Convolutional Neural Network (CNN) for gamma-ray selection in TAIGA-IACT (standalone monoscopic mode)

Evgeny Postnikov, Alexander Kryukov, Stanislav Polyakov, Dmitry Shipilov, Dmitriy Zhurov

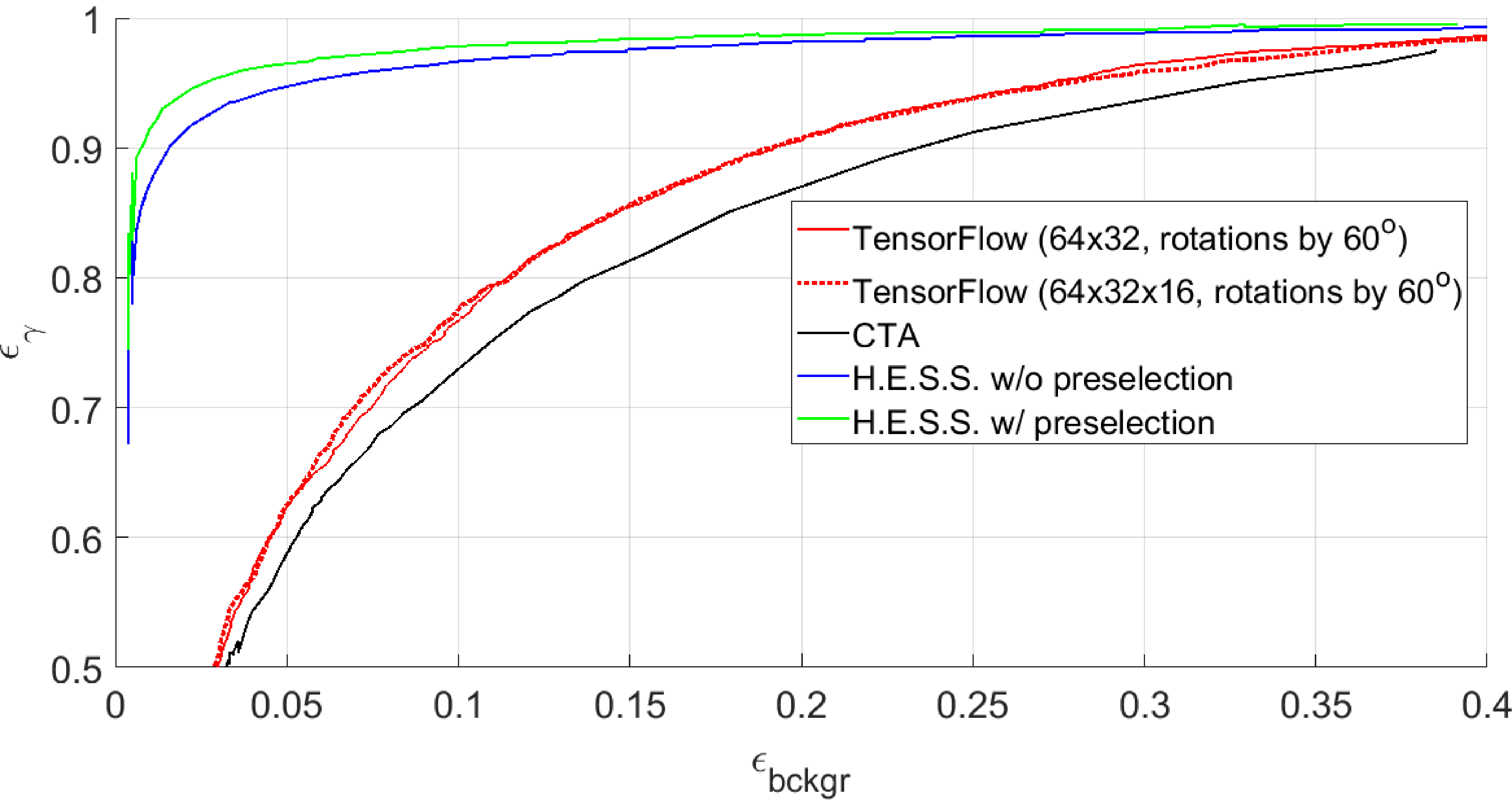
Why using CNN

- It’s a kind of ANN that uses a special architecture which is particularly well-adapted to classify images.
- Today, deep CNN or some close variant are used in most neural networks for image recognition.
- Feature extraction is automatic instead of manual choice (Hillas parameters).

CNN for IACTs [VERITAS, H.E.S.S., CTA]

IACT name	Sample size, $\times10^3$	Camera pixels		N of IACTs	E,TeV of MC	Pre-selection	Soft	Tasks						
		shape	N					selection			estimation			
								Part. name	Q	Q_{ref}				
VE-RI-TAS		Hex	499	1?	No MC		TF /Keras	μ			—	—	—	Image size, image radius
H.E.S.S	2000 600	Hex	960	4	0.02-100	Yes No	TF	γ	15 14	>5.5	—	Θ,φ	—	—
CTA SCT	200	Sq	>11000	1	1-10	Yes	Theano /Keras	γ	2.75	9	—	—	—	—
CTA LST	90.5	Hex	1855	100?	0.003-330	?	TF,PT	—			E	Θ,φ	EAS core	—

CNN for IACTs [VERITAS, H.E.S.S., CTA]



CTA impressions

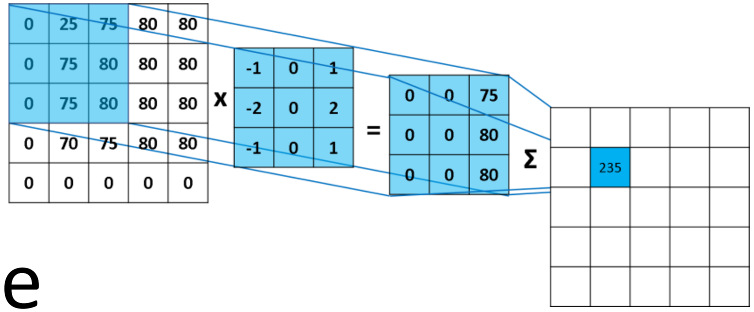
"CNN are capable of classifying simulated IACT images without any prior parametrization nor any assumption on the nature of the images themselves". [PoS(ICRC2017)809]

H.E.S.S. warning

- “During this study we have learned that CNNs trained on simulated events exhibit different performance when tested on a MC test-set and when analysing real-data.
- A network with the CRNN architecture that is trained to distinguish **between simulated proton images and real-data** images becomes astonishingly efficient at performing this task, with an accuracy of 99.5%.
- When testing the same classifier on a set comprised of **MC γ's** (which were not shown to the network during training) and **MC protons**, it assigns 99.6%.
- This illustrates the *risk of using simulations for training*, as DL methods for computer vision are able to easily find **features that do not exist in real-data** images”.
[arXiv 1803.10698]

How CNN works

- Convolutional layers apply a convolution operation (cross-correlation, or simply filtering) to the input, passing the result to the next layer, and so on.



- Special features of feedback avoid overfitting that was the problem for conventional ANN.

How CNN is implemented

Free libraries:

TensorFlow

PYTORCH

First effort – MC data ‘as is’

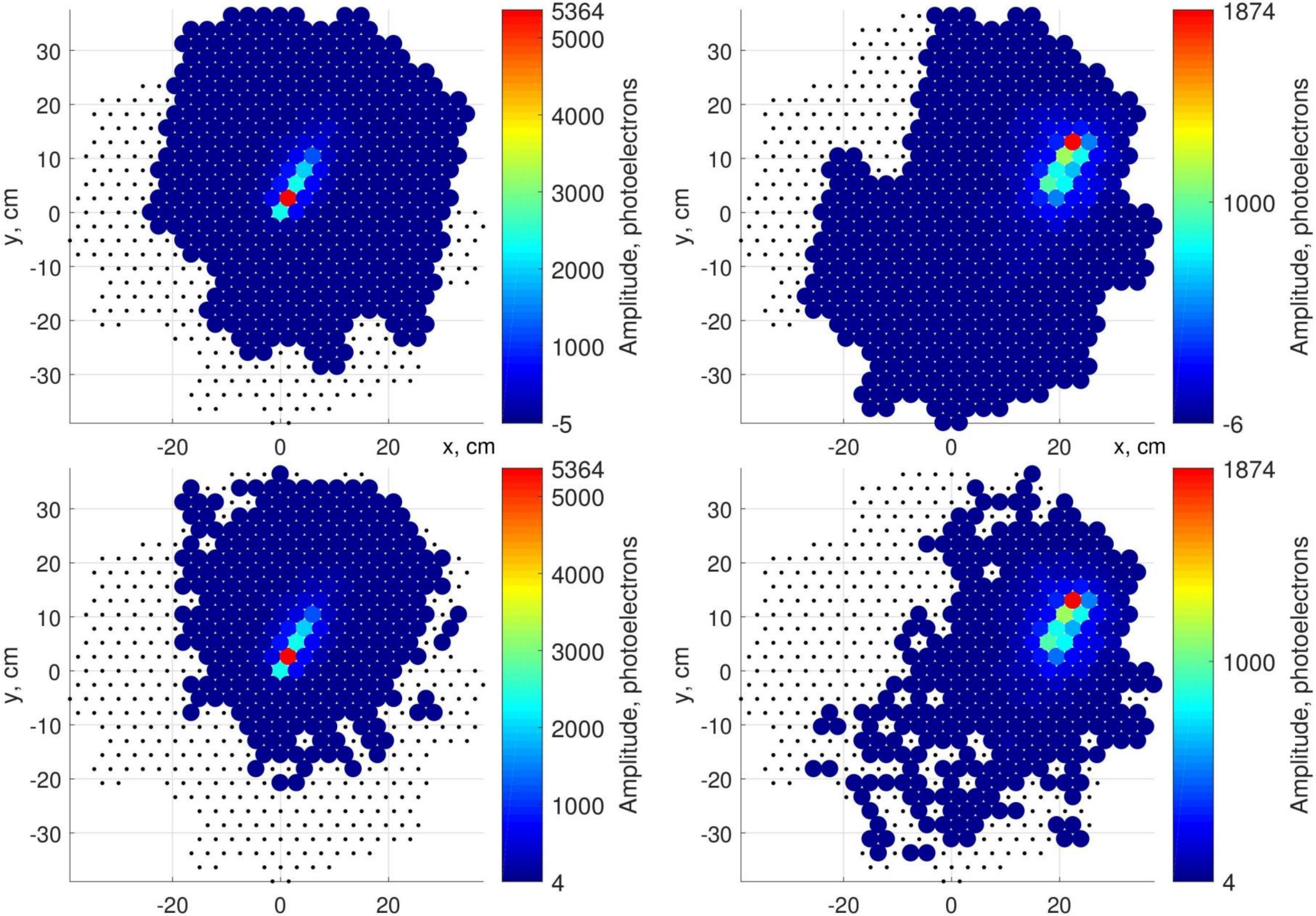
- Trying gamma-ray separation from proton background using Monte Carlo images without ‘image cleaning’ at all.
- For that purpose special Monte Carlo samples were prepared and given for analysis to both CNN packages (PyTorch, TensorFlow) as well as for a simple Hillas analysis using only two basic cuts.

Monte Carlo and blind analysis

- Training datasets: gamma-ray and proton images (Monte Carlo of TAIGA-IACT, 2-60 and 3-100 TeV respectively, exponent -2.6); NSB, trigger procedure and detector response added, but neither cleaning nor preselection applied.

- Test datasets: after CNN training, datasets (different from training ones) of gamma-ray and proton images in random proportion (blind analysis) were classified by each of the packages: TensorFlow and PyTorch. Each package output was ‘probability’ of any image to be gamma-ray of proton.

Simulated image example: γ (left), p (right); no cleaning (top), cleaned (bottom)



Particle identification quality

Quality factor

$$Q = \frac{\text{Significance of a } \gamma\text{-source after } \gamma \text{ separation}}{\text{Significance before separation}}$$

For Poisson distribution of hadron fluctuations:

$$Q = \frac{N_{\gamma\to\gamma}/N_{\gamma}}{\sqrt{N_{hadron\to\gamma}/N_{hadron}}} = \epsilon_{\gamma}/\sqrt{\epsilon_{bckgr}},$$

ϵ_{γ} is γ efficiency

	Simple 2-D technique (image width & orientation)	PyTorch	TensorFlow
W/o image cleaning/preselection:	Q=1.76	Q=1.74	Q=1.48
W/soft cleaning/preselection:	Q=1.70	Q=2.55	Q=2.99

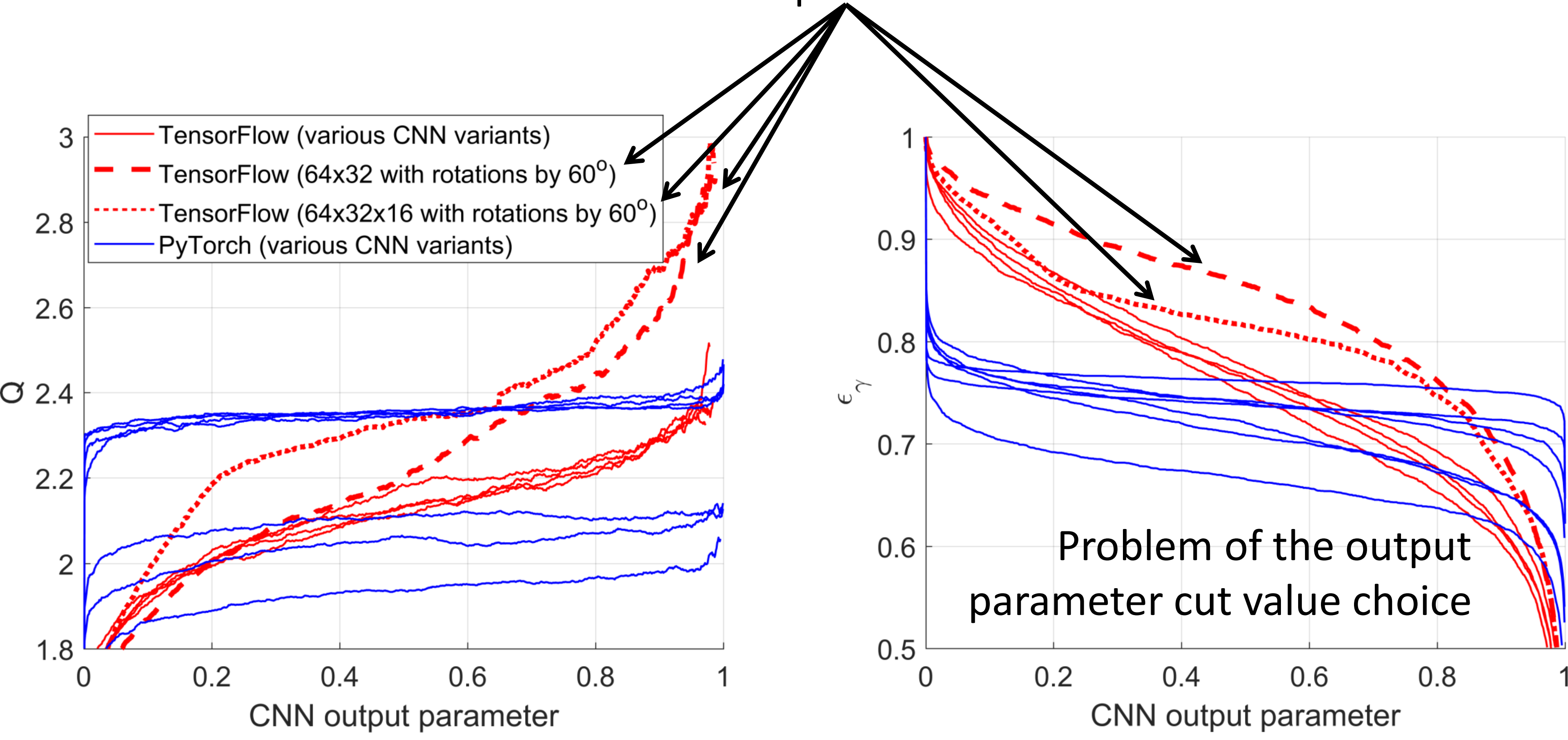
Particle identification quality

- Idea of deep learning application in our project (astroparticle.online, not TAIGA):
  no empirical cleaning or preselections at all
  => Q (and ROC curve) without preselection.
- To compare with other projects, the Q should be recalculated on a dataset subsample after preselection. E.g., with 8cm≤Rc≤25cm, size≥60p.e., npix≥6:
  - Q(TensorFlow)=4.10 (Q(Hillas)=2.76)

  And same but with the size≥100p.e.:
  - Q(TensorFlow)=5.43 (Q(Hillas)=3.14)

Q factor (left) and γ efficiency (right) vs CNN output parameter (*various CNN after cleaning*)

After additional rotations of learning sample by 60°, so that a sample size arouse from ~30 000 to ~180 000



Preliminary conclusions

- The standard image cleaning procedure even in a very soft variant led to significant improvement of the Q-factor.
- Another yet improvement in quality of identification is due to the additional image rotation in learning sample, which allows increasing sample size.
- To get higher Q, problem of choosing CNN output parameter value should be solved: the value should be taken as much as possible (almost 1), but to avoid losing more than 50% of gamma.
- Hexagonal pixel shape should be taken into account (H.E.S.S. recommendation is whether resampling the images to a square grid or applying modified convolution kernels that conserve the hexagonal grid properties).
- Verification using experiment data is required.
- Regression task (energy etc.) study is also required.
- Of course, larger sample size is also necessary.

Backup slides

IACT applications [VERITAS, H.E.S.S., CTA]

- VERITAS: selection of muon images, PoS(ICRC2017)826.

- H.E.S.S.: selection of gamma-ray events, stereo IACTs, 960 hexagonal pixels, arXiv 1803.10698.

  2*acosd(1-0.6/(2*pi))

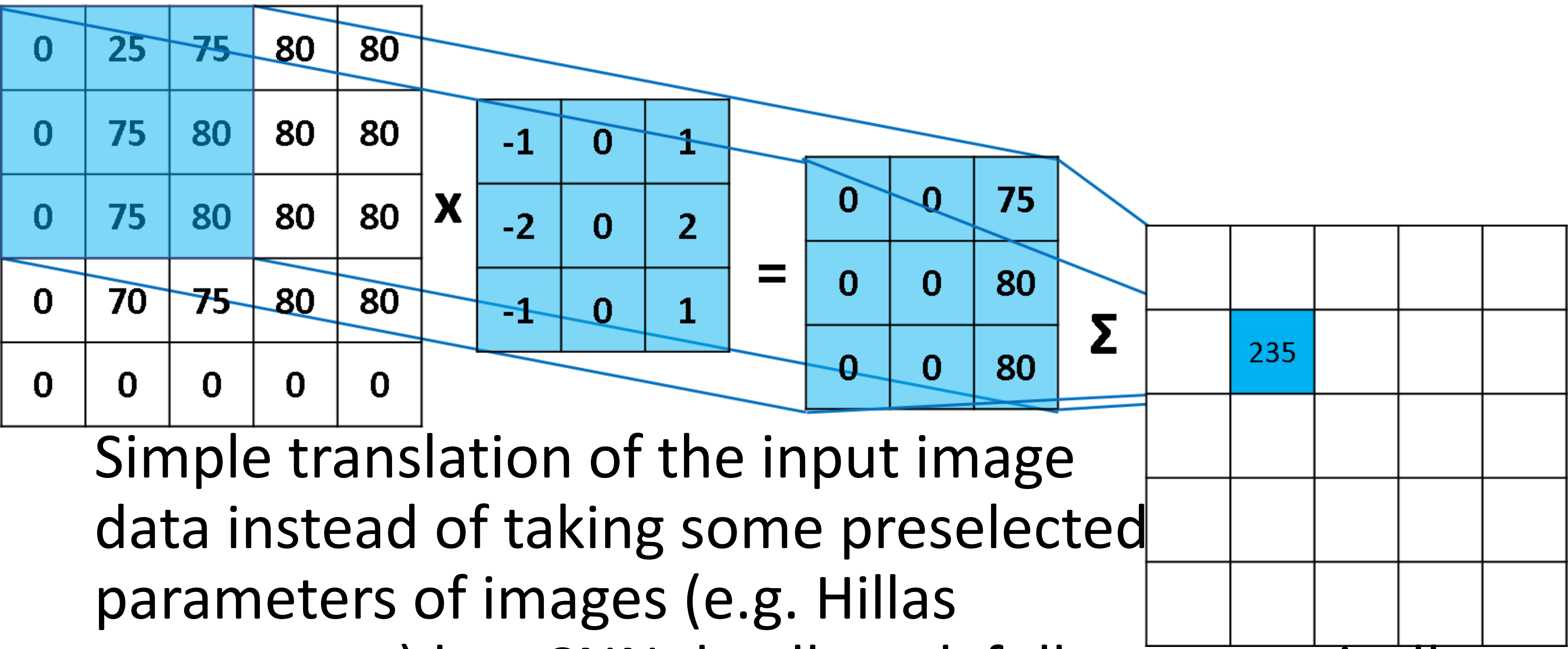
- CTA:
  - selection of gamma-ray events, standalone IACT, >11000 square pixels, PoS(ICRC2017)809.
  - energy, direction and impact point

CNN for IACTs [VERITAS, H.E.S.S., CTA]

	Camera pixels		Tasks			
	shape	number	selection	estimation		
VERITAS	hexagonal	499	muon			
H.E.S.S.	hexagonal	960	γ		direction	
CTA (SCTs)	square	>11000	γ	—	—	—
CTA (LSTs)	hexagonal	1855	—	energy	direction	EAS core

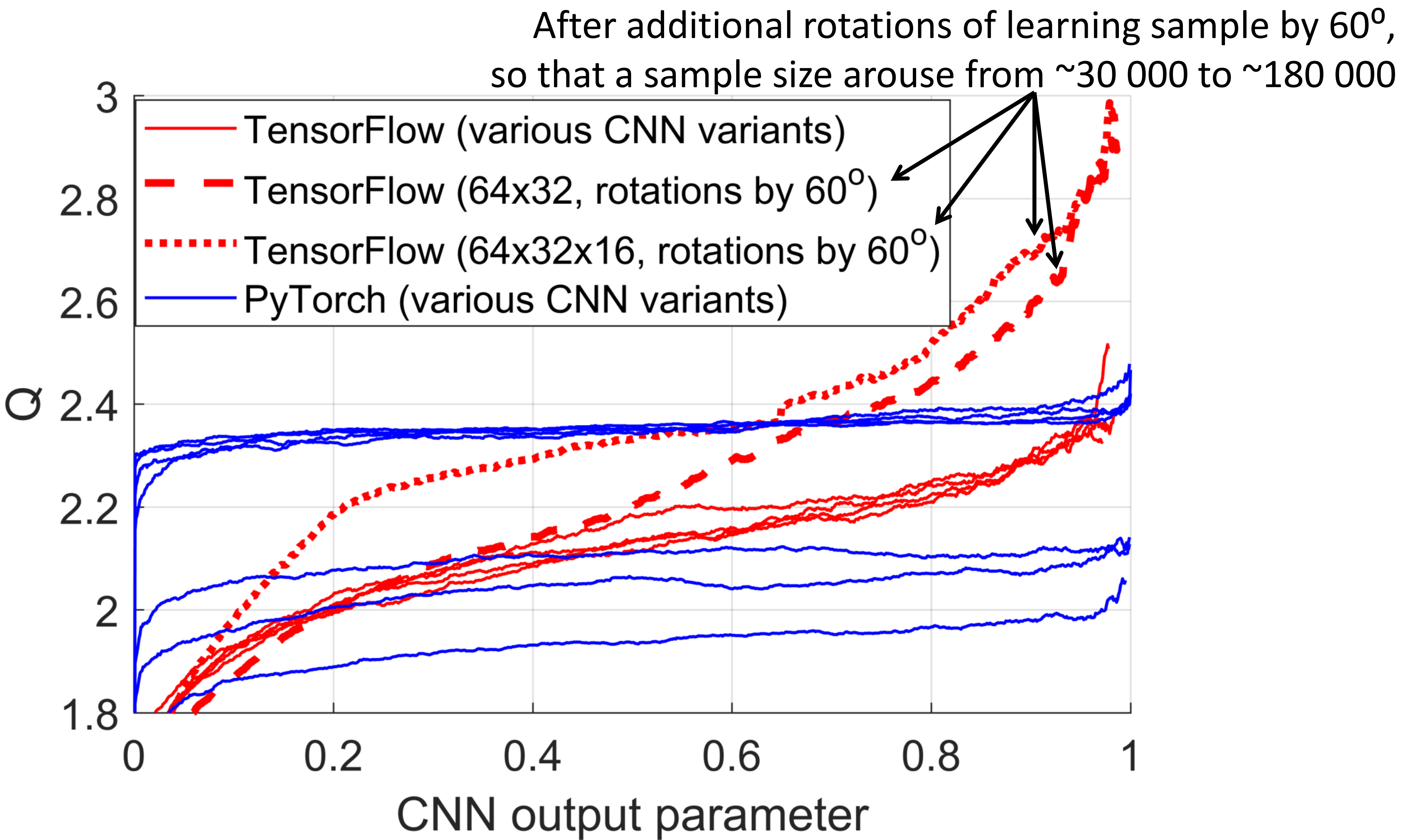
How CNN works

The idea of CNN is to behave in an invariant way across images.



Simple translation of the input image data instead of taking some preselected parameters of images (e.g. Hillas parameters) lets CNN do all work fully automatically.

Q vs CNN output parameter
(*various CNN after same soft cleaning*)



Number of correctly identified γ-rays vs CNN output parameter
(*Problem of the ‘cut value’ choice*)