneously with directional and energy reconstruction). Comparing our results between SSTCAM and VERITAS suggests that among the three CTA instrument classes (which will share a similar analysis structure) there may need to be differing analyses used per-class due to the potential effects of differing background spectra, different timing accuracies, and different plate scales/camera architectures. There is a reasonable argument that we should approach each of the three reconstruction tasks separately in order to definitively associate improvements in sensitivity with a particular analysis technique for a particular analysis task. Nonetheless, the potential for a stereoscopic, full-image-based, fast analysis for IACT arrays using deep learning remains a tantalising prospect that will without doubt remain a significant area of research for several years to come. Given the rapid advance of deep learning research, it is entirely feasible that analysis methods more robust to instrument noise and NSB than CNNs may become available in the near future, and our experience with CNN-type methods will provide invaluable lessons about how to approach such tasks. But we must be careful not to mistake deep learning model complexity for physical understanding of our instruments and the associated particle astrophysics in the meantime.

6.2 Future Prospects

To conclude, we will explore the potential avenues of further investigation that could be performed following the work in this thesis.

6.2.1 Bayesian Neural Networks and Probability Maps

As mentioned in Chapters 2 and 4, the current classification scores for individual events from CNN-type methods are not necessarily meaningful on their own. Neural network classifiers are subject to both epistemic uncertainty (uncertainty associated with data quality) and aleatoric uncertainty (associated with network models being imperfect). Research utilising Bayesian deep learning is a growing field in astronomy, but is more limited in astroparticle physics, with only preliminary conference paper

on the topic existing [205]. Iterative dropout methods are one such means of estimating deep-learning classifier uncertainty; they approximate an ensemble of different networks by leaving neurons free to drop out during testing (normally this is only done during training and the dropped-out neurons are frozen for testing). Some work has been performed on the subject of implementing 'Bayesian' iterative dropout methods for ConvLSTM architectures [206], but this work is at an early stage. Stable, open source implementations for this are not available yet. Our own experiments to implement these for ConvLSTM architectures for CHEC-S data produced classifier results no better than random-guessing, even on simulated test data.

One potential option that has seen use in certain Zooniverse projects [113], and would be significantly simpler to implement, is to simply train multiple classifiers with different starting seeds and then average the output scores for given events. With greater available GPU power this might become feasible in the near future for CTA.

Another option for directional and energy reconstruction has recently been proposed by Borquez et al. [in prep.]. The suggested technique is to perform upsampling on the output of a CNN with a dense backend to produce one or two dimensional probability maps for the task in question. These can be used to generate an stereoscopic uncertainty map for the position and energy of an event, however such techniques appear to currently be computationally costly and difficult to perform at scales necessary for CTA.

6.2.2 Data Augmentation

Data augmentation has been proposed [207] as a method of forcing neural networks to generalise to wider ranges of experimental conditions. However, in the specific case of stereoscopic IACT analysis, we posit that it's unlikely to help. One common means of performing such augmentation to improve robustness to noise is data augmentation with Gaussian noise [208], but as we have seen in Chapter 5 the NSB observed by IACT cameras is not Gaussian. One potential option would be to augment advanced NSB analysis, like that in Chapter 5, into training data generated

from *CORSIKA/sim_telarray*; given an NSB rate in Hz it is possible to compute the number of photoelectrons that would be detected in a ~ 96 ns observing window. But performing such an augmentation to give an accurate pixel-wise prediction (to the sensitivity level of a CNN-type method) for NSB rates would likely be highly challenging. This would involve significant computational cost, a result of the sensitivity of SiPMs to dim stars (requiring many stars to be modelled) and the potentially rapid shifts in NSB over time. Similarly, augmenting the data by rotating images (which is a way of creating robustness to rotation [94]) may not work in practice given the complex geometry of an IACT array.

6.2.3 Advanced Model Selection Methods

It should be noted that the Bayesian Optimisation presented in Chapter 4 is a comparatively simple method of model optimisation. In industry, more advanced methods of model selection are becoming available, such as evolutionary algorithms [209] (which simulate genetic mutation) and neural architecture searches [210]. These methods are slowly becoming more widely used. But, in general, such methods where the entire model architecture can be optimised (and not just hyperparameters) are extremely computationally intensive, even more so than the work in Chapter 4. At this level of optimisation complexity, end users such as the CTA collaboration cannot reasonably acquire sufficient compute power to perform these analyses. This might change in the coming years.

6.2.4 GAN-Based Simulations and Domain Adaptation

There is a wide body of computer science literature on using different variants of GAN (such as SimGAN [164] and cyclegan [211]) as a method of handling domain shifts (such as those between simulated and real observations). These typically rely on enforcing cyclic consistency between performance on different datasets. However, our own investigations of applying such methods to CHEC-S data, whereby we attempted to evaluate performance on a test dataset that had minor discrepancies (such as zeroed pixels) applied to it, were unsuccessful (it was better not to use

the GANs whatsoever). This might be a future source of potential methods to handle the real observations problem, but we believe it's unlikely. We found the difficulty of adapting GANs to different datasets, the complexity of training them, the difficulty interpreting their behaviour, as well as the additional computational cost, make these methods likely unfeasible. Another issue is the lack of a consistent framework to apply such methods in practice. Whilst there have been attempts at creating such a framework (the best of which is *Keras-GAN* [212]), there are simply too many alternative GAN models to explore every option in every use case.

Alternatively, GANs can be used as a fast pseudo-simulation method. Whilst this is unlikely to be useful for replacing *CORSIKA/sim_telarray* for simulating EAS (our attempts at this suffered from so called 'mode-collapse', whereby similar images are repeatedly generated), one potentially useful use of this (proportionately fast) technique would be as a simulator of NSB (using training data similar to that in Chapter 5). This is as GANs appear reasonably reliable at pseudo-simulation of noise (see for example results on CMB data in [213]); this might prove a faster means of NSB data augmentation than directly using *nsb* (as suggested earlier in this chapter).

6.2.5 Graph Networks

Given the results in Chapter 4, it seems likely that current CNN-based methods of event classification reliant on integrated charge data (and not utilising timing information as in Chapter 3) will in practice be limited by NSB and instrumental noise unless tailcut cleaning is applied.

In work outside this thesis, we recently proposed that an alternative to using new timing based methods with CNN-type methods would be using alternative deep learning classifiers with charge data. One such method is graph-based Chebyshev networks, where the image data is treated as a 2D connected graph and not a 2D Euclidean array. One advantage of this is that it could potentially skip the image mapping step needed for CNN-type analyses, though recent results from A. Jacobson [84] suggest that these methods are currently computationally prohibitive, being too VRAM hungry and slow to be of practical use. There is also no current means

of performing stereoscopic analysis with such techniques. This might change in upcoming years, with research into graph-based methods being currently extremely active to the extent that it leads to advances in GPU technology [214].

6.2.6 Transfer Learning

One potential solution to the hyperparameter problem is the option to take pre-trained and optimised classifiers such as ResNet50 [215] (which is trained on the standard computer science dataset ImageNet [216] consisting of images of everyday objects), and then slightly retrain the final weights in such a network on IACT data. This is similar to the approach used in a recent paper by Miener et al. [101]. This is very exciting work, and represents the first event classification results at realistic CTA scales. But techniques to adapt transfer learning methods to stereoscopic IACT analysis are in their relative infancy. Currently they don't appear to offer large improvements in effective area relative to BDT analysis as a result (on simulated data). We speculate that this is possibly because the weights in the network are currently frozen prior to the sequence analysis stage (this issue would be solvable with increased GPU power). That said, there are new deep-transfer-learning techniques (such as *Perceiver* models [217]) being developed by computer scientists for multi-modal data that may be suitable for IACT analysis that could be used 'out the box'. Development of such multi-model methods that support flexibility in the type and shapes of the input data will likely be highly useful in future work in the physical sciences.

6.2.7 Larger Arrays

Much of the work presented in Chapters 3 and 4 was GPU resource limited, meaning that using simulated arrays truly representative of the SST component of CTA were not possible. The Miener et al. [101] paper has shown the first CNN-based results from an array of 40 SSTs using the *CTLearn* framework, but this required significant time on the Wilkes-2 GPU cluster at Cambridge to perform. Optimising such networks beyond what has already been performed would be extremely difficult

and computationally intensive, but the power of future GPU clusters may make this tractable. But this still does not solve the problem with simultaneously utilising multiple telescope classes with Shilon-like deep learning event classification methods.

In the long term, even if CNN-type methods can only be demonstrated to offer increased background rejection power for isolated point sources and large arrays, this could aid in providing a sufficient signal-to-noise ratio to perform spectral analysis on short timescale observations. This might be useful for the observation of short GRBs from the ground.

6.2.8 FPGA Analysis and Photon Stream Analysis

FACT have recently presented results showing the potential use of Field Programmable Gate Arrays to enhance the speed of deep-learning-based event classification at inference time [218]. Whilst novel and interesting, the fact using FPGAs innately limits the analysis to monoscopic reconstruction limits the current potential benefits for sensitivity (since model complexity constraints are enforced when performing such analyses on such simple hardware). However, such FPGA-based methods may be an interesting avenue of investigation in the long term, such as in their use for muon tagging. This mono-analysis FACT paper also used timing information in a similar way to our CHEC methods to aid in event classification (albeit in a more computationally costly way than use in Chapter 3).

One promising avenue of investigation for SSTCAM is to implement the 'photon stream' analysis used by FACT [44]; this uses an iterative cross-correlation procedure to extract single photons from their SiPM waveforms, before applying DBSCAN clustering analysis to extract the Cherenkov light. This could lead to improved analysis methods for SSTCAM and might make transfer of such data from Cerro Paranal feasible given the resulting compression factor. This transfer probably wouldn't feasible for full waveforms as the data rate would hit 1Gb/s [119] if they were to be transferred off-site.

6.2.9 Upgrading Conventional Analysis

One largely unexplored area of IACT analysis is taking optimisation methods designed for deep learning analyses and applying them to conventional Hillas-parameter-based analysis. In particular, Bayesian optimisation (and more generally the field of AutoML, particularly packages such as *auto-sklearn* [180]) could be used to optimise Hillas parameter cuts and BDT/RF hyperparameters to achieve improved performance. Additionally, recent developments with the RAPIDS [219] library mean GPUs can be used to accelerate BDT and RF training, a useful tool given the data scales of CTA [119]. This is an area of research that should be prioritised, as it could lead to immediate, computationally feasible performance improvements for both the current generation of IACTs and CTA.

Appendices

A

VERITAS Cut Description

This appendix describes the VERITAS cut file used to generate the real observational results presented in Chapter 4. Essentially all bar the arraytheta2, arraymscw and arraymscl lines describe array selections that will accept all events in line with standard VERITAS trigger conditions. The values for arraytheta2 describe the θ^2 cut previously discussed. Finally, the arraymscl and arraymscw are cuts on the MSCW and MSCL that very loosely reject obviously hadronic events based on their widths.

Listing A.1: Cut Description Used for the real observations Analysis in Eventdisplay format

```
cut values for quality and gamma hadron cuts
=============================================================

* cutselection 70 0
* arraychi2 0. 1.e30
* arraycore -1. 350.
* cameraedge -1. 2.00
* sizesecondmax 200. 1.e30 V4
* sizesecondmax 200. 1.e30 V5
* sizesecondmax 200. 1.e30 V6
* emissionheight 0. 1.e10
* arrayltrig 0 0
* arrayltrig 1 0
* arrayltrig 2 0
* arrayltrig 3 1
* arrayltrig 4 0
* arrayltrig 5 0 V4 1234
* arrayltrig 5 1 V5
* arrayltrig 5 1 V6
* arrayltrig 6 1 V4
* arrayltrig 6 0 V5 1234
* arrayltrig 6 0 V6 1234
* arrayltrig 7 1
* arrayltrig 8 0
* arrayltrig 9 0 V4
* arrayltrig 9 0 V5 1234
* arrayltrig 9 0 V6 1234
* arrayltrig 10 0 V4 1234
* arrayltrig 10 1 V5
* arrayltrig 10 1 V6
* arrayltrig 11 1
* arrayltrig 12 1
* arrayltrig 13 1
* arrayltrig 14 1
* arrayltrig 15 1
* arraytheta2 -1. 0.008
* arraymscw -2. 2
* arraymscl -2 5
* arrayechi2 -99999. 99999.
* arraydE -99999. 99999.
* arrayerec 0. 1.e10
```

B

Summary NSB Statistics

This appendix briefly summarizes the results presented in Chapter 5 for easy reference.

Observation	Mean (MHz)	σ (MHz)	Maximum (MHz)	Minimum (MHz)
Dark 'Empty' Field	43.0	21.3	419.3	19.7
Eta Carinae No Moonlight	76.0	63.5	1048.1	35.0
Eta Carinae Half Moonlight	79.2	65.0	1158.7	31.2
Eta Carinae Full Moonlight	1043.3	77.1	2314.8	968.0

Table B.1: NSB rate statistics in MHz per pixel for the four Eta Carinae observing runs.

Observation	Mean (MHz)	σ (MHz)	Maximum (MHz)	Minimum (MHz)
Dark 'Empty' Field	43.0	11.7	148.3	28.1
Eta Carinae No Moonlight	76.0	37.4	398.0	40.1
Eta Carinae Half Moonlight	79.2	42.6	517.0	42.3
Eta Carinae Full Moonlight	1043.3	44.3	1371.5	972.0

Table B.2: Mean NSB rate statistics in MHz per superpixel for the four Eta Carinae observing runs.

Observation	Mean (MHz)	σ (MHz)	Maximum (MHz)	Minimum (MHz)
Dark 'Empty' Field	43.0	3.5	50.9	36.5
Eta Carinae No Moonlight	76.0	16.5	117.6	51.8
Eta Carinae Half Moonlight	79.2	16.6	133.9	56.2
Eta Carinae Full Moonlight	1043.3	22.9	1091.9	978.1

Table B.3: NSB rate statistics in MHz, now taken over mean values per TM for the four Eta Carinae observing runs.

Observation	Mean (MHz)	σ (MHz)	Maximum (MHz)	Minimum (MHz)
Dark 'Empty' Field	28.7	34.2	385.1	1.4
Eta Carinae No Moonlight	61.4	113.2	992.0	2.2
Eta Carinae Half Moonlight	60.4	116.5	1105.1	2.3
Eta Carinae Full Moonlight	99.4	216.6	1294.8	0.6

Table B.4: Maximum change in NSB rate in MHz between adjacent pixels for the four Eta Carinae observing runs.

Observation	Mean (MHz)	σ (MHz)	Maximum (MHz)	Minimum (MHz)
Dark 'Empty' Field	19.9	16.7	114.6	1.0
Eta Carinae No Moonlight	47.3	56.4	343.4	2.0
Eta Carinae Half Moonlight	51.0	70.7	453.8	1.6
Eta Carinae Full Moonlight	116.3	244.4	1121.7	1.5

Table B.5: Maximum change in NSB rate in MHz between adjacent superpixels, taken over mean values per superpixel for the four Eta Carinae observing runs.

Observation	Mean (MHz)	σ (MHz)	Maximum (MHz)	Minimum (MHz)
Dark 'Empty' Field	14.9	15.8	47.0	2.5
Eta Carinae No Moonlight	35.5	19.5	85.9	10.9
Eta Carinae Half Moonlight	36.7	19.5	74.9	9.1
Eta Carinae Full Moonlight	280.2	433.2	1069.7	11.9

Table B.6: Maximum change in NSB rate in MHz between adjacent TMs, taken over mean values per TM for the four Eta Carinae observing runs.

Observation	Mean (MHz)	σ (MHz)	Maximum (MHz)	Minimum (MHz)
Dark 'Empty' Field	-1.3	31.5	376.3	-413.2
Eta Carinae No Moonlight	0.3	86.8	995.2	-1037.9
Eta Carinae Half Moonlight	-2.4	93.6	1101.3	-1089.4
Eta Carinae Full Moonlight	-941.2	104.0	118.6	-2065.0

Table B.7: Change in mean NSB rate per pixel over 30 minutes for the four Eta Carinae observing runs.

Observation	Mean (MHz)	σ (MHz)	Maximum (MHz)	Minimum (MHz)
Dark 'Empty' Field	-1.1	15.9	109.9	-107.8
Eta Carinae No Moonlight	0.3	46.1	340.1	-298.6
Eta Carinae Half Moonlight	-2.1	54.0	430.7	-536.3
Eta Carinae Full Moonlight	-836.6	302.4	0.0	-1572.2

Table B.8: Change in mean NSB rate per superpixel over 30 minutes for the four Eta Carinae observing runs.

Observation	Mean (MHz)	σ (MHz)	Maximum (MHz)	Minimum (MHz)
Dark 'Empty' Field	-1.1	3.5	5.7	-12.4
Eta Carinae No Moonlight	0.3	15.2	36.0	-30.4
Eta Carinae Half Moonlight	-2.1	9.6	22.3	-21.5
Eta Carinae Full Moonlight	-836.6	296.6	0.0	-1000.8

Table B.9: Change in mean NSB rate per TM over 30 minutes for the four Eta Carinae observing runs.

References

[1] B. Acharya et al. *Science with the Cherenkov Telescope Array*. WORLD SCIENTIFIC, Feb. 2018. URL: http://dx.doi.org/10.1142/10986.

[2] J. Holder. *Atmospheric Cherenkov Gamma-ray Telescopes*. Preprint, https://arxiv.org/pdf/1510.05675.pdf. 2015.

[3] P. Morrison. "On gamma-ray astronomy". In: *Il Nuovo Cimento* 7.6 (Mar. 1958), pp. 858–865.

[4] W. Kraushaar et al. "Search for Primary Cosmic Gamma Rays with the Satellite Explorer XI". In: *Phys. Rev. Lett.* 8 (3 Feb. 1962), pp. 106–109. URL: https://link.aps.org/doi/10.1103/PhysRevLett.8.106.

[5] W. Kraushaar et al. "High-Energy Cosmic Gamma-Ray Observations from the OSO-3 Satellite". In: *apj* 177 (Nov. 1972), p. 341.

[6] C. Fichtel et al. "High-energy gamma-ray results from the second Small Astronomy Satellite." In: *apj* 198 (May 1975), pp. 163–182.

[7] C. Cesarsky et al. "Gamma ray astronomy and the COS-B satellite." In: *Endeavour* 8.1 (Jan. 1984), pp. 9–16.

[8] P. Dyal. "Particle and field measurements of the Starfish diamagnetic cavity". In: *Journal of Geophysical Research: Space Physics* 111.A12 (2006).

[9] R. Klebesadel et al. "Observations of Gamma-Ray Bursts of Cosmic Origin". In: *apjl* 182 (June 1973), p. L85.

[10] N. Gehrels et al. "The Compton Gamma Ray Observatory: mission status." In: *The Second Compton Symposium*. Vol. 304. American Institute of Physics Conference Series. Jan. 1994, pp. 3–11.

[11] MAGIC Collaboration. "Teraelectronvolt emission from the gamma-ray burst GRB 190114C". In: *Nature* 575.7783 (Nov. 2019), pp. 455–458. arXiv: 2006.07249 [astro-ph.HE].

[12] D. Hooper et al. *Two Emission Mechanisms in the Fermi Bubbles: A Possible Signal of Annihilating Dark Matter*. 2013. arXiv: 1302.6589 [astro-ph.HE].

[13] B. Abbott et al. "Multi-messenger Observations of a Binary Neutron Star Merger". In: *apjl* 848.2, L12 (Oct. 2017), p. L12. arXiv: 1710.05833 [astro-ph.HE].

[14] W. Galbraith et al. "Light Pulses from the Night Sky associated with Cosmic Rays". In: *Nature* 171 (1953), pp. 349–350.

[15] A. Abeysekara et al. "Sensitivity of the high altitude water Cherenkov detector to sources of multi-TeV gamma rays". In: *Astroparticle Physics* 50-52 (Dec. 2013), pp. 26–32. URL: http://dx.doi.org/10.1016/j.astropartphys.2013.08.002.

[16] A. Hillas et al. “The Spectrum of TeV Gamma Rays from the Crab Nebula”. In: *The Astrophysical Journal* 503.2 (Aug. 1998), pp. 744–759. URL: https://doi.org/10.1086/306005.

[17] F. Aharonian et al. “Evidence for TeV gamma ray emission from Cassiopeia A”. In: *aap* 370 (Apr. 2001), pp. 112–120. arXiv: astro-ph/0102391 [astro-ph].

[18] M. Mori. “CANGAROO-II and CANGAROO-III”. In: *AIP Conference Proceedings* (2001). URL: http://dx.doi.org/10.1063/1.1419521.

[19] P. Armstrong et al. *The University of Durham Mark 6 Gamma Ray Telescope*. Vol. 9. 2. Springer Science and Business Media LLC, 1999, pp. 51–80. URL: http://dx.doi.org/10.1023/A:1008018503904.

[20] J. Hinton. *Gamma-ray Astronomy*. 2007. arXiv: 0712.3352 [astro-ph].

[21] J. Jelley. “Čerenkov radiation: Its origin, properties and applications”. In: *Contemporary Physics* 3.1 (1961), pp. 45–57. URL: https://doi.org/10.1080/00107516108204445.

[22] I. Frank et al. “Coherent Visible Radiation of Fast Electrons Passing Through Matter”. In: *Selected Papers*. Ed. by Boris M. Bolotovskii, Victor Ya. Frenkel, and Rudolf Peierls. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 29–35. URL: https://doi.org/10.1007/978-3-642-74626-0_2.

[23] U. Katz. “Cherenkov light imaging in astroparticle physics”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 952 (Feb. 2020), p. 161654. URL: http://dx.doi.org/10.1016/j.nima.2018.11.113.

[24] T. Weekes. *Very high energy gamma-ray astronomy*. 2003.

[25] J. Sitarek et al. “Estimation of the height of the first interaction in gamma-ray showers observed by Cherenkov telescopes”. In: *Astroparticle Physics* 103 (Dec. 2018), pp. 108–114.

[26] J. Matthews. “A Heitler model of extensive air showers”. In: *Astropart. Phys.* 22 (2005), pp. 387–397.

[27] W. Heitler. *The Quantum Theory of Radiation*. Dover Books on Physics Series. Dover Publications, 1984. URL: https://books.google.co.uk/books?id=8jVRAAAAMAAJ.

[28] A. Abeysekara et al. “VERITAS Observations of the BL Lac Object TXS 0506+056”. In: *The Astrophysical Journal* 861.2 (July 2018), p. L20. URL: http://dx.doi.org/10.3847/2041-8213/aad053.

[29] A. Abeysekara et al. “Gamma-rays from the Quasar PKS 1441+25: Story of an Escape”. In: *The Astrophysical Journal* 815.2 (Dec. 2015), p. L22. URL: http://dx.doi.org/10.1088/2041-8205/815/2/L22.

[30] S. Klepser et al. “Hardware and software architecture of the upgraded H.E.S.S. cameras”. In: *Proceedings of 35th International Cosmic Ray Conference — PoS(ICRC2017)*. Sissa Medialab, Aug. 2017. URL: http://dx.doi.org/10.22323/1.301.0834.

[31] M. Holler et al. “Advanced search for the extension of unresolved TeV sources with H.E.S.S.: First measurement of the extension of the Crab nebula at TeV energies”. In: (2017). arXiv: 1707.04196 [astro-ph.HE].

[32] H. Abdalla et al. "Resolving acceleration to very high energies along the jet of Centaurus A". In: *Nature* 582.7812 (June 2020), pp. 356–359. URL: http://dx.doi.org/10.1038/s41586-020-2354-1.

[33] F. Temme et al. "FACT - First Energy Spectrum from a SiPM Cherenkov Telescope". In: *PoS* ICRC2015 (2016), p. 707.

[34] R. Brun et al. "ROOT An object oriented data analysis framework". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 389.1 (1997). New Computing Techniques in Physics Research V, pp. 81–86. URL: http://www.sciencedirect.com/science/article/pii/S016890029700048X.

[35] M. Amenomori et al. "First Detection of Photons with Energy beyond 100 TeV from an Astrophysical Source". In: *Phys. Rev. Lett.* 123 (5 July 2019), p. 051101. URL: https://link.aps.org/doi/10.1103/PhysRevLett.123.051101.

[36] Z. Cao et al. "Peta–electron volt gamma-ray emission from the Crab Nebula". In: *Science* 373.6553 (July 2021), pp. 425–430. URL: http://dx.doi.org/10.1126/science.abg5137.

[37] W. Atwood et al. "The Large Area Telescope on the Fermi Gamma-Ray Space Telescope Mission". In: *ApJ* 697.2 (June 2009), pp. 1071–1102.

[38] G. Maier et al. *Performance of the Cherenkov Telescope Array*. 2017. arXiv: 1709.01381 [astro-ph.IM].

[39] T. Hassan et al. "Monte Carlo performance studies for the site selection of the Cherenkov Telescope Array". In: *Astroparticle Physics* 93 (2017), pp. 76–85. URL: http://www.sciencedirect.com/science/article/pii/S0927650517300087.

[40] Cherenkov Telescope Array Observatory and Cherenkov Telescope Array Consortium. *CTAO Instrument Response Functions - prod5 version v0.1*. Version v0.1. Zenodo, Sept. 2021. URL: https://doi.org/10.5281/zenodo.5499840.

[41] R. White et al. "A Compact High Energy Camera (CHEC) for the Gamma-ray Cherenkov Telescope of the Cherenkov Telescope Array". In: *ICRC2017* (2017).

[42] T. Armstrong. "Monte Carlo studies of the GCT for the Cherenkov Telescope Array and the search for VHE AGN using cluster analysis". PhD thesis. University of Durham, 2016.

[43] J. Zorn et al. "Characterisation and testing of CHEC-M-A camera prototype for the small-sized telescopes of the Cherenkov telescope array". In: *Nucl. Instrum. Meth.* A904 (2018), pp. 44–63.

[44] S. Mueller et al. "Single Photon Extraction for FACT's SiPMs allows for Novel IACT Event Representation". In: 2017.

[45] R. White. "The Small-Sized Telescopes for the Southern Site of the Cherenkov Telescope Array". In: *PoS* ICRC2021 (2021), p. 728.

[46] L. Fortson. *Private Communication*. 2022.

[47] W. Benbow et al. "The H.E.S.S. Standard Analysis Technique". In: *Proc. Conf. Towards a Network of Atmospheric Cherenkov Detectors VII* (2005), pp. 163–172.

[48] P. Zyla et al. "Review of Particle Physics". In: *PTEP* 2020.8 (2020), p. 083C01.

[49] K. Greisen. "End to the Cosmic-Ray Spectrum?" In: *PRL* 16.17 (Apr. 1966), pp. 748–750.

[50] P. Grieder et al. *Extensive Air Showers*. Springer, 2010.

[51] F. Schmidt et al. *Corsika Visualizations*. `https://www.ikp.kit.edu/corsika/`.

[52] I. Lypova et al. "Investigation on gamma-electron air shower separation for CTA". PhD thesis.

[53] C. Leinert et al. "The 1997 reference of diffuse night sky brightness". In: *aaps* 127 (Jan. 1998), pp. 1–99.

[54] C. Benn et al. "Brightness of the night sky over La Palma". In: *Nar* 42.6-8 (Nov. 1998), pp. 503–507. arXiv: `astro-ph/9909153 [astro-ph]`.

[55] A. Weinstein. *The VERITAS Trigger System*. 2007. arXiv: `0709.4438 [astro-ph]`.

[56] J. Watson. "Calibration and analysis of the GCT camera for the Cherenkov Telescope Array". PhD thesis. University of Oxford, 2018.

[57] A. Daum et al. "First Results on the Performance of the HEGRA IACT Array". In: *arXiv e-prints*, astro-ph/9704098 (Apr. 1997), astro–ph/9704098. arXiv: `astro-ph/9704098 [astro-ph]`.

[58] S. Lombardi et al. "Advanced stereoscopic gamma-ray shower analysis with the MAGIC telescopes". In: *Proceedings of the 32nd International Cosmic Ray Conference, ICRC 2011* 3 (Sept. 2011).

[59] S. Lombardi. "Advanced stereoscopic gamma-ray shower analysis with the MAGIC telescopes". In: *International Cosmic Ray Conference*. Vol. 3. International Cosmic Ray Conference. Jan. 2011, p. 266. arXiv: `1109.6195 [astro-ph.IM]`.

[60] R. Lessard et al. "Wavelet imaging cleaning method for atmospheric Cherenkov telescopes". In: *Astroparticle Physics* 17.4 (July 2002), pp. 427–440. arXiv: `astro-ph/0110033 [astro-ph]`.

[61] K. Kosack et al. "ctapipe: A Low-level Data Processing Framework for CTA". In: *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36. International Cosmic Ray Conference. July 2019, p. 717.

[62] P. Chadwick. "Very high energy cosmic gamma rays from radio and X-ray pulsars". PhD thesis. Durham University, 1987.

[63] T. Weekes et al. "Observation of TeV Gamma Rays from the Crab Nebula using the Atmospheric Cherenkov Imaging Technique". In: *APJ* 342 (1989), p. 379.

[64] D. Berge et al. "Background modelling in very-high-energy gamma-ray astronomy". In: *Astronomy and Astrophysics* 466.3 (Apr. 2007), pp. 1219–1229. URL: `http://dx.doi.org/10.1051/0004-6361:20066674`.

[65] S. Ohm et al. "Gamma/hadron separation in very-high-energy gamma-ray astronomy using a multivariate analysis method". In: *Astroparticle Physics* 31.5 (June 2009), pp. 383–391.

[66] G. Maier et al. "Eventdisplay: An Analysis and Reconstruction Package for Ground-based Gamma-ray Astronomy". In: *35th International Cosmic Ray Conference (ICRC2017)*. Vol. 301. International Cosmic Ray Conference. Jan. 2017, p. 747. arXiv: `1708.04048 [astro-ph.IM]`.

[67] E. Aliu et al. "Improving the performance of the single-dish Cherenkov telescope MAGIC through the use of signal timing". In: *Astroparticle Physics* 30.6 (2009), pp. 293–305. URL: http://www.sciencedirect.com/science/article/pii/S0927650508001497.

[68] M. de Naurois et al. "A high performance likelihood reconstruction of Gamma-rays for imaging atmospheric Cherenkov telescopes". In: *Astroparticle Physics* 32.5 (Dec. 2009), pp. 231–252.

[69] A. Mitchell et al. *Detection of extended TeV emission around the Geminga pulsar with H.E.S.S.* 2021. arXiv: 2108.02556 [astro-ph.HE].

[70] G. Maier. *VERITAS: Status and Latest Results.* 2007. arXiv: 0709.3654 [astro-ph].

[71] F. Acero et al. "Prospects for Cherenkov Telescope Array Observations of the Young Supernova Remnant RX J1713.7-3946". In: *The Astrophysical Journal* 840.2 (May 2017), p. 74. URL: http://dx.doi.org/10.3847/1538-4357/aa6d67.

[72] S. Le Bohec et al. "A new analysis method for very high definition Imaging Atmospheric Cherenkov Telescopes as applied to the CAT telescope". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 416.2-3 (Nov. 1998), pp. 425–437. URL: http://dx.doi.org/10.1016/S0168-9002(98)00750-5.

[73] M. Lemoine-Goumard et al. "Selection and 3D-reconstruction of Gamma-ray-induced air showers with a stereoscopic system of atmospheric Cherenkov telescopes". In: *Astroparticle Physics* 25.3 (2006), pp. 195–211.

[74] R. Parsons et al. "A Monte Carlo template based analysis for air-Cherenkov arrays". In: *Astroparticle Physics* 56 (Apr. 2014), pp. 26–34.

[75] K. Kosack et al. *ctapipe.* https://github.com/cta-observatory/ctapipe. 2021.

[76] M. Longair. *High Energy Astrophysics.* 2011.

[77] G. Blumenthal et al. "Bremsstrahlung, Synchrotron Radiation, and Compton Scattering of High-Energy Electrons Traversing Dilute Gases". In: *Rev. Mod. Phys.* 42 (2 Apr. 1970), pp. 237–270. URL: https://link.aps.org/doi/10.1103/RevModPhys.42.237.

[78] O. Klein and T. Nishina. "Über die Streuung von Strahlung durch freie Elektronen nach der neuen relativistischen Quantendynamik von Dirac". In: *Zeitschrift fur Physik* 52.11-12 (Nov. 1929), pp. 853–868.

[79] J. Aleksić et al. "The major upgrade of the MAGIC telescopes, Part II: A performance study using observations of the Crab Nebula". In: *Astroparticle Physics* 72 (Jan. 2016), pp. 76–94.

[80] G. Giacinti et al. "Halo fraction in TeV-bright pulsar wind nebulae". In: *Astronomy and Astrophysics* 636 (Apr. 2020), A113. URL: http://dx.doi.org/10.1051/0004-6361/201936505.

[81] Alice K. Harding. *Gamma-ray pulsars: Models and predictions.* 2001. URL: https://doi.org/10.1063%2F1.1370785.

[82] M. Spir-Jacob et al. *Detection of sub-100 GeV gamma-ray pulsations from PSR B1706-44 with H.E.S.S.* 2019. arXiv: 1908.06464 [astro-ph.HE].

[83] M. Ajello et al. "The Origin of the Extragalactic Gamma-Ray Background and Implications For Dark Matter Annihilation". In: *The Astrophysical Journal* 800.2 (Feb. 2015), p. L27. URL: http://dx.doi.org/10.1088/2041-8205/800/2/L27.

[84] A. Jacobson. *Exploratory Analysis of the Use of ChebNets for Event Classification with the Cherenkov Telescope Array*. University of Oxford Masters Thesis. 2021.

[85] Defferrard M. et al. *Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering*. 2016. arXiv: 1606.09375 [cs.LG].

[86] J. Clark. *Google research chief: 'Emergent artificial intelligence? Hogwash!'* https://www.theregister.com/2013/05/17/google_ai_hogwash/. Accessed: 2021-10-20.

[87] G. Wilson et al. *A Survey of Unsupervised Deep Domain Adaptation*. 2020. arXiv: 1812.02849 [cs.LG].

[88] I. Goodfellow et al. *Deep Learning*. Adaptive Computation and Machine Learning series. MIT Press, 2016. URL: https://books.google.co.uk/books?id=Np9SDQAAQBAJ.

[89] K. Albertsson et al. "Machine Learning in High Energy Physics Community White Paper". In: *arXiv e-prints* (July 2018). arXiv: 1807.02876 [physics.comp-ph].

[90] W. Rawat et al. "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review". In: *Neural Computation* 29.9 (2017), pp. 2352–2449.

[91] A. Pandya et al. *Pattern Recognition with Neural Networks in C++*. CRC Press, 1996.

[92] A. Aniyan et al. "Classifying Radio Galaxies with the Convolutional Neural Network". In: *The Astrophysical Journal Supplement Series* 230.2 (June 2017), p. 20. URL: http://dx.doi.org/10.3847/1538-4365/aa7333.

[93] T. Ryuu. *Introduction to Multilayer Neural Networks with TensorFlow's Keras API - Learn how to build and train a multilayer perceptron using TensorFlow's high-level API Keras*. https://morioh.com/p/b988929cd561. Accessed: 2021-09-28.

[94] F. Chollet et al. *Keras*. https://keras.io. 2015.

[95] D. Mahajan et al. "Exploring the Limits of Weakly Supervised Pretraining". In: *Lecture Notes in Computer Science* (2018), pp. 185–201. URL: http://dx.doi.org/10.1007/978-3-030-01216-8_12.

[96] T. Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006). ROC Analysis in Pattern Recognition, pp. 861–874. URL: http://www.sciencedirect.com/science/article/pii/S016786550500303X.

[97] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[98] Rupesh K Srivastava, Klaus Greff, and Jurgen Schmidhuber. "Training very deep networks". In: *Advances in neural information processing systems* 28 (2015).

[99] K. Hornik et al. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (1989), pp. 359–366. URL: https://www.sciencedirect.com/science/article/pii/0893608089900208.

[100] M. Jacquemont. "Cherenkov Image Analysis with Deep Multi-Task Learning from Single-Telescope Data". PhD thesis. Annecy, LAPP, LIRIS, Lyon, 2020.

[101] T. Meiner et al. "Reconstruction of stereoscopic CTA events using deep learning with CTLearn". In: *PoS* ICRC2021 (2021), p. 730.

[102] K. Fukushima et al. "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position". In: *Pattern Recognition* 15.6 (1982), pp. 455–469. URL: https://www.sciencedirect.com/science/article/pii/0031320382900243.

[103] D. Wright et al. "Multi-column deep neural networks for image classification". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), pp. 3642–3649.

[104] S. Patel et al. *Introduction to Deep Learning: What Are Convolutional Neural Networks?* https://uk.mathworks.com.

[105] S. Hochreiter et al. "Long Short-Term Memory". In: *Neural Computation* 9 (8 1997), pp. 1735–1780.

[106] A. Xavier. *An introduction to ConvLSTM*. https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7. Accessed: 2021-06-28.

[107] I. Shilon et al. "Application of deep learning methods to analysis of imaging atmospheric Cherenkov telescopes data". In: *Astroparticle Physics* 105 (Feb. 2019), pp. 44–53.

[108] Ashish Vaswani et al. *Attention is all you need*. 2017.

[109] J. Bergstra et al. "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA, USA: JMLR.org, 2013, pp. I-115–I-123. URL: http://dl.acm.org/citation.cfm?id=3042817.3042832.

[110] M. Zeiler. "ADADELTA: An Adaptive Learning Rate Method". In: *CoRR* abs/1212.5701 (2012). URL: http://arxiv.org/abs/1212.5701.

[111] D. Kingma et al. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[112] B. Chen. *Early Stopping in Practice: an example with Keras and TensorFlow 2.0*. https://towardsdatascience.com/a-practical-introduction-to-early-stopping-in-machine-learning-550ac88bc8fd. Accessed: 2021-07-28.

[113] M. Walmsley et al. "Galaxy Zoo: probabilistic morphology through Bayesian CNNs and active learning". In: *Monthly Notices of the Royal Astronomical Society* 491.2 (Oct. 2019), pp. 1554–1574.

[114] Y. Gal. *Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference*. 2015. arXiv: 1506.02158 [stat.ML].

[115] J. Brownlee. *A Gentle Introduction to Batch Normalization for Deep Neural Networks*. https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/. Accessed: 2022-03-17.

[116] Q. Feng et al. "The analysis of VERITAS muon images using convolutional neural networks". In: *Proceedings of the International Astronomical Union* 12.S325 (Oct. 2016), pp. 173–179. URL: http://dx.doi.org/10.1017/S1743921316012734.

[117] M. Laraia et al. "Muon Hunter 2.0: efficient crowdsourcing of labels for IACT image analysis". In: *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36. International Cosmic Ray Conference. July 2019, p. 678.

[118] L. Fortson. *Private Communication*. 2019.

[119] M.P. Arribas et al. for the CTA Consortium. "Trigger and data rates expected for the CTA observatory". In: *AIP Conference Proceedings*. AIP, 2012. URL: http://dx.doi.org/10.1063/1.4772376.

[120] R. Clark. *Categorising simulated very-high-energy events from the Cherenkov Telescope Array using Machine Learning*. University of Oxford Masters Thesis. 2020.

[121] R. Parsons et al. "Background rejection in atmospheric Cherenkov telescopes using recurrent convolutional neural networks". In: *European Physical Journal C* 80.5, 363 (May 2020), p. 363. arXiv: 1910.09435 [astro-ph.IM].

[122] T. Li et al. "Analysis methods for results in gamma-ray astronomy." In: *apj* 272 (Sept. 1983), pp. 317–324.

[123] A. Brill et al. "Investigating a Deep Learning Method to Analyze Images from Multiple Gamma-ray Telescopes". In: *2019 New York Scientific Data Summit (NYSDS)* (June 2019). URL: http://dx.doi.org/10.1109/NYSDS.2019.8909697.

[124] S. Mangano et al. "Artificial Neural Networks in Pattern Recognition". In: *Lecture Notes in Computer Science* (2018). URL: http://dx.doi.org/10.1007/978-3-319-99978-4.

[125] S. Mangano et al. "Extracting gamma-ray information from images with convolutional neural network methods on simulated Cherenkov Telescope Array data". In: *Artificial Neural Networks in Pattern Recognition, Lecture Notes in Computer Science* (2018). URL: http://dx.doi.org/10.1007/978-3-319-99978-4.

[126] S. Koppula et al. *A Deep Learning Approach for Characterizing Major Galaxy Mergers*. 2021. arXiv: 2102.05182 [astro-ph.GA].

[127] T. Holch et al. "Probing Convolutional Neural Networks for Event Reconstruction in Gamma-Ray Astronomy with Cherenkov Telescopes". In: *ICRC2017* (2017).

[128] D. Wright et al. "A transient search using combined human and machine classifications". In: *MNRAS* 472 (2 2017), pp. 1315–1323.

[129] M. Ohishi et al. "Effect of the uncertainty in the hadronic interaction models on the estimation of the sensitivity of the Cherenkov telescope array". In: *Journal of Physics G: Nuclear and Particle Physics* 48.7 (June 2021), p. 075201. URL: http://dx.doi.org/10.1088/1361-6471/abfce0.

[130] M. Erdmann et al. *Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks*. 2018. arXiv: 1802.03325 [astro-ph.IM].

[131] L. Oakes et al. *Combined Dark Matter searches towards dwarf spheroidal galaxies with Fermi-LAT, HAWC, HESS, MAGIC and VERITAS*. 2019. arXiv: 1909.06310 [astro-ph.HE].

[132] E. Kain. *Bad News: Graphics Card Prices Are Skyrocketing And There's No End In Sight*. https://www.forbes.com/sites/erikkain/2021/01/06/bad-news-graphics-card-prices-are-skyrocketing-and-theres-no-end-in-sight/?sh=169aff9e5594. Accessed: 2021-06-28.

[133] D. Molloy. *The great graphics card shortage of 2020 (and 2021)*. https://www.bbc.co.uk/news/technology-55755820. Accessed: 2021-06-28.

[134] NVidia. *Nvidia 2080Ti User Guide*. https://www.nvidia.com/content/geforce-gtx/GEFORCE_RTX_2080Ti_User_Guide.pdf. Accessed: 2021-06-28.

[135] A. Nur et al. *The De-democratization of AI: Deep Learning and the Compute Divide in Artificial Intelligence Research*. 2020. arXiv: 2010.15581 [cs.CY].

[136] D. Nieto et al. *Exploring deep learning as an event classification method for the Cherenkov Telescope Array*. 2017. arXiv: 1709.05889 [astro-ph.IM].

[137] J. Holder. "Exploiting VERITAS Timing Information". In: *29th International Cosmic Ray Conference (ICRC29), Volume 5*. Vol. 5. International Cosmic Ray Conference. Jan. 2005, p. 383. arXiv: astro-ph/0507450 [astro-ph].

[138] M. Erdmann et al. "A deep learning-based reconstruction of cosmic ray-induced air showers". In: *Astroparticle Physics* 97 (Jan. 2018), pp. 46–53.

[139] D. Shipilov et al. "Signal recognition and background suppression by matched filters and neural networks for Tunka-Rex". In: *arXiv e-prints* (Dec. 2018). URL: https://doi.org/10.1051/epjconf/201921602003.

[140] M. Huennefeld. "Deep Learning in Physics exemplified by the Reconstruction of Muon-Neutrino Events in IceCube". In: *PoS* ICRC2017 (2018), p. 1057.

[141] N. Choma et al. *Graph Neural Networks for IceCube Signal Classification*. 2018. arXiv: 1809.06166 [cs.LG].

[142] D. Heck et al. "CORSIKA: A Monte Carlo code to simulate extensive air showers". In: *FZKA* (Feb. 1998).

[143] K. Bernlohr. "Simulation of imaging atmospheric Cherenkov telescopes with CORSIKA and sim_telarray". In: *Astroparticle Physics* 30.3 (2008), pp. 149–158. URL: http://www.sciencedirect.com/science/article/pii/S0927650508000972.

[144] G. Maier. "Performance of the Cherenkov Telescope Array". In: *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36. International Cosmic Ray Conference. July 2019, p. 733.

[145] O. Le Blanc et al. "Final characterisation and design of the Gamma-ray Cherenkov Telescope (GCT) for the Cherenkov Telescope Array". In: *Ground-based and Airborne Telescopes VII*. Ed. by Heather K. Marshall and Jason Spyromilio. Vol. 10700. International Society for Optics and Photonics. SPIE, 2018, pp. 299–318. URL: https://doi.org/10.1117/12.2313158.

[146] S. Spencer et al. "Prospects for the Use of Photosensor Timing Information with Machine Learning Techniques in Background Rejection." In: *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36. International Cosmic Ray Conference. July 2019, p. 798.

[147] A. Brill et al. *dl1-data-handler*. `https://github.com/cta-observatory/dl1-data-handler`. 2018.

[148] M. Holler et al. "Run-Wise Simulations for Imaging Atmospheric Cherenkov Telescope Arrays". In: *35th International Cosmic Ray Conference (ICRC2017)*. Vol. 301. International Cosmic Ray Conference. Jan. 2017, p. 755. URL: `https://ui.adsabs.harvard.edu/abs/2017ICRC...35..755H`.

[149] A. Oppenheim et al. *Discrete-Time Signal Processing (2nd Ed.)* USA: Prentice-Hall, Inc., 1999.

[150] S. Lam et al. "Numba: A LLVM-based Python JIT Compiler". In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. LLVM '15. Austin, Texas: ACM, 2015, 7:1–7:6. URL: `http://doi.acm.org/10.1145/2833157.2833162`.

[151] The HDF Group. *Hierarchical Data Format, version 5*. http://www.hdfgroup.org/HDF5/. 1997.

[152] L. Arrabito et al. "DIRAC framework evaluation for the Fermi-LAT and CTA experiments". In: *Journal of Physics: Conference Series* 513.3 (June 2014), p. 032003.

[153] Martin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. `https://www.tensorflow.org`. 2015.

[154] J. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. URL: `http://doi.acm.org/10.1109/MCSE.2007.55`.

[155] X. Shi et al. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting". In: *CoRR* abs/1506.04214 (2015). URL: `http://arxiv.org/abs/1506.04214`.

[156] M. Jacquemont et al. "Indexed Operations for Non-rectangular Lattices Applied to Convolutional Neural Networks". In: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5 VISAPP: VISAPP,* INSTICC. SciTePress, 2019, pp. 362–371.

[157] P. Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272.

[158] D. Pan et al. "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching". In: *Bioinformatics* 22.17 (July 2006), pp. 2059–2065. URL: `https://doi.org/10.1093/bioinformatics/btl355`.

[159] M. Ancona et al. "Towards better understanding of gradient-based attribution methods for Deep Neural Networks". In: *arXiv e-prints*, arXiv:1711.06104 (Nov. 2017). arXiv: `1711.06104 [cs.LG]`.

[160] M. Jacquemont et al. "Single Imaging Atmospheric Cherenkov Telescope Full-Event Reconstruction with a Deep Multi-Task Learning Architecture". In: *Astronomical Data Analysis Software and Systems ADASS XXX*. Granada, Spain, Nov. 2020. URL: https://hal.archives-ouvertes.fr/hal-03043005.

[161] M. Ancona et al. "Towards better understanding of gradient-based attribution methods for Deep Neural Networks". In: *ICLR 2018 Conference Blind Submission* (2018).

[162] C. Adams et al. "Detection of the Crab Nebula with the 9.7m prototype Schwarzschild-Couder telescope". In: *Astroparticle Physics* 128 (Mar. 2021), p. 102562. URL: http://dx.doi.org/10.1016/j.astropartphys.2021.102562.

[163] S. Sankaranarayanan et al. *Learning from Synthetic Data: Addressing Domain Shift for Semantic Segmentation*. 2018. arXiv: 1711.06969 [cs.CV].

[164] A. Shrivastava. *Learning from Simulated and Unsupervised Images through Adversarial Training*. 2017. arXiv: 1612.07828 [cs.CV].

[165] R. Nagai et al. *Focal Plane Instrumentation of VERITAS*. Oct. 2007.

[166] D. Nieto et al. *Private Communication*. 2018.

[167] D. Nieto et al. "Studying deep convolutional neural networks with hexagonal lattices for imaging atmospheric Cherenkov telescope event reconstruction". In: (2019). URL: https://arxiv.org/abs/1912.09898.

[168] Francois Henault et al. "Design of light concentrators for Cherenkov telescope observatories". In: *Nonimaging Optics: Efficient Design for Illumination and Solar Concentration X* (Sept. 2013). Ed. by Roland Winston and Jeffrey Gordon. URL: http://dx.doi.org/10.1117/12.2024049.

[169] *Awesome Domain Adaptation*. https://github.com/zhaoxin94/awsome-domain-adaptation. Accessed: 2019-09-24.

[170] N. Otte et al. *CARE*. http://otte.gatech.edu/care/. Accessed: 2021-09-03.

[171] M. Krause. "Studies of the influence of the geomagnetic field on the sensitivity of gamma-ray observatories". PhD thesis. Brandenburgische Technische Universitat Cottbus, 2011.

[172] A. Hillas. "Cerenkov Light Images of EAS Produced by Primary Gamma Rays and by Nuclei". In: *19th International Cosmic Ray Conference (ICRC19), Volume 3*. Vol. 3. International Cosmic Ray Conference. Aug. 1985, p. 445.

[173] M. Holler et al. "Advanced search for the extension of unresolved TeV sources with H.E.S.S." In: *35th International Cosmic Ray Conference (ICRC2017)*. Vol. 35. International Cosmic Ray Conference. Jan. 2017, p. 676. arXiv: 1707.04196 [astro-ph.HE].

[174] N. Dawe et al. *root_numpy*. http://scikit-hep.org/root_numpy/. 2021.

[175] J. Bergstra et al. "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA, USA: JMLR.org, 2013, I–115–I–123.

[176] W. Koehrsen. *A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning*. https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f. Accessed: 2021-05-24.

[177] M. Pumperla et al. *Hyperas*. https://github.com/maxpumperla/hyperas. Accessed: 2021-05-12.

[178] A. Bouvier et al. *Prospects of GRB observations for CTA from a phenomenological model*. Sept. 2011.

[179] MongoDB Contributors. *MongoDB*. https://github.com/mongodb/mongo. 2021.

[180] automl. *auto-sklearn*. https://github.com/automl/auto-sklearn. 2021.

[181] M. Buechele et al. *nsb*. https://pypi.org/project/nsb/. 2020.

[182] M. Ahnen et al. "Performance of the MAGIC telescopes under moonlight". In: *Astroparticle Physics* 94 (Sept. 2017), pp. 29–41. URL: http://dx.doi.org/10.1016/j.astropartphys.2017.08.001.

[183] A. Nagai et al. "SiPM behaviour under continuous light". In: *Journal of Instrumentation* 14.12 (Dec. 2019), P12016–P12016. URL: http://dx.doi.org/10.1088/1748-0221/14/12/P12016.

[184] A. Asano et al. "Evaluation of silicon photomultipliers for dual-mirror Small-Sized Telescopes of Cherenkov Telescope Array". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 912 (2018). New Developments In Photodetection 2017, pp. 177–181. URL: https://www.sciencedirect.com/science/article/pii/S0168900217312032.

[185] Imen Al Samarai et al. *Calibration and operation of SiPM-based cameras for gamma-ray astronomy in presence of high night-sky light*. 2019. arXiv: 1908.06860 [astro-ph.IM].

[186] D. Guberman et al. "Silicon photomultipliers in Very High Energy gamma-ray astrophysics". In: *Journal of Instrumentation* 15.05 (May 2020), pp. C05039–C05039. URL: http://dx.doi.org/10.1088/1748-0221/15/05/C05039.

[187] K. Bernlohr. "Simulation of imaging atmospheric Cherenkov telescopes with CORSIKA and sim_telarray". In: *Astroparticle Physics* 30.3 (Oct. 2008), pp. 149–158. URL: http://dx.doi.org/10.1016/j.astropartphys.2008.07.009.

[188] K. Krisciunas et al. "A model of the brightness of moonlight". In: *Publications of the Astronomical Society of the Pacific* 103 (Sept. 1991), p. 1033. URL: https://doi.org/10.1086/132921.

[189] K. Gorski et al. "HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere". In: *The Astrophysical Journal* 622.2 (Apr. 2005), pp. 759–771. URL: http://dx.doi.org/10.1086/427976.

[190] Gaia Collaboration. "The Gaia mission". In: *aap* 595, A1 (Nov. 2016), A1. arXiv: 1609.04153 [astro-ph.IM].

[191] Gaia Collaboration. "Gaia Data Release 1. Summary of the astrometric, photometric, and survey properties". In: *aap* 595, A2 (Nov. 2016), A2. arXiv: 1609.04172 [astro-ph.IM].

[192] Jacques M. Beckers. "Estimating the effects of nitrate mining activities on the astronomical site quality of the Cerro Paranal Observatory". In: *Advanced Technology Optical Telescopes V*. Ed. by L. Stepp. Vol. 2199. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. June 1994, pp. 817–823.

[193] S. Preuss et al. "Study of the photon flux from the night sky at La Palma and Namibia, in the wavelength region relevant for imaging atmospheric Cherenkov telescopes". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 481.1-3 (Apr. 2002), pp. 229–240. URL: http://dx.doi.org/10.1016/S0168-9002(01)01264-5.

[194] J. Tregloan-Reed et al. "First observations and magnitude measurement of Starlink's Darksat". In: *aap* 637, L1 (May 2020), p. L1. arXiv: 2003.07251 [astro-ph.IM].

[195] S. Gallozzi et al. "Concerns about ground based astronomical observations: quantifying satellites' constellations damages". In: (2020). arXiv: 2003.05472 [astro-ph.IM].

[196] M. Gaug et al. *Impact of E-ELT laser light on Cherenkov Telescope Array cameras*. 2013. arXiv: 1307.3056 [astro-ph.IM].

[197] Robert L. Wildey. "The Nyquist Criterion in CCD Photometry for Surface Brightness". In: *Publications of the Astronomical Society of the Pacific* 104.674 (1992), pp. 285–289. URL: http://www.jstor.org/stable/40679860.

[198] Astropy Collaboration. "The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package". In: *aj* 156.3, 123 (Sept. 2018), p. 123. arXiv: 1801.02634 [astro-ph.IM].

[199] L. Bradley et al. *astropy/photutils: 1.0.0*. Version 1.0.0. Sept. 2020. URL: https://doi.org/10.5281/zenodo.4044744.

[200] J Watson et al. *sstcam-simulation*. https://github.com/sstcam/sstcam-simulation. 2021.

[201] F. Ochsenbein et al. *The VizieR database of astronomical catalogues*.

[202] H. E. S. S. Collaboration. "Detection of very-high-energy γ-ray emission from the colliding wind binary η Car with H.E.S.S." In: *Aap* 635, A167 (Mar. 2020), A167. arXiv: 2002.02336 [astro-ph.HE].

[203] S.M Bradbury and H.J Rose. "Pattern recognition trigger electronics for an imaging atmospheric Cherenkov telescope". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 481.1 (2002), pp. 521–528. URL: https://www.sciencedirect.com/science/article/pii/S0168900201013419.

[204] Thomas Armstrong et al. "Monte Carlo Simulations and Validation of NectarCAM, a Medium Sized Telescope Camera for CTA". In: *Proceedings of 37th International Cosmic Ray Conference — PoS(ICRC2021)*. Sissa Medialab, July 2021. URL: https://doi.org/10.22323%2F1.395.0747.

[205] C. De Bom et al. "Bayesian Deep Learning for Shower Parameter Reconstruction in Water Cherenkov Detectors". In: *Proceedings of 37th International Cosmic Ray Conference — PoS(ICRC2021)* (2021).

[206] M. Strickland et al. *Deep Predictive Models for Collision Risk Assessment in Autonomous Driving*. 2018. arXiv: 1711.10453 [cs.RO].

[207] M. Shigeru et al. "DADIL: Data augmentation for domain-invariant learning". In: *Data Science and Pattern Recognition* 4.2 (2020), pp. 33–49.

[208] C. Theobald et al. *A Bayesian Convolutional Neural Network for Robust Galaxy Ellipticity Regression*. 2021. arXiv: 2104.09970 [cs.LG].

[209] A. Martin et al. "EvoDeep: A new evolutionary approach for automatic Deep Neural Networks parametrisation". In: *Journal of Parallel and Distributed Computing* 117 (2018), pp. 180–191. URL: https://www.sciencedirect.com/science/article/pii/S0743731517302605.

[210] T. Elsken et al. *Neural Architecture Search: A Survey*. 2018. arXiv: 1808.05377 [stat.ML].

[211] J. Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. 2020. arXiv: 1703.10593 [cs.CV].

[212] E. Lindernoren et al. *Keras-GAN*. https://github.com/eriklindernoren/Keras-GAN. 2018.

[213] G. Montefalcone et al. "Inpainting CMB maps using Partial Convolutional Neural Networks". In: *JCAP* 03 (2021), p. 055. arXiv: 2011.01433 [astro-ph.IM].

[214] P. Brown et al. *Graphcore sets new AI Performance Standards with MK2 IPU Systems*. https://www.graphcore.ai/posts/graphcore-sets-new-ai-performance-standards-with-mk2-ipu-systems. 2021.

[215] K. He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[216] J. Deng et al. "Construction and Analysis of a Large Scale Image Ontology". In: Vision Sciences Society. 2009.

[217] A. Jaegle et al. *Perceiver: General Perception with Iterative Attention*. 2021. arXiv: 2103.03206 [cs.CV].

[218] S. Buschjäger et al. "On-Site Gamma-Hadron Separation with Deep Learning on FPGAs". In: *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track*. Springer International Publishing, 2021, pp. 478–493. URL: https://doi.org/10.1007%2F978-3-030-67667-4_29.

[219] rapids.ai. *RAPIDS*. https://github.com/rapidsai. 2021.