# Convolutional Neural Networks for H.E.S.S.

Master's Thesis in Physics

Presented by
**Tobias Fischer**
30.04.2018

Erlangen Centre for Astroparticle Physics
Friedrich-Alexander-Universität Erlangen-Nürnberg


Supervisor: Prof. Dr. Stefan Funk

## Abstract

The H.E.S.S. experiment utilizes a set of Imaging Atmospheric Čerenkov Telescopes (IACT) to detect very high energy (E>100 GeV) $\gamma$-rays. The collected data is passed through a multi stage reconstruction chain, to suppress cosmic ray background and recovering direction and energy of $\gamma$-rays. Calibration of the chain is achieved via Monte Carlo simulations of particle showers, their propagation through the atmosphere and the detector response.

While older standard methods are based on hand engineered features, Convolutional Neural Networks (CNN) learn features by themselves. CNN are in principle capable of IACT image analysis, but can only partially compete with state of the art methods. This work presents some basic considerations for preprocessing but also difficulties arising with more sophisticated image analysis methods.

CNN can also distinguish simulations from real data exceptionally well, suggesting a strong discrepancy. It is well known that high energy particle interaction simulations are merely an approximation, uncertainties of atmospheric propagation and telescope hardware simulation have not yet been thoroughly studied. While there are some ways to alleviate the issue, CNN can ultimately be a tool to increase simulation realism.

# Contents

# 1 Introduction

The detection of cosmic rays by Victor Hess in the early 19th century started a new era in particle physics. At the time no large scale particle accelerators existed [1], thus cosmic rays were the sole source of very high energy radiation. The discovery of the positron and muon by Carl Anderson was just the beginning of a series of new discoveries - which soon started to pile up and form a whole zoo of particles which was later unified and described by the standard model of particle physics.

As of today, cosmic rays carry energies multiple orders of magnitude higher than any anthropogenic accelerator can provide. A few dozen particles of $E > 10^{20}$ eV have been detected [2], which is $\approx 10^7$ times the LHCs center of mass energy. However, high luminosity, precisely controlled primary particle properties and mature detector technology of todays colliders render them more capable of detecting and measuring new particle properties.

The scientific community usually distinguishes cosmic rays from $\gamma$-rays - a reasonable choice, since charged cosmic rays are deflected by electromagnetic fields during their propagation through space. Uncharged, massless $\gamma$-rays travel rather straight and can thus be used for astronomical purposes to produce an energy dependent skymap. Cosmic rays and $\gamma$-rays are accelerated by complex mechanisms throughout our universe, making those mechanisms a subject of study by themselves. Typical candidates are e.g. active galactic nuclei, pulsars, supernova remnants leading to diffuse or shock acceleration. An overview is given in [3, 4].

Different classes of $\gamma$-ray detectors like the space-based Fermi-LAT [5, 6], or ground based experiments like HAWC [7] and Imaging Atmospheric Čerenkov Telescopes (IACT) are utilized to detect $\gamma$-ray sources. A multitude of IACT is currently in operation, including H.E.S.S., MAGIC and VERITAS [8–10] - all of which are actually IACT Arrays (IACTA) to benefit from stereoscopic vision; with CTA [11] a larger array is planned at two locations on the northern and southern hemisphere. There are also joint observations utilizing multiple detector systems [12]. Interestingly enough, IACT can in principle also be used for neutrino-astronomy [13] or entirely new physics. Searches for magnetic monopoles [14], axion like particles and multiple other non-standard physics phenomena were conducted [15, 16]. A comprehensive overview of $\gamma$-ray detectors - including but not limited to the aforementioned - can be found in [17].

IACT require sophisticated image analysis algorithms in order to distinguish $\gamma$-rays from cosmic rays and to reconstruct $\gamma$-ray origin and energy. Some approaches have been around for decades (i.e. the Hillas method [18]), but can limit the instruments resolution [19]. It is noteworthy that any new analysis chain can be applied to all past data. This includes events that were formerly ignored due to preselection cuts for other methods.

More advanced methods than Hillas have been developed, utilizing state of the art learning algorithms like boosted decision trees [20–22]. Machine learning itself is an incredibly active field of study, empowered by the exponential growth of computing power and amount of data [23]. Currently the most powerful image processing methods utilize Convolutional Neural Networks (CNN). Unlike most of its predecessors algorithms, CNNs are fed with raw image data instead of some derived values (i.e. features), thus acting as a feature extractor [24].

Training a learning algorithm in a supervised fashion requires annotated data. Such data is not naturally available for IACT, but can be synthesized via sophisticated simulations. This includes shower development and light propagation in the atmosphere, detector optics, and finally behavior of instrument hardware. For H.E.S.S. this is achieved via CORSIKA and the `sim_telarray` package [25].

In this work simulations of H.E.S.S. events are utilized to train convolutional neural networks. Standard tasks are the classification of primary particle type (i.e. background suppression) and regression of primary particle properties (e.g. initial energy and direction). These tasks have been tackled before [26–30], this work is an evolution of the methods used in previous works.

This work shows that it is possible to train a network to distinguish between real data and simulations that should resemble the same kind of data. Such problematic behavior could be alleviated by either reducing the discrepancy between simulation and real data (i.e. increasing simulation realism) or by blinding the reconstruction method to the differences.

# 2 $\gamma$-ray astronomy with IACT

The first breakthrough in $\gamma$-ray astronomy was the detection of a constant TeV $\gamma$-ray flux from the crab nebula by the Whipple observatory in 1989 [31]. Since then over 200 sources have been detected [32], 60 alone during the H.E.S.S. galactic plane survey [33]. Continuous monitoring yields observations of flares [34, 35], i.e. sudden high flux outbursts of known sources. Combination of multiple instruments' observations enables the classification of source type. Even if a large part is not classified yet, most sources are pulsar wind nebulae (PWN) or supernova remnants (SNR) [32].

## 2.1 Air Showers

Air showers are particle cascades emerging from interaction of primary high energy cosmic rays and $\gamma$-rays with the terrestrial atmosphere. A quantitative overview of different cosmic ray source types, respective spectra and composition can be found in [36]. $\gamma$-rays are produced via interaction of high energy particles with matter and radiation fields. The two main mechanisms are $\pi^0$-decay and inverse Compton scattering [37]. $\pi^0$s are created during scattering of relativistic nuclei (including protons), with a dominant decay channel $\pi^0 \rightarrow 2\gamma$ [38]. Inverse Compton up-scattering of low energy photons (e.g. the cosmic microwave background and optical light) with relativistic electrons also yields $\gamma$-rays. Other production processes like bremsstrahlung and synchrotron radiation are discussed in [4].

High energy particles reaching earth scatter with atmospheric nuclei, resulting in a shower of secondary particles. For $\gamma$-rays the main processes are bremsstrahlung and pair-production resulting in an electromagnetic shower [39]. An incident $\gamma$-ray is converted into a charged lepton-antilepton pair in a nuclear Coulomb field. A conversion without recoil partner is impossible due to conservation of energy and momentum. For the relevant energies the leptons are almost exclusively of the first generation, the second generation is strongly suppressed [40]. The electron and positron carry equal amounts of energy on average. They will emit bremsstrahlung in form of a $\gamma$-ray upon passing some recoil partner similar to pair production. The bremsstrahlung's $\gamma$-ray will then produce another pair and so forth. This exponential production of particles stops at electron energies of $\approx 80$ MeV, where ionization starts to dominate the energy loss. Note that primary cosmic ray electrons and positrons can also initiate an electromagnetic shower.

For protons and heavier nuclei, hadronic interactions - mediated via the strong force - have to be considered additionally. Compared to leptons, hadrons show a much richer spectrum of interactions and intermediate states. Thus hadronic showers are very diverse and usually contain multiple hadronic and electromagnetic sub-showers. Intermediate particles are nuclear fragments, mesons (mostly $\pi$) and baryons (including hyperons).

While uncharged $\pi^0$ immediately decay into two photons, $\pi^\pm$ decay less quickly and can interact with the atmosphere before decaying via $\pi^\pm \rightarrow \mu^\pm + \nu_\mu/\bar{\nu}_\mu$. Another notable difference to electromagnetic showers is the relatively high traverse momentum of secondaries. A comparison of hadronic and electromagnetic mechanisms is shown in Fig. 2.1.

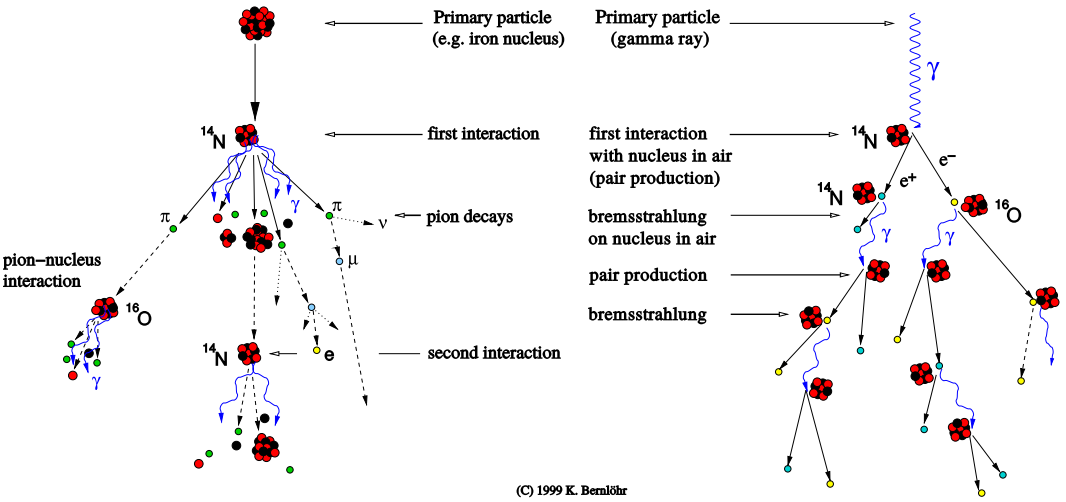


**Figure 2.1:** Illustration of hadronic and electromagnetic shower development in the atmosphere. **Left:** Hadronic shower containing multiple hadronic interactions, particle decays and γ-ray production. **Right:** Electromagnetic shower evolving by successive pair-production and bremsstrahlungs processes. Graphics adapted from [41], courtesy of Konrad Bernlöhr.

Secondary charged particles will emit Čerenkov radiation as they travel faster than light in the medium. The particle polarizes the surrounding medium, yielding radiation on relaxation of the polarization. The phase velocity of light in a medium with refractive index $n$ is $c_{medium} = c/n$. A coherent conic wavefront with opening angle $\cos(\Theta) = 1/\beta n$ forms behind the particle, see Fig. 2.2. This optical light propagates through the atmosphere. Light is scattered by large (i.e. $d \approx \lambda$) and small ($d \ll \lambda$) particles via Mie- and Rayleigh scattering respectively. Absorption is caused by multiple compounds, notably ozone which strongly suppresses light of $\lambda \lesssim 300$nm [42].

The analytical calculation of every possible shower is impossible since the process is inherently statistical. Thus it is common practice to utilize large ensembles of Monte Carlo simulations of single showers. Some examples can be seen in Fig. 2.3 and 2.4. It has to be noted that the models used for hadronic interactions are based on extrapolated accelerator measurements. Even though they are improved continuously, large uncertainties remain. The study of hadronic matter is a very active field (e.g. pentaquarks [44,45] and baryonic resonances [46] are not yet well understood). The existence of discrepancies between simulations and data is well known and is subject of numerous studies [47–50]; different simulation packages also disagree substantially. By utilizing recent data from powerful accelerators like the LHC uncertainties can be reduced [48].

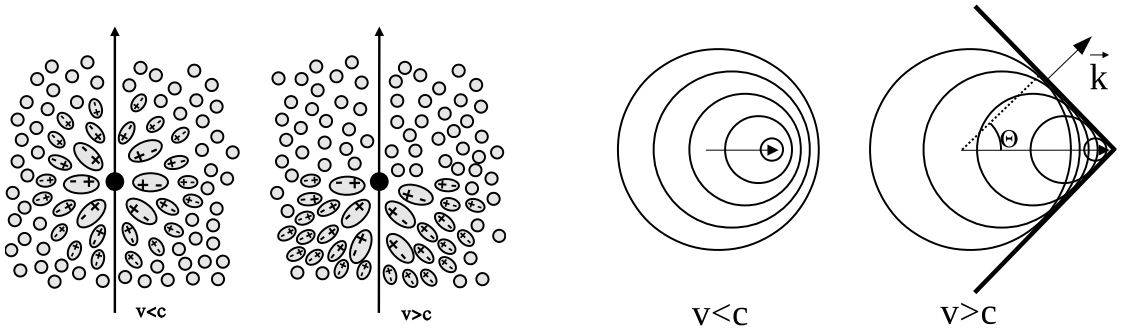


**Figure 2.2:** Illustration of the basic mechanism leading to emission of Čerenkov radiation. **Left:** Negatively charged particle traverses medium, causing polarization. **Right:** Construction of the Čerenkov wave front. Taken from [43]

## 2.2 Working principle of IACT

With their high sensitivity and low field of view, IACT are a very directional $\gamma$-ray detection instrument and have to be pointed specifically at a source. Large faceted mirrors reflect Čerenkov light onto a camera of photomultiplier tubes (PMT), yielding an image of the shower as depicted in Fig. 2.6. A centrally triggered array of multiple IACT provides good suppression of background events due to night sky and muons. If multiple telescopes are illuminated by Čerenkov light, an event is registered. Even though the timescale of illumination is in the order of nanoseconds, the readout is fast enough to register individual showers. The obtained images are still covered in noise. Hardware noise and night sky can be filtered out by image cleaning. Cleaned images can then be fed into the analysis software.

H.E.S.S. is an IACTA located in Namibia consisting of four smaller telescopes (CT1-4) and was extended by a bigger fifth telescope (CT5). The telescopes can be pointed anywhere in the sky, usually parameterized by zenith and azimuth angles. CT1-4 are identical twins arranged in a 120m side-length square while CT5 is placed in the array center. CT1-4 each utilize a 960 PMT camera. The PMTs are arranged hexagonally for high packing density and coverage. Winston cones are installed to collect light that would otherwise fall into gaps between PMTs. Thus each PMT effectively monitors a hexagonal tile of the camera plane. CT5 works similarly, but has a bigger camera with 2048 PMTs. The small cameras were upgraded to match the performance of the newer big camera [52]. The cameras are now not only able to collect intensity information (i.e. photoelectron count), but also time information. This includes time of maximum, time over threshold and even actual waveforms with nanosecond resolution (sample mode). Detailed information is available in [53].

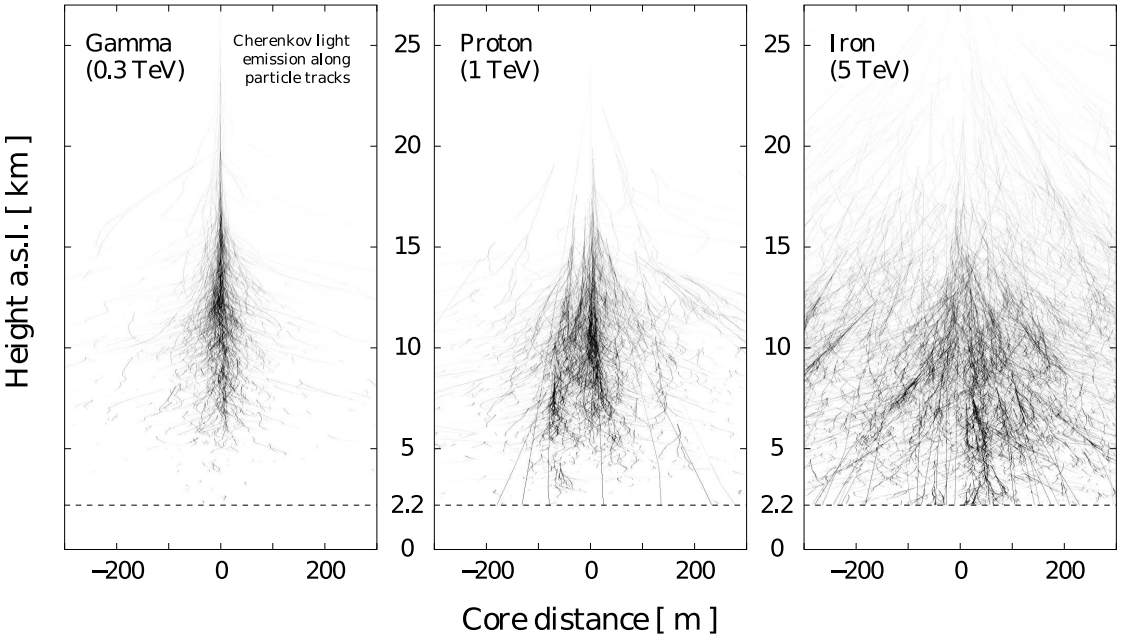


**Figure 2.3:** Lateral extension of air showers by different particle types and energies for ground level 2.2km above sea level (a.s.l.). The illustrations need to be highly stretched to illustrate the shapes. The darker the particle track, the more Čerenkov light is emitted. Taken from [25]

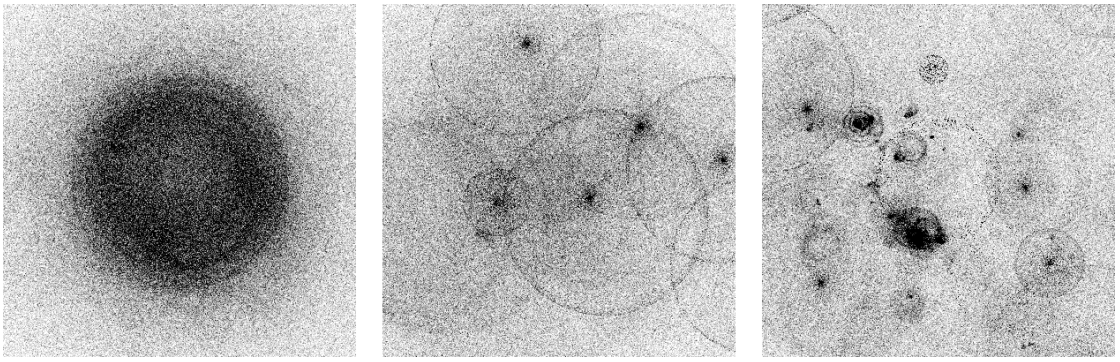

**Figure 2.4:** Patterns of Čerenkov light on the ground below the showers from Fig. 2.3. The depictions are 400m x 400m. The darker, the more light hit the ground. Very dark spots in the hadronic patches emerge from particles hitting the ground. Adapted from [51]

## 2.3 Image analysis for IACT

Air shower images taken by IACT are generally of elliptical shape. While $\gamma$-ray shower images can be reasonably approximated by an ellipse, hadronic ones are rather irregular. The Hillas parameters [18] can be used in various ways to get to reconstruct $\gamma$-ray

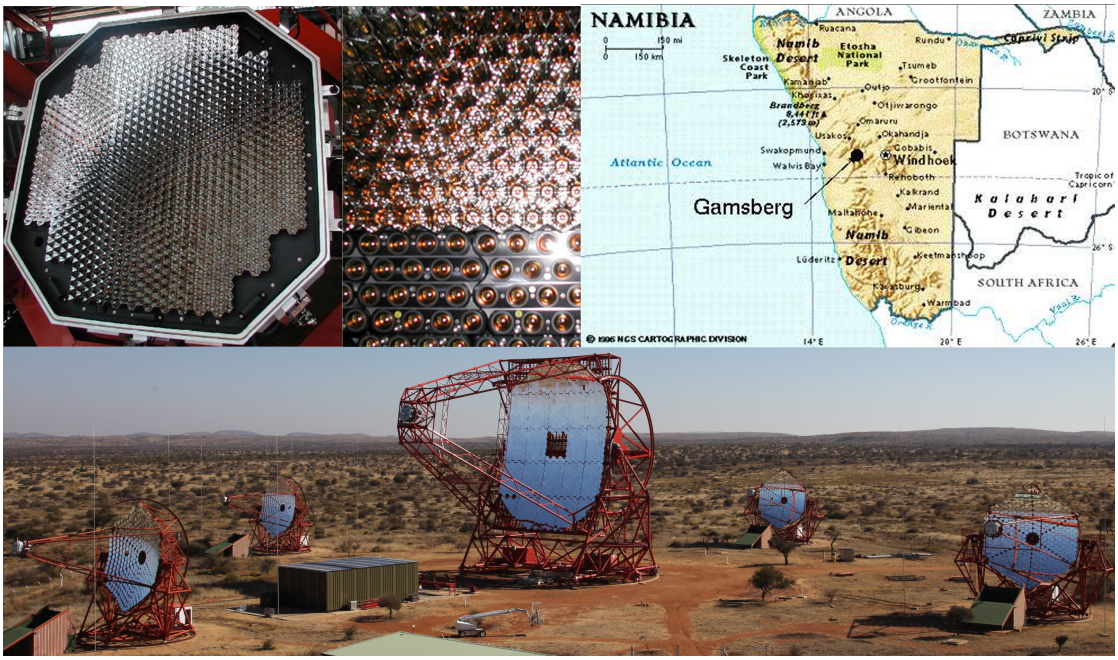


**Figure 2.5:** Collage of H.E.S.S.: From l. to r., t. to b.: Small H.E.S.S. camera (from [54]); Closeup of partially installed winston cones (from [55]); Location of H.E.S.S. in Africa (from [55]); Panorama of the H.E.S.S. site (Credits to Clementina Medina)

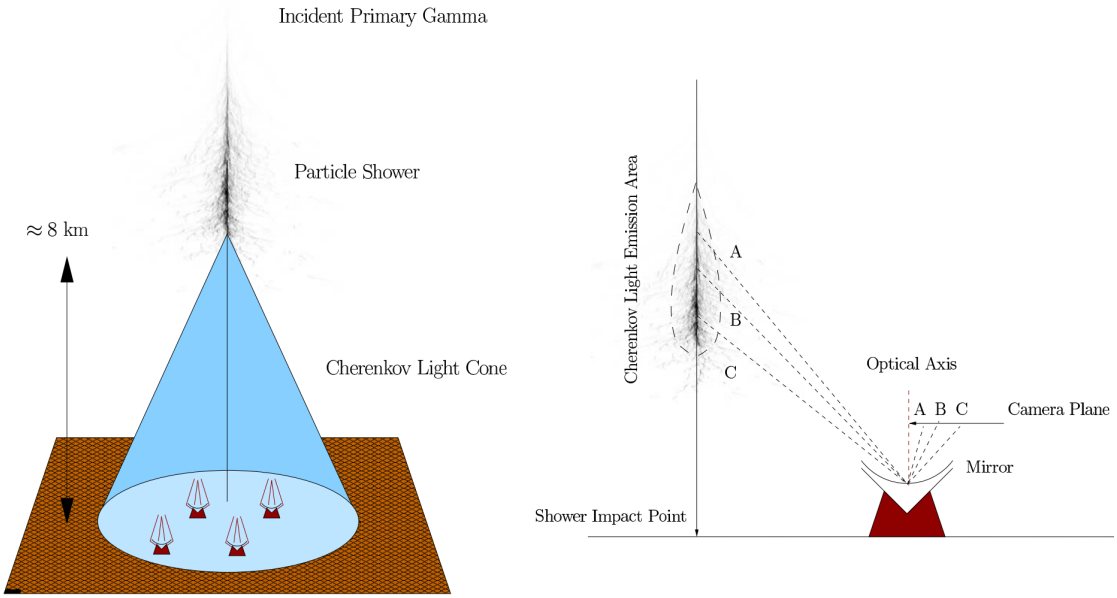


**Figure 2.6:** Illustration of the working principle of IACT. **Left:** Telescopes are collecting Čerenkov light from a particle shower. **Right:** A specially shaped mirror projects the light into the camera plane. Taken from [55]

origin [56], see Fig. 2.7. Using first and second order moments of the image, the Hillas parameters can be calculated analytically. For example the center of gravity can be calculated via a weighted average over all pixels $i$ with the pixel position $x_i$ and $y_i$ and the photoelectron count $q_i$:

$$\langle x \rangle = \frac{\sum_i x_i q_i}{\sum_i q_i} \qquad \langle y \rangle = \frac{\sum_i y_i q_i}{\sum_i q_i} \tag{2.1}$$

The calculation of $\langle x^2 \rangle, \langle y^2 \rangle$ and $\langle xy \rangle$ is similar. Using these values as a basis one can directly calculate the Hillas parameters shown in Fig. 2.7. The full calculation can be found in [57, 58]. Since the moments are influenced strongly by outliers, image cleaning is very important. To get rid of bright outlier pixels e.g. from a bright star in the field of view two thresholds $t_1, t_2$ are defined. Low intensity ($< t_1$) pixels are disregarded. Pixels with intensity bigger than $t_1$ are kept only if they have a neighbor with intensity larger than $t_2$ and vice versa.

Using the Hillas parameters by intersecting the ellipse major axes in the camera plane the source position is obtained. The position needs to be transformed from camera plane coordinates to galactic coordinates (or any other coordinate system of choice). Other algorithms can yield better performance and estimates of reconstruction error, as shown in Fig. 2.8. An overview is given in [56].

While direction reconstruction is a purely geometric method, energy reconstruction is based on Monte Carlo simulations. This requires the total image intensity and impact distance of the particle. Impact distance is calculated in array coordinates similarly to the particle origin calculation in camera coordinates. The impact point is where the extrapolated path of the primary particle hit the ground. Lookups of energy for given intensity and impact distance are created in zenith-bins using Monte Carlo simulations. Interpolation of the lookup table for observed shower parameters allows the estimation of $\gamma$-ray energy.

The separation of background and signal, i.e. cosmic rays and $\gamma$-rays is of uttermost importance for $\gamma$-ray astronomy. The number of cosmic ray induced events outweighs the actual signal by typically four orders of magnitude [41]. This ratio varies with the actual source brightness. Separation can be achieved by using the average aspect ratio of showers. The so called mean reduced scaled width (MRSW) and length (MRSL) are calculated from Hillas parameters based on ensembles of shower simulations [59].

Background suppression can be drastically improved by learning algorithms: A boosted decision tree (BDT) fed with Hillas parameters and trained on Monte Carlo simulations is presented in [20, 21]. The Hillas parameters are not the only basis for analysis. There are several other methods, including analytical shower models and other Monte Carlo based approaches. For example fitting shower parameters to an analytical model of actual shower images was used for the CAT experiment [60] and later adapted for H.E.S.S. [61]. The analytical model can be replaced by large ensembles of shower simulations, yielding even better performance [62] and has also been adapted for other experiments [63].

While artificial neural networks (ANN) like a multilayer perceptron could be used similarly to boosted decision trees, they do not promise strong improvements. In fact,

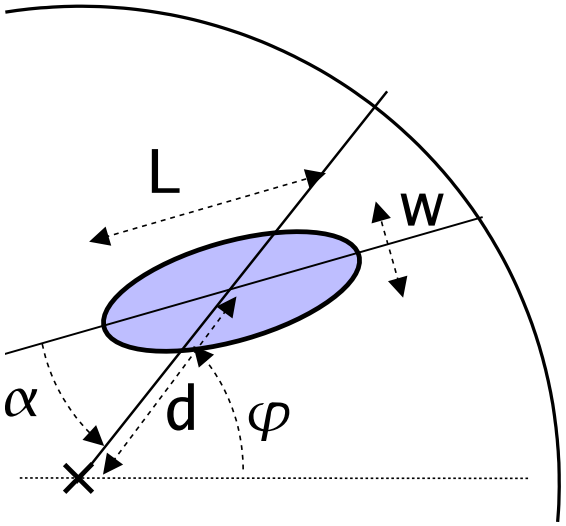


**Figure 2.7:** Depiction of a shower ellipse and its Hillas parameters. Taken from [58]. The ellipse shape is fully parameterized by its length and width. Position and orientation in the camera plane can be written in terms of multiple variables. The center of gravity can be expressed in Cartesian or polar coordinates $(d, \phi)$ while the ellipse orientation can be measured relative to the camera's x-axis or the radial center line $(\alpha)$.

decision trees and neural networks are not as dissimilar as one might expect. For example, any decision forest can be recast into a two layer neural network yielding the exact same predictions [64]. Furthermore decision trees are often said to be more transparent than neural networks, since they rely on combinations of binary decisions instead of high dimensional vector spaces. It is however possible to construct a decision tree from a neural network in order to better understand the decision process [65].

Convolutional neural networks (CNN) are an extension of ANN. A key point is that they are not fed hand engineered features like ANNs and BDTs, but take raw images as input. Convolutional kernels inside the network capture recurring distributions in the images, yielding learned features. The hierarchical structure of CNNs allows capturing rather abstract representations of image properties that can be neither easily visualized nor understood. This property can lead to some skepticism towards the physical reasoning behind learned features. However CNNs are not a black box per se, since any computation inside can be traced. The next chapter provides an overview of neural network structures and the training process used in deep learning.

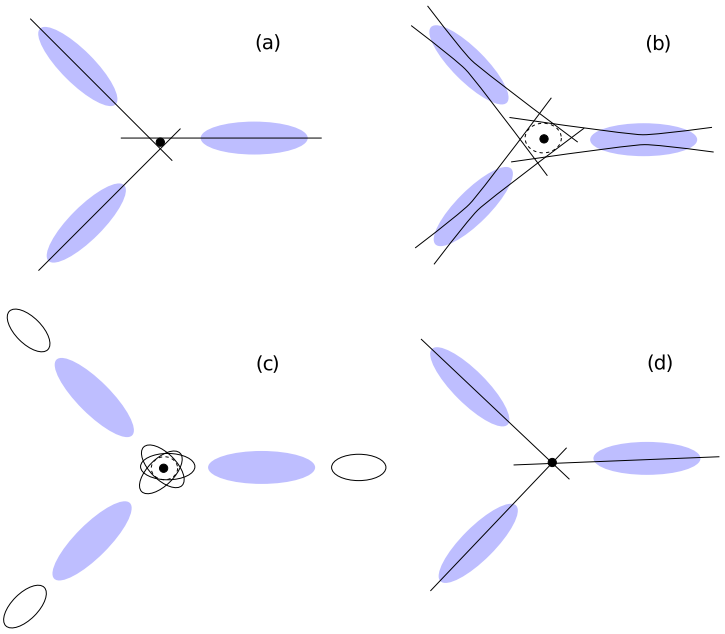


**Figure 2.8:** Illustration of the usage of Hillas parameters for stereoscopic reconstruction. Illustration of the Hillas parameters and their usage for direction reconstruction. Four geometric reconstruction algorithms based on the Hillas parameters. Adapted from and explained in detail in [56] **(a):** Intersection of Ellipse major axes **(b):** Incorporation of uncertainties **(c):** Utilizing ellipse aspect ratio to estimate source position **(d):** Optimization of shower geometry to fit observed data

# 3 Deep learning basics

While deep learning is a relatively new concept, probably every physicist had contact with one of the most basic learning algorithms throughout his or her career. Linear regression was already applied in the late 19th century to predict children's height from their parents' height by Francis Galton, yielding an estimate for an unknown based on statistical data [66, 67]. The result was obtained by drawing a line on a scatter plot by eye, lacking a well defined measure for the 'goodness of fit'. The data was later submitted to a Cambridge mathematician for rigorous analysis, yielding similar results. Todays sophisticated numerical algorithms and semiconductor technology enable us to fit incredibly complex models to inconceivable amounts of data. This is usually formulated as a minimization problem via some loss function. For linear regression a popular loss is the sum of squared residuals.

State of the art predictive models utilize deep learning to extract abstract features from data. Learning can be used synonymous with the - at least in a physics context - more familiar term fitting. The next sections contain a mathematical description of the methods used for this work and possible future developments.

## 3.1 History of learning algorithms

To understand deep learning it is crucial to first grasp the general concept of learning, and going deep afterwards. There are several classes of learning algorithms. For this work we consider a function minimization problem. Assume a linear model to predict some value $y$ for a given value $x$, similar to the aforementioned parent and children height. The linear model $f$ parameterized by some learnable weights $\boldsymbol{\omega} = \{\omega_1, \omega_2\}$ is of the form:

$$y = f(x; \boldsymbol{\omega}) \coloneqq \omega_0 + \omega_1 x \tag{3.1}$$

Utilizing a set of $n$ known data points $\mathcal{D} = \{x_i, y_i\}$ with $i \in \{1, ..., n\}$ one can measure the performance of $f$. For given weights $\boldsymbol{\omega}$ and dataset $\mathcal{D}$ the loss function $\mathcal{L}$ yields a scalar (the 'loss'). A common choice is the sum of squared residuals:

$$\mathcal{L}(f; \boldsymbol{\omega}, \mathcal{D}) \coloneqq \frac{1}{n} \sum_{i=1}^{n} \left[ y_i - f(x_i; \boldsymbol{\omega}) \right]^2 \tag{3.2}$$

Optimization of the models performance for given data $\boldsymbol{x}$ and ground truth $\boldsymbol{y}$ corresponds to a minimization of $\mathcal{L}$ by varying $\boldsymbol{\omega}$. The $\boldsymbol{\omega}$ that minimizes the loss is usually found by some random initial guess and a gradient descend method. In practice more

sophisticated optimization techniques are used. However they still rely on gradient computation. A single gradient descend step with some scalar learning rate $\alpha$ is defined as:

$$\boldsymbol{\omega}_{\text{new}} = \boldsymbol{\omega}_{\text{old}} - \alpha \frac{\partial \mathcal{L}}{\partial \boldsymbol{\omega}_{\text{old}}} \tag{3.3}$$

The update is repeated until convergence. In deep learning, this is called backpropagation. The learning rate is a hyperparameter that cannot be learned but is chosen empirically. It is common practice to decay the learning rate to aid convergence.

## 3.2 Basic building blocks

Since we now know how learning works, we can go deep. It is possible to stack linear models while putting some nonlinearity in between. The nonlinearity is also called activation function, while the linear operation is often called fully connected or dense layer. Stacking lots of layer yields deep neural networks. Common choices of activation functions are tanh or the relu [68]. Investigations of ensembles of activation functions were conducted [69], empirically comparing several activation functions. It is also common practice to add a bias vector after the linear transformation. A dense layer with input vector $\boldsymbol{x}$, output vector $\boldsymbol{y}$, weight matrix $\boldsymbol{W}$ and bias vector $\boldsymbol{b}$ with point-wise application of some activation function $\sigma$ is condensed in the operation

$$\boldsymbol{y} = \sigma(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \tag{3.4}$$

The dimension of $\boldsymbol{W}$ has to fit the input and determines the output dimension. The output dimension is often referred as the number of neurons in that layer. Note that the order of entries in $\boldsymbol{x}$ does not matter for a dense layer, as a permutation of $\boldsymbol{x}$ can be trivially compensated by applying the same permutation to $\boldsymbol{W}$. This fact clearly shows that it is a sensible choice to enforce some structure in $\boldsymbol{W}$ if $\boldsymbol{x}$ contains structured data. The structure could for example be image data (2D) or constant length time series data (1D). A perfectly fit structured linear operation is the discrete convolution of an image with some kernel denoted by $*$. Containing lots of zeros and repeated entries, the usage of such an operator reduces the memory footprint drastically. A simple example of 1D convolution of some data $\boldsymbol{x} = (x_0, x_1, x_2, x_3)$ with a kernel $\boldsymbol{k} = (k_1, k_2)$ shows that a linear operator $\boldsymbol{H}$ is equivalent to the convolution $\boldsymbol{x} * \boldsymbol{k}$:

$$\boldsymbol{x} * \boldsymbol{k} = \begin{pmatrix} x_0k_0 + x_1k_1 \\ x_1k_0 + x_2k_1 \\ x_2k_0 + x_3k_1 \end{pmatrix} = \begin{pmatrix} k_0 & k_1 & 0 & 0 \\ 0 & k_0 & k_1 & 0 \\ 0 & 0 & k_0 & k_1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \boldsymbol{H}\boldsymbol{x} \tag{3.5}$$

The free parameters of this proto-convolution layer are the kernel size and padding of the input. The output is called a feature map. Convolution was originally used as a network layer because of its similarity to the biological visual nervous system [70] and later trained with backpropagation [71, 72]. A full convolution layer takes multiple

feature maps as input and yields multiple feature maps as output, enabling the stacking of convolution layers. There is a kernel for each pair of input and output feature maps. Earlier work contained selective connectivity between input and output feature maps [73], however this property is not present anymore in state of the art models. Adding biases is optional.

Another important property of many datasets is the translational invariance of features. Embedding this invariance into the network structure is achieved by pooling operations. The geometrical interpretation is similar to a convolution - shifting some window over the input data - but is inherently nonlinear. Pooling returns e.g. the maximum or the average value inside the window instead of a linear combination of kernel and input patch.

The network output is either a set of scalar values in case of a regression problem or a set of class probabilities for a classification task. The last layer of a regression problem can simply be a dense layer with respective dimensions. For classification, a dense layer with output dimensionality equal to the number of classes is followed by a softmax layer. This ensures that the sum of class probabilities is normalized to one:

$$\text{softmax}(\boldsymbol{x})_i \coloneqq \frac{e^{x_i}}{\sum_i e^{x_i}} \tag{3.6}$$

Using this kind of normalization instead of a simpler approach is justified by the symbiotic interplay with cross entropy as a loss function for classification.

## 3.3 Neural network training

The training of a deep neural network is very similar to finding the best parameters for a simple linear model. Due to their high capacity of free parameters precautions have to be made in order to prevent memorization of training examples, so called overfitting. Some dataset $\mathcal{D}$ is randomly split into three sets: $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{valid}}$ and $\mathcal{D}_{\text{test}}$. The training set is used for direct minimization of the loss value. During training, the loss is also calculated for the validation set in order to detect overfitting, tuning hyperparameters and check for convergence. The test set is held back until all hyperparameters are set and should be evaluated as rarely as possible and only to report the final performance.

The choice of loss function strongly depends on the network's task. For classification the most common choice is the cross entropy. Given a ground truth class vector $\hat{\boldsymbol{y}}$ and some prediction out of a softmax layer $\boldsymbol{y}$, the cross entropy $\mathcal{C}$ is:

$$\mathcal{C}(\hat{\boldsymbol{y}}, \boldsymbol{y}) \coloneqq -\sum_i \hat{y}_i \log(y_i) \tag{3.7}$$

Datasets are commonly way too big to fit into memory. Thus the loss and its gradient are calculated on a subset of the dataset, a 'batch': $\mathcal{B} \subset \mathcal{D}_{\{\text{train, valid, test}\}}$. This method also has the advantage of enabling parallel processing of multiple batches and gradient aggregation. The influence of batch size on network performance has been investigated thoroughly [74–76].

The trajectory of network weights through the whole configuration space during gradient descent can be suboptimal. More sophisticated optimization algorithms utilize adaptive learning rates, averaged gradients and can lead to faster and more stable convergence. Large gradients can catapult the network weights way out of a reasonable range. Adding regularization terms to the loss function mediates this issue. In principle any data independent but weight depended term can be used for regularization. A common choice are the sum of L1 or L2 norms of weights. Another practice is dropout [77], zeroing a certain part of a layers activations during training. Randomly zeroing some values prohibits single values to gain too much importance. During network evaluation all values are kept. The more sophisticated technique dropconnect [78] zeros connection values instead of activations. An overview of several optimization and regularization techniques is given in [79].

Since the order of batches and weight initialization are random, the resulting trained networks will produce slightly different predictions after each training run. To measure reproducibility and robustness of a given network and training schedule, ensembles of trained networks can be compared. Ensembles can also be used to increase the prediction performance by averaging over the network outputs.

# 4 Deep learning concepts for H.E.S.S.

## 4.1 Image preprocessing

The H.E.S.S. telescope cameras consist of arrays of photomultipliers arranged in a hexagonal grid (synonymous with 'triangular grid'). Current high performance implementations of neural network primitives for convolution, pooling etc. are implemented for square grids [80]. Also the camera pixel arrangement is not convex but has empty corners. Thus it is necessary to preprocess images taken by H.E.S.S. in order to feed them into modern software via images with square pixel shape. The raw mathematical formulation and logical reasoning behind transformations used in deep learning could in principle handle hexagonal images. Yet, it is not backed by decades of low level hardware-software co-engineering. However there are some ways to formulate hexagonal convolutions and pooling via square grid computations.

Notable differences between square and hexagonal lattices are the differing symmetries. Lattices are sets of points that get mapped onto themselves by certain discrete symmetry transformations. Hexagonal lattices are symmetric under $n\frac{\pi}{3}$ rotations, square lattices are subject to $n\frac{\pi}{2}$ rotational symmetry for $n \in \mathbb{Z}$. Thus, any conversion between them implies breaking of rotational symmetry. The same holds true for translational symmetry. Square lattices obey symmetry under two orthogonal translation directions, while the hexagonal lattice symmetry directions are not orthogonal.

There are different approaches to generate square images from hexagonal ones. Assigning pixel intensity to the pixel center, a single point in the camera plane, opens up a multitude of interpolation methods including linear and cubic methods. An overview can be found in [81]. Interpreting the camera's PMTs as hexagonal histogram bins collecting single photons is more realistic than a center point approximation and can be rebinned into a square histogram. It is also possible to rearrange the camera pixel grid (e.g. via shearing of the camera plane or shifting every second pixel row) while keeping the original pixel values without any interpolation. Utilizing certain maskings to convolution kernels, the symmetries of a hexagonal grid can be kept while using computations optimized for square grids [82]. Also additional information that is harder to interpolate than intensity (e.g. timing of pixel activations) can be incorporated easier using hexagonal kernels.

### 4.1.1 Standard interpolation

Interpolation methods like linear and cubic interpolation are widely used in image processing. Pixel values can be interpreted as discrete samples of some continuous function, the image intensity distribution in case of H.E.S.S. . Constructing such a function al-

lows sampling the original image at an arbitrary set of points. The interpolated values obey some relationship to neighboring original image values depending on the choice of interpolation method. Linear and cubic interpolation can be achieved by partitioning the camera plane into simplices (i.e. triangles for 2D) with corners at the camera pixels' centers. For any point inside the convex hull of the camera the intensity can be calculated via interpolation of the triangle corner intensities. The de facto standard segmentation method for interpolation is the Delaunay triangulation [83]. Problematic behavior arises in the corners of H.E.S.S. cameras because of the large distance between hexagonal pixel centers, see Fig. 4.1. Artificial zero valued hex pixels are introduced to mask the camera corners.

Interpolation is only possible inside the pixel centers' convex hull - any value outside would have to be extrapolated. Filling the corners yields a square convex hull, fitting the desired square image. The resolution of the resulting image is arbitrary. It has to be noted that neither linear nor cubic interpolation preserve the total image intensity. Cubic interpolation can lead to negative pixel values, which is also unphysical.

### 4.1.2 Rebinning

Camera pixel values are related to the number of photons collected by the corresponding PMT's photocathode. Interpreting the image as a histogram is thus more physical than concentrating the pixel value to a single point. For our purposes we would like to know what an image taken with a camera consisting of a square grid of square PMTs would look like. This can be achieved by rebinning the hexagonal histogram into a square one. This allows using an arbitrary resolution and implies the conservation of total image intensity.

Rebinning can be formulated as a single sparse matrix-vector multiplication, resulting in low computation times. The matrix entries are pairwise overlaps of old and new histogram bins. Overlapping squares with hexagons results in irregularly shaped polygons, as seen in Fig. 4.1. The vast majority of bins do not overlap, yielding a sparse matrix.

### 4.1.3 Hexagonal kernel

The most elegant approach to preprocessing would be keeping the original pixel values and converting the image to square shape in order to profit from recent advancements in high performance computing. A triangular lattice can be mapped onto a square lattice via an affine transformation. A special convolution kernel can then be used to keep the topological relationships of the original image.

Nonlinear transformations can also yield square grids. Shifting every second pixel row by half a pixel distance and stretching the image results in a much smaller square grid than the one obtained by a linear transformation. Constructing convolution kernels for such images is more complex as it requires multiple convolution operations and interleaving of intermediate results.

This method has the major advantage of easier incorporation of harder to interpolate pixel information like timing. While the sample mode data could be interpreted

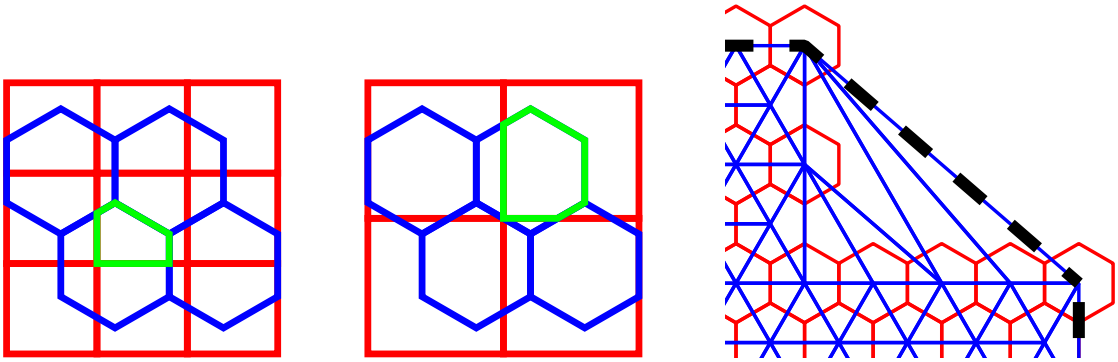

**Figure 4.1:** Illustration of geometric properties of the rebinning and interpolation preprocessing. **Left, Middle:** Rebinning a hexagonal histogram (blue) to a square one (red). The redistribution of bin contents is determined by bin overlaps (green). Two different output resolutions are shown. **Right:** Corner of a H.E.S.S. camera. The centers of hex pixels (red) can be Delaunay triangulated (blue). Stretched narrow triangles imply relations between rather distant pixels. The convex hull is shown in dashed black.

as a video, the time of maximum and time over threshold values are much harder to reasonably interpolate. Rebinning time values is obviously meaningless. Two members of the task group provide a reference implementation of hexagonal convolutions and pooling [82], accompanied by a manual and illustrations.

### 4.1.4 Comparing preprocessing methods

Since any preprocessing will introduce some kind of bias into the image, the influence of aforementioned methods on artificial telescope images is studied in the following. Shower images are elliptic blobs in the camera plane. A simple parameterized analytical model function $f$ can be used to create shower ellipses of varying shape and position. $f$ is chosen to be a general two dimensional Gaussian function.

$$f(x, y; x_0, y_0, \sigma_x, \sigma_y, \theta) \coloneqq \exp\left(-\frac{1}{2} X^\mathsf{T} C X\right) \tag{4.1}$$

$$\text{with} \qquad X = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \qquad \text{and} \qquad C = \operatorname{diag}\left(\sigma_x^{-2}, \sigma_y^{-2}\right)$$

$f$ is still subject to border effects at the camera's edge. Therefore an envelope function $g$ is introduced to smoothly set any values beyond the camera radius $r_0$ to zero. With $r = \sqrt{x^2 + y^2}$:

$$g(f(x, y); r_0) \coloneqq \begin{cases} f(x, y) \left[\frac{1}{2}\left(1 + \cos(\pi \frac{r}{r_0})\right)\right]^{\frac{1}{8}} & \text{if} \quad r < r_0 \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

| Parameter | $x_0$ | $y_0$ | $\sigma_x$ | $\sigma_y$ | $\theta$ |
|---|---|---|---|---|---|
| lower bound | -0.35 | -0.35 | 0.02 | 0.11 | 0 |
| upper bound | 0.35 | 0.35 | 0.1 | 0.2 | $\pi$ |

**Table 4.1:** Bounds of uniform distributions of sample function parameters.

Random parameter sets $P = \{x_0, y_0, \sigma_x, \sigma_y, \theta\}$ are generated in order to create an ensemble of artificial shower images. Each element is chosen from a uniform distribution with certain lower and upper bounds. $g$ is sampled at each camera pixel's center coordinate, producing a hexagonal grid of pixel values. Applying the different preprocessing methods yield square grids and interpolated pixel values.

Earlier work [28] used a $\chi^2$ method via $P$ to fit $g$ to the interpolated images. The fitting procedure is time intensive and occasionally shows convergence issues. While it is possible to throw away bad samples and obtain basic results with the fitting method, another way was explored. Direct calculation of the Hillas parameters is several orders of magnitudes faster while representing similar values. The parameters $x_0, y_0$ correspond to the Hillas center of gravity, while $\sigma_x, \sigma_y$ are equal to ellipse width and length. The orientation angle is represented by $\theta$.

The Hillas parameters of preprocessed images can be compared to the original image parameters. The absolute difference of original image parameters and preprocessed image parameters yields residuals that can be used as a quantitative measure of performance. Apart from different preprocessing methods, the only free parameter is the resolution of preprocessed images. As the H.E.S.S. cameras aspect ratios are irrational and non-square, the resulting images are also non-square. Image resolution is from now on the resolution in x-direction, while the y resolution is chosen to fit the camera's aspect ratio. Two different resolutions are used: 32 pixel images roughly conserve the total pixel count. 64 pixel images compensate the information loss due to smearing during preprocessing by denser sampling. The grids used for this study are shown in Fig. 4.2.

A study comparing linear and cubic interpolation with rebinning is performed for $5.6 \times 10^6$ random parameter combinations. The parameters are distributed uniformly with bounds as in Tab. 4.1. The resulting parameter residuals are shown in Fig. 4.3. It has to be noted that the distribution shapes depend on the parameter bounds. However a qualitative comparison is still valid, as the differences in parameter preservation performance are rather large. This is especially the case for cubic interpolation, which excels at ellipse shape preservation. Yet cubic interpolation introduces other difficulties, like slow processing and negative pixel values.

Ellipses in preprocessed telescope images are strongly stretched by linear interpolation and rebinning. The ellipse minor axis (width) is stretched more than the major axis (length) because of the more steep intensity profile. For rebinning, the effect stems from the distibution of intensity to neighboring bins. Linearly interpolated gaussians are spread due to the under-estimation of intensity inside a convex 'cup' around $(x_0, y_0)$, and the over-estimation outside. This can be shown using a example gaussian $E(\boldsymbol{x})$ and the condition for convexity, the Hessian $H_E(\boldsymbol{x})$ being positively semidefinite:

$$E(\boldsymbol{x}) \coloneqq e^{-(x_1^2+x_2^2)} \tag{4.3}$$

$$H_E(\boldsymbol{x}) \coloneqq \left(\frac{\partial^2 E}{\partial x_i \partial x_j}(\boldsymbol{x})\right)_{i,j=1,2} = -2E(\boldsymbol{x}) \begin{pmatrix} 2x_1^2 - 1 & 2x_1x_2 \\ 2x_1x_2 & 2x_2^2 - 1 \end{pmatrix} \tag{4.4}$$

$$\boldsymbol{x}^\intercal H_E(\boldsymbol{x})\boldsymbol{x} = -\underbrace{2E(\boldsymbol{x})}_{>0\,\forall \boldsymbol{x}\in\mathbb{R}^2} (2(x_1^2+x_2^2)^2 - (x_1^2+x_2^2)) \overset{!}{\geq} 0 \tag{4.5}$$

$$\Rightarrow \sqrt{x_1^2+x_2^2} \leq \frac{1}{\sqrt{2}} = r_{\text{convex cup}} \tag{4.6}$$

Incorporating the parameters from $f$ yields a scaled, rotated and translated ellipse instead of a circle. Even though the envelope function $g$ further influences the shape, some part of the function will still be convex if realistic parameters for $f$ are chosen. This is also true for more sophisticated analytical shower models [61] and Monte Carlo simulations [62].

Differences between $\Delta\text{cog}_x$ and $\Delta\text{cog}_y$ arise from different grid alignments of hexagons vs. squares. While linear and cubic interpolation are generally improving at higher resolution, rebinning only gains accuracy in y direction - beating cubic interpolation. Linear interpolation evidently produces larger shifts and thus performs consistently worse than cubic interpolation.

Orientation of shower ellipses is also best preserved by cubic interpolation. While rebinning and linear interpolation are on par at lower resolution, rebinning gains less from higher resoulution images. The errors in rotation are roughly on the order of millirad.

This phenomenological study does not aim for thorough explanations of the observed distributions of residuals. The phase space of possible resolutions and parameter ranges is far too large to be completely explored. However it could be shown that cubic interpolation consistently outperforms linear interpolation and rebinning. The Gaussian image model and light collection process (i.e. sampling the function in the pixel center) are reasonable enough to show the difference in performance. Using analytical shower models (e.g. [61]) and integrating the resulting light intensity after passing the nonlinear mirror optics for each hex and square pixel is beyond the scope of this work.

It has to be noted that the performance of different preprocessing methods can only be estimated by such a toy model. The actual performance for a CNN application strongly depends on the preservation of internal intensity distributions. This fact also suggests that a CNN utilizing hexagonal kernels should in principle outperform any network relying on the presented preprocessing methods. This comes with the cost of reimplementing not only convolution operations, but also pooling for hexagonal images.

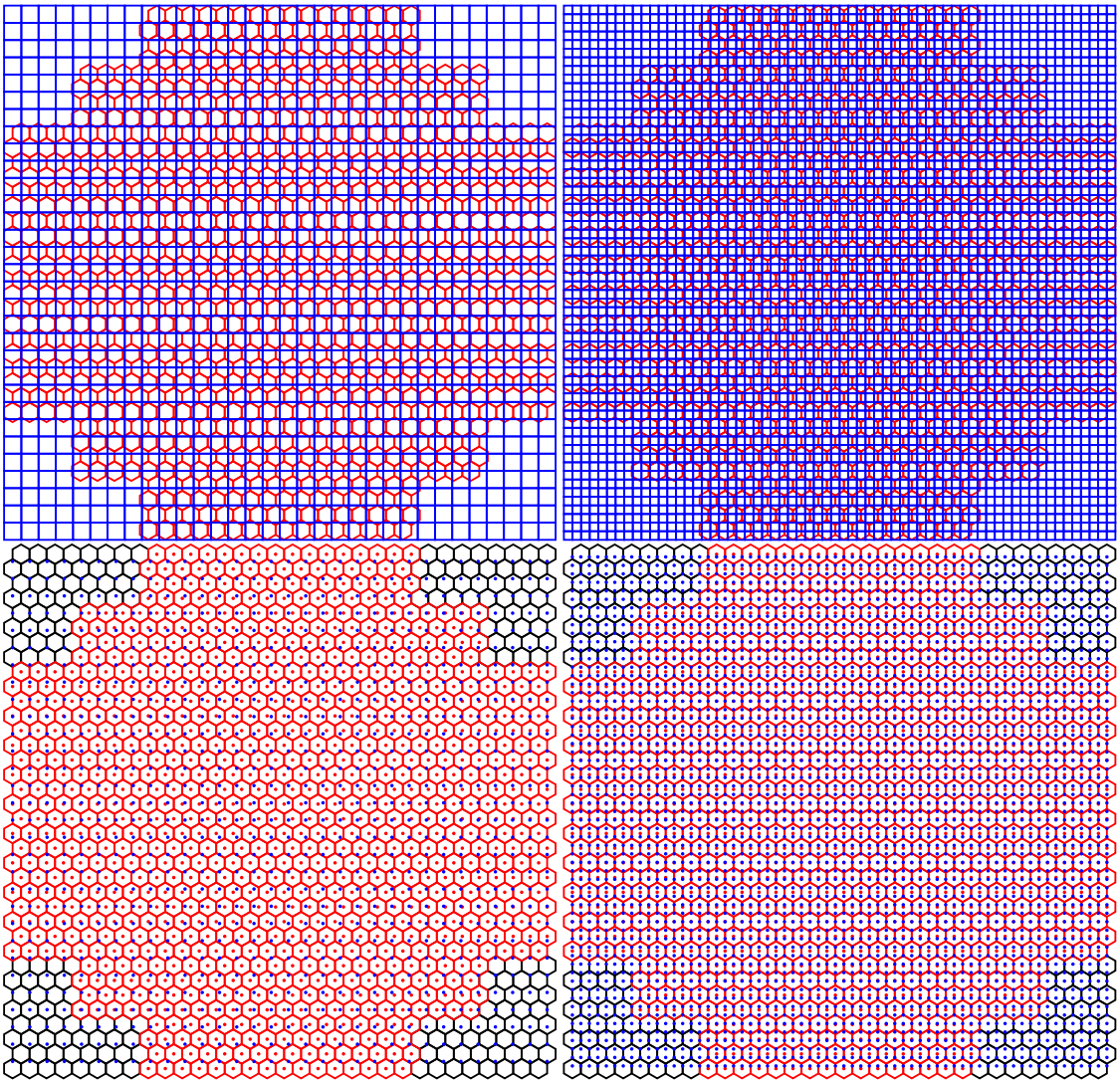

**Figure 4.2:** Geometry of preprocessing methods. **Left column:** 32 pixel resolution. **Right column:** 64 pixel resolution. **Top row:** Rebinning. Square bins are shown in blue, hexagonal bins in red. **Bottom row:** Interpolation. The original hexagonal pixels and their centers are shown in red while additional padding pixels are black. The blue dotted interpolation grids are used for both linear and cubic interpolation. For resolution 64, the x coordinates of the hexagonal image and the interpolation grid align nicely.

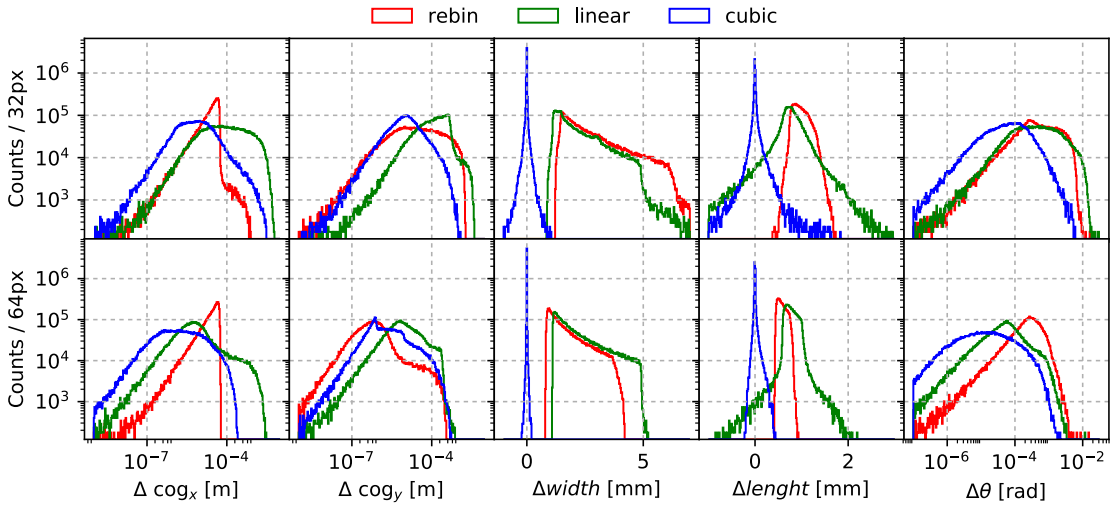


**Figure 4.3:** Hostigrams of Hillas parameter residuals for 32px (top) and 64px (bottom) resolution. Different preprocessing methods are depicted in colors. The x axis of each column is aligned. Count axes are logarithmic. The cog and $\theta$ plots show the absolute value of residuals, as their distributions are symmetric. The width and length distributions are not symmetric, requiring linear x-axes. Width, length and $\theta$ parameters are preserved much better by cubic interpolation. Utilizing higher resolution images generally aids parameter preservation. Distribution shapes depend on the parameter ranges (see Tab. 4.1 for the ones used here), yet the qualitative result stays the same.

## 4.2 Network architectures

Historical artificial neural networks had rather few layers. The number of layers started to rise with computational power and better understanding of inner workings of neural networks. A notable breakthrough was the introduction of ResNet [84], which allows to efficiently train hundreds of layers. More recent variants still gain accuracy beyond 1200 layers [85]. Such networks are only possible due to sophisticated initialization and training techniques combined with appropriate network architecture. Also the amount of arithmetic operations and memory required to train and apply such networks is a concern. Novel approaches use reinforcement learning or evolution to explore new network architectures, yielding state of the art results while using less space and computation power [86, 87].

However, these networks have grown on a very narrow selection of datasets for computer vision problems, composed of millions of everyday images for several hundred classes. Data taken by H.E.S.S. is less diverse but requires higher reconstruction accuracy in order to produce scientifically relevant results. The best approach to deep learning for H.E.S.S. is - in the spirit of Occams razor - to start with a very basic approach. Earlier work was based on combined event displays, summing all four tele-

| Layer type (kernel, stride) | Output shape |
|---|---|
| Input | $1 \times 32 \times 31$ |
| Conv2D (3, 1) | $16 \times 32 \times 31$ |
| Conv2D (3, 1) | $16 \times 32 \times 31$ |
| MaxPool (3, 2) | $16 \times 16 \times 16$ |
| Conv2D (3, 1) | $32 \times 16 \times 16$ |
| Conv2D (3, 2) | $32 \times 8 \times 8$ |
| MaxPool (3, 2) | $32 \times 4 \times 4$ |
| Flatten | 512 |
| Dense | 128 |

| Layer type | Output shape |
|---|---|
| Input | $4 \times 128$ |
| Flatten | 512 |
| Dense | 512 |
| Dropout (0.8) | 512 |
| Dense | 256 |
| Dropout (0.8) | 256 |
| Dense | 2 |

**Table 4.2:** Network architectures of the siamese twins (left) and the combination part (right). Each twin gets a single telescope image as input. All four twins outputs are concatenated and fed into a larger dense network that ultimately yields the desired output. After each conv and dense layer, the activation relu is applied. For the last dense layer softmax is used instead.

scope images into a single image. The issue of feeding a hexagonal image into existing frameworks was resolved by an approach of Gaussian sampling similar to interpolation via a radial basis function. A CNN consisting of four convolution layers followed by a dropout layer was found to perform best after a basic architecture search [26]. A basic investigation (a predecessor of this work, see section 4.1.4) showed that the Gaussian sampling strongly distorts the image [28]. Ongoing research of the task group includes multichannel input images (one channel per telescope) and recurrent neural networks for variable length inputs (only triggered telescopes) [30]. Also hexagonal convolutions were implemented to keep the undistorted original image information [82].

This work explores another way to combine multiple telescope images. Siamese networks [88] are basically multiple twins of CNN, sharing their weights. This architecture is motivated by the fact that the four telescopes are identical twins. Each telescope in an IACTA collects the same kind of information and is thus fed into the same network. The supposed network takes a single telescope image as input but shares weights for all telescopes. Concatenation of each network output (one per telescope) yields a single combined feature vector. Feeding this vector of intermediate features into another network that predicts a desired quantity enables end to end learning of the whole stack. Dropout after each layer prevents overfitting to the training data. The last layers neuron count equals the number of classes. See Tab. 4.2 for the complete architecture.

The architecture used for this work is rather shallow in order to reduce training times and hardware requirements. While absolute prediction accuracy is higher for deeper networks, relative accuracies can still be used to investigate certain properties of IACT data. Architecture search and hyperparameter optimization like the choice of optimizer are purposefully omitted in this work. Architectures designed for maximum significance in $\gamma$-ray astronomy can be found in [30].

# 5 Applied deep learning

Utilizing the presented data processing and network architecture for analysis of H.E.S.S. data in order to obtain astronomical results is only of secondary nature for this work. During the search for optimal regressors and classifiers [30], the possibility to separate real data from simulations unreasonably well was discovered. The following chapter will shed some light into the issue and possible mitigation strategies.

## 5.1 Training data and objectives

Training deep neural networks for classification or regression requires labeled data. Such data is not naturally available for IACT and needs to be synthesized using Monte Carlo (MC) simulations. A well established simulation tool chain for H.E.S.S. is provided by `CORSIKA` with addition of the `sim_telarray` package. For this study three datasets are used: Monte Carlo protons (`MCp`), Monte Carlo $\gamma$s (`MCg`) and real data (`Real`). The Monte Carlo simulations were not produced as specifically for this work, but are taken from the H.E.S.S. collaborations common pool. They are all diffuse, no pointsources were used in this work. The simulations are for muon efficiency [89] phase 1, with a zenith angle of 20°. The muon efficiency unites light collection efficiencies of telescope mirrors, Winston cones and PMTs and degrades with aging telescope hardware. The `Real` dataset consists of four phase 1 runs on PKS2155, with a zenith angle roughly matching the simulations. Two pairs of chronologically close (`34855, 34881`) and distant (`21932, 75053`) runs are investigated. Real data includes only extremely few photons, every single real data event is thus labeled as a real proton.

As some datasets contain CT5 images, they are pruned in order to keep only events with more than two triggered CT1-4 telescopes. The dataset images are calibrated and cleaned by the H.E.S.S. Analysis Package (HAP).

As CNNs process raw image data, the input phase space is much richer than the one for previous methods based on derived features (i.e. Hillas parameters). During the task groups studies, the possibility to classify between simulations and real data was discovered. For a perfect simulation it should be impossible to distinguish between MC and real data. However the uncertainties in hadronic interaction models, atmosphere properties and detector behavior might introduce image artifacts that are not present in real data.

The datasets are split and mixed into subsets according to each task. In order to find out more about the simulation vs. real data discrepancy, several tasks were performed:

1. Background rejection: Classification of `MCg` vs. `MCp`

2. Data type classification: `Real` vs. `MCp`
3. Network validation: `Real` A vs. `Real` B

Loss based learning algorithms like CNN naturally require balanced datasets [90]. Any dataset used for these studies contains a 50-50 split of class labels. Also the datasets were split into 60% training, and each 20% validation and test sets. While the Monte Carlo datasets contain each roughly 5 million events, the real data runs contain only a few hundred thousand events each.

## 5.2 Preprocessing

The preprocessing methods of 4.1 are applied to real data and simulations in order to obtain square images required as a neural network input. The pixel intensity distributions before and after preprocessing are investigated. The comparison was carried out between cubic interpolation because of the superior conservation of image parameters and rebinning for the extremely quick processing. The resolution was set to 32 pixels to keep a lower pixel count again for quicker processing. Two histograms of pixel intensities of events from a run on PKS2155 are displayed in Fig. 5.1. The distributions are similar for every dataset (`MCp`, `MCg`, `Real`). Cubic interpolation not only introduces strongly negative pixel intensities but also blows up the dataset size by an order of magnitude due to lots of near zero values. Increased dataset size, slow processing and the difficulty of dealing with artifacts renders cubic interpolation unfit for further usage. Thus the following experiments are performed with rebinning, which is superior not only due to several orders of magnitude lower computation times and storage requirements but also the straight forward physical interpretation.

## 5.3 Results

The tasks of Sec. 5.1 are tackled with the aforementioned siamese network architecture and rebinning preprocessing method. For the evaluation, classes are thought to be separated at a predicted class label $\zeta$ of 0.5. Accuracy is the portion of correctly classified events with this $\zeta$-cut. For a real world application the $\zeta$-cut must be chosen to maximize significance, as a higher value will get rid of more background but also discards signal. For this qualitative work the accuracy is only calculated to allow comparability with other work.

In order to showcase the networks basic abilities first, a MC based classification task is performed. Separating cosmic from $\gamma$-rays is of uttermost importance for any IACT analysis chain. The proposed network can distinguish between simulated signal and background with a test set accuracy of 93.4%. While this accuracy can clearly be improved (compare to 95.4% in [30]), it demonstrates the networks ability to capture image features.

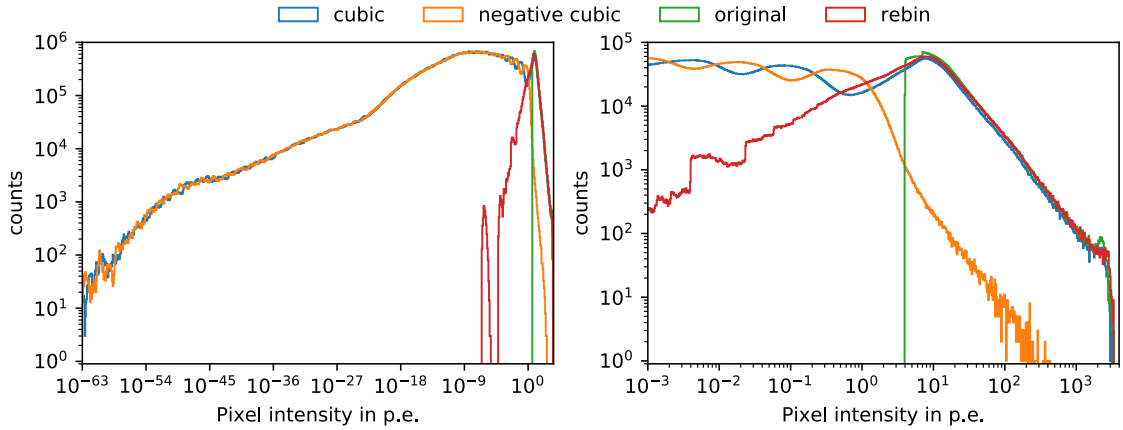


**Figure 5.1:** Histograms of pixel intensities for original hex images (green), rebinning (red) and cubic interpolation. Negative intensities (orange) obtained by cubic interpolation are shown separately from positive entries (blue). **Left:** The whole dynamic range of the image. Cubic interpolation clearly produces lots of near zero fluctuations. Below a certain intensity, negative and positive values are equally frequent. **Right:** Closeup of higher pixel intensity realm. The original intensity distribution is cut off at the lower image cleaning threshold and has a small peak at the upper threshold. All three positive pixel value distributions agree in shape with a similar power law above the cleaning threshold. The osciallations of cubic pixel intensity values stem from the curvature minimization during interpolation. Such small intensity values are rather insignificant, which is also the case for rebinning. However the negative arch of cubic pixel intensities extends far into higher intensities, even though such values are much less common than positive ones.

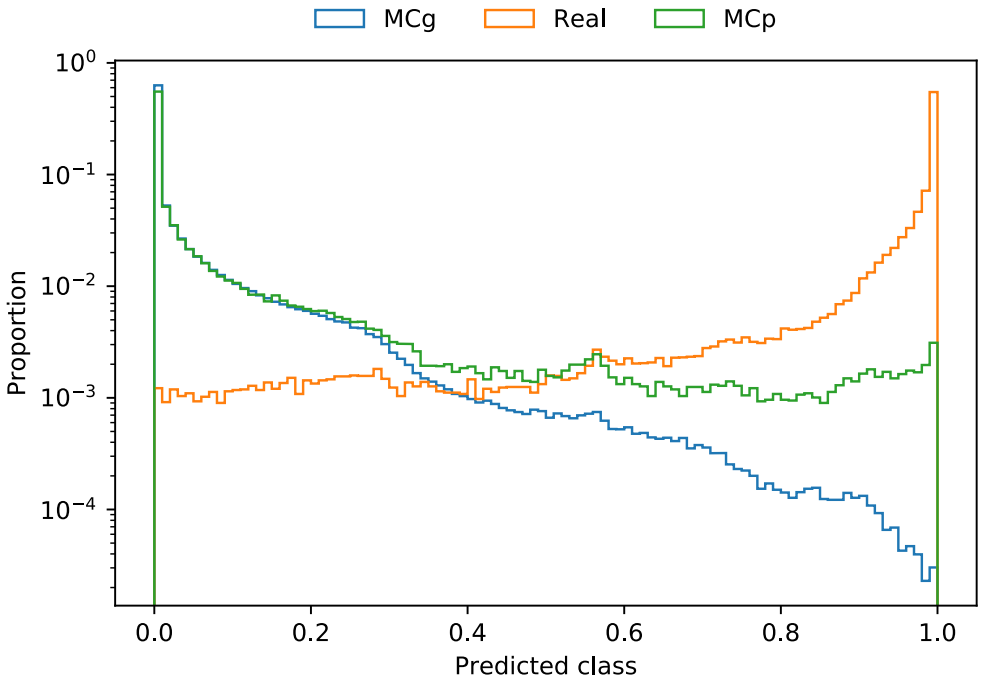


**Figure 5.2:** Predicted class value for a network trained to distinguish between `Real` and `MCp`. While the separation of real and simulated protons is easily possible, testing on a set of `MCg` shows even better classification results. The y-axis is normalized to the total event count in each dataset. The `MCg` dataset is roughly ten times larger than the training set. Thus the histogram is more smooth due to better statistics.

For the second task, the network is trained to classify `Real` from `MCp`. After the training converges at 93% accuracy, the network is tested by a large set of `MCg`. The distribution of predicted class labels is shown in Fig. 5.2.

Even though no training on simulated $\gamma$-ray events was performed, separation is clearly possible. This suggest the presence of MC exclusive features, detected by the CNN. The origin of such features is unclear as they can emerge at any point in a complex simulation pipeline.

The third task is a simple test to make sure that some changes of telescope configuration over time do not introduce features that can be used to easily distinguish between some runs. For the close pair as well as the distant pair of `Real` runs no separation is possible, the accuracy is stuck at 50% from the beginning and does not improve.

## 5.4 Outlook

This work provides an experimental deep learning implementation, built on simulated events for the H.E.S.S. experiment. The possibility to distinguish between real data and simulations using a CNN shows that simulated events contain different features than real events.

There are several strategies to identify the source of the discrepancy: Analyzing the features themselves could yield insights on the faulty block of the simulation. Frameworks like layerwise relevance propagation [91, 92] can provide insights about important image regions. A human expert could identify the origin of discrepancies by investigation of produced explanation heatmaps.

While it is well known that hadronic interactions cannot be modeled precisely, the separability of `MCg` suggests another source of uncertainty. Still one could look into classifying events simulated with different interaction models. Apart from the interaction model several other parameters influence the final image content. Adversarial play with all these parameters could yield more realistic shower images. While one network is trained to separate real data from simulations, another one is simultaneously trained to set simulations parameters that make this task difficult. However, this would require an incredible amount of simulation and network training time.

There are also several mitigation strategies, not resolving the issue at its root. Training a deep autoencoder [93] to reproduce real event images and evaluation of simulated images could allow the mapping of simulation data to real event images. Generative adversarial networks [94] could create images that look like real data but are synthesized based on simulation properties like particle type, energy and origin.

Further improvements are also possible on the technical side: Cleaned images contain mostly zeros ($\approx$99%), so a sparse network architecture could accelerate the computation [95, 96]. Architecture search and hyperparameter optimization rely on large initial investment of computational resources but reduce the time effort required for later training and analysis runs. However any more sophisticated architecture will be more susceptible to differences between real and simulated images.

Extending the demonstrated techniques to CTA is not easy. There are two main obstacles: Several different telescope types yielding different image sizes and shapes have to be united. Also the large amount of possible triggered combinations for a few dozen telescopes is an issue. This could be alleviated by recurrent neural networks as used in [30], accepting variable length and size input sequences. As CTA also heavily relies on simulated events, the discrepancy to real data continues to be an issue.

# Bibliography


[1] P. J. Bryant. A brief history and review of accelerators. In *CERN Accelerator School: Course on General Accelerator Physics Jyvaskyla, Finland, September 7-18, 1992*, pages 1–16, 1992.

[2] A. Codino. About the consistency of the energy scales of past and present instruments detecting cosmic rays above the ankle energy. *ArXiv e-prints*, October 2017.

[3] C. Grojean and M. Spiropulu. Proceedings of the 2009 CERN-Latin-American School of High-Energy Physics, Recinto Quirama, Colombia, 15 - 28 March 2009. *ArXiv e-prints*, October 2010.

[4] M. S. Longair. *High Energy Astrophysics*. Cambridge University Press, 1992.

[5] M. Ajello, W. B. Atwood, L. Baldini, et al. 3FHL: The Third Catalog of Hard Fermi-LAT Sources. ApJS, 232:18, October 2017.

[6] R. Rando and for the Fermi LAT Collaboration. Post-launch performance of the Fermi Large Area Telescope. *ArXiv e-prints*, July 2009.

[7] A. U. Abeysekara, R. Alfaro, C. Alvarez, et al. Sensitivity of the high altitude water Cherenkov detector to sources of multi-TeV gamma rays. *Astroparticle Physics*, 50:26–32, December 2013.

[8] G. Pühlhofer and for the H.E.S.S. collaboration. H.E.S.S. highlights. *ArXiv e-prints*, January 2018.

[9] J. Sitarek and the MAGIC Collaboration. Highlights of the MAGIC AGN program. *ArXiv e-prints*, August 2017.

[10] J. Holder. Latest results from VERITAS: Gamma 2016. In *6th International Symposium on High Energy Gamma-Ray Astronomy*, volume 1792 of *American Institute of Physics Conference Series*, page 020013, January 2017.

[11] CTA Consortium. Cherenkov Telescope Array: The Next Generation Gamma-ray Observatory. *ArXiv e-prints*, September 2017.

[12] N. Park, for the VERITAS Collaboration, Fermi-LAT Collaboration, and HAWC collaboration. VERITAS and Fermi-LAT observations of TeV gamma-ray sources from the second HAWC catalog. *ArXiv e-prints*, August 2017.

[13] D. Gora, M. Manganaro, E. Bernardini, et al. Search for tau neutrinos at PeV energies and beyond with the MAGIC telescopes. *ArXiv e-prints*, October 2017.

[14] G. Spengler and U. Schwanke. Signatures of Ultrarelativistic Magnetic Monopoles in Imaging Atmospheric Cherenkov Telescopes. In *Proceedings, 32nd International Cosmic Ray Conference (ICRC 2011): Beijing, China, August 11-18, 2011*, volume 5, page 105, 2011.

[15] M. Doro. Rare Events searches with Cherenkov Telescopes. In *European Physical Journal Web of Conferences*, volume 136 of *European Physical Journal Web of Conferences*, page 01003, March 2017.

[16] M. Doro. Gamma-ray, Particle and Exotic Physics at TeV energies with the MAGIC telescopes. *ArXiv e-prints*, June 2017.

[17] J. Knödlseder. The future of gamma-ray astronomy. *Comptes Rendus Physique*, 17:663–678, June 2016.

[18] A. M. Hillas. Cerenkov light images of EAS produced by primary gamma. *International Cosmic Ray Conference*, 3, August 1985.

[19] Werner Hofmann. Performance limits for Cerenkov instruments. In *Towards a network of atmospheric Cherenkov detectors VII, Palaiseau, April 27-29, 2005*, 2006.

[20] S. Ohm, C. van Eldik, and K. Egberts. $\gamma$/hadron separation in very-high-energy $\gamma$-ray astronomy using a multivariate analysis method. *Astroparticle Physics*, 31:383–391, June 2009.

[21] Stefan Ohm. *Development of an advanced $\gamma$/hadron separation technique and application to particular $\gamma$-ray sources with H.E.S.S.* PhD thesis, Heidelberg, 2010.

[22] Maria Krause, Elisa Pueschel, and Gernot Maier. Improved $\gamma$/hadron separation for the detection of faint $\gamma$-ray sources using boosted decision trees. *Astropart. Phys.*, 89:1–9, 2017.

[23] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. *ArXiv e-prints*, July 2017.

[24] Yann Lecun, Leon Bottou, Y Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324, 12 1998.

[25] K. Bernlöhr. Simulation of imaging atmospheric Cherenkov telescopes with CORSIKA and sim_ telarray. *Astroparticle Physics*, 30:149–158, October 2008.

[26] Tim Lukas Holch. A novel approach to $\gamma$-hadron separation for h.e.s.s. based on convolutional neural networks. Master's thesis, Humbold-Universität zu Berlin, 2016.

[27] Q. Feng, T. T. Y. Lin, and VERITAS Collaboration. The analysis of VERITAS muon images using convolutional neural networks. In M. Brescia, S. G. Djorgovski, E. D. Feigelson, G. Longo, and S. Cavuoti, editors, *Astroinformatics*, volume 325 of *IAU Symposium*, pages 173–179, June 2017.

[28] T. Lukas Holch, I. Shilon, M. Büchele, et al. Probing convolutional neural networks for event reconstruction in $\gamma$-ray astronomy with cherenkov telescopes. *ArXiv e-prints*, November 2017.

[29] D. Nieto, A. Brill, B. Kim, T. B. Humensky, and f. t. Cherenkov Telescope Array. Exploring deep learning as an event classification method for the Cherenkov Telescope Array. *ArXiv e-prints*, September 2017.

[30] I. Shilon, M. Kraus, M. Büchele, et al. Application of Deep Learning methods to analysis of Imaging Atmospheric Cherenkov Telescopes data. *ArXiv e-prints*, March 2018.

[31] T. C. Weekes, M. F. Cawley, D. J. Fegan, et al. Observation of TeV gamma rays from the Crab nebula using the atmospheric Cerenkov imaging technique. ApJ, 342:379–395, July 1989.

[32] S. P. Wakely and D. Horan. TeVCat: An online catalog for Very High Energy Gamma-Ray Astronomy. *International Cosmic Ray Conference*, 3:1341–1344, 2008.

[33] H.E.S.S. Collaboration, :, H. Abdalla, et al. The population of TeV pulsar wind nebulae in the H.E.S.S. Galactic Plane Survey. *ArXiv e-prints*, February 2017.

[34] F. Aharonian, A. G. Akhperjanian, A. R. Bazer-Bachi, et al. An Exceptional Very High Energy Gamma-Ray Flare of PKS 2155-304. ApJ, 664:L71–L74, August 2007.

[35] H.E.S.S. Collaboration, A. Abramowski, F. Aharonian, et al. H.E.S.S. observations of the Crab during its March 2013 GeV gamma-ray flare. A&A, 562:L4, February 2014.

[36] S. Thoudam, J. P. Rachen, A. van Vliet, et al. Cosmic-ray energy spectrum and composition up to the ankle: the case for a second Galactic component. A&A, 595:A33, October 2016.

[37] F. A. Aharonian. *Very high energy cosmic gamma radiation : a crucial window on the extreme Universe*. World Scientific Publishing Co, 2004.

[38] C. Patrignani et al. Review of Particle Physics. *Chin. Phys.*, C40(10):100001, 2016.

[39] Walter Heitler. *The Quantum Theory of Radiation*. Monographs on Physics. Oxford University Press, 1954.

[40] A. Fassò and J. Poirier. Spatial and energy distribution of muons in $\gamma$-induced air showers. *Phys. Rev. D*, 63:036002, Dec 2000.

[41] Joachim Hahn. *Supernova Remnants with H.E.S.S.: Systematic Analysis and Population Synthesis*. PhD thesis, Heidelberg, 2014.

[42] K. Bernlöhr. Impact of atmospheric parameters on the atmospheric Cherenkov technique. *Astroparticle Physics*, 12:255–268, January 2000.

[43] M. de Naurois and D. Mazin. Ground-based detectors in very-high-energy gamma-ray astronomy. *Comptes Rendus Physique*, 16:610–627, August 2015.

[44] R. Aaij, B. Adeva, M. Adinolfi, et al. Observation of J $/\psi$ p Resonances Consistent with Pentaquark States in $\Lambda_b{}^0 \to$J $/\psi$ $K^-$p Decays. *Physical Review Letters*, 115(7):072001, August 2015.

[45] Q.-S. Zhou, K. Chen, X. Liu, Y.-R. Liu, and S.-L. Zhu. Doubly heavy pentaquarks. *ArXiv e-prints*, January 2018.

[46] V. Crede and W. Roberts. Progress towards understanding baryon resonances. *Reports on Progress in Physics*, 76(7):076301, July 2013.

[47] T. Pierog. Open issues in hadronic interactions for air showers. In *European Physical Journal Web of Conferences*, volume 145 of *European Physical Journal Web of Conferences*, page 18002, June 2017.

[48] T. Pierog, R. Engel, D. Heck, and G. Poghosyan. Future of Monte Carlo simulations of atmospheric showers. In *European Physical Journal Web of Conferences*, volume 89 of *European Physical Journal Web of Conferences*, page 01003, March 2015.

[49] Sergey Ostapchenko and Marcus Bleicher. Constraining pion interactions at very high energies by cosmic ray data. *Phys. Rev.*, D93(5):051501, 2016.

[50] L. B. Arbeletche, V. P. Goncalves, and M. A. Muller. Investigating the influence of diffractive interactions on ultra - high energy extensive air showers. *ArXiv e-prints*, January 2017.

[51] Air shower cherenkov light simulations. https://www.mpi-hd.mpg.de/hfm/CosmicRay/ChLight/ChLat.html. Accessed: 2018-02-26, Courtesy of Konrad Bernlöhr.

[52] G. Giavitto, T. Ashton, A. Balzer, et al. A major electronics upgrade for the H.E.S.S. Cherenkov telescopes 1-4. In A. S. Borisov, V. G. Denisova, Z. M. Guseva, et al., editors, *34th International Cosmic Ray Conference (ICRC2015)*, volume 34 of *International Cosmic Ray Conference*, page 996, July 2015.

[53] S. Bonnefoy, T. Ashton, M. Backes, et al. Performance of the upgraded H.E.S.S. cameras. *ArXiv e-prints*, August 2017.

[54] Arnim Balzer. Systematic studies of the h.e.s.s. camera calibration. Master's thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2014.

[55] Stefan Funk. *A new population of very high-energy $\gamma$-ray sources detected with H.E.S.S. in the inner part of the Milky Way.* PhD thesis, Heidelberg, 2005.

[56] W. Hofmann, I. Jung, A. Konopelko, et al. Comparison of techniques to reconstruct VHE gamma-ray showers from multiple stereoscopic Cherenkov images. *Astroparticle Physics*, 12:135–143, November 1999.

[57] Mathieu De Naurois. *Very High Energy astronomy from H.E.S.S. to CTA. Opening of a new astronomical window on the non-thermal Universe.* Habilitation à diriger des recherches, Université Pierre et Marie Curie - Paris VI, March 2012.

[58] P. T. Reynolds, C. W. Akerlof, M. F. Cawley, et al. Survey of candidate gamma-ray sources at TeV energies using a high-resolution Cerenkov imaging system - 1988-1991. ApJ, 404:206–218, February 1993.

[59] F. Aharonian, A. G. Akhperjanian, A. R. Bazer-Bachi, et al. Observations of the Crab nebula with H.E.S.S. A&A, 457:899–915, October 2006.

[60] S. Le Bohec, B. Degrange, M. Punch, et al. A new analysis method for very high definition imaging atmospheric Cherenkov telescopes as applied to the CAT telescope. *Nuclear Instruments and Methods in Physics Research A*, 416:425–437, October 1998.

[61] M. de Naurois and L. Rolland. A high performance likelihood reconstruction of $\gamma$-rays for imaging atmospheric Cherenkov telescopes. *Astroparticle Physics*, 32:231–252, December 2009.

[62] R. D. Parsons and J. A. Hinton. A Monte Carlo template based analysis for air-Cherenkov arrays. *Astroparticle Physics*, 56:26–34, April 2014.

[63] S. Vincent for the VERITAS Collaboration. A Monte Carlo template-based analysis for very high definition imaging atmospheric Cherenkov telescopes as applied to the VERITAS telescope array. *ArXiv e-prints*, September 2015.

[64] Johannes Welbl. Casting random forests as artificial neural networks (and profiting from it). In *GCPR*, 2014.

[65] N. Frosst and G. Hinton. Distilling a Neural Network Into a Soft Decision Tree. *ArXiv e-prints*, November 2017.

[66] Michael Bulmer. *Francis Galton - Pioneer of Heredity and Biometry.* The Johns Hopkins University Press, 2003.

[67] Francis Galton. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246–263, 1886.

[68] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. In *14th International Conference on Artificial Intelligence and Statistics*, volume 15, pages 315–323, Fort Lauderdale, United States, April 2011.

[69] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for Activation Functions. *ArXiv e-prints*, October 2017.

[70] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[71] Y. LeCun, B. Boser, J. S. Denker, et al. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[72] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representation by error propagation. In David E. Rumelhart, James L. McClelland, and CORPORATE PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[73] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[74] P. Goyal, P. Dollár, R. Girshick, et al. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *ArXiv e-prints*, June 2017.

[75] V. Patel. The Impact of Local Geometry and Batch Size on the Convergence and Divergence of Stochastic Gradient Descent. *ArXiv e-prints*, September 2017.

[76] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le. Don't Decay the Learning Rate, Increase the Batch Size. *ArXiv e-prints*, November 2017.

[77] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[78] Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, and Rob Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1058–III–1066. JMLR.org, 2013.

[79] P. Murugan and S. Durairaj. Regularization and Optimization strategies in Deep Convolutional Neural Network. *ArXiv e-prints*, December 2017.

[80] S. Chetlur, C. Woolley, P. Vandermersch, et al. cuDNN: Efficient Primitives for Deep Learning. *ArXiv e-prints*, October 2014.

[81] Erik Meijering. A chronology of interpolation: From ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, 90:319 – 342, 04 2002.

[82] Tim Lukas Holch and Constantin Steppa. Hexagdly - hexagonal convolutions with pytorch, February 2018.

[83] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, second edition, 2000.

[84] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *ArXiv e-prints*, December 2015.

[85] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Weinberger. Deep Networks with Stochastic Depth. *ArXiv e-prints*, March 2016.

[86] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning Transferable Architectures for Scalable Image Recognition. *ArXiv e-prints*, July 2017.

[87] E. Real, A. Aggarwal, Y. Huang, and Q. V Le. Regularized Evolution for Image Classifier Architecture Search. *ArXiv e-prints*, February 2018.

[88] Pierre Baldi and Yves Chauvin. Neural networks for fingerprint recognition. *Neural Computation*, 5(3):402–418, 1993.

[89] R. Chalme-Calvet, M. de Naurois, J.-P. Tavernet, and for the H.E.S.S. Collaboration. Muon efficiency of the H.E.S.S. telescope. *ArXiv e-prints*, March 2014.

[90] M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *ArXiv e-prints*, October 2017.

[91] Sebastian Bach, Alexander Binder, Grégoire Montavon, et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140, 07 2015.

[92] A. Binder, G. Montavon, S. Bach, K.-R. Müller, and W. Samek. Layer-wise Relevance Propagation for Neural Networks with Local Renormalization Layers. *ArXiv e-prints*, April 2016.

[93] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

[94] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, et al. Generative Adversarial Networks. *ArXiv e-prints*, June 2014.

[95] B. Graham and L. van der Maaten. Submanifold Sparse Convolutional Networks. *ArXiv e-prints*, June 2017.

[96] B. Graham. Sparse 3D convolutional neural networks. *ArXiv e-prints*, May 2015.