Deep Learning on MAGIC: a Performance Evaluation for Very High Energy Gamma-Ray Astrophysics

Ettore Mariotti

Department of Information Engineering

Università degli Studi di Padova

A thesis submitted for the degree of

Laurea Magistrale In Telecommunication Engineering

8 April 2019

Academic Year 2018-2019

Supervisor: **Alessandro Chiuso**
Co-Supervisor: **Alberto Testolin**

Co-Supervisor: **Rubén López Coto**
Istituto Nazionale di Fisica Nucleare

It is only when you really understand something
that you can realize how little you know about everything.
— Zen saying

Alla mia famiglia.

Abstract

γ rays represent the ideal messenger of the ultra violent non-thermal universe because in virtue of having a null electric charge, their path is not deflected by intergalactic magnetic fields. When a very high energy γ ray enters in the atmosphere, it interacts with it producing an extensive air shower of secondary particles, generating a pool of Cherenkov light that can be detected by ground-based telescopes. MAGIC is an Imaging Atmospheric Cherenkov telescope belonging to the current generation of instruments of this type. Unfortunately for gamma-ray astronomy, the universe is full of high energy cosmic particles that interact with the atmosphere in a very similar way to the γ rays. As the signal-to-noise ratio is extremely low (<1:2000 even for the brightest sources), the standard analysis uses methods of machine learning on a parametrization of the images of the captured events in order to discriminate the signal from the background. As any parametrization, it implies an irreversible loss of information. A deep learning approach that works from the pixel information could build resilient abstract representations that have the potential of enhancing the analysis.

This thesis proposes to carry out a full reconstruction using convolutional neural networks for solving the problem of separating γ from non-γ events (binary classification), the reconstruction of their energy (single-variate regression) and their direction (multi-variate regression). With the aim of maximizing the performances, a novel boosting technique called Transfer Snapshot Ensemble is presented and evaluated. Using architectures modified from the state of the art in the computer vision domain, the new designed pipeline shows significant improvements both in energy reconstruction ($\sim$ 30% above 1 TeV) and in the direction reconstruction ($\sim$ 20%) over the standard MAGIC analysis.

The whole pipeline has been used to evaluate the performance on a reference source dataset taken from observations of the Crab Nebula. By applying the analysis developed in this work, it is possible to reach an integral sensitivity of 1.12% of the Crab for energies above 150GeV.

The thesis is organized as following: In chapter 1 I give a general introduction to cosmic ray physics, in chapter 2 I delve into how the MAGIC telescope works. In chapter 3 I briefly introduce the deep learning field and convolutional neural networks, in chapter 4 I explain the details of the analysis and finally chapter 5 concludes the work with a glimpse into possible future works.

Contents

1

Short introduction to cosmic ray and γ-ray astronomy

Observing the sky we detect particles with such high energies that they cannot be produced in typical thermal processes and must be originated in violent non-thermal processes. These high energy particles have been given the name of cosmic rays.

After traveling cosmic distances, when they reach the Earth and hit the atmosphere, they produce cascades of secondary particles called extensive air showers (EAS). If the velocity of these particles is greater than the speed of light in the atmosphere, they emit Cherenkov radiation that can be detected at ground level

1.1 Cosmic rays

Cosmic rays are high energy charged particles produced by galactic and extragalactic objects. Their composition is mostly made of atomic nuclei ($> 99\%$) and in a smaller quantity of electrons, positrons, neutrino and photons ($< 1\%$). Although they can interact with matter in their path to Earth, they are able to reach it because of the low density of matter in the space. The energy of cosmic rays spans from 10^{10} to 10^{20} eV following a power-law distribution. The technology for the detection of cosmic rays depends on their energy. They can be detected directly or indirectly by balloons- or satellite-borne experiments (lower energies), or by ground-based telescopes (higher energies).

Being charged particles, their journey is affected by the magnetic fields encountered along the way. For this reason the measured directions carry no information about the place of the event that originated them.

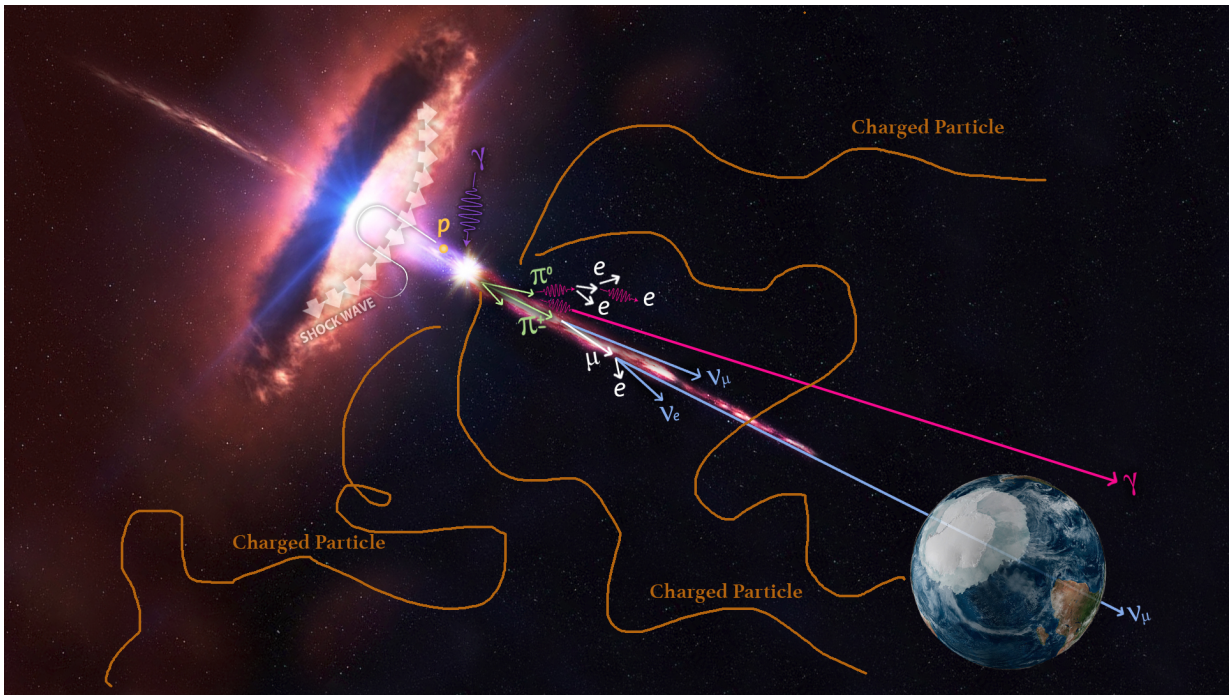


Figure 1.1: Unlike charged particles, the path of photons (and neutrinos) is not deflected by intergalactic magnetic fields. This open the possibility to pinpoint the source of the events that generates them.

1.2 γ rays

γ rays are electromagnetic radiation produced by relativistic particles. Since they are not electrically charged, they are not deflected by the intergalactic magnetic fields. Because of this, they keep intact the information needed to pinpoint the sources originating them. This is one of the reasons why γ-ray astronomy became increasingly important in recent times (Figure 1.1).

1.2.1 Sources

There are a number of galactic and extragalactic sources of γ rays. Among extragalactic sources there are active galactic nuclei and γ-ray bursts, while galactic sources are supernova remnants, pulsars, pulsar wind nebulae and binary systems.

- ⊙ **Supernova remnants** are the material and shock waves produced after a supernova explosion. Particles are thought to be accelerated with energies up to ~PeV by the expanding shock wave of the ejected materials.

- ⊙ A **pulsar** is a dense, fast rotating neutron star which emits a beamed electromagnetic radiation, regularly visible when the beam is pointing towards the observer.The radiation direction is fixed by the magnetic axis and the particle acceleration occurs around intense field zones.

A **Pulsar Wind Nebula (PWN)** is a bubble of electrons, positrons and magnetic field directly fed by a pulsar. In this case particles are accelerated in the wind termination shock up to energies in the PeV domain. The most studied objects in the sky is the Crab Nebula PWN.

⊙ **Binary systems** are composed by a massive star and a compact object (such as black holes, neutron stars, white dwarfs) orbiting around the common center of mass. The compact object can accumulate matter from the companion feeding its accretion disk. The accumulated gas can heat up reaching extreme conditions and therefore it can produce γ rays.

⊙ **Active galactic nuclei (AGN)** are actively growing supermassive ($> 10^6$ solar masses) black holes in the center of a galaxy. Perpendicular to the galactic plane there are relativistic outflows called jets where the charged particle acceleration (in the GeV-TeV domain) takes place.

⊙ **Gamma-ray bursts** are explosions of γ rays with energies up to TeV domain. They take place at cosmological distances. They are uniformly distributed in the sky and have a very short duration, which varies from fractions of a second to minutes. Their origin is still unclear, but they are believed to be originated during a huge system collapse or major merger events.

1.3 Interaction with the atmosphere

The Earth atmosphere is not transparent for γ rays and cosmic rays: they interact in the high atmosphere with the air molecules and produce chain reactions called extensive air showers (EAS). The particles that interact with the atmosphere are called primary particles and they cannot be detected by ground-based experiments. Ground-based telescopes can detect the products of the reaction between primary particles and atmosphere, i.e. the so called secondary particles. Air showers can be electromagnetic, if the process starts with a high energy photon, electron or positron, or hadronic, if the incoming particle is a proton or a nucleus.

⊙ **Electromagnetic air shower** When a primary γ ray interacts with an atmosphere nucleus, it undergoes the process called pair-production, generating an electron-positron pair. The generated electrons and positrons interact with the electromagnetic field of atmosphere nuclei further emitting γ rays due to a process called bremsstrahlung. By subsequent pair production and bremsstrahlung processes a cascade of particles is formed. The electromagnetic air shower stops when the energy of electrons is lower than a critical value, for which the dominant process for electron energy loss is ionization.

⊙ **Hadronic air shower** An hadronic air shower begins with an inelastic scattering between a hadron and an atmospheric nucleus, producing in particular electromagnetic subshowers. When hadronic interactions occur, particles like pions can be produced and the original cosmic ray can generate more hadronic interactions. For a more in depth analysis of high

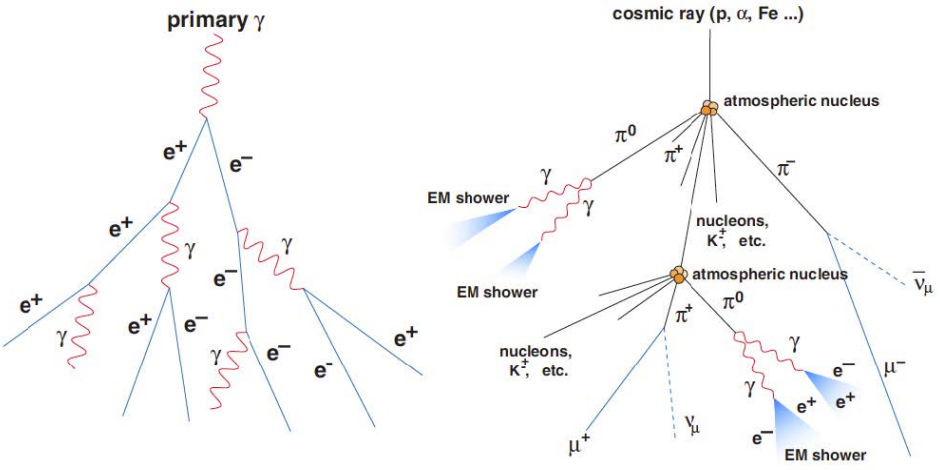


Figure 1.2: Schematic representation of a γ ray (left panel) versus hadronic induced (right panel) showers. Credit: (23).

energy hadronic interactions, refer to (Engel et al.). Hadronic showers constitute the main background signal that we want to be able to filter out.

1.4 Cherenkov radiation

Whenever a particle moves with a speed v faster than the speed of light in the medium $c' = \frac{c}{n}$ (where c is the speed of light in vacuum and n is the refraction index of the medium), Cherenkov light is produced. This is because when a charged particle travels in matter, it polarizes the medium producing spherical electromagnetic waves along its track. If $v > c'$ the particle moves faster than the propagation of the waves, producing constructive interference at an angle Θ_C, also called Cherenkov angle. By calling $\beta = \frac{v}{c}$ and with basic geometric reasoning (Figure 1.3) it is possible to compute this angle as:

$$\cos(\Theta_C) = \frac{c'}{v} = \frac{c/n}{\beta c} = \frac{1}{\beta n} \tag{1.1}$$

With the equation above, the wave shock will propagate as a cone with angle Θ_C (whose average value in air is $\sim 1°$) and cancel out in every other direction due to destructive interferences. This effect have an analogy in acoustics when an object travels at supersonic speed (Mach waves).

As the refraction index of air changes with altitude (due to its non-uniform density), we have slightly different cones of light that reach the ground (see Figure 1.4). The shower as a whole generate a circular pool of light, product of the superposition of every singular interactions that fire a donut-like flash of light. Since in general the primary particle don't fall perpendicularly to the ground, the shape of the pool is an ellipsoid. This is the signal that can be detected by ground-based experiments such as MAGIC.

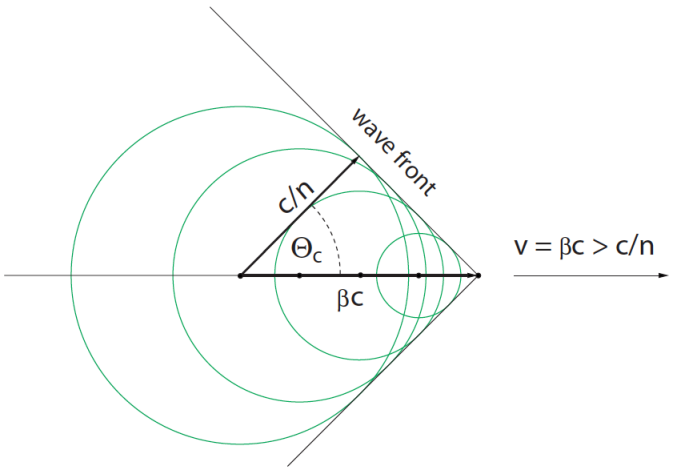


Figure 1.3: Schematic view of Cherenkov spherical emission

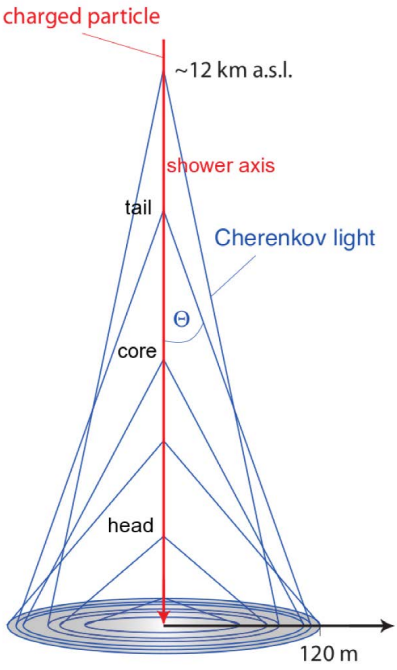


Figure 1.4: Sketch of the atmospheric Cherenkov angle variation with altitude. Image taken from (17)

2 IACT and the MAGIC Telescopes

There are several techniques to detect very-high energy particles. Experiments can be made in the high atmosphere with balloons or can even take place in the outer space with satellites. Due to the power-law nature of the energy distribution of cosmic rays and the limited dimensions for a detector on board, these experiments can only detect the lower energies of the spectrum. In order to observe VHE events, a different approach based on ground based telescopes called Imaging Atmospheric Cherenkov Technique (IACT) can be used. In the first section the general characteristics of IACT telescopes are presented, while in the second section the features of the MAGIC telescopes are discussed.

2.1 IACT Technique

Imaging Atmospheric Cherenkov Telescopes (IACTs) telescopes detect the Cherenkov light emitted in extensive air showers initiated by VHE γ-rays hitting the atmosphere. They are located at altitudes between 2000-3000 m.

The amount of Cerenkov photons arriving to the ground is around 10 to 20 photons/m^2 for a 100 GeV γ-ray shower, increasing almost linearly with the energy. For this reason detectors with a large collection area are needed: the current generation of IACT telescopes have a mirror diameter up to 28 m to be sensitive to low photon densities at the ground.

Cerenkov light brightens quite uniformly an area with a radius of 120 m: if the telescope is inside the Cerenkov lightpool, the light can be reflected and focused by the mirrors and collected by a camera composed of several photosensors, usually photomultiplier tubes (PMTs).

Photons collected by telescopes are produced at different heights and they reach the mirrors at various angles. As a consequence, they hit different PMTs. Through conversion in electronic

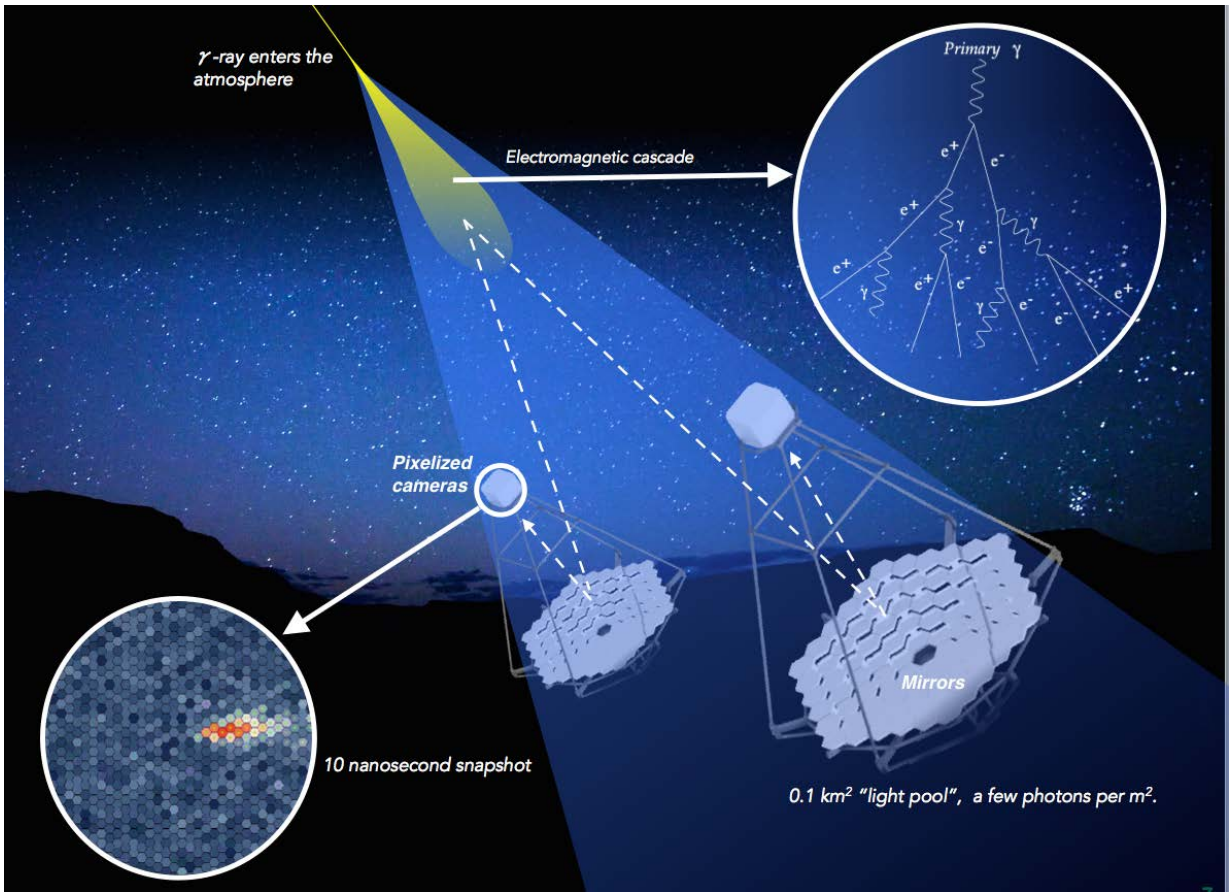


Figure 2.1: The IACT Technique in a nutshell. The few photons of Cherenkov light produced by the electromagnetic shower initiated by a high-energy particle are focused by the mirrors on the camera. This signal is converted to an electronic one by photomultipliers and will be used for further analysis. Image taken from (5)

signal an image of the EAS can be obtained. The shower image in the camera reflects the shape of the particle cascade.

Each image represents an air shower induced by one primary particle. The background is composed by cosmic rays: those with similar energy as a γ-ray can produce a similar reaction in the atmosphere, but with higher intensity. For the purpose of fully characterizing an astronomical source of γ rays one need to reconstruct the direction and the energy of the original γ rays. A source is considered to be detected when the statistical significance of the events contained in the signal region over those contained in the background region is larger than five standard deviations.

Figure 2.2: Picture of two MAGIC telescopes located in La Palma at El Roque de los Muchachos. Image taken from https://magic.mpp.mpg.de/

2.2 The MAGIC Telescopes

The Major Atmospheric Gamma ray Imaging Cerenkov telescopes (MAGIC) are located on the Canary island of La Palma at 2200 m above the sea level. MAGIC I is operative since late 2003; it can work in stereoscopic mode thanks to its twin telescope MAGIC II since 2009. Both MAGIC I and II have 17 m diameter reflectors: the large collective areas assure a low energy threshold of the order of 50 GeV.

The telescopes are not surrounded by a protective dome, as usual for optical telescopes, because of their large dimensions. This causes the mirror surface to be affected by the constant exposure to the external ambient. Therefore the surface of the mirrors is protected by a coat of quartz, which preserves them from chemical and mechanical damages. The mechanical design of the telescopes is made of an innovative carbon fiber structure which allows very fast movements.

Since the typical duration of Cherenkov signals is less than 10 ns, very fast electrical response and readout systems are required. The readout electronics is not integrated in the camera structure but it is located in a separated control house. As a consequence, any drawbacks for the electronics to be located in a counting house and the weight of the camera is reduced.

The stereoscopic view allows to get more informations about each shower, like a 3D shape and its location, and to strongly suppress the background thanks to the coincidence between the two telescopes.

2.2.1 Camera and Signal

The camera of both telescopes is supported by a single aluminium tubular arch and it is located almost at the focal length distance (17 m) of the mirrors. Both cameras are composed by 1039 PMTs with a collection area of 0.1° grouped in 169 clusters.

The signal collected by PMTs is converted into optical signal and sent through optical fibers to the counting house. There the signal is converted back into an electric one and sent to the trigger branch. The trigger is comprised of three levels:

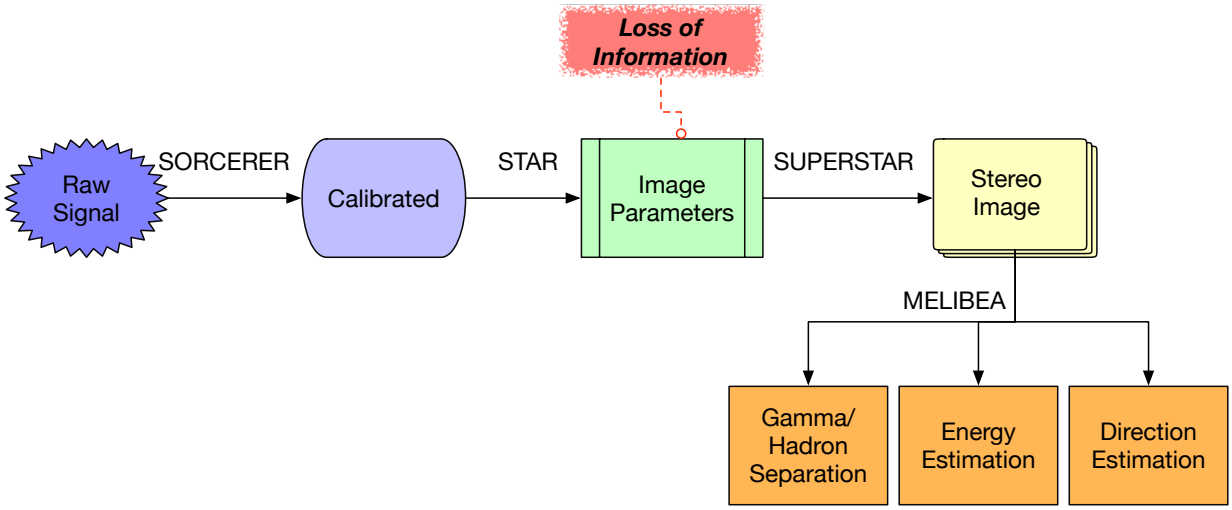


Figure 2.3: Schematic view of the standard MAGIC analysis

- **Level 0**: an analogical discriminator with a programmable threshold, i.e. only signals exceeding the threshold can pass to the next level. Usually the threshold consists on 5 to 8 overlapping photons.
- **Level 1**: a digital filter that reconstructs the topological time coincidence of the signal in the camera. A level 1 trigger signal is generated when a certain number of next-neighbor pixels are activated synchronously
- **Level 3**: a stereo trigger that activate when both telescopes fire a level 1 trigger in a 180 ns window.

Triggered signals are digitized and stored with a 1.66 GHz sampling data acquisition system (DAQ) storing the charge and time information with 50 slides of 12 bit for each triggered signal.

2.2.2 Analysis

MAGIC Analysis is performed using the MARS software. MARS is a multi-purpose software environment based on ROOT/C++ libraries. The first challenge of the analysis is to detect the events generated by γ rays and discern them from the background of hadron-induced showers (γ/hadron separation). This is an important step as there is a strong class imbalance in the events that are selected by the trigger, typically with a ratio of 10000:1 hadrons to gammas. Once the gammas are selected the goal is to reconstruct the direction and energy of the γ-ray that generated the shower. The whole pipeline, represented in Figure 2.3, develops as follows:

1. **Calibration** of the signal into phe, performed by `SORCERER` program (for DRS4 data). Here the the 50 slides of sampled signal detected by the photomultiplier is processed in order to have a neat estimate of the global number of photons and their averaged time arrival.

2. **Image cleaning** and **Image parameters** calculation computed by the program `star`. In this phase the complex event is reduced to a set of carefully hand-crafted parameters. See Figure 2.6 for reference.

3. **Stereo image parameters** with the program `superstar`. This step merge the parameters of the two telescopes.

4. Train of a **Random Forest (RF)** model for the γ/hadron separation, produce the look-up tables for the energy reconstruction (**Nearest Neighbor**) and compute a set of parameters for the reconstruction of the arrival direction. This training, carried out by Coach (stereo), needs simulated Monte Carlo (MC) gamma-ray events and a data sample of real background data (observations with no gamma ray-emitter in the field of view).

5. **Apply** the machine learning models to the real data in order to compute hadronness, reconstructed energy and arrival direction. Also applied to the MC simulations. The program used for this task is `melibea`.

6. Computation of signal **significance** with Odie, skymaps with Caspar and spectra and light curves with Fluxlc or Flute.

2.2.2.1 Monte Carlo Simulations

In order to be able to train the machine learning models, a dataset of synthetic simulations (Monte Carlo simulation, also referred as MC) where the ground-truth of the events are known is necessary.

In MAGIC, gamma rays are simulated in two ways:

- ⊙ as *diffuse* (Figure 2.4): for the study of extended sources or sources shifted from the nominal position. In this case, gamma rays are uniformly simulated covering a circle of 1.5° radius. For this a total of 24 960 000 events were simulated, of which 1 456 096 triggered.

- ⊙ as *ringwobble* (Figure 2.5): simulates a ring of 0.4° radius (with a width of $< 0.1°$) from the camera center, accounting for the 0.4° offset used in the standard wobble mode. It is used for the analysis of point-like sources. For this a total of 4 989 000 events were simulated, of which 462 848 triggered.

The energy spectrum of the simulated events follow a power-law distribution of the same kind as the observed data and is also lower bounded by the trigger technology, with a peak in the number of triggered events around 100 GeV. The models are in general trained on the diffuse MC (paritioning the dataset in 1 092 072 train events and 364 024 validating events) and then tested on the ringwobble one.

2.2.2.2 Why making two different simulations?

When looking for a new source in the sky, the telescope points the location of interest with an offset of 0.4° in multiple acquisitions. This is useful for canceling some systematic error in

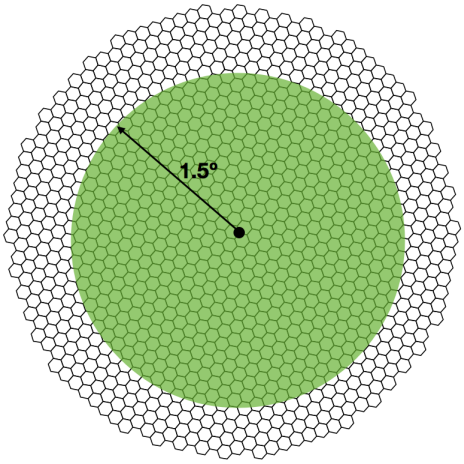


Figure 2.4: Montecarlo *diffuse* simulation

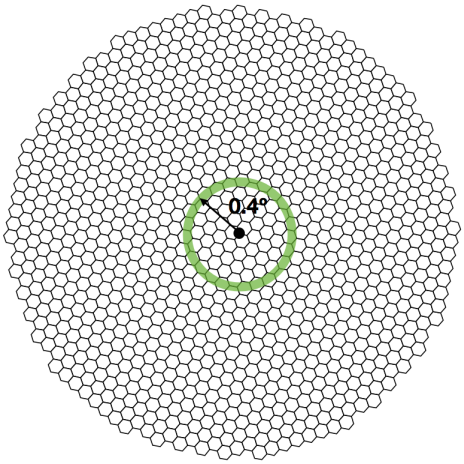


Figure 2.5: Montecarlo *ringwobble* simulation

the final measure. For this fact we expect to reconstruct each signal coming from a point-like source in that region. But since we have a strong background of γ-like events induced by high energy electrons and the presence of γ rays not coming from the source we don't want to bias our analysis with the strong a-priori constraint of point-like events.

Moreover, being able to correctly reconstruct the direction of diffuse γ like events allow us to more reliably estimate the background, and thus having a finer estimate of the significance of a given data acquisition.

For these reasons each classifier and regressor is trained on the *diffuse* dataset, but their performance abilities are computed on the *ring-wobble*.

2.2.2.3 Signal Calibration

When the trigger is issued, the data acquisition system stores the information of each camera pixel in the raw data digitalizing the analog signal with a *Flash Analog to Digital Converter* (FADC). From these slices are computed the following quantities of interest:

- ⊙ ***phe*** is the number of photoelectrons captured by each PMT, and is computed thought the F-Factor method proposed by (15).

- ⊙ ***time*** is computed as the average number of time steps passed from the trigger absolute time, weighted by the value of the digitized signal.

In order to keep the quality of data high, many calibration event runs are recorded during normal data taking. Calibration runs are used to calibrate the pedestal subtracted charge of the signal.

2.2.2.4 Image cleaning

After the calibration, the images are cleaned, keeping only significant signals, and parametrized by the `star` program. Although after the signal pre-processing, charge and arrival time for each

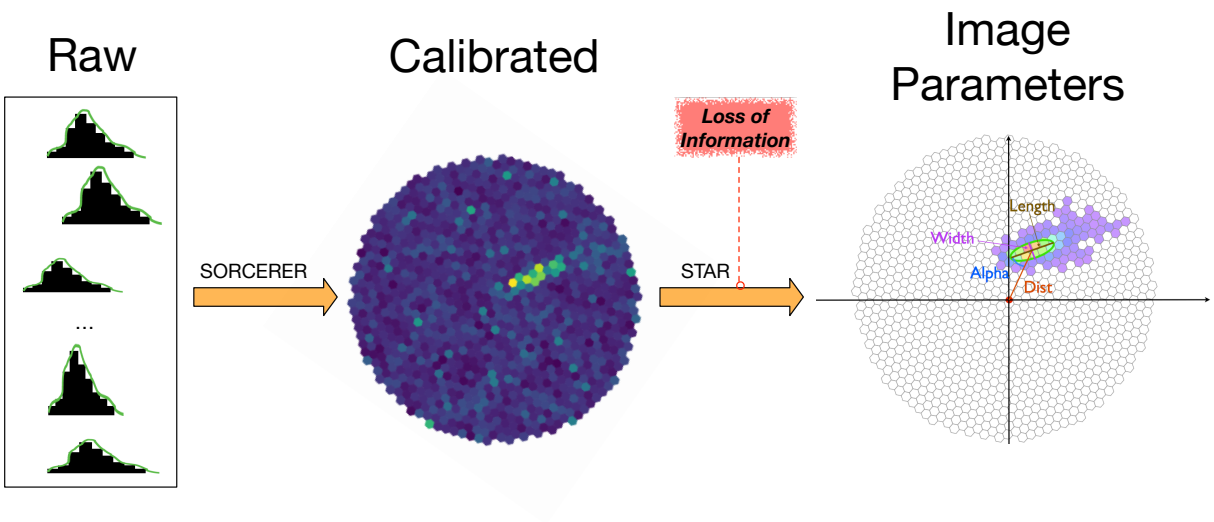


Figure 2.6: Visual representation of the signal processing pipeline. Notice that whenever we perform a parametrization, we have a loss of potentially useful information.

PMT is available, not all pixel contain useful information. Most of them contain only noise, useless for the signal analysis. Thus, the image cleaning algorithm aims to keep the pixels in which a significant amount of Cherenkov photons produced signal from the shower, discarding those pixels that, below a certain arrival time and amplitude thresholds, do not contain useful information of the shower image. More in particular two scalars, ζ and κ (with $\kappa < \zeta$) are defined. A pixel is considered to contain significative signal if:

- ⊙ It detects a number of photons $> \zeta$, or
- ⊙ It detects a number of photons $> \kappa$ and is adjacent to a pixel that detected more than ζ photons.

2.2.2.5 Image parametrization

After the image cleaning, a set of parameters is computed by fitting an ellipse on the surviving pixels. The main parameters calculated are:

- ⊙ Size: It corresponds to the sum of the charges in phe of each surviving pixel. For a given impact distance of the event, the size is correlated to the energy of the primary gamma ray if the event is contained in the Cherenkov light pool of radius 120 m.
- ⊙ Length: Longitude of the major semi-axis of the ellipse. It is related with the longitudinal development of the cascade.
- ⊙ Width: Longitude of the minor semi-axis of the ellipse. It is a measurement of the lateral development of the cascade.

- ⊙ Conc(N): Fraction of the image charge contained in the N brightest pixels. It gives the compactness of the image, which for EM cascades is larger than for hadronic showers. The used value is Conc(2).
- ⊙ Dist: Angular distance between the position of the source and the center of gravity of the image. The larger the dist value, the larger the impact parameter of the shower in the ground.
- ⊙ Alpha: Angle between the major axis of the ellipse and the imaginary line connecting the source position and the center of gravity of the image.

also **Time-dependent parameters** are computed as they can be useful to discriminate between EM and hadronic showers, given that the former develops faster (3 ns compared to the 10 ns):

- ⊙ Time RMS: RMS of the arrival time of the surviving pixel, which is smaller for gamma ray-induced cascades.
- ⊙ Time gradient: Slope of the linear fit applied to the arrival time projection along the major axis, which gives the direction of the shower development.

Other parameters are used to estimate the image quality. Thus, very noise images or images not well-contained in the camera can be discarded.

- ⊙ LeakageN: Fraction of the shower light contained in the N outermost rings of PMTs of the camera. This parameter measures how much image is contained in the camera.
- ⊙ Number of islands: Number of separated groups of pixels after image cleaning. Hadronic showers usually produced sub-showers that are reflected in the camera as separated images.

Finally, there are the so-called directional parameters. They are used to differentiate between the head (top of the cascade) and tail (bottom of the cascade). Atmospheric showers present higher charge in the head part, since particles in the top have higher energies.

- ⊙ Asymmetry: Direction of the line between the center of gravity of the image and the pixel with the highest charge. The EM cascades present positive asymmetry, i.e. pixels with the highest charge are located close to the source position.
- ⊙ M3Long: Following the same criterion as the asymmetry parameter, M3Long is the third moment of the image along its major axis.

While all these parameters have been successfully used for the complete reconstruction of the events, any parametrization lead to an inevitable loss of information (Figure 2.6). This is what motivates the pursue of an alternative approach, working directly on the calibrated data.

3

Deep Learning and Convolutional Neural Networks

Machine learning is a branch of computer science that aim to solve classification and regression problems without explicitly having a human intervention for tuning some parameters of a model, but rather the model is automatically learning its best configuration from the data. While the use of machine learning is not new in the pipeline analysis of MAGIC, the possibility of fully exploiting all the data available has the potential to enhance current performances. For this reason the recently developing field of deep learning could be an interesting direction to pursue.

In the next sections I am going to introduce how and why deep learning was born and how it can be applied to the full characterization of γ ray events.

3.1 History and Context of Deep Learning

For decades computer scientists have developed tools and models to solve classification and regression machine learning problems. This turned out to work really well for many real world problems, but it always required a careful hand-crafted parametrization of the raw data. Most of the work in solving these classification and regression problems was indeed about building a sophisticated feature engineering that required a lot of effort and domain-specific knowledge. This allowed to transform complex high-dimensional representations into a suitable internal description from which a designed model could capture patterns of the input.

Representation learning is instead the task of automatically learn features from the raw data. Deep learning is a way of solving representation learning with the use of gradient-based techniques with a multi-layered level of representation. Many relatively simple non linear differen-

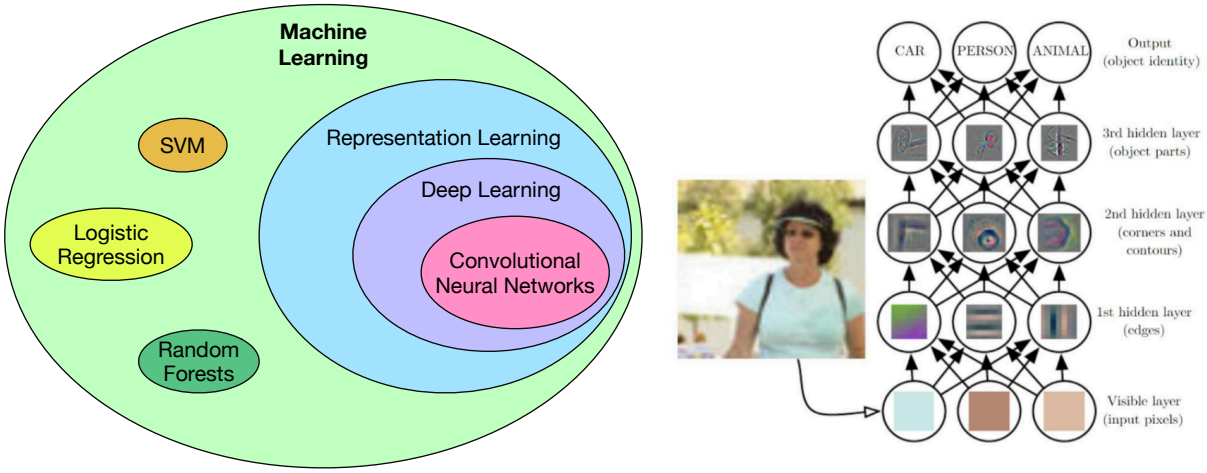


Figure 3.1: Deep Learning is a form of representation learning where each feature is automatically computed by the optimization of differentiable layers. This moves the burden of feature engineering from humans to machines.

tiable submodules are stacked one upon each other in order to construct more and more abstract patterns characterizations. With a sufficient number of simple layers, very complex functions can be approximated arbitrarily well[1]. The success of this approach is that in this way we are building an hierarchical set of features. In the case of an image for example, the first layers detect edges and corners, the second layer assembles corners and edges to detect motifs, while the third layer may collect and combine motifs to create familiar patterns. The fundamental paradigm shift from the traditional approach here is that this kind of hierarchical representation is not designed by a human engineer, but it is instead captured from data with a general-purpose learning procedure.

3.1.1 Supervised Learning

When for the problem at hand we have the true value that we want to reconstruct, we are doing supervised learning. Deep supervised learning works by setting up a differentiable compositional model and an objective function (loss) that measures the discrepancy of the prediction from the true value. At training time the machine computes the error of the prediction and then propagates its gradient with respect to the parameters using the chain rule. This gradient information is then used by the model to modify the internal adjustable weights in order to reduce the error. This algorithm is called back-propagation and is an efficient way of minimizing highly non-convex functions. The variant that is usually adopted by practitioners in the field is stochastic gradient descent (SGD): the gradient is not computed on the whole data, but instead on a few samples (often called a *batch*). It is called stochastic because the resulting vector point to the direction of the closest minima plus a random term. This random term is very important as it helps the optimization process not to get stuck on a saddle point (which are provably many).

[1]Theoretic results state that actually any function can be approximated arbitrarily well with just one layer (but without stating anything about the number of inner states). In practice deep neural network are more effective in solving classification and regression tasks.

When the optimization phase converges to a stable low value, the resulted model is applied on an unseen dataset in order to asses its generalization performances.

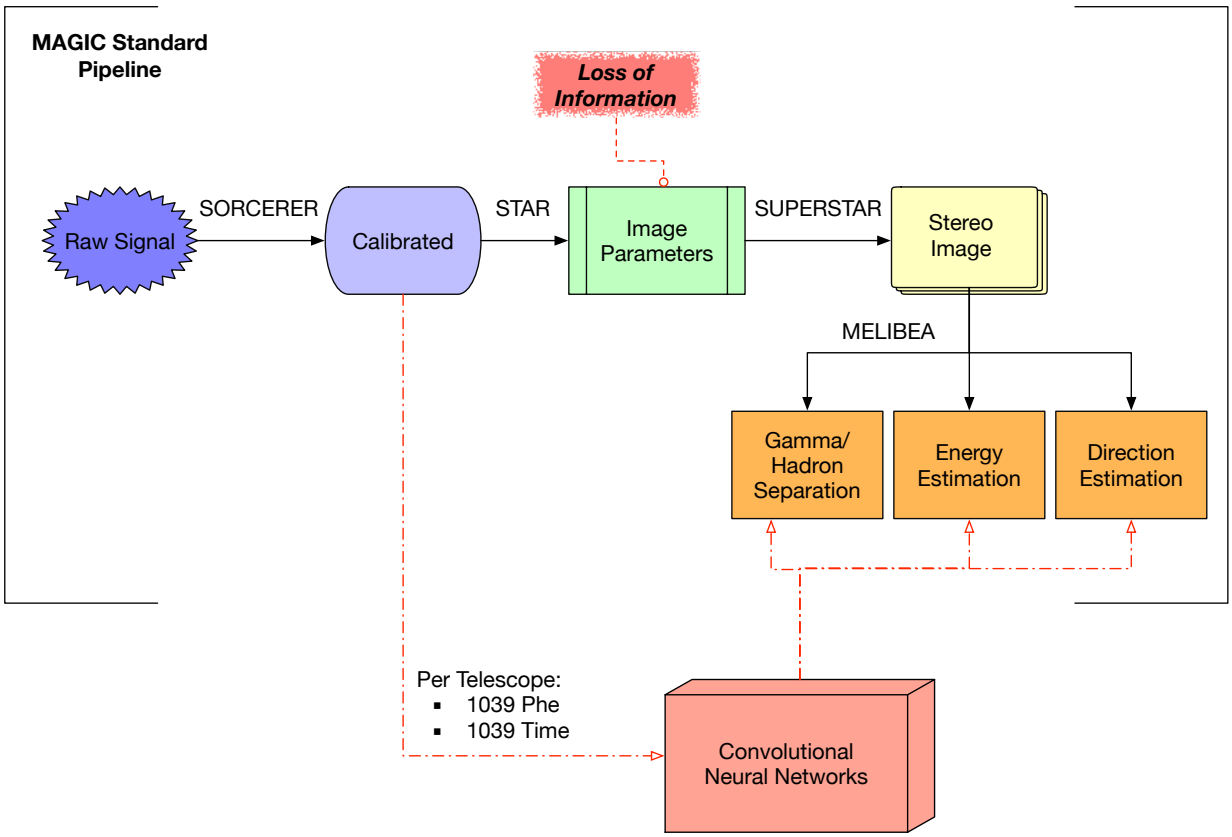


Figure 3.2: The new approach proposed in this thesis: a complete analysis by working directly from the calibrated data.

3.1.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are a particular class of neural networks that are designed to work on array data. Many kind of data can be seen as structured array data: 2D spectrograms for audio, RGB images as a 3D values, 1D for time sequences (including language). Some of the most important characteristics for the success of the CNNs can be summarized as:

⊙ The convolution operation, which produce translational-invariant feature detectors

⊙ Local connections, which help to find patterns in highly correlated data

⊙ Many layers, which gradually builds more abstract feature maps

A typical CNN is structured in a series of stages alternating convolutional layers to pooling layers. Convolutional layers are responsible of detecting a particular combination of features

from the layer below. They are composed of units that apply the convolution operation of the so called kernel on a local patch of the feature map from the previous layer. The result of this combination is then passed to a non-linear function, typically a Rectified Linear Unit (ReLU) (defined as $\max(x, 0)$).

The role of the pooling layers on the other hand is to merge similar features, providing robustness to variability in the input data. Because the relative positions of the features forming a motif can vary somewhat, we can reliably detect the motif by sub-sampling the position of each feature by taking the maximum over a certain patch (*max pooling*). Once the architecture is set up, the model can be trained with back-propagation just as any other neural network.

In essence, deep neural networks exploit the property that data typically manifest in compositional hierarchies, in which higher-level features are obtained by composing lower-level ones. In pictures for example, local combinations of corners and edges form structured patterns, and structured patterns form objects. Similar hierarchies exist in other kind of data. The pooling allows resilience to little changes in position and appearance of element of the previous layer.

3.1.2.1 Brief Historical Review

Historically, CNNs first appeared in computer-vision, in (14) where the author demonstrated their effectiveness on the problem of recognizing handwritten digit. After a while the work of (13) won the ImageNet competition (which consisted in the classification of 1 million of images) with a CNN called AlexNet, greatly improving over the other competing methods. He was the first to present a CNN approach in this kind of challenge and proposed GPUs as the technological support on which speeding up computations. This marked the beginning of the deep learning revolution in computer vision, where features extractions and classification was all embedded in one single algorithm driven by gradient.

After that, VGG-net of (19) demonstrated how the size of the convolution kernel was not so important, but instead how with increased network depth and the smallest kernel size (3×3 so that it could be preserved a notion of up/down/left/right) they were able to generalize much better. This motivated research into deeper and deeper networks. Nevertheless, it was soon clear that when depth is increased, the optimization process becomes much more delicate and sometimes unstable. Moreover, when a network is too deep, it faces (among others) the problem of the vanishing gradient: the gradient used to update the parameters becomes rapidly close to zero. For solving this problem a number of innovations came into place:

- ⊙ *ReLU* (defined as $\max(x, 0)$) non linearity proposed by (2) instead of the sigmoid function
- ⊙ *Batch Normalization* by (10): a strategy that consist in collecting statistics of the output of each layer batch per batch in order to normalize it. This help a lot the stability of the training, allowing the possibility of using much higher learning rates providing thus a more rapid convergence of the loss function.
- ⊙ *Residual Connections*: introduced by (6), where the idea is to sum the output of layer i with the output of the layer $i - k$. This residual passing vastly reduced the problem of the vanishing gradient, allowing the training of much deeper (and powerful) networks.

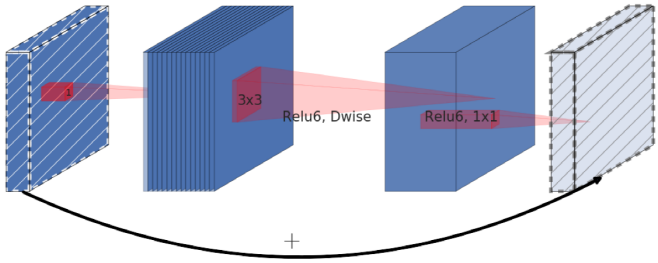
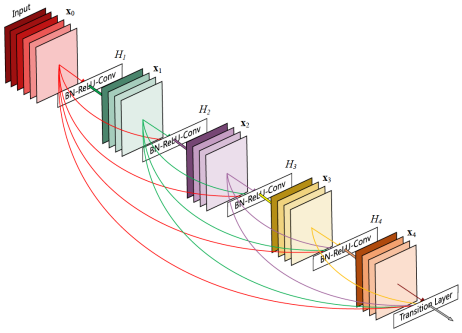


Figure 3.3: An inverted residual block connects narrow layers with a skip connection while layers in between are wide. Dashed tensors have linear activations. Image taken from (16)



(a) Sketch of the DenseNet architecture. The densely connected residuals allow to greater network depth and feature reuse

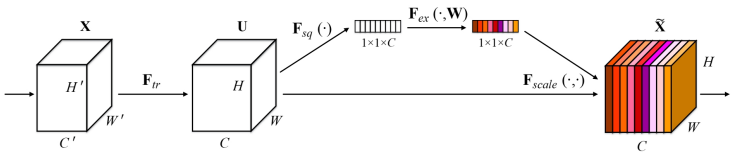


Figure 3.4: The Squeeze-and-Excitation block. Image taken from (7)

Following this path came along the work of (16) with MobileNetV2, an architecture which used an innovative building block: inverted linear residual blocks (Fig. 3.3). It consist of 1×1 blocks to expand the feature space in high dimensions, followed by separable depth-wise convolution and by another linear 1×1 layer that works as a bottleneck. A residual connection is also added from the first to the last 1×1 block. With this configuration the number of parameters and operations needed to be computed is drastically reduced, while still exhibiting extraordinary generalization performances comparable with models that are orders of magnitudes more complex in terms of number of parameters and operations needed.

Another successful attempt to enhance the representation power of deep CNNs was made by (9) with DenseNet. A DenseNet block (Figure 3.4a) consisted of a set of layers densely connected by residual connections. This allowed to greatly lessen the vanishing gradient problem while also introducing the capability of feature-reusing from the lower layers. DenseNets were able to reach depth of 121, 169, 201 layers and beat the state of the art in many computer vision standardized datasets.

One of the most recent innovation is not in the architecture, but in the block design. In the work of (7) the authors purpose the Squeeze-and-Excitation (SE) block. The purpose of this block is to provide a self-attention mechanism to each channel of the convolutional layers. This new block allowed already existing architectures to reach a new level of performances.

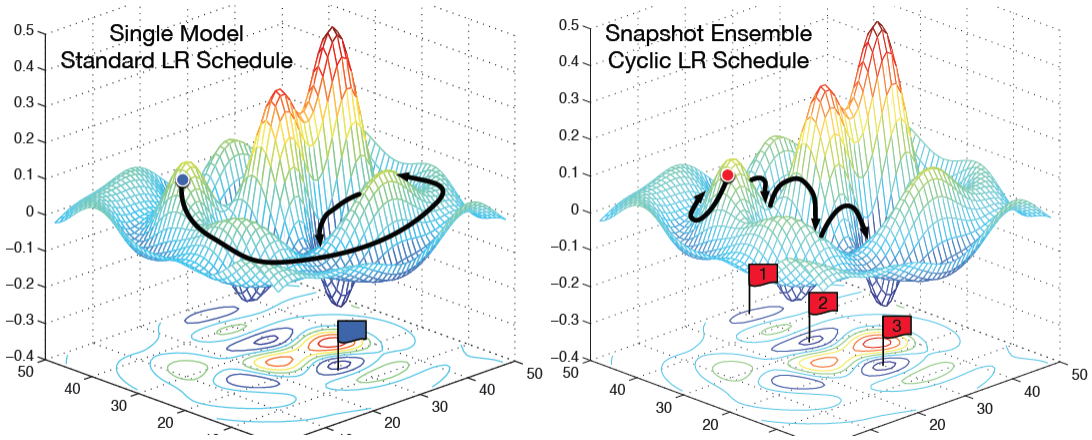


Figure 3.5: Snapshot Ensemble works by saving a version of the network whenever it reaches a local minimum. These snapshots exhibit significantly different weights structure. This diversity is a richness that can be exploited with ensemble. Image taken from (8)

3.1.3 Optimization

The optimization of these deep neural network makes use of the gradient, but there exist different strategies on how to handle this process. The most trivial one is Stochastic Gradient Descend (SGD), which simply computes the gradient of the loss with respect to the parameters and then use it, multiplied by an hyper-parameter α called Learning Rate (LR), to update the weights. This can be done in batches so that we have a noisy estimate of the direction of the minimum, allowing to escape the trap of local minima or saddle points. The addition of a momentum component for the learning rate was proven to help convergence in some cases. On this there were developed a number of adaptive policies such as ADAM, AdaDelta, Adagrad that dynamically lower the learning rate as training progresses.

A disruptive work on optimization was pursued by (20) with Cyclical Learning Rate (CLR), where it was showed how with a learning rate that cycle between a very high value and a low one it can be achieved what is called Superconvergence: convergence of the network to its maximum capacity after far less iterations than usual. Based on this work Snapshot learning by (8) was built, born from the observation that while training with a CLR the network explores different local minima. As learning progresses the different explored minima gets lower and lower, but most importantly each minimum result in a significantly different network that nonetheless achieve similar performances. This can be exploited by "taking a snapshot" of each minimum explored and then using a collection of snapshots in order to make an ensemble of neural networks while just having to train once (see Figure3.5).

Building on this work, (11) purpose Stochastic Weight Averaging (SWA). This is a technique that try to conjunctively leverage the diversity of the founded minima and to cut down inference time. Their proposal is to use a final model which weights are the average of every saved snapshot. SWA tend to find a minimum in a flat basin, which have been correlated with better

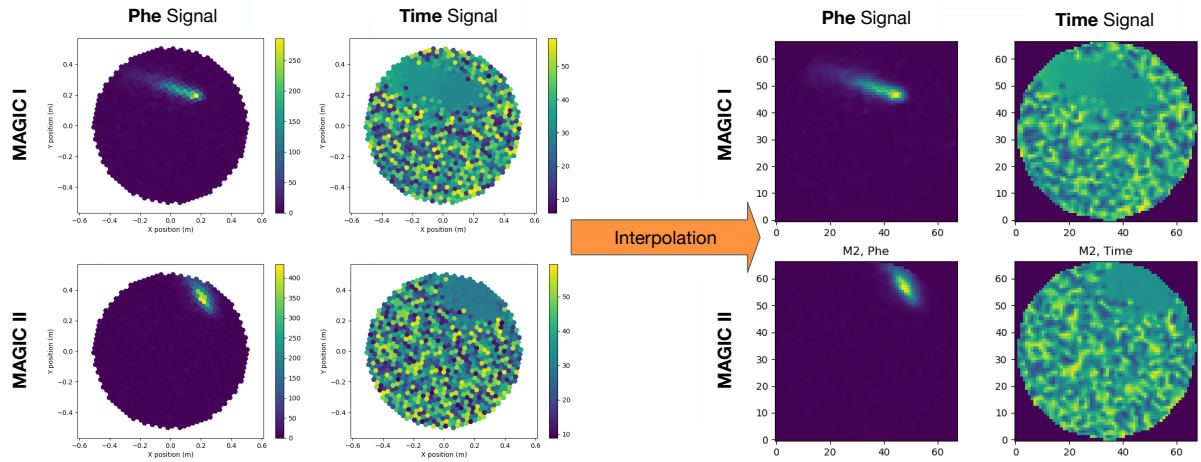


Figure 3.6: The interpolation step, needed for being able to interface with modern deep learning frameworks

generalization performances since (12).

3.2 A New MAGIC Analysis

Since any parametrization implies an inevitable loss of information, one could argue that a deep learning approach could indeed learn new and useful features from the calibrated data (Figure 3.2), even without the cleaning process described in 2.2.2.

The availability of huge datasets of MC simulations and of real data taken from the telescopes could be leveraged by CNNs in order to gain a better representation of the underlying event and to eventually compute a more precise analysis.

In order to investigate this approach one must be able to communicate between different frameworks and standards: from the ROOT files in which the data is contained following the hexagonal disposition of the photomultipliers, to an interpolation on a regular rectangular-lattice and a tensor-representation that is required by modern deep learning frameworks such as Tensorflow or PyTorch.

3.2.1 Data reading and Interpolation

The data, written in ROOT files, see (3), needs to be read, transformed in an information-preserving way and saved to an appropriate format. The data is read with `uproot`, a library that allows to read ROOT files without the need of installing non-standard software and custom libraries needed to read the data..

Once in memory, the data is interpolated using `scipy`. Of utmost importance was the choice of the interpolation step: a step too large would lead to information loss, one too small would blow up storage requirements. By recalling the Shannon-Nyquist sampling theorem: any band-

limited signal can be reconstructed from its samples, if the original signal has no frequencies above 1/2 the sampling frequency. In the space domain, this translate to an interpolation step smaller or equal to the maximum distance between two close sampled points. Being the camera an array of pixels on an hexagonal lattice where the distance between two adjacent is $\simeq$ 32mm, the chosen interpolation step was set to a conservative 15mm. A linear interpolation method was chosen for its good speed-quality trade-off. A view of the result of the process can be seen in Figure 3.6. The interpolated data was the *phe* and *time* signal of each event that triggered both telescopes. The result was then saved as a `numpy` tensor of dimensions $(67 \times 68 \times 4)$. Due to the heavy-computing and big-data involved, the whole interpolation process was parallelized with the multiprocessing python library.

3.2.2 Software Architecture

Due to the fact that the interpolated dataset would not fit in RAM memory (> 300GB), an appropriate `Keras` generator was set up in order to load it iteratively from the disk. Each event was saved in a separate `numpy` file, a format chosen for its lightness and low overhead when loading. Each event have an unique ID which comes from a combination of simulation/data gathering parameters and an MD5 checksum of the tensor. Labels were instead saved in a dictionary data structure where each key was the unique ID and the value the corresponding true-value (hadron/gamma label, energy value, direction coordinates). This allow a fast and efficient retrival of the needed informations.

The Keras train generator task was to select a certain (fixed) number of ID lables at random, loading the corresponding files from the disk and sending them to the GPU (along with the appropriate labels) for the optimization process. The generator is built in such a way that in one pass over the data an event is not chosen twice, but in each pass the reading order is different (very important for stochastic gradient-based optimization techniques). The generator also support loading data with multiple workers, lowering the dead time due to overheads.

3.2.3 Hardware Architecture

The number of operation needed in CNNs for inference and backpropagation is relatively large, but it is also massively parallelizable. For this task GPUs (Graphic Processing Unit) have shown to be really effective. Their cores can perform just a restricted set of operation, but a single GPU can contain thousands of cores which can operate in parallel.

For this work it was used a server with a Titan Xp and another server with a Titan V[1].

As the dataset was too big to be put in RAM, data reading from the magnetic disk technology was an issue that needed to be addressed as it was the bottleneck of the computation pipeline.

With an hardware upgrade of SSD-NVMe (Solid-State Disk), the non-sequential reading performances was more than one order of magnitudes better and allowed to cut down the training time by a factor $\sim$ 20. In this configuration the GPU is fully utilized and thus the system is optimized to its best capabilities.

[1]The Titan V and Titan Xp used for this research were donated by the NVIDIA Corporation.

4
Reconstruction

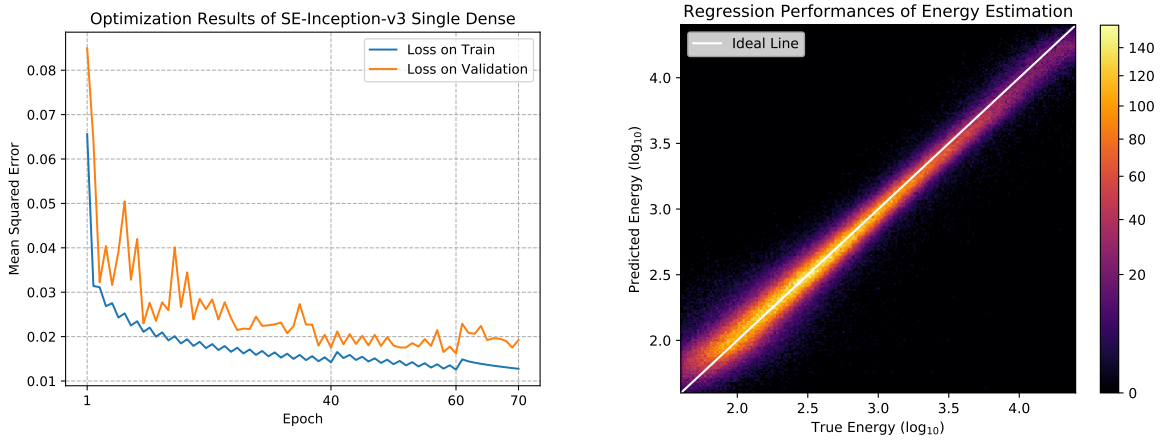
In order to claim the discovery of a new γ ray source, a statistically significant number of γ rays are needed to be detected as coming from the same direction in the sky. This is a challenging task due to the extremely low signal to noise ratio of IACT experiments which is of the order of 1:2000 even for the brightest sources. The dataset is thus extremely noisy and we need to use several techniques to separate between signal and background. This call for the need of a good separation strategy and an accurate direction reconstruction. Additionally, to properly study the sources, we need to reconstruct their spectra and for this we need the energy reconstruction. In essence, there is the need to address the following challenges:

1. To separate γ-induced from hadron-induced events that triggered in the telescopes (a *binary classification* problem).

2. To reconstruct the energy of the γ ray that originated the cascade (a single-output *regression* problem)

3. To reconstruct the direction of the γ ray that originated the cascade (a double-output *regression* problem)

The aim of this thesis is to perform a full reconstructions by exploiting the pattern recognition power of deep learning. Other works like (18) have already tried to explore this approach with the use of CNNs and while reporting good and promising results, their chosen architectures did not exploited many of the recent discoveries and innovations in the field. In the next sections I will discuss the application of slightly modified state-of-the-art architectures taken from the computer vision literature for the analysis of data taken by the MAGIC telescopes.



(a) Mean Squared Error as a function of the Epoch **(b)** 2D histogram of the predicted energy vs true energy

Figure 4.1: The optimization evolution of the network and the double histogram of regression performances for the model that reached the lowest validation error

4.1 Energy Reconstruction

The MC data described in 2.2.2.1 was used for the reconstruction of the energy of the original gamma. As explained in 2.2.2.2, diffuse MC was used for training, while ring-wobble was kept for testing. Both the *phe* and *time* signals described in 2.2.2.3 were considered and since the data is taken by two telescopes, the input of the neural network is a tensor of shape $(66 \times 67 \times 4)$.

4.1.1 SE-Inception v3 Single Dense

Different architectures were tested but the one showing the best performance was the Inception v3 developed by (21). Using that as a starting point, I modified the network with the application of SE blocks after each inception block and by the addition of another dense layer (64 neurons) before the final output. This last layer is a single neuron connected by linear weights on which an L2 regularization is imposed. The network have been set-up to learn the base 10 logarithm of the energy, as the linear value was making the learning unstable.

The model was trained for a total of 80 epochs (see Figure 4.1a) with a cosine annealing learning rate schedule inspired by the work of (8). In particular the first 40 epochs were done with an initial LR=0.05, with a cycle length of 2 epochs. From epoch 40 to 60 the initial LR was slightly lowered to 0.045. From epoch 60 to 70 the LR was lowered to 0.040 and the cycle length was set to 1 epoch (so to save one snapshot per epoch).

From this training, two models were proposed:

1. Minimum validation snapshot: the snapshot that reached the lowest loss on the held-out validation data during the entire training history (see Figure 4.1b)

2. The SWA of the last 10 snapshots: as suggested by (11), this model was built by averaging layer-per-layer the weights of the snapshots of the last 10 epochs.

4.1.2 Transfer Snapshot Ensemble: a novel boosting technique

While SWA is a good way enhance the performances of the trained model at a low computational expense by simply averaging the weights of some good snapshots, it is possible to exploit the richness of the diversity of the various snapshots in a different way. By recalling the philosophy of transfer learning: if we have a model trained on a vast amount of data, it is possible to fine-tune the pre-trained model on new slightly different data with little effort. See (22) for more details. It is then possible to create a big network built with k snapshots that are linked together by a differentiable layer. In this context the use of a cosine annealing learning rate technique makes available good and different snapshots at no extra computational expense. As the new resulting model is differentiable, it is possible to fine-tune this ensemble of pre-trained networks on the same data. The rationale behind this approach is that each snapshot have a significantly different weight configuration, see (11). It is thus reasonable to expect that different (meaningful) features are extracted by different snapshots. The problem of appropriately weight each different feature can be solved by gradient techniques. We coin this approach Transfer Snapshot Ensemble (TSE).

As a way of further boosting the performances of TSE, the new model can be trained again with a cosine annealing LR and its last snapshots can be averaged in what could be called TSE-SWA (see Figure 4.2).

4.1.2.1 Learning Rates

For the TSE-SWA starting from the same initialization two different LR have been explored, an high one (0.05) and a low one (0.004) for a total of 10 epochs. The results of the training can be seen in Figure 4.3. The training with a low LR made the model descent in a gradual lower training and validation loss, while with the high LR the model likely explored more significantly different configurations after each epoch. It could have been the case that with more training iteration the high LR would have led to a significantly lower training error. More in general, if one fixes the time budget, it seems more appropriate to set a low learning rate in this phase. More experiments on TSE are left as a future work.

4.1.3 Results

In order to objectively evaluate the performances of the different optimization techniques explored, each model is evaluated on an unseen test set (ring-wobble MC). On this is computed a global test loss as the relative linear error is computed as:

$$Loss = \frac{1}{N} \sum_{i}^{N} \frac{(y_i - \hat{y}_i)}{\hat{y}_i}$$

where y_i is the reconstruction of the i-th event, while y_i is the real value. In Figure 4.4 the results of this evaluation are shown.

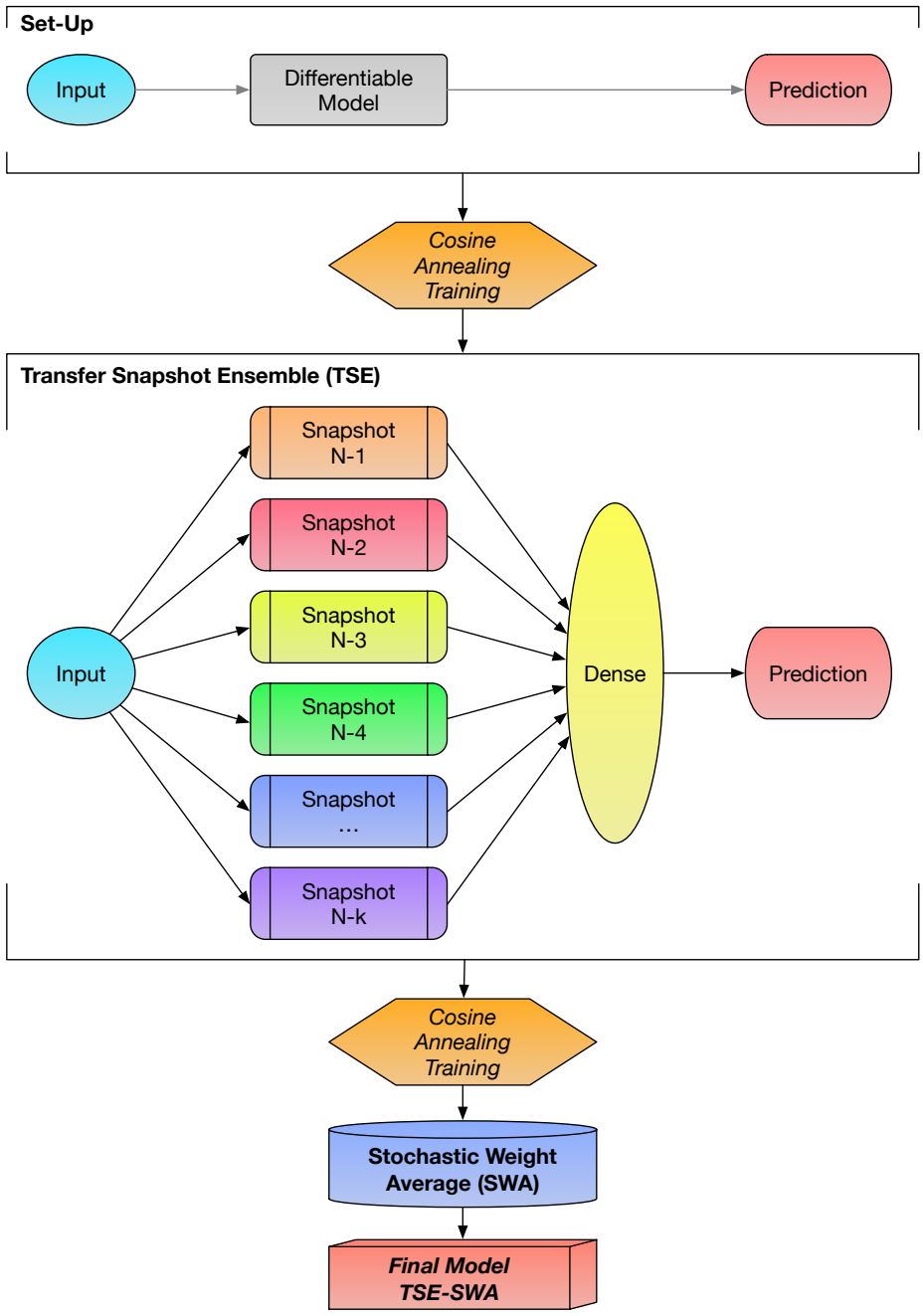


Figure 4.2: Illustration of how TSE-SWA works. After training a differentiable model with a cosine annealing learning schedule for N epochs, one ends with the last k (with$k < N$) models being significantly different from one another. This means that each model captures a different useful aspect of the data for the problem at hand. In order to exploit this richness it is possible to build a new bigger model by linking all the snapshots with a new common hidden dense layer before the prediction. After training it again with a cosine annealing schedule it is possible to take the SWA in of the last epochs in order to maximize the generalization performances.

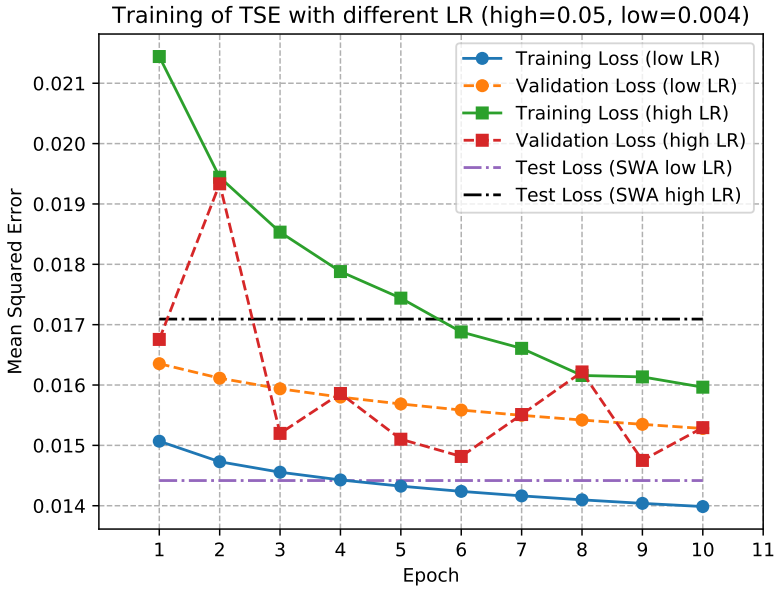


Figure 4.3: High LR vs Low LR

Looking at the picture it can be appreciated how both the SWA and TSE strategies bring an independent benefit to the final generalization ability. The TSE-SWA model is the one with minimum test loss.

As performances varies as a function of energy (it is usually easier to reconstruct the high-energy events), the relative error is studied for different energy bins. The whole energy spectrum is partitioned in logaritmically-spaced energy bins. For each bin the relative error is computed. We expect that its distribution will be gaussian-like with mean μ and standard deviation σ. In order to be resilient to outliers, μ have been computed with the 50-th percentile, while σ from half the difference between the 16^{th} and the 84^{th} percentile (so to get bounds that could hold 68% of the distribution, as the standard deviation). The error distribution for a single model, along with a fitted gaussian for reference, can be seen in Figure 4.5.

Figure 4.4 shows the resulting μ and σ for different energies for the various models. It is interesting to notice that the high LR TSE-SWA network has the smallest bias, but the low LR TSE-SWA has a minimum global reconstruction error. In order to compare the relative enhancement with respect to (1), the improvement defined as $\frac{\sigma_{Aleksic}-\sigma_{NN}}{\sigma_{Aleksic}}$ is shown in the same Figure 4.4. After 1 TeV the CNN reconstruction perform significantly better, achieving almost a 30% improvement for the highest energies, but with a worst performance at the lowest ones.

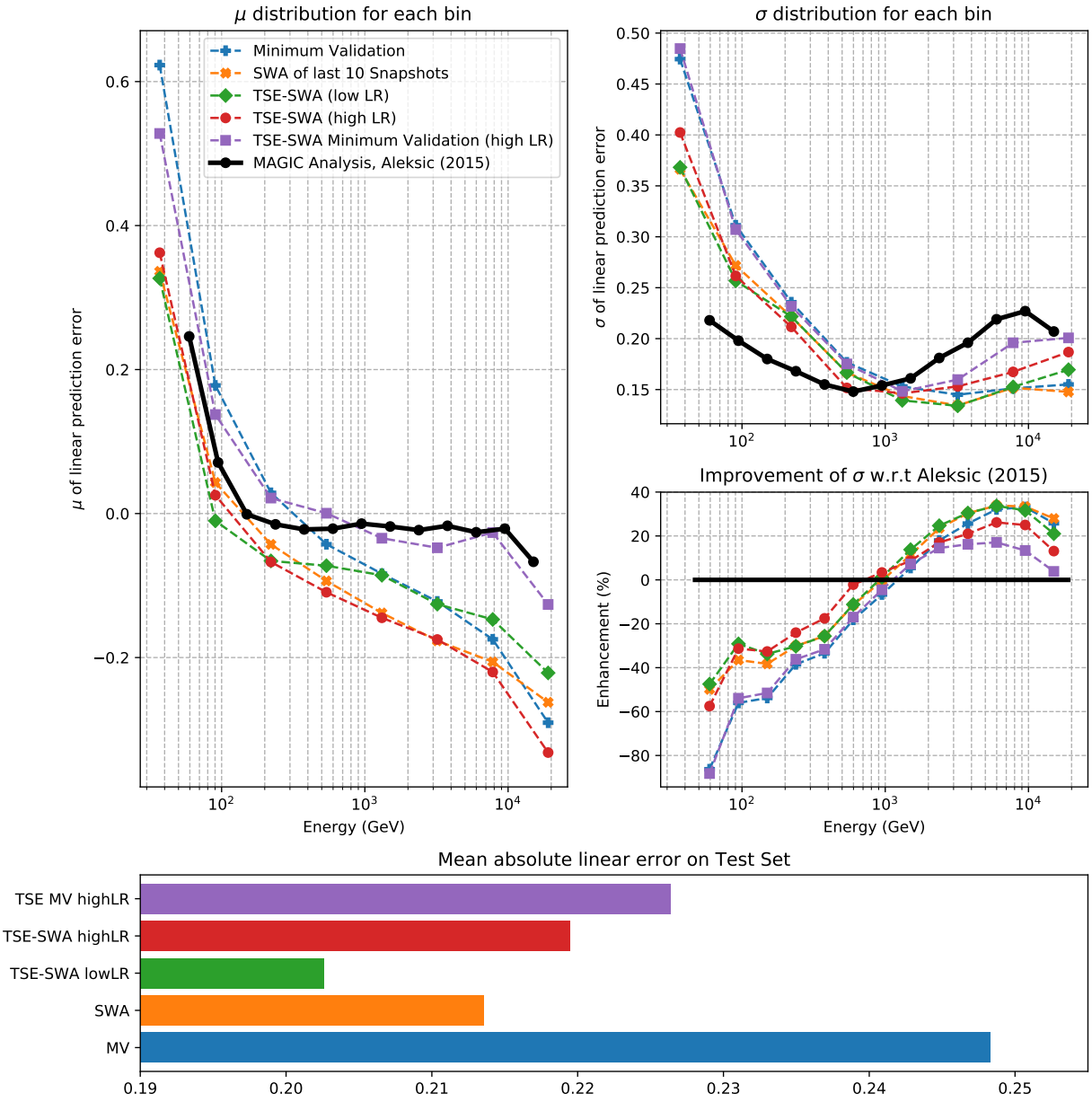


Figure 4.4: Performances of energy reconstruction for the different optimization techniques of the SE Inception-v3 architecture. In the top panels can be seen the distribution of μ and σ of the relative error and the relative improvement of the σ of each model with respect to MAGIC standard analysis. At the bottom the global mean reconstruction error for every energy.

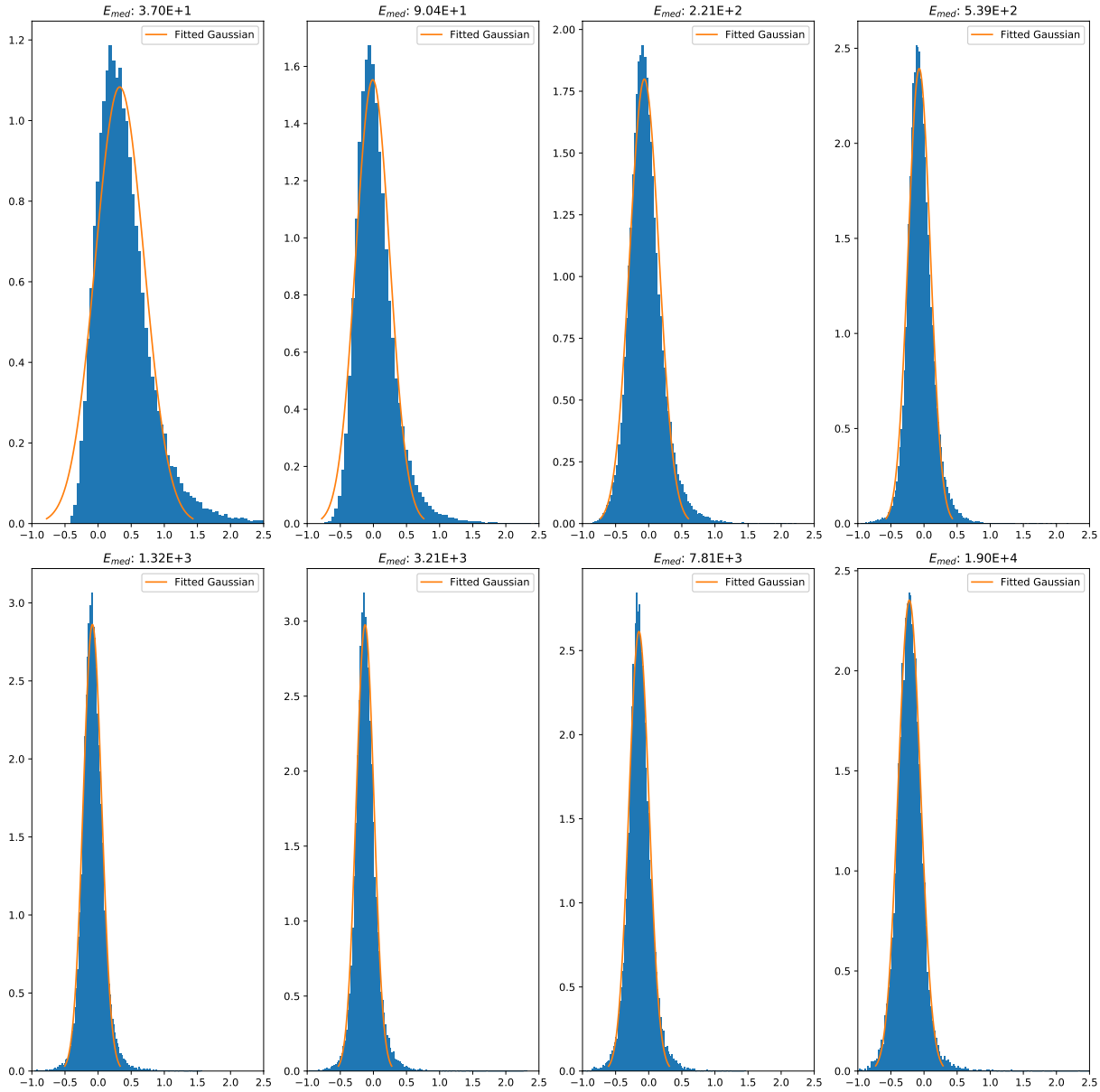
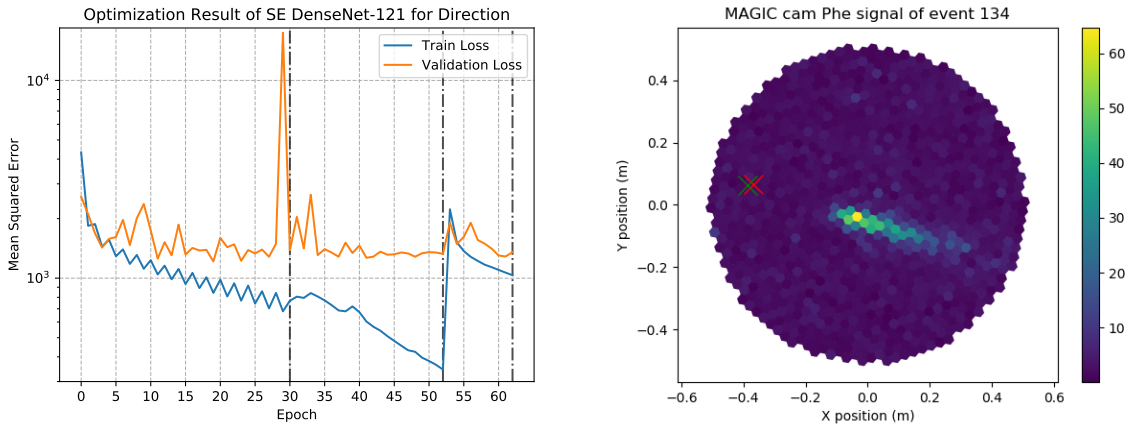


Figure 4.5: How the error decomposition was made. The sigma is the 68-percentile containment of the data, mimicking a Gaussian distribution. In this picture is illustrated the error distribution of the TSE-SWA low LR SE-Inception v3 model.



(a) Training history of SE DenseNet-121

(b) Example of a γ-ray event

Figure 4.6: In (a) the training history of SE DenseNet 121 is depicted, while panel (b) shows how the direction is represented (red cross) and reconstructed (green cross). The camera coordinates then maps to sky coordinates as a function of the direction in which the telescope is pointing.

4.2 Direction Reconstruction

For the reconstruction of the direction, as it was for the energy, the training and validation datasets composed of diffuse MC, while the test dataset was composed by ring-wobble MC. In this case the performance was analyzed by training the models with both *phe* and *time* signal both also with *phe* without *time*.

4.2.1 SE-Densenet 121

Several architectures were tried, but the most effective one was the DenseNet 121, proposed by (9). This 121-layer deep network make vast use of the dense block, promoting feature reuse and lessening the problem of the vanishing gradient. As a way of boosting its performances, a SE block was added at the end of each dense block. The last layer (2 neurons, one for each coordinate) was regularized with an L2 kernel constraint.

The network was set up to learn the direction of the primary particle that caused the EAS. This direction is represented as a tuple of coordinates expressed in mm from the center of the camera (see figure 4.6b).

In order to assess the importance of the *time* information, the network was once trained with it, and once without it (thus only with *phe*). The training with just *phe* lasted 20 epochs with a LR set to 7.5×10^{-4} and a cycle length of one epoch. The training with both *phe* and *time* was split in three rounds (see figure 4.6a):

- ⊙ **Training I**: LR set to 10^{-4} and a cycle length of 2 epochs
- ⊙ **Training II**: LR kept the same but the cycle length was cut down to 1 epoch.

⊙ **Training III**: The LR was set to 1.5×10^{-3} while the cycle length was kept fixed to 1. This have been done in order to see if the network was blocked in a saddle point difficult to escape.

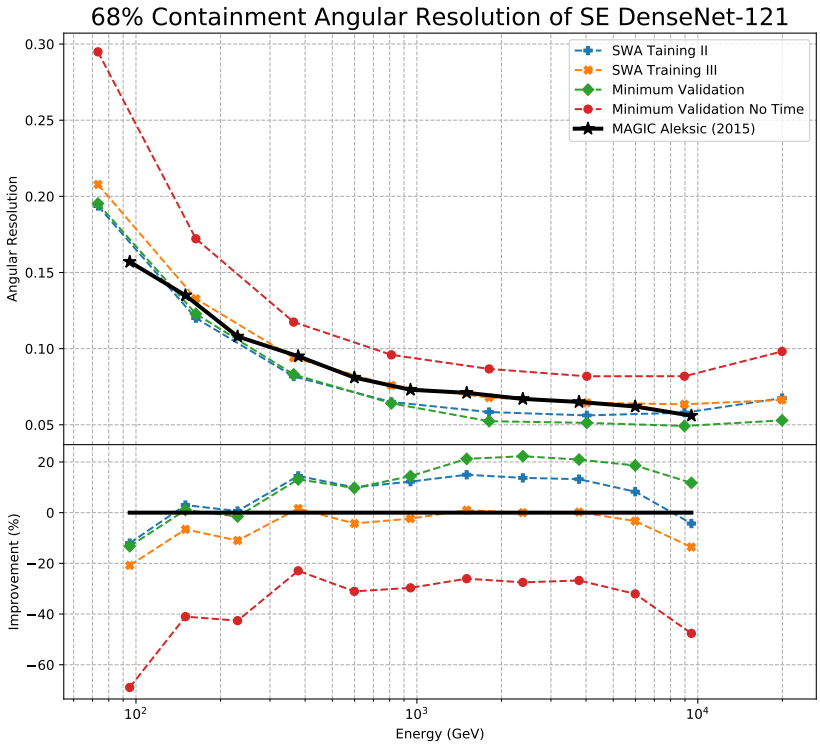


Figure 4.7: Performances in terms of Θ_{68}. The angular resolution is plotted on the top panel, while on the bottom one we can appreciate the relative improvement (in %) with respect to (1).

4.2.2 Results

In order to evaluate the performances of the reconstruction, for different bands of energies have been computed the Θ^2, defined as:

$$\Theta_i^2 = (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2$$

where x_i and y_i represent the true coordinates (in degrees) of the i-th event while $\hat{x}_i$ and $\hat{y}_i$ are the reconstructed ones. This is a measure of the error of the estimate that is linearly proportional to the area of the circle centered at the true position and with radius the reconstructed one. In this way the sensitivity of the instrument have a linear dependence with this parameter.

In order to characterize the performances, for different energy bins we compute a quantity called Θ_{68}, defined as cut in Θ at which the 68% of the distribution is contained. In Figure 4.7 we show the angular resolution of the models as a function of the energy.

It is possible to confidently claim that the time information play an important role in the precise reconstruction of the direction. Compared to the MAGIC standard analysis of (1), the minimum validation model with both *phe* and *time* perform up to 20% better. We remark here that the performance of the SWA network does not improve that of the minimum validation loss as one would expect and was shown for the Energy reconstruction. After testing several hypothesis, the reason remains unclear, although it could point to the different nature of the regression problem we are trying to tackle in this section

4.3 Separation

As mentioned at the beginning of the chapter, in ground-based IACT experiments such as MAGIC is to discriminate between events induced by a γ-ray versus the one induced by an hadronic primary particle. Hadronic showers are much more frequent (with a ratio of at least 2000:1) and they constitutes our background noise. Thus being able to separate them from γ-induced signals in an enhancement of the signal-to-noise ratio (SNR).

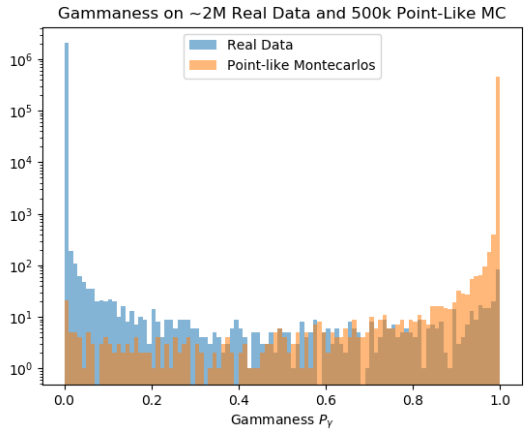
The events containing only accidental photons from the Night Sky Background are rejected by the different level of triggers described in 2.2.1. Despite this all cascades are recorded and therefore a separation needs to be applied. I have described in section 2.2.2 the machine learning techniques used by the MAGIC collaboration to distinguish between gamma and hadron initiated images. In this work, I propose a different and complementary high-level strategy for filtering out events triggered by hadronic showers based on a state-of-the-art CNNs for classification. As we do have available labeled data, this is a supervised binary classification problem where the neural network have been set up to optimize a binary-crossentropy loss. The dataset was constructed with synthetic γ-ray events from a Monte Carlo simulation (see 2.2.2.1) for the γ class and with real acquisition data for the hadronic class. This choice is motivated by a number of reasons:

- ⊙ The MC hadronic simulation is much more computationally intensive with respect to γ simulations, moreover the simulation will always be somewhat *different from reality*.

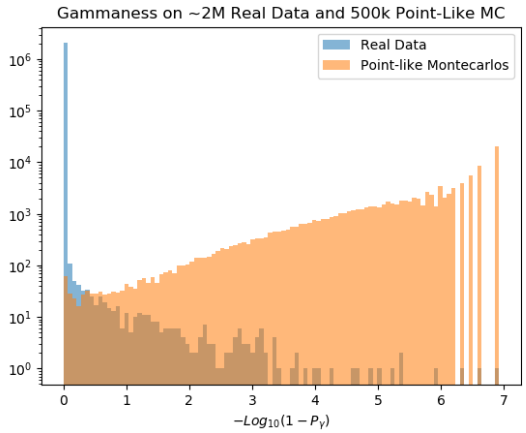
- ⊙ When pointing the telescope to a sky region without a gamma source (as the ones selected), the probability of getting a γ-like event are $< 1\%$. Thus even though the hadronic dataset can be considered "noisy", as it may contain some events of the other class, it is negligibly so.

- ⊙ It is a customary to do so in the standard MAGIC analysis.

The MC γ events *diffuse* were chosen for training (~1 million events), while the *point-like* for testing (~500k events). The hadronic train and validation datasets were built with data taken in different days (so to be robust to changes of atmospheric conditions) and pointing to different parts of the sky. The data was taken from the **Cyg-X3** (7^{th} July 2018) and **1ES2037** (3^{rd} October 2018). The hadronic test class were the data taken on a different direction in the sky in a different day: **SS433** (6^{th} October 2018).



(a) Gammaness Histogram



(b) Logarithmic domain for gammaness

Figure 4.8: Separation power of MobileNetV2 trained on raw data (notice Log scale). The results are evaluated on an unseen test set of 2 milion real data points and more than 500k MC ringwobble events. Results are represented using the Gammaness and the variable χ as suggested in Parsons et al. (to be submitted)

4.3.1 Processing

For consideration that will be made in later, three different sets were produced:

1. Raw set: the data after the calibration step.

2. Mild cleaning: each event is processed with the cleaning procedure of parameters $6, 3.5$. This is the standard cleaning used in MAGIC pipeline

3. Hard cleaning: each event is processed with the cleaning procedure of parameters $10, 5$. This configuration is set to clean almost all the noise in the camera.

4.3.2 First approach: MobileNetV2 on raw data

For the first approach the chosen architecture was a MobileNetV2 from the work of (16), with $\alpha = 1$ and one-class sigmoid activation, optimized using the snapshot technique with binary crossentropy loss. In this way, the CNN outputs a single scalar bounded in [0,1]. Since in the training phase it was chosen to label γ-like events as 1 and hadrons as 0, the output of the network can be interpreted as "how much" that the event is a gamma. In this sense we shall call this final output "gammaness". This architecture was chosen as it reaches high scores in standard computer vision datasets, while keeping the memory footprint and computational load small. This also implies a faster training procedure with back-propagation.

After a few epochs the network reached perfect separation of the training data and a validation accuracy $> 99.99\%$, a plot of the distribution of the gammaness of MC and real data can be seen

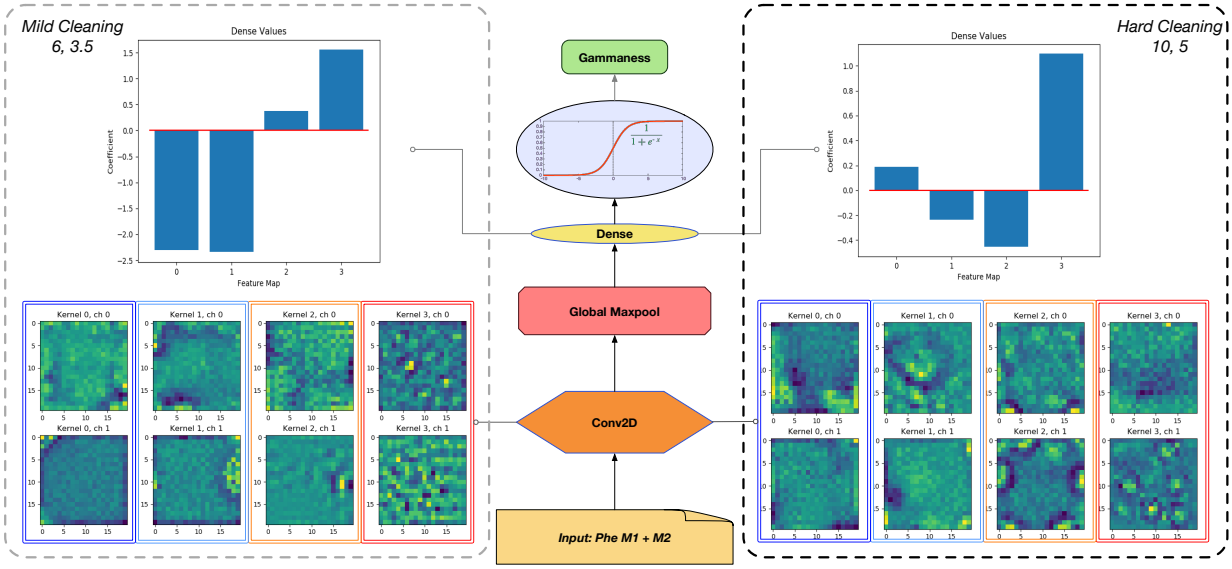


Figure 4.9: SimplicioNet structure and weight disposition after training. On the left, the final network after being exposed to the data cleaned with parameters 6, 3.5 (mild cleaning). On the right, the model after convergence reached with data cleaned with parameters 10, 5 (hard cleaning).

in Figure 4.8a. These results are unrealistically good as we do expect more gamma-like events in the real-data caused by high energy electrons which produce the exact same image as a γ.

Consequently, the hypothesis is that the neural network is indeed finding a perfect way to separate the classes, but not by discriminating γ from hadrons. The model is separating simulation from reality.

4.3.3 An interpretable Neural Network: SimplicioNet

In order to try understand how the CNN approach was so efficient in making the separation I built the most simple possible CNN. This CNN, called *SimplicioNet*[1], is made of just one layer of 4 kernels of 20x20 pixels followed by a ReLU activation. Over this, the information flows in a maxpool that selects the most prominent kernel activation. This is then combined by a linear layer in a single output and is followed by a sigmoid activation. The sigmoid has the property of squeezing the output between 0 and 1. Because of this construction, we can claim that the feature maps that are weighted by a positive number contribute to the gamma class and the ones weighted by a negative number contribute to the hadron class. The network was trained once with *phe* signal only from mild-cleaned data (6, 3.5) and once with *phe* signal only from hard-cleaned data (10, 5). The structure is showed in figure 4.9.

[1]The name is a tribute to the work of Galileo Galilei "Dialogo sopra i due massimi sistemi del mondo"

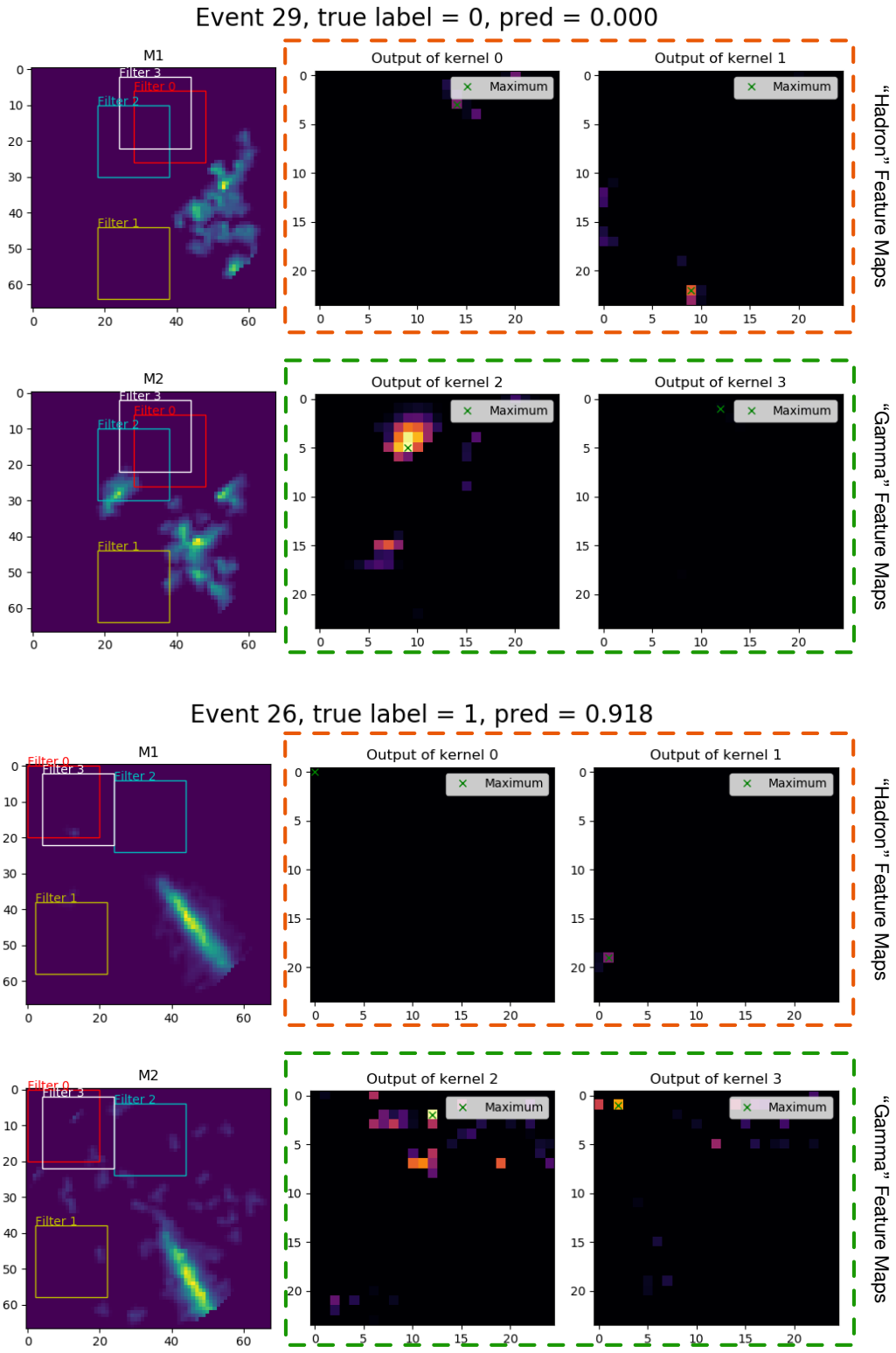


Figure 4.10: Receptive fields and activation maps of SimplicioNet trained and evaluated with data mild cleaned (parameters 6, 3.5). We represent two events seen by the two different MAGIC telescopes. Notice how the receptive field of the kernels is not looking at the significant signal, but is mainly focused on the background.

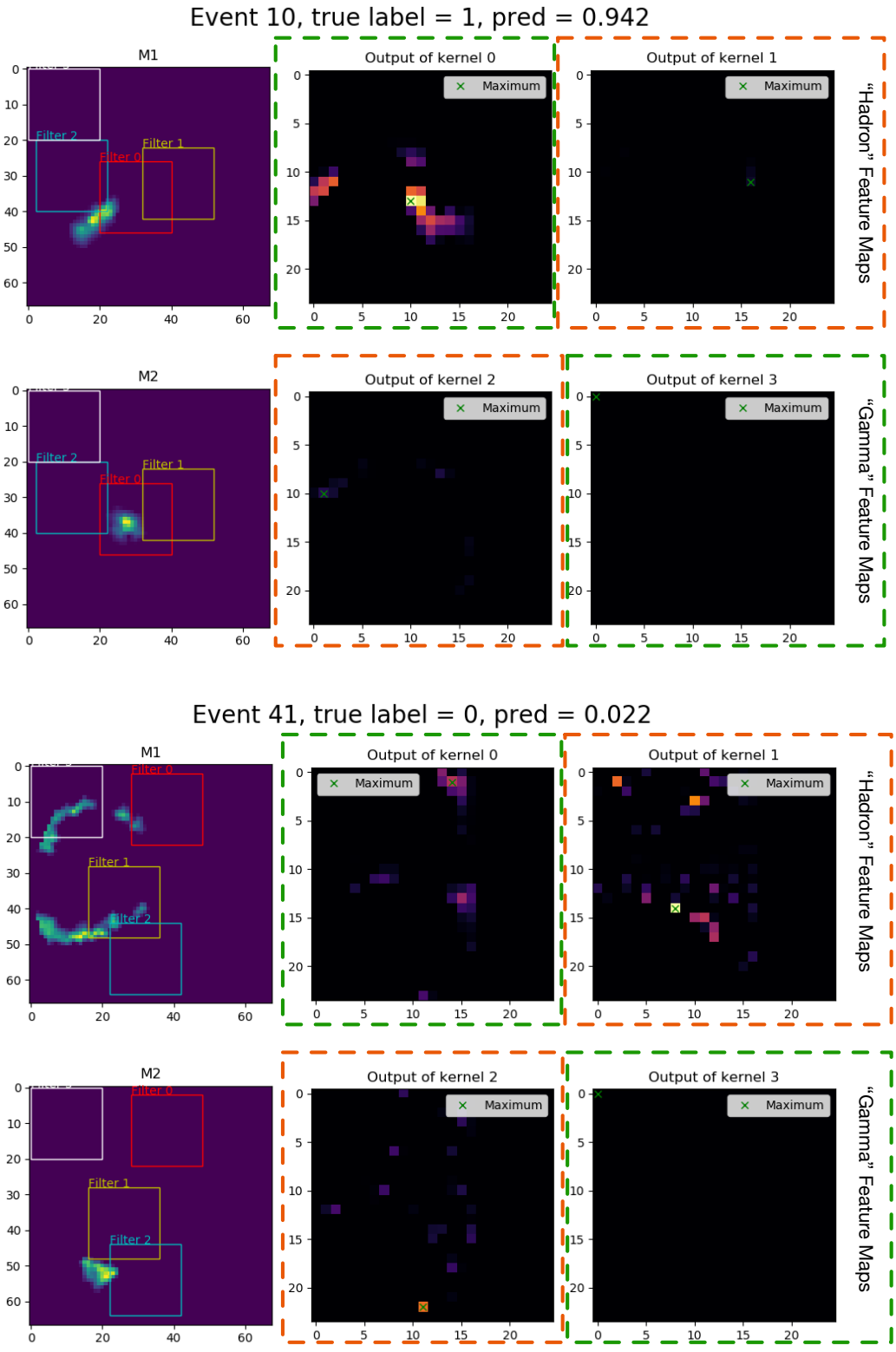


Figure 4.11: Receptive fields and activation maps of SimplicioNet trained and evaluated with data hard cleaned (parameters 10, 5). We represent two events seen by the two different MAGIC telescopes. Notice how the receptive field of the kernel is now focusing on the significant signal.

4.3.3.1 Mild-cleaning SimplicioNet

Trained on the mild cleaning, SimplicioNet's Dense values tell us that kernel 0 and 1 are responsible for the election of class 0 (hadrons), while kernel 2 and 3 are responsible for the detection of class 1 (gamma like).

The network reached a validation accuracy > 95% in just 7 epochs. For better understanding how such accuracies were reached, a plot of the receptive field of the decisive features is shown for various events. As can be seen by the pictures in Figure 4.10, the kernels focused not on the significative signal, but rather on some discriminative feature of the background noise that made the simulation separable from the real data.

4.3.3.2 Hard-cleaning SimplicioNet

SimplicioNet was then trained again from scratch on the hard-cleaned (10, 5) dataset. By observing the dense values, this time kernel 0 and 3 are responsible of the gamma class while 1 and 2 of the hardron. In this context it reached a validation accuracy of $\sim 83\%$. When investigating on what its filters were focusing on, it was more evident that the receptive fields was on some part of the significative signal. Some examples of these filters are shown on Figure 4.11

4.3.4 MobileNetV2 on hard-cleaned data

Being reassured by the fact that with the hard-cleaning the artifacts of the simulation were at least significantly suppressed, the MobileNetV2 was trained from scratch on it. At convergence it reached a validation accuracy of $\sim 93\%$, becoming a good candidate for a valid model.

4.4 Final Test: Evaluation of the whole new pipeline on the Crab Nebula

We have shown how the models tested performed on simulated datasets. Reality is different from the simulation (as section 4.3.2 testify), thus a full analysis on real data should be carried out in order to fairly evaluate them. In this particular domain, the Crab Nebula (a supernova remnant) is appropriate as it is the most studied object in the sky from many different instruments and the standard candle and calibration source in VHE gamma-ray astronomy (4). Being also an active point source of gamma rays with a Poisson distribution, it is the the perfect candidate to evaluate the whole pipeline developed up until now.

A dataset of 1774 seconds (roughly 30 minutes) of observation taken from two different acquisitions have triggered 471727 events. These have been interpolated as they were for the reconstruction of energy and direction, while for the application of the separation algorithm they were first cleaned with the hard-cleaning procedure.

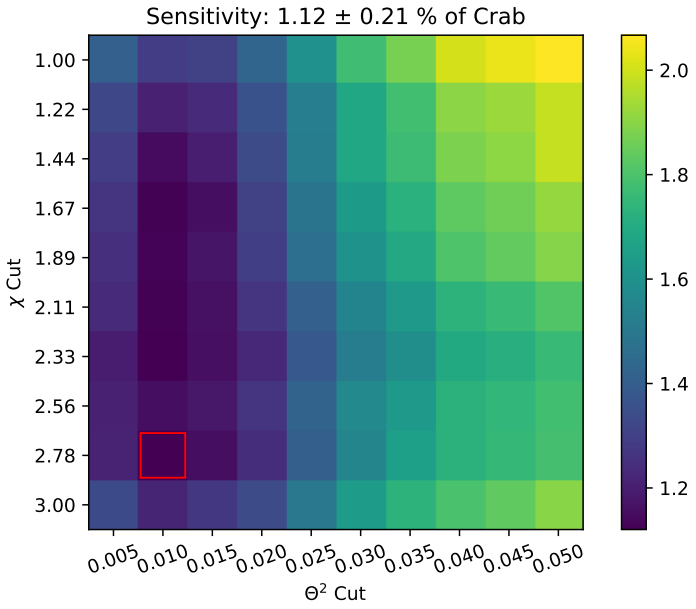


Figure 4.12: Grid search optimization for finding the best cut in gammaness and Θ^2 in order to minimize the sensitivity, measured as percentage of the Crab Nebula. By filtering in Θ^2 we are selecting the gamma-like events coming from the source direction. The search is done in the variable $\chi = -\log(1-gammaness)$) and by considering only events whose reconstructed energies were larger than 150GeV. The sensitivity is computed from the significance for which, while respecting the condition $N_{signal} - N_{background} > 10$ and $N_{signal} - N_{background} > 0.05 \cdot N_{background}$

4.4.1 Condition for Detection

In order to to claim the detection of any point-like source it is necessary to detect with a statistical confidence of 5σ the presence of a signal. Since the signal and background events follow a Poissonian distribution, the significance can be approximated as

$$S = \frac{N_{signal} - N_{background}}{\sqrt{N_{background}}}$$

where N_{signal} is the number of events classified as gammas and reconstructed with a $\Theta^2 < \Theta^2_{cut}$, while $N_{background}$ is the number of events classified as gammas and reconstructed in a background region of the same size as the signal one, distant from the expected source. In order to have a more reliable estimate of $N_{background}$ three areas are considered, corresponding to the rotation of the true source position by 90°-180°-270° with respect to the center of the camera. $N_{background}$ is then computed as the sum of the events reconstructed in these locations and weighted by α, defined as the ration between the area of the expected signal and the area of the background region, in this case $\alpha = 1/3$.

4.4.2 Results

The homogeneous distribution of all the reconstructed events in the top left panel of Figure 4.13 testify the good direction reconstruction power of SE DenseNet-121. As we apply a more and more discriminating gammaness cut, defined as the value above which an event is considered a gamma from the classifier, the surviving directions are reconstructed only close to the true position of the Crab Nebula (known from other kind of messengers).

In order to compute the optimal values of the gammaness cut and the Θ^2_{cut}, a coarse grid search for the maximization of the statistical significance of the detection (which correspond to the minimization of the sensitivity) is performed and is shown in Figure 4.12. The results of this optimization on the Crab did not partition the dataset into train and test subsets because we did not have enough data (not because it has not been taken, but because of thesis time constrains). We tried not to optimize too much on fluctuations by performing loose cuts, but we understand that this is not fair. In chapter 5 I reference how a more fair comparison should be done with a larger Crab dataset, and is thus left as a future work. Nonetheless the whole pipeline reaches a sensitivity of $1.12 \pm 0.21\%$ of the Crab nebula, meaning that it can detect in 50 hours a source which has a flux as low as $1.12 \pm 0.21\%$ of the Crab.

The detection power of the pipeline can be appreciated in Figure 4.14 where Θ^2 is plotted for the signal against the background (also called "off"). In 1774 seconds the Crab is detected with a significance of $44 \pm 8\sigma$, meaning that it can be detected (with a statistical confidence of 5σ) in $t = 1774 * (5/44)^2 = 22.63$ seconds.

We did not calculate the sensitivity using MC because the neural network, even after the 10-5 separation was applied, was still identifying gammas extremely good (many gammas kept even for very tight cuts in gammaness). As a future work in chapter 5 we propose either to calculate the sensitivity directly using the Crab real data or train the CNN on MC.

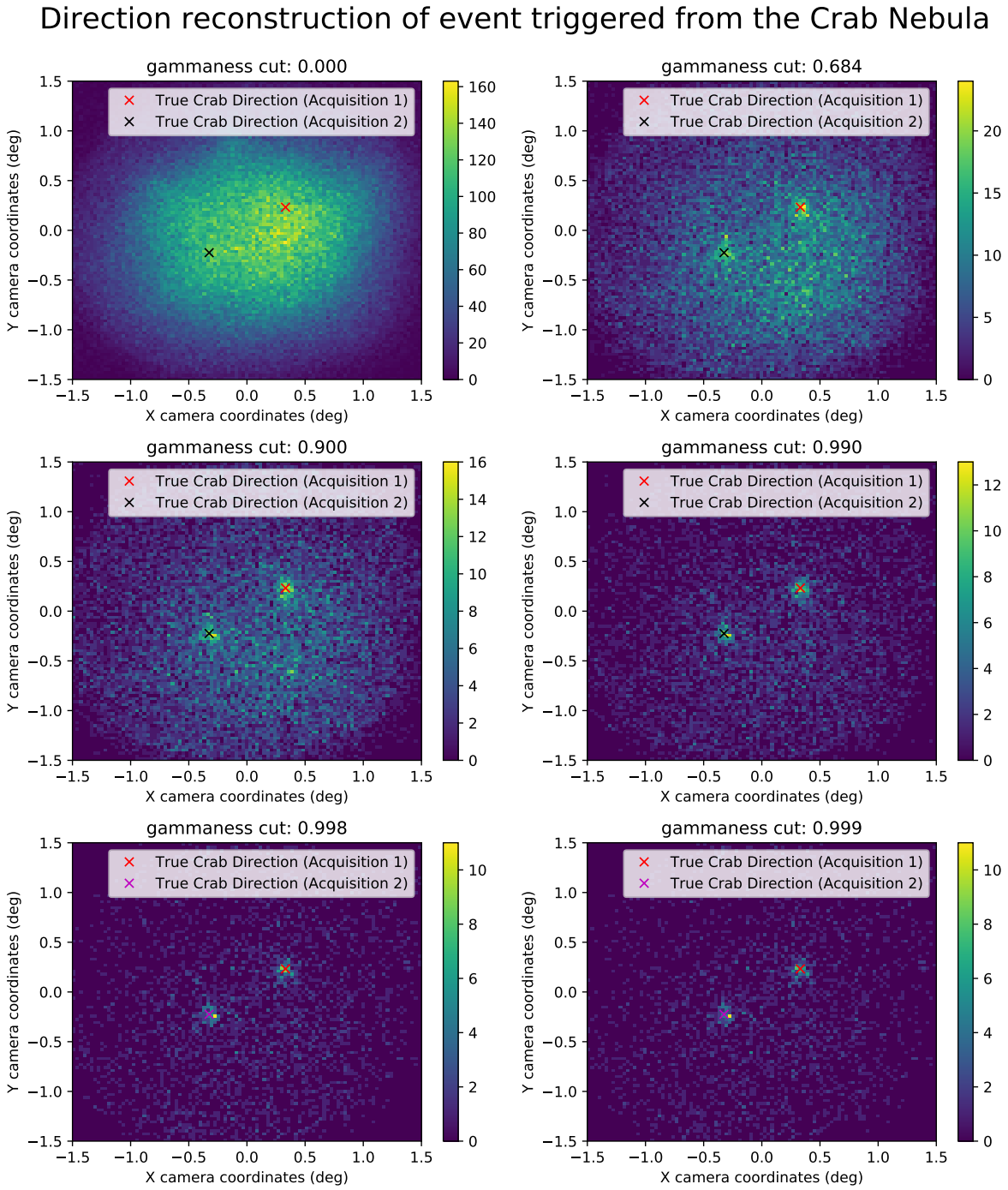


Figure 4.13: 2D histogram of the reconstructed direction of the Crab Nebula as a function of the gammaness cut

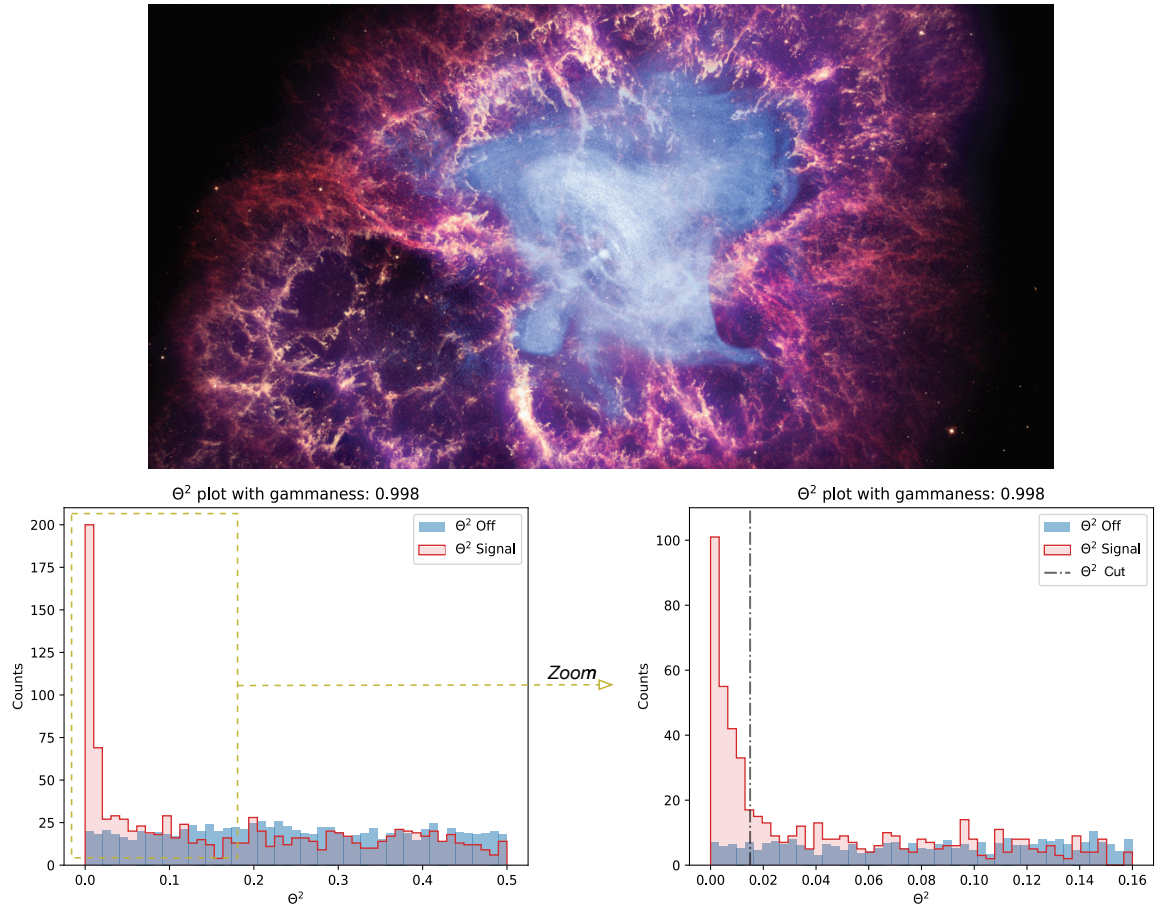


Figure 4.14: Top: the Crab Nebula as seen by combining the information from the X-Ray (Chandra), optical (Hubble) and infrared (Spitzer) data. Bottom: the Crab Nebula detection with the analysis developed in this thesis. The plot depicts the histogram distribution of Θ^2, which represents the number of events that happen to be reconstructed close to the true position of the nebula. With a cut in $gammaness > 0.998$, $\Theta^2 < \Theta^2_{cut} = 0.01$, Energy > 150 GeV we have $N_{signal} = 196$, $N_{background} = 50$ (with three $N_{background}$). By considering $\alpha = 1/3$ (as the ratio of the size of the signal with the size of the background) for appropriately weighting the signal against the three backgrounds, the Crab is detected with a significance of $44 \pm 8\sigma$ in 1774 seconds of observation.

5
Conclusions and Outlook

In this work I designed and performed a novel full analysis for the MAGIC telescopes using CNNs. This has been done by setting up the reconstruction of energy and direction as regression problems while the γ/hadron separation as a binary classification problem.

In the pursue of the best possible performances a novel boosting technique, TSE, has been proposed and evaluated on the energy reconstruction problem, leading to a significant improvement of $\sim 30\%$ with respect to the standard implemented analysis of (1) for events with energies above 1 TeV. We reserve as a future work a more in-depth study of the presented TSE boosting technique.

The reconstruction of the direction resulted in an improvement of almost the 20% with respect to the state of the art. It is worth mentioning that any improvement in the angular resolution translates directly in an improvement of the sensitivity. In this problem the most generalizing model was the one selected with the criterion of the minimum loss on the validation set, a different result with respect to the experiments performed on the energy reconstruction. The TSE boosting technique was not implemented on this task and is reserved as a future work.

The separation was a challenging task due to the significative differences in the simulated versus real datasets that the neural networks were able to model. In order to assess this fact, a simple and interpretable model was built: *SimplicioNet*. It helped to show how the network was using features in the background for the separation of the two classes. As a way around, a standard cleaning procedure from literature was used in order to minimize the simulation artifacts and to provide a reasonable separation of the events. This is suboptimal as the power of deep learning relies on being able to extract meaningful patterns from raw data. As a future work, two paths are possible in order solve this problem:

1. Use a simulated datasets also for the background class. This would help eliminating any

real life feature the network can be taking to differentiate between real data and MC simulations.

2. Superimpose the pedestal noise recorded from the real data on the MC (20 events of pedestal per second is taken with random trigger every data run). Like that, we could eliminate any possible mismatch between the simulated and measured noise.

In order to evaluate the new reconstruction pipeline as a whole, a full analysis was conducted on real data taken from the Crab Nebula. The source is indeed detected with a statistical significance of 44σ in 1774 seconds of observations, leading to an overall integral sensitivity of 1.12% of the Crab Nebula flux above 150 GeV. This means that in 50 hours of observations it is possible to claim the existence of a γ-ray source in the sky which has a flux as low as 1.12% of the Crab Nebula with a confidence of 5σ. This final test consecrate the validity of the new approach.

With an enhancement of the separation power, as it is expected by the proper application of deep learning methods on appropriate datasets, the overall sensitivity is expected to lower below the actual MAGIC analysis which is currently of 0.84 ± 0.02% (as the direction reconstruction done in this work is significantly better). In order to better characterize this new approach a new full analysis on 50 hours of Crab observation will provide enough statistics to confidently compare the differential sensitivity with the actual analysis.

References

[1] Aleksić, J., Ansoldi, S., Antonelli, L., Antoranz, P., Babic, A., Bangale, P., Barceló, M., Barrio, J., González, J. B., Bednarek, W., et al. (2015). The major upgrade of the magic telescopes, part ii: A performance study using observations of the crab nebula. *Astroparticle Physics*, 72:76–94.

[2] Arora, R., Basu, A., Mianjy, P., and Mukherjee, A. (2016). Understanding deep neural networks with rectified linear units. *CoRR*, abs/1611.01491.

[3] Brun, R. and Rademakers, F. (1997). ROOT: An object oriented data analysis framework. *Nucl. Instrum. Meth.*, A389:81–86.

[4] Bühler, R. and Blandford, R. (2014). The surprising crab pulsar and its nebula: a review. *Reports on Progress in Physics*, 77(6):066901.

[Engel et al.] Engel, R., Heck, D., and Pierog, T. Extensive air showers and hadronic interactions at high energy, journal = Ann. Rev. Nucl. Part. Sci., volume = 61, year = 2011, pages = 467-489, doi = 10.1146/annurev.nucl.012809.104544, slaccitation =.

[5] Fernández Barral, A. (2017). *Extreme particle acceleration in microquasars jets and pulsar wind nebulae with the MAGIC telescopes / author: Alba Fernndez Barral ; director: Dr. Oscar Blanch Bigas ; tutor: Dr. Enrique Fernndez Snchez.* PhD thesis. Bibliografia.

[6] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[7] Hu, J., Shen, L., and Sun, G. (2017). Squeeze-and-excitation networks. *CoRR*, abs/1709.01507.

[8] Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. (2017). Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*.

[9] Huang, G., Liu, Z., and Weinberger, K. Q. (2016). Densely connected convolutional networks. *CoRR*, abs/1608.06993.

[10] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org.

[11] Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2018). Averaging weights leads to wider optima and better generalization. *CoRR*, abs/1803.05407.

[12] Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, abs/1609.04836.

[13] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[14] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324.

[15] Mirzoyan, R. (1997). On the Calibration Accuracy of Light Sensors in Atmospheric Cherenkov Fluorescence and Neutrino Experiments. *International Cosmic Ray Conference*, 7:265.

[16] Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., and Chen, L. (2018). Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381.

[17] Schultz, C. (2013). *Development of New Composite Mirrors for Imaging Cherenkov Telescopes and Observations of the Two Blazar Objects 1ES 0806+524 and 1ES 1011+496 with MAGIC*. PhD thesis, Padua U.

[18] Shilon, I., Kraus, M., Büchele, M., Egberts, K., Fischer, T., Holch, T. L., Lohse, T., Schwanke, U., Steppa, C., and Funk, S. (2019). Application of deep learning methods to analysis of imaging atmospheric cherenkov telescopes data. *Astroparticle Physics*, 105:44–53.

[19] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

[20] Smith, L. N. and Topin, N. (2017). Super-convergence: Very fast training of residual networks using large learning rates. *arXiv preprint arXiv:1708.07120*.

[21] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826.

[22] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning. *CoRR*, abs/1808.01974.

[23] Wagner, R. M. (2006). *Measurement of very high energy gamma-ray emission from four blazars using the MAGIC telescope and a comparative blazar study*. PhD thesis, Max-Planck-Institut für Physik, Föhringer Ring 6, 80805 München, Germany.

Acknowledgements

Nessuno mai lavora da solo e questa tesi sicuramente non è un'eccezione. Questa tesi è il frutto di un grande lavoro messo in piedi mattoncino per mattoncino grazie agli stimoli (più o meno consapevoli) delle persone che mi sono state intorno. Vorrei innanzitutto ringraziare l'Università degli Studi di Padova, in particolare il dipartimenti di ingegneria dell'informazione, per avermi permesso di seguire un percorso di altissimo livello formativo lasciandomi ampia libertà di esplorare temi per cui ho sempre provato un forte interesse.

Ringrazio il professore Alessandro Chiuso per aver accettato il ruolo di relatore per questo lavoro i cui temi trattati sono diversi da quelli di una tipica laurea in ingegneria. Ringrazio il professore Alberto Testolin per i preziosi consigli sul Deep Learning e per essere sempre stato disponibile a un confronto sulle possibili metodologie da utilizzare. Un grandissimo ringraziamento va al mio co-relatore dell'INFN Rubén López Coto, la persona che da più vicino mi ha seguito nell'arco degli ultimi mesi. Oltre alla sua enrome disponiblità e prontezza nella risposta, lavorare al suo fianco è stata una potente fonte di stimoli e crescita. Egli mi ha infatti insegnato, oltre a mille piccoli dettagli importanti, qualcosa che non si può imparare sui libri: un profondo senso critico verso il proprio lavoro e una forte fedeltà nella ricerca della verità empirica, sopratutto quando essa si discosta dai risultati di una sudata analisi a cui è facile affezionarsi. Questa è forse l'essenza più pura del metodo scientifico.

Un forte ringraziamento ed abbraccio va alla mia famiglia che non ha mai smesso di credere in me. Un grazie a mia sorella Emma per il suo costante supporto nonostante i momenti difficili della vita, un grazie a mia madre per avermi insegnato l'importanza della bellezza nelle cose e un grazie a mio padre per avermi cresciuto trasmettendomi il fascino per l'universo e il desiderio di riolvere i misteri di come esso funziona. Ringrazio mia nonna per essermi sempre stata vicina, non ultimo nel preparami ogni giorno gustosi pranzetti in cui mi dava l'opportunità di spiegarle i miei progressi, spesso accendendo alcune scintille che sono state chiave per i risultati di questa tesi.

Un caro grazie ai miei colleghi di università, il cui confronto e dialogo è sempre stato apprezzato come il migliore dei regali. In ultimo, ma non di importanza, un grazie a tutti i miei amici e compagni di ventura che hanno condiviso con me il mio tempo libero. La vostra amicizia è il mio tesoro più prezioso.

Aprile 2019