Identification of Direct Cherenkov Pixels using Boosted Decision Trees

Robert Stein

May 17, 2016

Abstract

When Cosmic Rays travel through the atmosphere, the primary particle will often emit some Cherenkov light before the much brighter Extended Air Shower is generated. Unlike the broader shower component of telescope images, all of the Direct Cherenkov light will usually be concentrated in a single pixel. A new method for identifying these Direct Cherenkov pixels was developed, relying on machine learning. A set of Cosmic Ray telescope images were simulated, both with and without the Extended Air Shower background, for use in classifier training. With reference to the background-free telescope images, the Direct Cherenkov pixel in each corresponding full-shower telescope image could be reliably identified. A Boosted Decision Tree to identify these Direct Cherenkov pixels was trained, using data from the full-shower training images. The Boosted Decision Tree performance was then tested on a second set of full-shower telescope images, and compared to existing methods of Direct Cherenkov pixel identification.

1 Introduction

A Cosmic Ray will often emit Direct Cherenkov (DC) light in the upper atmosphere, before generating an Extended Air Shower (EAS) through interaction with the lower atmosphere. Analysis of the DC light from a shower can indicate primary particle energy, charge and position. It is consequently very useful to identify and measure the quantity and direction of DC light emission, for use in event reconstruction. There are numerous Telescope Arrays which currently image Cherenkov Light emission from Cosmic Rays in the atmosphere, including the HESS, MAGIC and VERITAS Experiments. In these telescope images, the DC light is usually concentrated in a single 'DC pixel'. Identifying this pixel is challenging, because the brighter EAS Cherenkov light background often overlaps with the DC pixel.

At present, the DC pixel candidate can be identified by applying a number of cuts to pixels in an image, such as those used by the HESS collaboration [1]. The variable Q_{DC} is defined as the ratio of the largest neighbouring pixel intensity to the intensity of a given pixel, and from the subset of pixels passing these cuts, the DC candidate is simply the pixel with the smallest Q_{DC}. However, this method is neither reliable nor efficient. An better method would aim to increase the number of correctly identified events, and enable cuts which better discriminate between correctly and incorrectly identified events.

Classifiers provides an alternative method of identification, making use of supervised machine learning to find structure in datasets. To train a classifier, we require a set of training data entries containing several variables, as well as the correct class for each of the entries in the dataset. Once trained, a classifier can be used to predict the class for any such data entry. In this case, the classifier must learn to distinguish between non-DC and DC pixels.

The CORSIKA package [3] was used to generate Cosmic Ray events, while the sim_telarray package [2] was used to generate corresponding HESS array telescope images. Simulation with EAS background was used to produce training pixel sets, while corresponding simulation without EAS background was used to determine the true class of each pixel in the training sets. The data was used to train a Boosted Decision Tree (BDT) classifier to identify DC pixels. The data was provided in the form of individual pixel entries, rather than as discrete sets for images or events.

Once trained, the BDT was applied to all pixel entries in a separate 'testing' set of simulated telescope images. For each pixel, the BDT assigned a 'Signal Probability', P_{signal}, indicating the likelihood of the pixel being a DC pixel. For each image, the pixel with the largest P_{signal} was identified as the sole DC pixel candidate. As before, the true class of each pixel was determined from a second EAS-free simulation. Thus, the accuracy of BDT identification for test telescope images was calculated.

2 Image Simulation

The full simulation of air showers was performed using the CORSIKA package atmospheric profile derived from measurements conducted at the HESS site in Namibia. The simulated particles were Fe^{56}, within the Energy Range of $35 - 135$ TeV and a spectrum $\phi \propto E^{-2.7}$. For each set of simulated event, 4 unique random number seeds were used to generate the shower. An altitude of 1800m was assumed, again corresponding to the HESS site. The simulated zenith angle ranged from $0° < \theta < 2°$, while the simulated azimuth angle ranges from $-2° < \phi < 2°$. The four smaller HESS-phase-1 telescopes were arranged in a cross along the x/y axis with the larger HESS-phase-2 'CT5' telescope placed at the center. The length of each cross arm was $85m$. The simulated target region of the cores was chosen to be a square centered on CT5, with each 300m-long side bisecting the x/y axis.

To determine the true class of each pixel, a simulation was initially run with an energy cut of 10 PeV on all muons and electrons. Because this cut exceeded the primary particle energy, neither daughter muons and electrons, nor the photons they would have emitted, were simulated. Thus the hadronic Cherenkov Light from the primary particle and daughter fragments, but not the EAS light, was present in the camera image. A second identical 'EAS Simulation' was run including the same random seeds, but without the energy cut on muons and electrons. This gave a complete EAS image including identical DC light.

With the sim_telarray package, the expected HESS hardware response to each air shower was simulated. Among other things, the program accounts for atmospheric transmission and density, mirror positions, sizes and reflectivities, camera shadowing and triggering, quantum efficiency and pulse responses. For the full-shower image, the night sky background was also simulated by sim_telarray. Due to the comprehensive and detailed nature of these hardware simulations, the resultant images can be considered 'realistic' camera images. However, sim_telarray introduces various sources of random noise to the simulation, leading to some divergence between the EAS-free and full-shower images.

The various pixel entry variables were then found from the sim_telarray output. The HESS telescope pixels have a high gain Channel 0 and a low gain Channel 1, with both voltages undergoing a Flash Analogue-to-Digital Conversion (FADC). The simulated value of the FADC Voltage for each channel was found. Alongside the pedestal and gain, the quantity $Intensity = (FADC - Pedestal) \times Gain$ was calculated. Due to possible saturation of the high gain FADC, only the low gain $Intensity$ was used.

Sim_telarray also derives various Hillas whole-image parameters. These include the image width and length measured in degrees, from which the aspect ratio $A.R = \frac{width}{length}$ was calculated. The reconstructed shower direction and the shower center of gravity were also calculated, as positions in azimuth and zenith. Additionally the estimated energy and distance from each telescope to core, r_{core}, were found.

For every pixel, in addition to the $Intensity$, its location within the telescope image was determined using the standard HESS layout. The variables $\Delta_{C.o.G}$, $\Delta_{Direction}$ and Δ_{Line} were defined as the distance from the pixel to the shower center of gravity, shower direction, and the line joining those two points. Furthermore, the nearest neigbouring pixel IDs were calculated for every pixel position, enabling the $Intensity$ in each neighbouring pixel to be found. The largest neighbouring intensity was identified, and the ratio $Q_{DC} = \frac{Intensity_{N.N.max}}{Intensity}$ was derived. Similarly the largest neighbouring FADC was found, and the ratio $raw_Q = \frac{FADC_{N.N.max}}{FADC}$ was calculated. In addition, the Nearest Neighbour Mean Intensity $Mean_{N.N}$ was recorded. The variable $DC_{Signal} = Intensity - Mean_{N.N}$ was defined as an rough guess of the 'DC signal' component in the pixel.

3 DC Pixel Identification

As a basis for comparison, the original HESS cuts listed in 1 were replicated for a set of test data. For every image, the total image amplitude I_{tot} was used alongside the zenith angle θ to determine a unique Q_{DC} cut, reducing the number of accepted DC candidates. For each image, the smallest Q_{DC} among any remaining pixels was used to identify as the DC pixel candidate. Because many images had no pixel that passed all cuts, the Q_{DC} method was frequently unable to identify a DC pixel. In the original analysis, an additional cut r_{core}>$40m$ was applied. However, the uncertainty in determining the core position through Hillas Analysis is typically of the order of $\pm 30m$. Consequently, this particular cut was omitted, along with the Impact Parameter cuts.

The candidates were checked against the true DC pixels identified in the EAS-free images. Of 2000 events it was found that there were 6000 triggered images. Of these images, 17% passed the required cuts. The Q_{DC} was found to be 68% accurate in identifying the DC pixel in those passing images, as

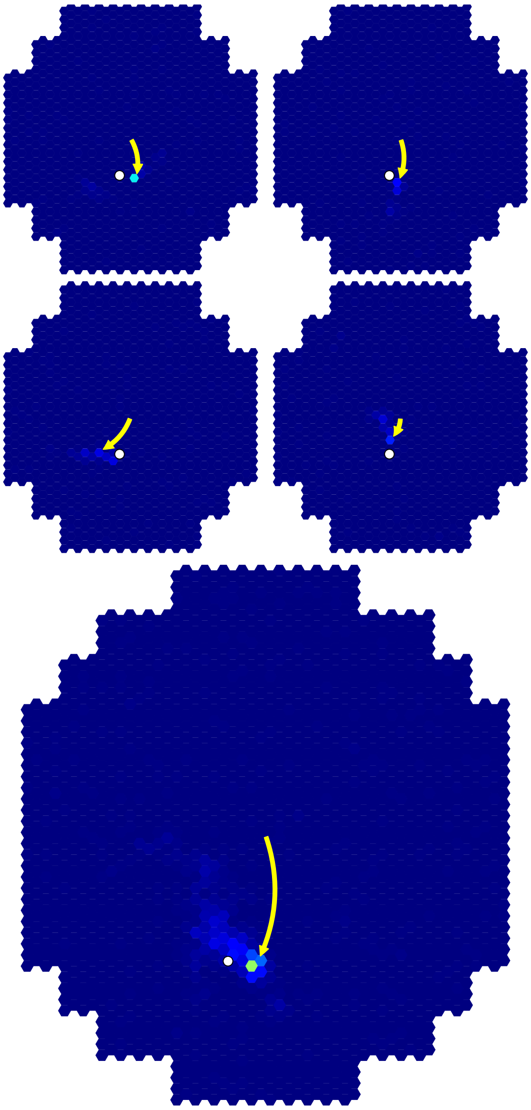

Figure 1: A typical camera image without the EAS shower. The DC light is visible in every telescope, indicated by the yellow arrow. The shower direction is represented by the white circle. The largest telescope is CT5, but the relative image sizes are not done to scale

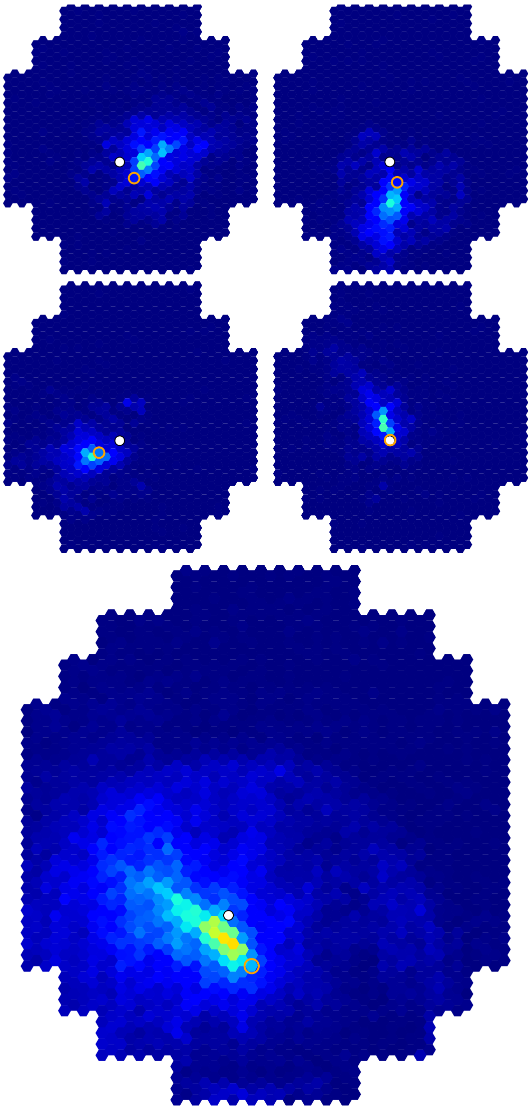

Figure 2: The same shower as in 1 is shown here with the inclusion of the EAS shower. The DC light is pixels are again denoted by an orange circle. The white star represents the BDT-selected DC pixel, while the white cross represents the shower center of gravity.

Table 1: Cuts applied to image pixel sets, used by HESS collaboration [1]

Variable	Cut
$\Delta_{C.o.G}$	>0.17
$\Delta_{C.o.G}$	<0.91
$\Delta_{Direction}$	<0.45
Δ_{Line}	<0.23
Aspect Ratio	<0.75
Q_{DC}	$<0.14\times\log(\frac{I_{tot}}{161\times\cos\theta})$

shown in 3. These values served as a benchmark for BDT performance.

The BDT was trained with the Scikit Learn Python package [4]. A set of 2000 CORSIKA events was selected, and randomly split through use of the Python random.random() function into the two subsets for training and testing. Within the subset of training events, every triggered HESS 1 image was used. For each of the 3.6 million triggered image pixels, an entry was formed of the variables listed in table 2. A score of 0 was assigned to every pixel to indicate background, with the exception of those that were identified as the true DC pixel through the Shower-Free simulation. These pixels were included with a score of 1, identifying it as a signal pixel. Having created a dataset, the BDT was then trained with a maximum depth of 5, and 100 trees generated.

Table 2: Relative Feature Importance in BDT training

Variable	Relative Importance
DC_{Count}	0.38
$Mean_{N.N}$	0.22
$\Delta_{Direction}$	0.15
Q_{DC}	0.11
raw_{Q}	0.06
Δ_{Line}	0.04
$Intensity$	0.04

The relative importance of each ‘feature’ is automatically calculated by the Scikit Learn package, and is also recorded in table 2. The variable DC_{Count} was consistently the most importance variable across many combinations of included variables and BDT training parameters. It was found that, under the conditions listed above, the BDT was 99.94 % accurate for the entire training dataset, and 99.93 % accurate for the testing dataset. This indicates that the BDT was not significantly overtrained, which would otherwise be manifested by a large divergence in accuracy between testing and training data.

Having trained the BDT successfully, it was then applied to the same separate dataset as for the classic QDC identification. In each camera image, the event with the largest BDT score was deemed to be ‘most signal-like’, and thus selected as the DC pixel candidate. A cut was applied, requiring $P_{signal} > 0.5$ for the DC candidate to be accepted. A second cut requiring $DC_{Count} > 200$ removed many incorrectly identified events.

Application of this combined cut greatly increases the successful identification rate. Of the 2000 events, 84% of events passed all of the required cuts. The BDT was found to be 90 % accurate in identifying the DC pixel in those passing events. This represents a very significant improvement in DC pixel identification over the previous QDC method, and corresponds to a fivefold increase in the number of HESS data events that can be studied using the LPD method.

HESS2!

4 Conclusion

The use of BDT identification has been shown to be superior to the traditional Q_{DC} method, by providing five times as many correctly identified events, once cuts have been applied. Furthermore, the resultant sample has only half the contamination of the Q_{DC} method, with 15% of the sample being incorrectly identified events rather than 30% for Q_{DC}.

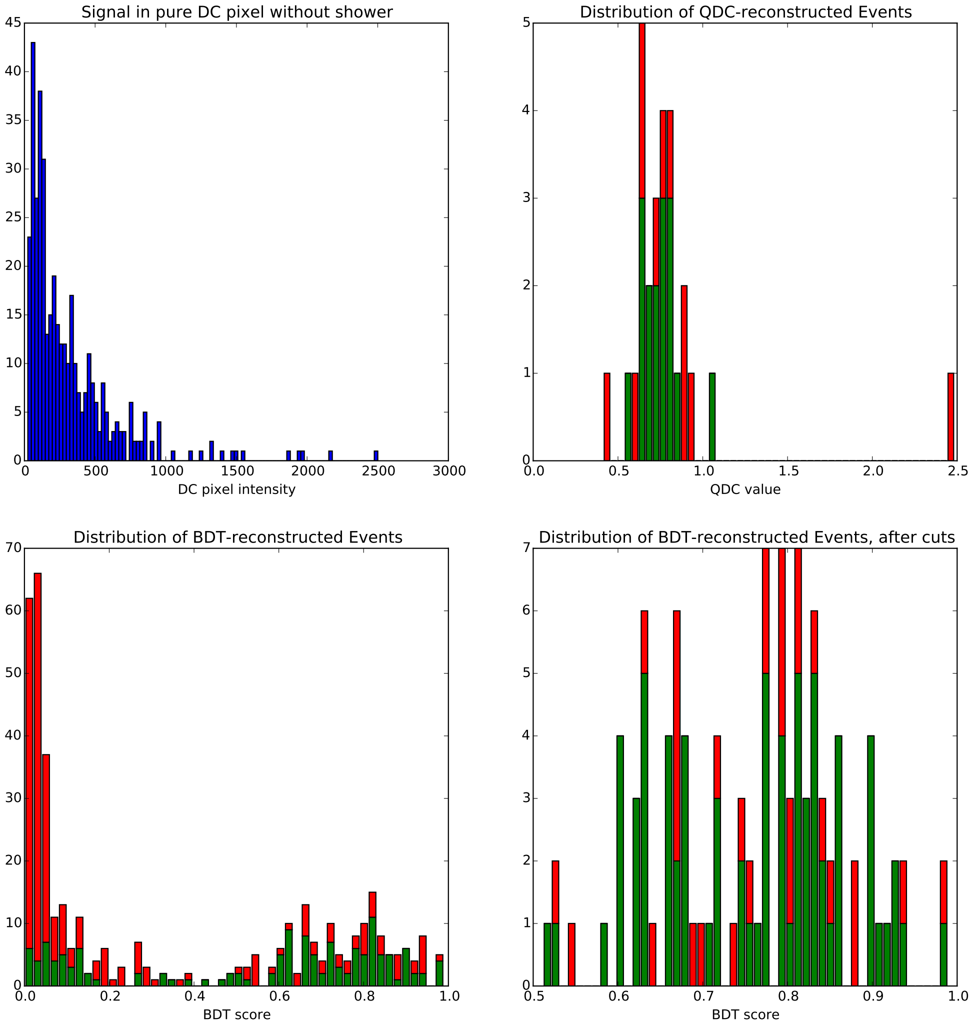

Figure 3: The true $Intensity_{DC}$ in the EAS-free image is shown in the top left, with a broad exponential decay in count as DC intensity increases. Sim_telarray requires a minimum of 20 photoelectrons in order for a telescope to be triggered, leading to a discontinuity at low intensities in the graph. In the top right- the distribution of the dataset is shown, once all Q_{DC} cuts have been applied. In the bottom left, the P_{signal} (BDT score) distribution is shown before any cuts. On the bottom right, we see the same distribution after both DC_{Count} and P_{signal} cuts are applied. All green events are ones in which the DC pixel has been correctly identified, while red events are ones that have been incorrectly identified. We desire both a large number of green events, and a good degree separability of red and green events.

References

[1] F. Aharonian et al. First ground based measurement of atmospheric Cherenkov light from cosmic rays. *Phys. Rev.*, D75:042004, 2007.

[2] Konrad Bernlohr. Simulation of Imaging Atmospheric Cherenkov Telescopes with CORSIKA and sim_telarray. *Astropart. Phys.*, 30:149–158, 2008.

[3] D. Heck, G. Schatz, T. Thouw, J. Knapp, and J. N. Capdevielle. CORSIKA: A Monte Carlo code to simulate extensive air showers. 1998.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.