technische universität
dortmund

Arbeit zur Erlangung des akademischen Grades
Bachelor of Science

Gamma-Hadron Separation with Deep Learning for the First G-APD Cherenkov Telescope

Jan Moritz Behnken
geboren in Achim

2017

Lehrstuhl für Experimentelle Physik V
Fakultät Physik
Technische Universität Dortmund

Erstgutachter:	Prof. Dr. Dr. Wolfgang Rhode
Zweitgutachter:	Prof. Dr. Kevin Kröninger
Abgabedatum:	15. September 2017

Abstract

High energy particles from cosmic sources reach the atmosphere and interact with it. The resulting particle showers emit Cherenkov radiation, which can be recorded by telescopes at ground-level.

This thesis is motivated by the study of such images taken by the First G-APD Cherenkov Telescope (FACT) through Deep Learning. The purpose of this study is, to increase the sensitivity of FACT. To identify the source of the particles a separation of images caused by gamma rays and hadrons will be performed. For that reason a simulated dataset is used to train a Convolutional Neural Network (CNN). To achieve optimal performance, 30 different network architectures and regularizations are being compared to each other.

Afterwards a comparison is made between the performance of the currently used classifier and the CNN. Although the CNN performs comparably on a test dataset, it clearly fails to separate gamma rays from hadrons on real measured images. This can be attributed to a systematic error between simulated and real datasets (Monte Carlo Mismatch).

Kurzfassung

Von kosmischen Quellen erreichen energiereiche Teilchen die obersten Schichten der Erdatmosphäre und wechselwirken mit dieser. Dabei entstehende Teilchenschauer senden Tscherenkov-Strahlung aus, welche von Teleskopen am Erdboden aufgenommen werden kann.

Motivation für diese Arbeit ist die Untersuchung dieser Bilder vom First G-APD Cherenkov Teleskop (FACT) mittels Deep Learning. Ziel dabei ist eine Verbesserung der Empflindlichkeit von FACT. Um die Quellen der Teilchen ausfindig zu machen, werden die durch Gammastrahlung und Hadronen ausgelösten Bilder separiert. Hierfür wird auf simulierten Datensätzen ein Convolutional Neural Network (CNN) trainiert. Um eine optimale Performance zu erreichen, werden 30 unterschiedliche Netzwerkarchitekturen und Regularisierungen miteinander verglichen.

Anschließend wird ein Vergleich der Performance zwischen dem bisher eingesetzten Klassifizierer und dem CNN gezogen. Obwohl das CNN auf Testddaten eine vergleichbar gute Leistung erreicht, zeigt sich ein deutliches Versagen beim Separieren von real gemessenen Bildern. Dies wird auf einen systematischen Fehler zwischen simulierten und echten Daten zurückgeführt (Monte Carlo Mismatch).

Contents

1 Introduction

The topic of this thesis falls under an area of astrophysics, investigating some of the most extreme phenomena in the universe. For example, supernovae and black holes are known to be sources of high energy particles, but the processes emitting this radiation are still a subject of research. To deepen the knowledge concerning these processes the emitted gamma rays and hadrons can be examined by telescopes.

Whereas the direction of hadrons is scrambled by magnetic fields while passing through galaxies and nebulae, gamma rays are unaffected by such fields and propagate in straight lines. When the sub-atomic particles arrive at the Earth, the direction of the gamma rays is preserved and can be used to detect their source and measure its properties.

When entering the atmosphere both gamma rays and hadrons collide with other particles, transferring some of their energy to their collision partners. Both types of particles cause air showers which emit Cherenkov radiation by reason of the high energies involved. Ground-level telescopes can measure the gamma rays and hadrons indirectly by observing the brief Cherenkov flash.

Afterwards, machine learning algorithms classify the recorded images as being induced by a gamma ray or a hadron. Using only the measurements caused by gamma rays, their preserved direction reveals their cosmic source. Further investigation of these gamma events can unveil source specific characteristics.

This thesis will apply Deep Learning and Convolutional Neural Networks to FACT images and will compare the network's performance to the currently-used classification algorithm. To achieve this, different network architectures will be implemented in Google's Python library `TensorFlow`. Afterwards the network's hyperparameters are optimized and these networks are evaluated. Finally, the best CNN model will be used to analyse Crab nebula observations and the results will be compared to the currently-used Random Forest.

2 Essentials

2.1 Cosmic Radiation

The cosmos is filled with particles of many kinds and sources. Whereas the direction of travel of charged particles is constantly bent by electromagnetic fields on their journey through space, uncharged particles such as gamma rays preserve their direction during their voyage. These circumstances allow the sources to be identified. Source specific characteristics can be measured by determining the particle's properties.

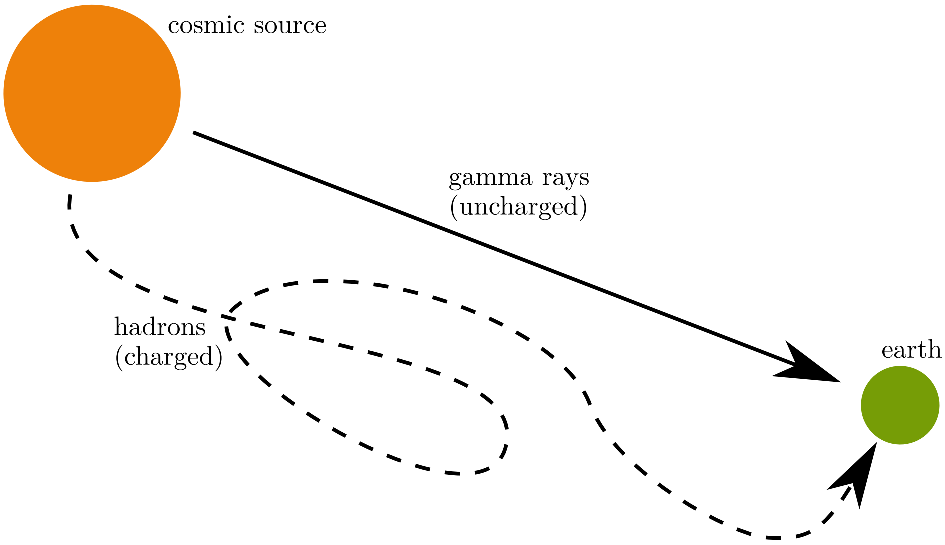


Figure 2.1: Uncharged particles propagate in straight lines and charged particles propagate unpredictably through space.

Neither gamma particles nor hadrons can pass through the Earth's atmosphere and for this reason collide with it. In this way the cosmic particle transfers some of its energy to its collision partner. This causes an air shower by reason of the high energies in the MeV range involved. Since the speed of light in air is lower than in space, some particles in an air shower can be faster than the speed of light in air without violating the speed of light in vacuum; these particles emit a cone of Cherenkov radiation. When a high energy cosmic particle interacts with the atmosphere, a short flash of Cherenkov light can be detected at ground-level [1].

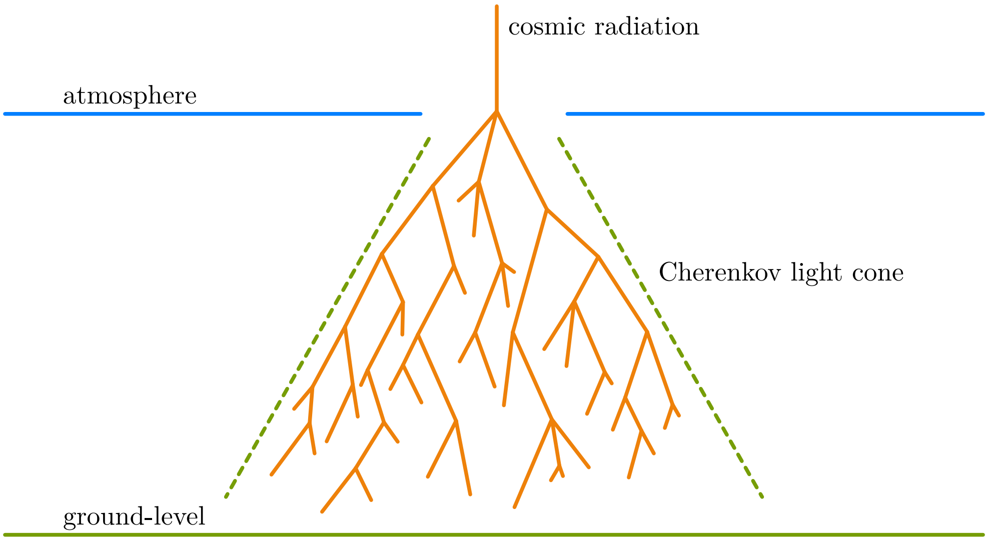


Figure 2.2: Cosmic particles cause air showers in the atmosphere by colliding with it. These showers emit Cherenkov light

2.2 The First G-APD Cherenkov Telescope

Among other telescopes, the First G-APD Cherenkov Telescope (FACT) on La Palma records these light flashes. It uses Geiger-mode avalanche photodiodes (G-APDs) as optical sensors for a test benchmark of this technology. With a comparatively small mirror surface area of $9.5\,\mathrm{m}^2$ it has operated since 2011 and takes images of air showers caused by cosmic radiation in the TeV energy range [7].

The 1440 camera pixels form a hexagonal grid. This hexagonal pixel structure represents a challenge for further image processing as software for image classification was developed for images with square pixels. Therefore, the camera image has to be transformed by using the pixel IDs to map from the hexagonal to a square grid structure [5]. Skewing the image by translating it to a different coordinate system and padding it with empty pixels enables further processing without developing special software. One drawback of this procedure is the loss of some direct neighborhood information for every pixel (hexagonal: 6 neighbors, square: 2 times 4 equivalent neighbors).

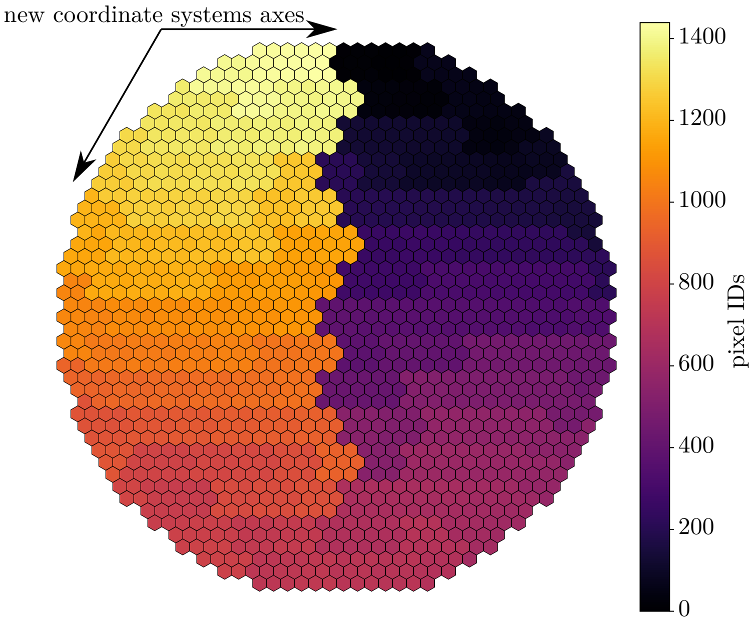


Figure 2.3: Translating the FACT camera image to a new coordinate system by using the axes shown allows a transformation from a hexagonal to a square grid structure.

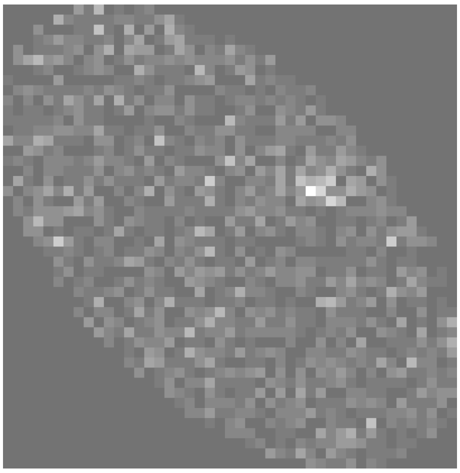


Figure 2.4: The transformed camera image will be padded with zeros and has a final size of 45 by 46 pixels.

2.3 Convolutional Neural Networks

In recent years Deep Learning has evolved at an incredible rate and made impressive progress in image classification tasks. In light of this, this thesis tests the utility of Convolutional Neural Networks (CNN) for classifying the skewed camera images by the particle type which induced the air shower. For differentiating between images caused by gammas or hadrons, there are many possible architectures for the CNN. With this in mind, every architecture is composed of layers each with a different role. The image will then be passed from layer to layer, undergoing transformations at each one.

- Convolution layers (denoted in the plots by "c") act as feature generators. These layers allow for the translation invariant recognition of patterns.
- Pooling layers reduce the feature space by selecting the most important features. They follow convolution layers and will not be denoted in the plots.
- Fully-connected layers (denoted by "f") will make up the final layers of the net. They combine the computed features and classify the image.

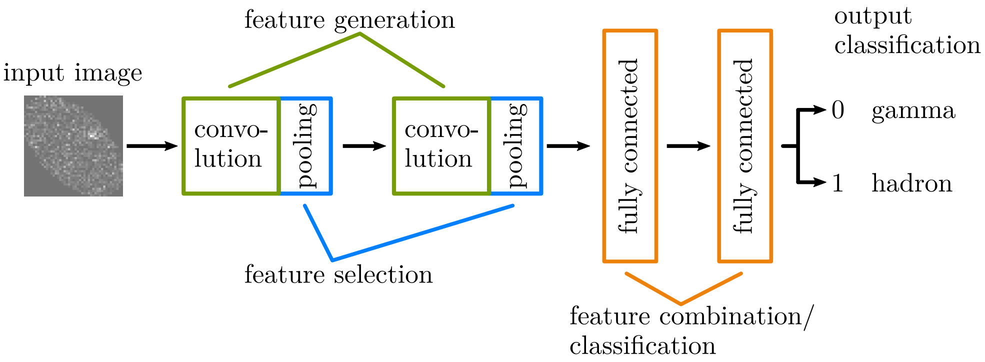


Figure 2.5: The input image will be transformed by each layer and passed onto the next one until it finally reaches the output layer, which classifies the image.

To minimize overfitting behavior and maximize generalizability of the network, different approaches can be used. To attain evenly-distributed pixel values, each batch of images fed into the network is preprocessed to have a mean of 0 and a standard deviation of 1 (batch normalization).

At the expense of longer training times, dropout layers ("d") can be inserted after any layer. This will drop some of the information flowing through the network and force it to learn distributed representations of every feature, making the network more robust.

To enable deeper networks and faster training, pretraining will be employed. After training shallow networks for a short time, a new, untrained layer will be attached. This process will be repeated until the network's growth reached its final depth. While most of the layers only have to adapt slightly, the new layers can adjust their behavior according to the pretrained network quickly. Additional insights into Convolutional Neural Networks and Deep Learning can be found in Ian Goodfellow's book "Deep Learning" [3].

For implementing the layers and the architectures Google's Python library `TensorFlow` [6] has been used. GPUs were used for training to take advantage of their parallel computing capabilities and thereby cutting down the training time. This results in training times fluctuating between 15 min and 90 min for a single network.

3 Optimizing the CNN

3.1 Processing the Images

For the following steps, a simulated dataset containing roughly 800 000 gamma events and 400 000 hadron events has been used. Each recorded event holds the counts of every arrived photon for all 1440 pixels for 100 consecutive time frames of 0.5 ns. To reduce these 144 000 variables, the counts of photons for each pixel have been summed along the time axis (time series summation).

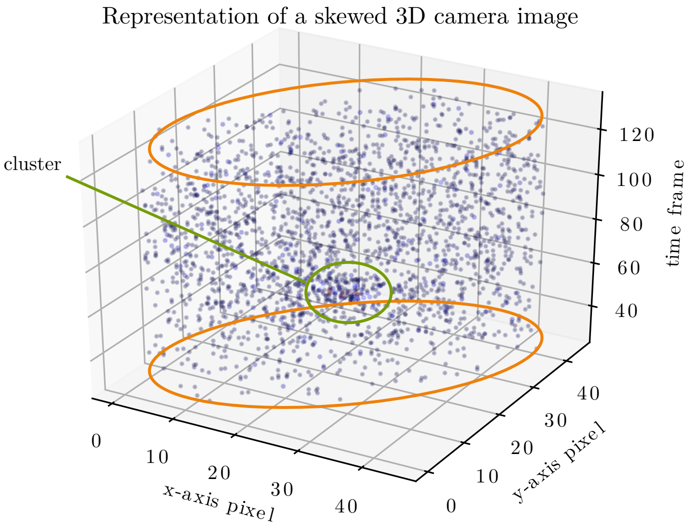


Figure 3.1: An event contains 100 consecutive images of the Cherenkov flash. To minimize the number of variables, all values for a single pixel have been summed up for each event.

Naturally, many background photons are contained in the image. Since the telescope triggers all events in approximately the same time frame, the photons of the air shower can be cut out by removing preceding and succeeding time frames filled with noise. This results in cleaner, denoised images for the pattern detection in the CNN.

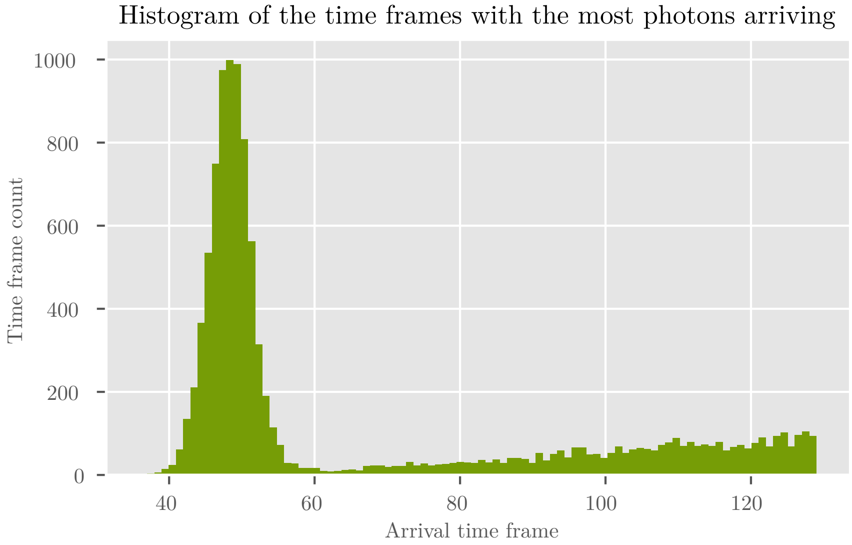


Figure 3.2: The brightest time frames for 10 000 events are shown in the histogram. The distribution peaks around the triggering time. By cutting out this cluster, noise can be reduced.

For 10 000 events, the time frame with the most photons arriving was computed. These bright time frames most likely contain the events to examine. Creating a histogram of these time frames, highlights that the telescope triggers the events between frames 35 and 60 nearly every time. As a result, only this range of frames will be used for the denoised images, dropping all other frames, which contain mostly noise.

In the following paragraph, all architectures will be evaluated on the images containing all photons as well as the denoised ones.

3.2 Comparing Architectures

In this section each tested network architecture is composed of a convolution layer followed by a pooling layer (c) and fully-connected layers (f). The architecture notation follows this example:

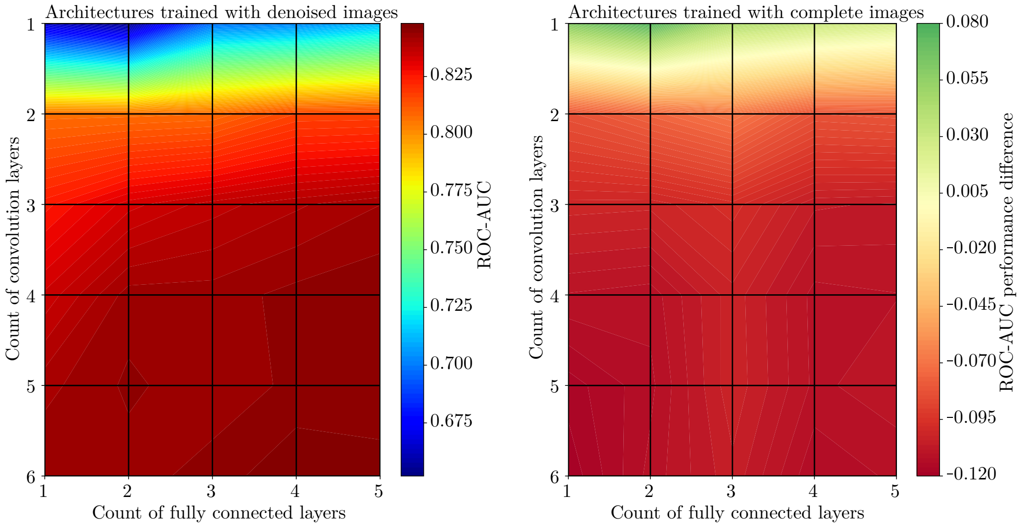
- A "`3c_2f`" architecture translates to a network starting with three convolution-and-pooling layers and ending with two fully connected layers.

There are four important hyperparameters for networks of this kind:

- The number of images in one batch fed to the network (batch-size)
- The size of the patch/kernel in the convolution layers (patch-size)
- The number of feature maps the convolution layers compute (depth)
- The number of neurons contained in the fully-connected layers (neurons)

For all parameters a random grid search was performed over the course of the following steps. To reduce the impact of random fluctuations in the network's performance caused inter alia by the grid search, 50 networks have been trained for each architecture.

The common procedure of using a separate training, validation and testing dataset has been adopted. To measure the network's performance, the area under the Receiver-Operating Characteristic (ROC-AUC) [2] is used. The training of a network is terminated when the ROC-AUC-score of the validation dataset does not rise anymore (early stopping).



(a) CNNs on denoised images

(b) CNNs on noisy images

Figure 3.3: The ROC-AUC performances of 30 architectures are compared on both denoised and noisy images. Since denoised images perform better, the second plot shows the difference between the denoised and noisy images' ROC-AUC scores.

By comparing the networks trained on noisy and denoised images, it becomes apparent that denoised images produce a much more reliable and therefore better network.

Looking at the architectures, it appears unambiguous that deeper networks perform much better than shallow ones. The number of feature-generating convolution layers seems to have a greater impact than the number of fully connected layers on the ROC-AUC score.

By examining the best-performing denoised deep network architectures, only sleight differences can be seen. Since a sample of only 50 trained networks guarantees no statistical certainty, no distinct best architecture can be proclaimed.

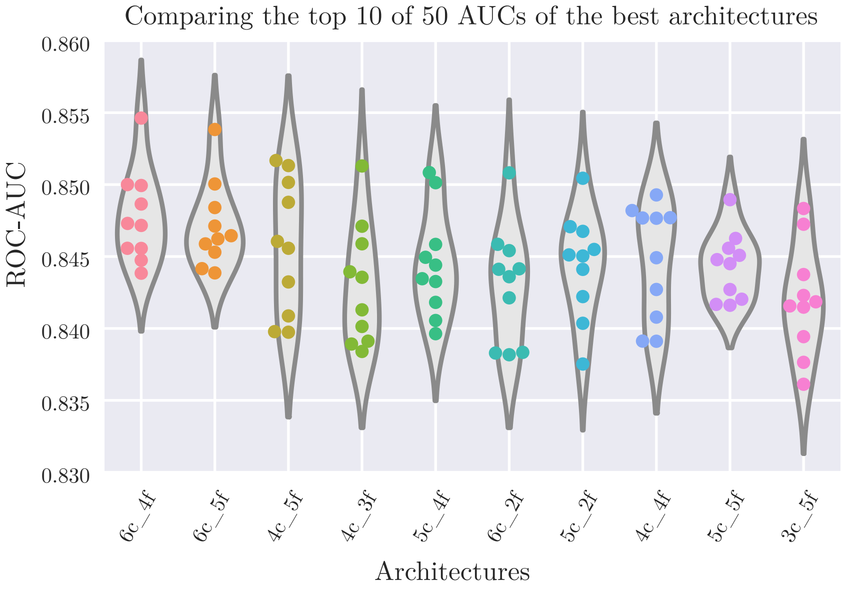
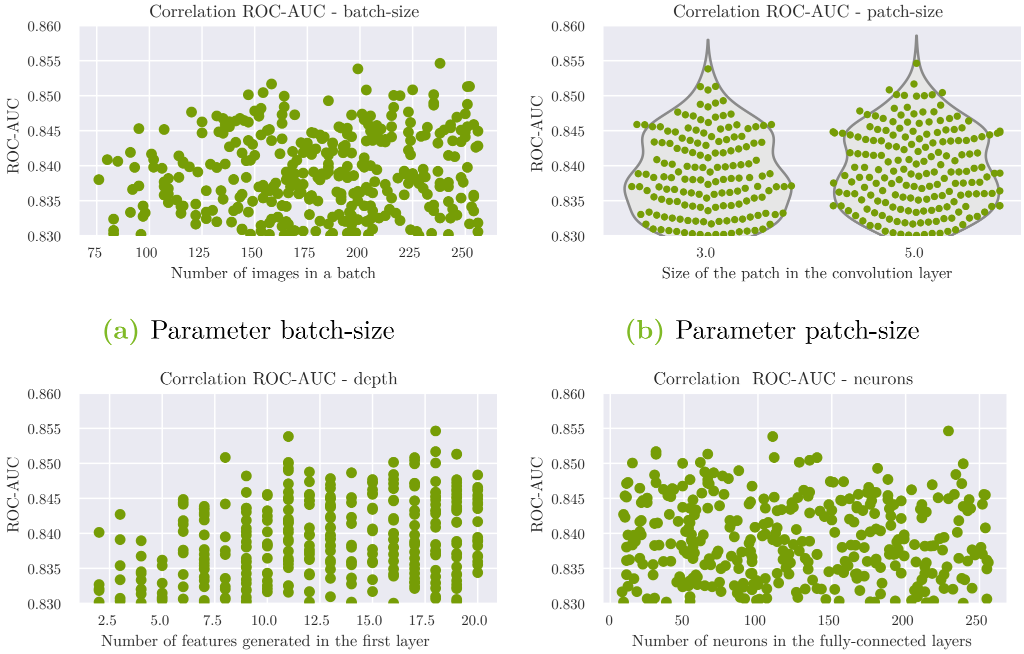


Figure 3.4: For the comparison of the best denoised architectures the 10 best performing networks from 50 are being compared in this plot.

It can be seen that deeper networks perform better on the given task and the more convolution layers included in the architecture, the higher the ROC-AUC score. Additionally, using only the important part of the recorded events and dropping some noise, increases the network's performance as well.

For a meaningful conclusion concerning the architectures, the hyperparameters of the networks will be investigated. Since the optimal values for the hyperparameters could not be determined beforehand, all parameters for the above investigations were randomly chosen from a suitable value range. Therefore a conclusion can be drawn afterwards to narrow this range down or investigate a more promising value range.

Since only the best networks are of interest, the hyperparameters of the high performing ones are investigated. For this analysis, the performance of all computed networks are compared regardless of their architecture. As all parameters (batch-size, patch-size, depth and neurons) show only a small impact on the ROC-AUC score, only small gains in performance can be achieved through optimizing the hyperparameters. For this reason the random grid search will be performed for further investigations.



(a) Parameter batch-size

(b) Parameter patch-size

(c) Parameter depth (first c-layer)

(d) Parameter hidden-nodes

Figure 3.5: Only a small impact on the ROC-AUC can be detected by comparing the hyperparameters. Since optimizing the parameters specifically appears complicated, the random grid search will be utilized henceforth.

3.3 Regularizing the Network

The best-performing "`6c_4f`" architecture is fixed for further investigations so that the feature space to explore is reduced. To stabilize behavior of the network, regularizing dropout layers will be integrated into the network's architecture.

Since the layer can be combined with every existing layer in the network and can likewise adopt different values for the amount of data to drop out in every layer, a large feature space has to be evaluated. As adding dropout to a network extends the training time many times over, only one dropout layer at once can be tested at every position. In the first round of testing, a dropout rate of 50 % will be used. Investigating every possible permutation of multiple dropout layers is not possible because of the size of the feature space that must be explored.

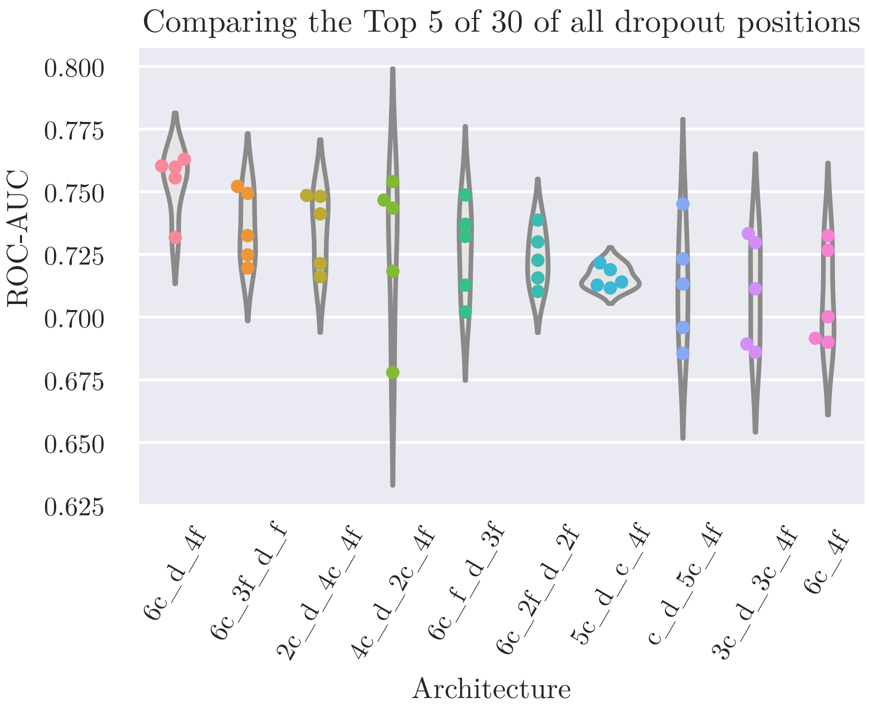


Figure 3.6: Using dropout layers increases the performance every time, but inserting the layer between the convolution and the fully connected layers, unlocks its potential the most.

Positioning the dropout layer between the convolution layers and the following fully-connected layers appears to have the most positive impact on the network's performance. Regularizing the fully connected layers seems to strengthen the network as well. On the contrary, dropping too much information in the convolution layers weakens the network overall. Therefore, the dropout rate is most important, when it is combined with the layer's position.

It has been decided that dropout layers shall be implemented after every layer, but using different dropout rates. The rates in table 3.1 were taken from corresponding literature [3].

Table 3.1: The chosen dropout rates for the different dropout positions

Dropout position	Dropout rate
c- d -c	0.90
c- d -f	0.75
f- d -f	0.50

Since dropout prolongs the network's training cycle and increases the difficulty of training deep networks in general, pretraining is used to enable efficient training of many layers.

A short network containing dropout layers is trained for a few epochs and the layers adjust their inner values using gradient decent. Afterwards new layers are appended to the network and the short training epoch starts again. In this case training epochs consisting of 0, 1000, 2000, 5000 and 8000 batches are being compared. To complete the training after the network has grown to its full depth, a normal training epoch with early stopping is used.

Using different amounts of pretraining batches, the results illustrate an increase in the performance when compared to the networks without pretrained layers. By pretraining the network too much, there seems to be an decrease in the performance through overfitting. Therefore, it is advisable to choose the pretraining amount carefully. In this case 5000 batches of images have been chosen.

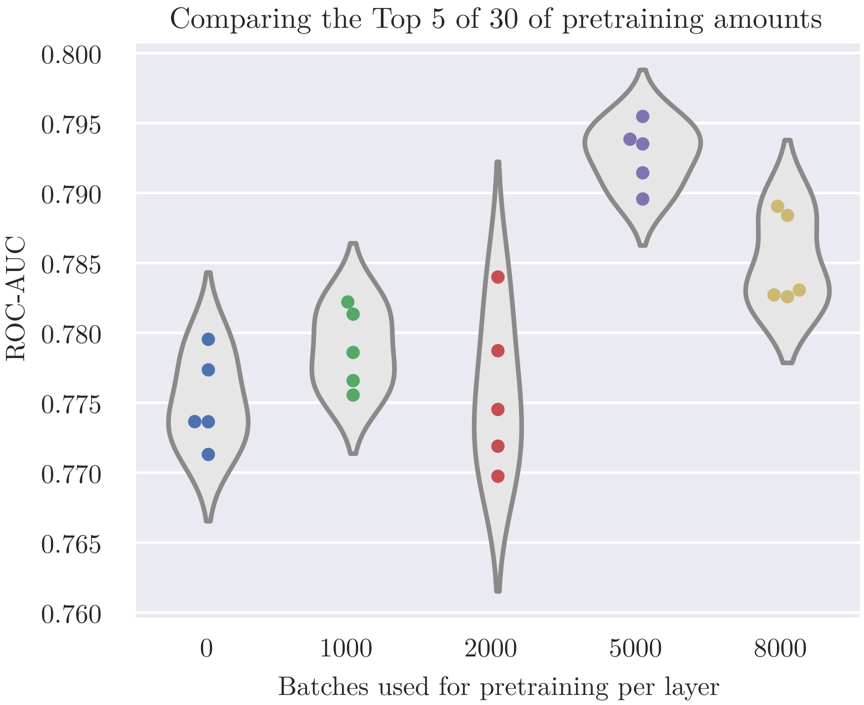


Figure 3.7: Implementing pretraining in the training process increases the network's performance in general. Since too much pretraining appears to be counterproductive, 5000 batches will be used henceforth.

4 Results

After the optimizing process 20 CNNs were trained with the gained insights. A "`6c_4f`" architecture was trained with the mentioned dropout rates and 5000 batches of pretraining for every layer. The hyperparameters were set by a random grid search.

For reviewing the best, trained network, a so far unused simulated dataset has been used. As a result, the separation of hadrons and gamma rays through the CNN's processing can be observed and evaluated. By processing 500 000 events of both classes and plotting the resulting predictions, a separation of those classes can be observed. Although many events overlap and are therefore not separable, both curves are slightly skewed to their respective end of the axes. With this dataset, a ROC-AUC score of 0.817 can be achieved.

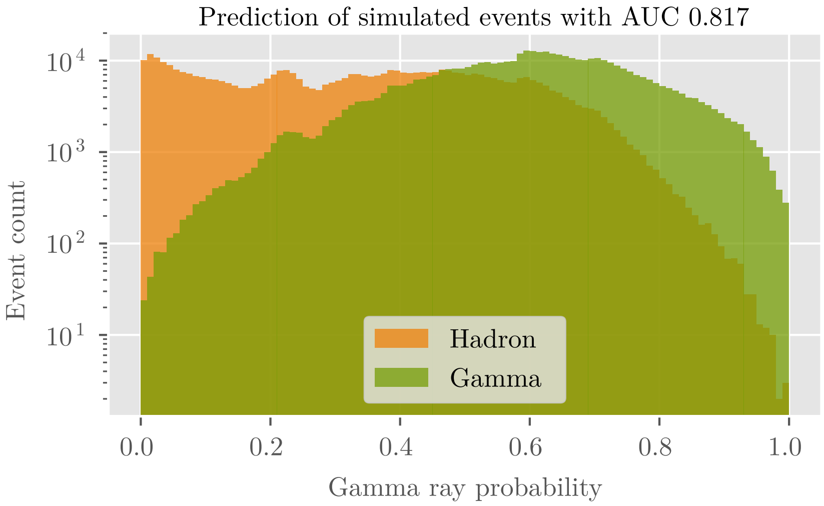


Figure 4.1: Using an unseen, simulated dataset for the evaluation of the CNN's performance, a separation of the two classes is visible.

In the best case a real-image dataset displays a similar distribution of the classes, only distinguished from the simulated events by the ratio of hadrons to gamma rays.

The real-image dataset consists of data, which has been recorded between 2013 and 2014 and shows the activity of the Crab nebula. It contains roughly 20 000 000 events.

By comparing the histograms of the simulated and real datasets, a distinctly smaller occurrence of gamma rays can be noted. This could arise from a small ratio in the cosmic radiation or a poor performance of the CNN caused by a mismatch between the real and simulated datasets.

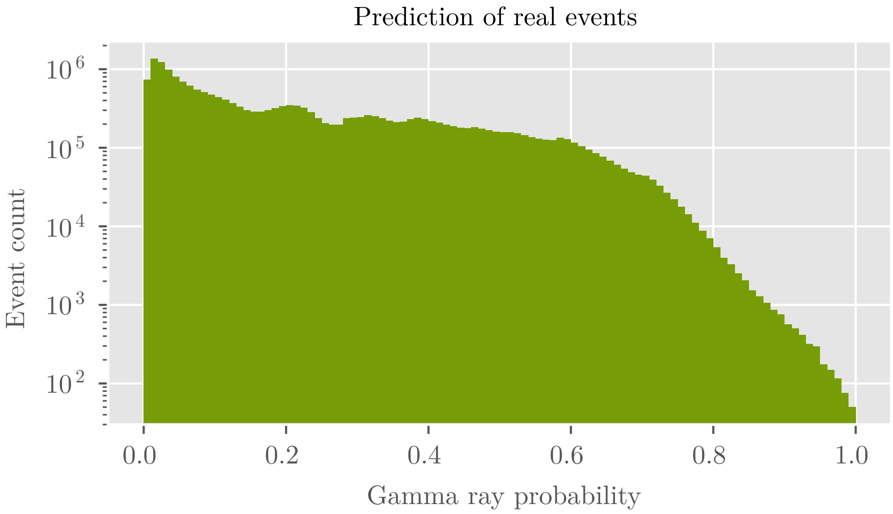


Figure 4.2: Using a real dataset of the Crab nebula, the predictions imply a high rate of hadrons and a small rate of gamma rays in the cosmic radiation.

Since the Crab nebula is a well surveyed source by many telescopes around the globe, it should be detectable with the newly predicted gamma rays at its position. Since every gamma ray can be retraced to its origin, the gamma ray activity of six opposing positions in FACT's field of view will be compared. The first position ("On") is the calculated position of the Crab nebula and contains activities of the source and the background radiation. The five other positions ("Off") lie circular and equally spaced next to the first one and therefore contain only background radiation. In this way a difference of activity at the Crab nebula position can be detected.

The positions will be compared using a Theta-plot. Both position's centers are placed at zero on the x-axis. The x-axis measures the angular distance to this position, while the y-axis depicts the activity of the cosmic region. The crucial significance value summarizes this difference. In this case, the Crab nebula can be detected with 24.4 σ. Since comparable approaches with currently used Random Forest classifiers reach 39.89 σ [8], a poor performance of the CNN can be assumed.

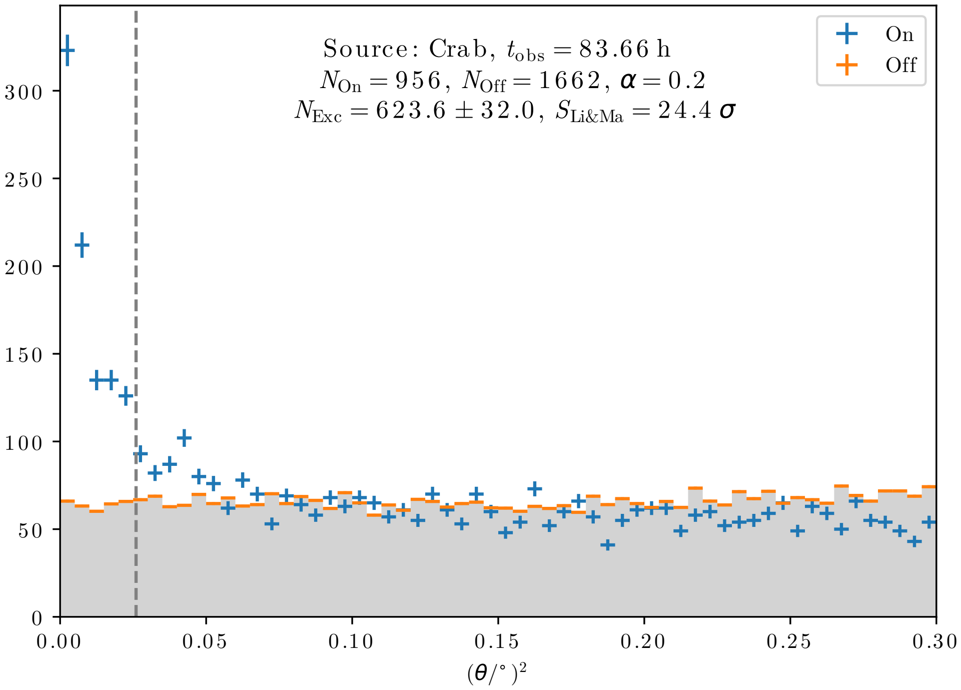


Figure 4.3: After 83.7 h of observation time the Crab nebula can be seen with the CNN with a significance of 24.4 σ. To optimize the significance, an angular cut at 0.026 and a threshold of 0.766 have been used for this plot.

This poor performance can derive from two distinct origins: either the network can be optimized to overcome the performance difference or the simulated input data contains features which the CNN utilizes, that are absent in the real images (Monte Carlo Mismatch). Since much effort has been put into optimizing the network, only small prospective improvements can be expected from further optimization. Therefore, the big performance difference seems to emerge from a mismatch between the training dataset and the real images.

5 Conclusion

5.1 Potential Improvements

To enhance the network's performance, there are two different approaches: on the one hand, there is the possibility of changing the input data; on the other hand, there are further options to improve the network's structure and data handling. Options to improve results will be described in the following paragraph. The order follows the data's pathway through the network and its programs.

First and foremost a simulated dataset has been used and the downside of this has been described. The mismatch between simulated and real data seems to arise a malfunction when predicting real images. Improving this dataset would be a complex and time-consuming task without any improvement guarantees. As this may not solve the problem, a more promising approach could be to switch from generated images to real images. The current classifier could predict labels for each real image, and these labels and images would form the training data for the network. In contrast to the first approach, uncertain labels would be the main challenge to overcome.

Additionally, the flat hexagonal structure has been skewed to fit into a flat square structure. Alternatively, a two-dimensional hexagonal structure could be transformed to a three-dimensional cubic structure without losing any neighborhood information at all. Furthermore, the time series information has been lost in this thesis by summing it up. In contrast to a two-dimensional convolution with loss of information, as performed in this thesis, a four-dimensional convolution with all information could be performed for the feature generation. For this, the library `TensorFlow` will not be sufficient as such high dimensional convolutions are not yet supported.

Although many architectures have been implemented and many hyperparameters have been independently evaluated, only a small feature space could be investigated in this thesis. Naturally, expanding the hyperparameter optimization could yield better results. Since all hyperparameters form a feature space themselves, a machine learning algorithm could be designed to predict a network's performance by using the hyperparameters as input. To optimize a network's performance during the training, a reinforcement learning algorithm (Q-Learning) could increase the network's performance. Rewarding the reinforcement learner for increasing the best performance, could achieve a more optimized network overall [4].

As the mismatch in the data will not be overcome by increasing the performance of the network itself, the training data must first of all be exchanged. After that, the network proposed in this thesis could investigate the performance using the new data instead to the simulated data used here. Once this has been successful, the other suggestions could be implemented.

5.2 Outlook

To classify images originating from cosmic radiation, a Convolutional Neural Network has been trained on a simulated dataset. This network has been optimized, evaluated and performs comparably to the currently-used Random Forest on this task. By predicting real images and computing the significance of the Crab nebula, a crucial performance difference between the CNN and the other classifiers emerges. Since CNNs have outperformed other classifiers in image classification tasks in the past, this gap cannot be explained by a poorly optimized network alone. The known mismatch in the simulated training dataset and the real images can be again confirmed. To overcome this problem, an improved training dataset is the most promising option. As simulated datasets contain the threat of mismatches every time, training the CNN on real images could be the solution. This introduces the challenge of creating a dataset containing real images with reliable labels. As a first step, the current classifiers could be used to label the images. The most reliable images could serve as the training data for a CNN. This CNN's performance could yield new insights as to whether this approach is encouraging.

Bibliography

[1] Demtröder. *Kern-, Teilchen- und Astrophysik*. Springer, 2017.

[2] Tom Fawcett. *An introduction to ROC analysis*. 2005. URL: `http://people.inf.elte.hu/kiss/12dwhdm/roc.pdf`.

[3] Courville Goodfellow Bengio. *Deep Learning*. The MIT Press, 2016.

[4] Samantha Hansen. *Using Deep Q-Learning to Control Optimization Hyperparameters*. 2016. URL: `https://arxiv.org/pdf/1602.04062.pdf`.

[5] *Hexagonal grids. Transforming a hexagonal grid to a quadratic one*. 2015. URL: `http://www.redblobgames.com/grids/hexagons/`.

[6] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: `https://www.tensorflow.org/`.

[7] Sebastian Achim Mueller et al. “Single Photon Extraction for FACT’s SiPMs allows for Novel IACT Event Representation”. In: *Proceedings of the 35th ICRC*. 801. 2017.

[8] Thomas Fabian Temme. *On the hunt for photons: analysis of Crab Nebula data obtained by the first G-APD Cherenkov telescope*. 2016. URL: `https://eldorado.tu-dortmund.de/handle/2003/35745`.

Eidesstattliche Versicherung

Ich versichere hiermit an Eides statt, dass ich die vorliegende Abschlussarbeit mit dem Titel “Gamma-Hadron Separation with Deep Learning for the First G-APD Cherenkov Telescope” selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ort, Datum

Unterschrift

Belehrung

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50 000 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden (§ 63 Abs. 5 Hochschulgesetz –HG–).

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird ggf. elektronische Vergleichswerkzeuge (wie z. B. die Software “turnitin”) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen.

Ort, Datum

Unterschrift