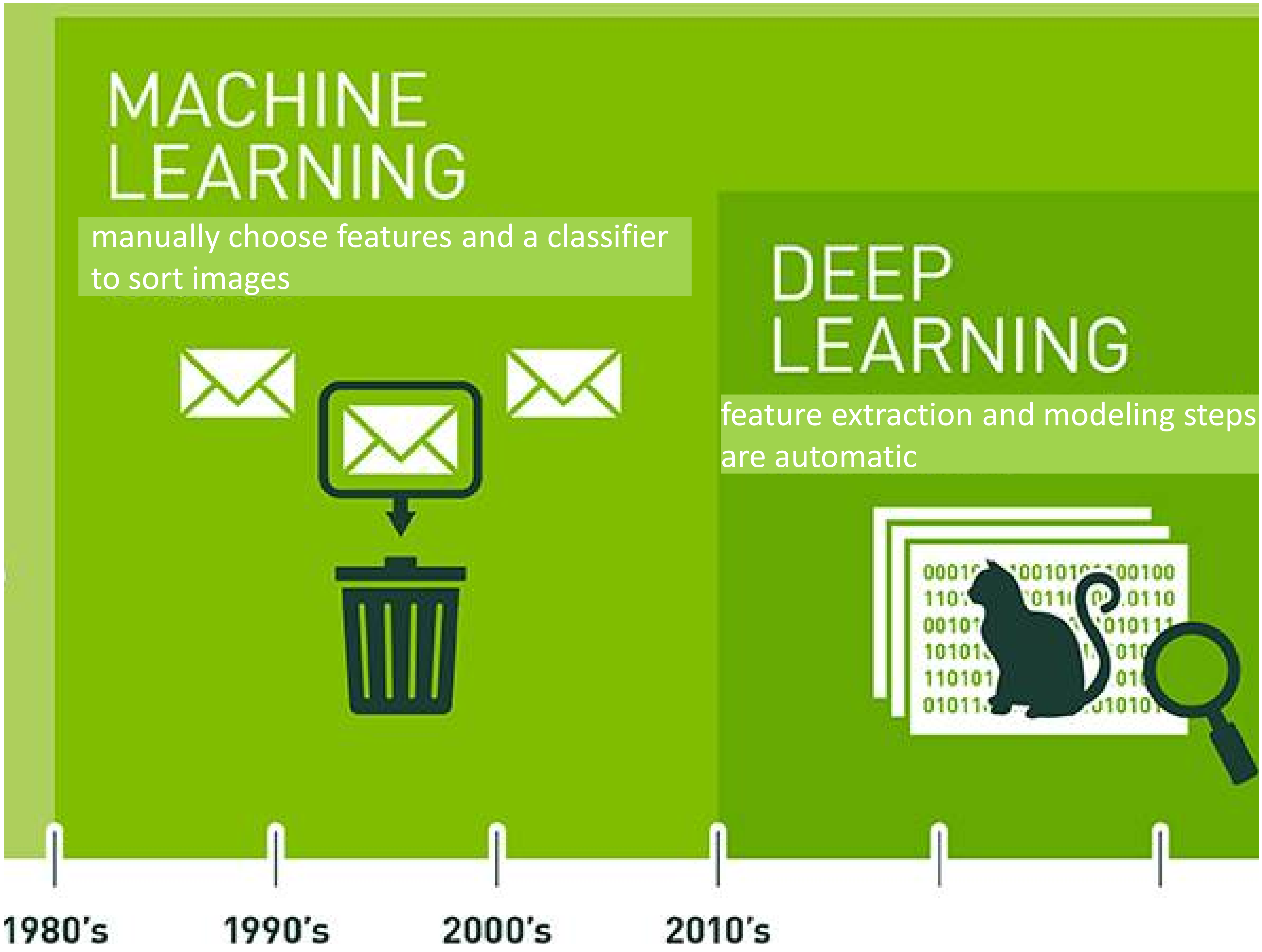
Deep learning for energy estimation and particle identification in gamma-ray astronomy

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


MACHINE LEARNING
manually choose features and a classifier to sort images
DEEP LEARNING
feature extraction and modeling steps are automatic
1980's
1990's
2000's
2010's

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 background in other gamma-ray astronomy projects

- VERITAS
- H.E.S.S.
- CTA
  - Schwarzschild-Couder Telescopes (medium telescopes)
  - Large-Sized Telescopes

- Deep learning techniques (CNN) have previously been developed to select gamma-ray events in the TAIGA experiment

A good quality of selection was achieved as compared with the conventional approach

TAIGA telescope

- TAIGA is an array of telescopes (currently only the 1st one installed) designed to detect the very high energy gamma-rays ($>1\times10^{12}$ eV) through their interaction with the Earth's atmosphere. The gamma-rays produce a shower of particles that travel through the atmosphere, emitting Cherenkov light which is then detected by our telescope (8.5 m^2 mirror area) and projected onto the photomultiplier-based camera (560 photomultipliers ~ 560 pixels of the image).

TAIGA telescope

Purpose of image analysis

- Particle identification:
  - gamma ray

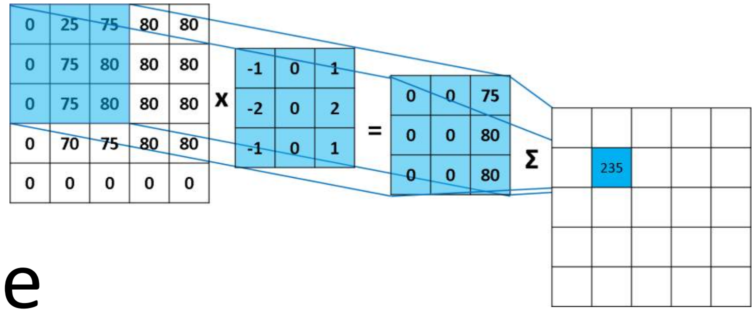
  VS
  - charged cosmic ray, mostly proton.

The idea behind CNN

- 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. dimensions and orientation) lets CNN do all work fully automatically ("capable of classifying IACT images without any prior parametrization", CTA).

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

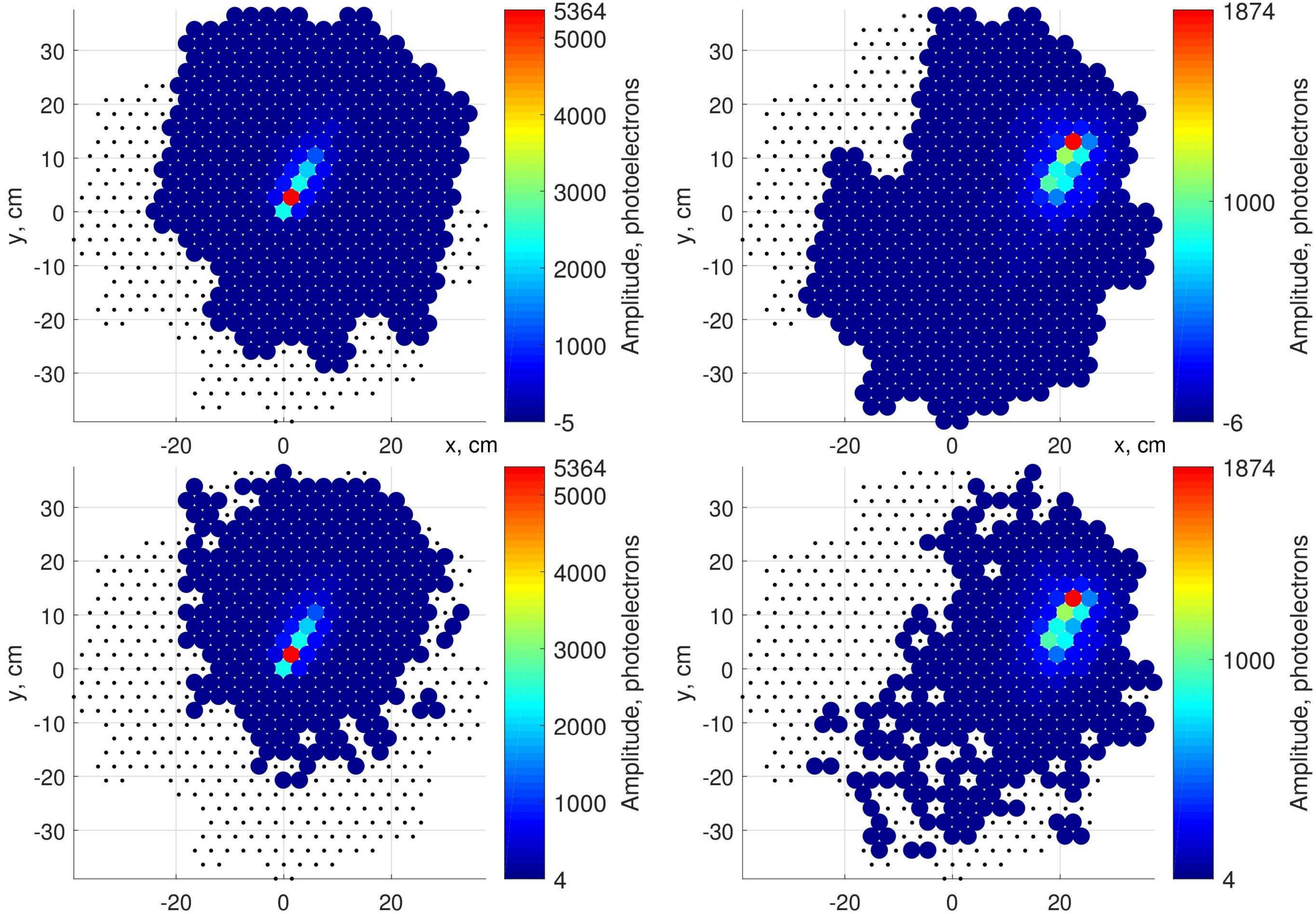
Free libraries: TensorFlow PYTORCH

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); night sky background, trigger procedure and detector response added; no image cleaning (or very soft cleaning) applied; no preselection.

- 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:

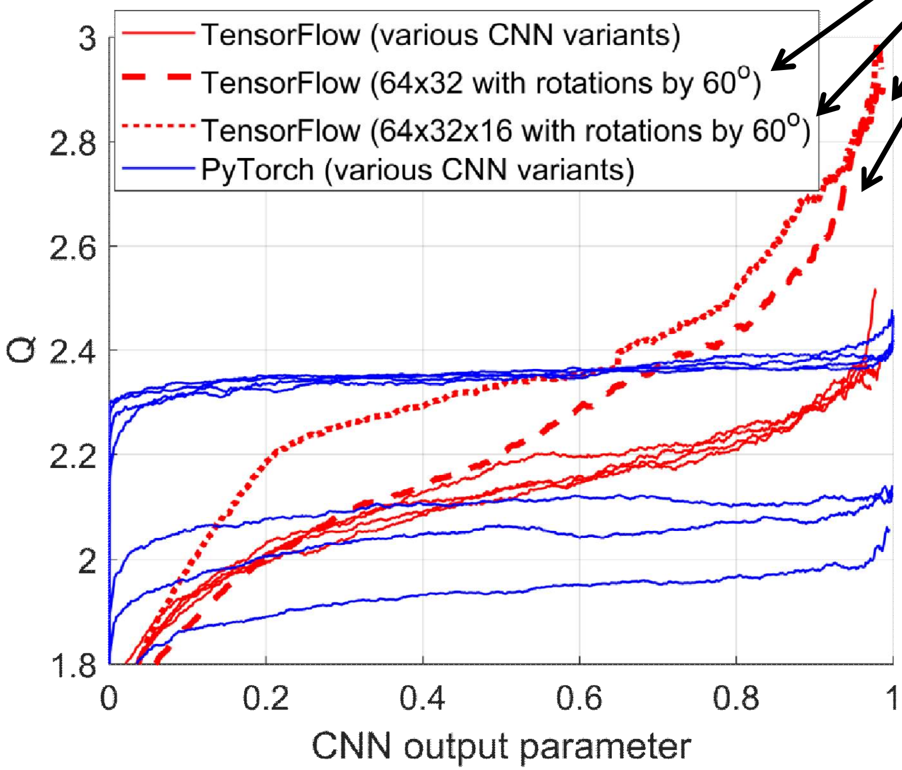
$$Q = \frac{N_{\gamma\to\gamma}/N_\gamma}{\sqrt{N_{hadron\to\gamma}/N_{hadron}}} = \epsilon_\gamma/\sqrt{\epsilon_{bckgr}},$$

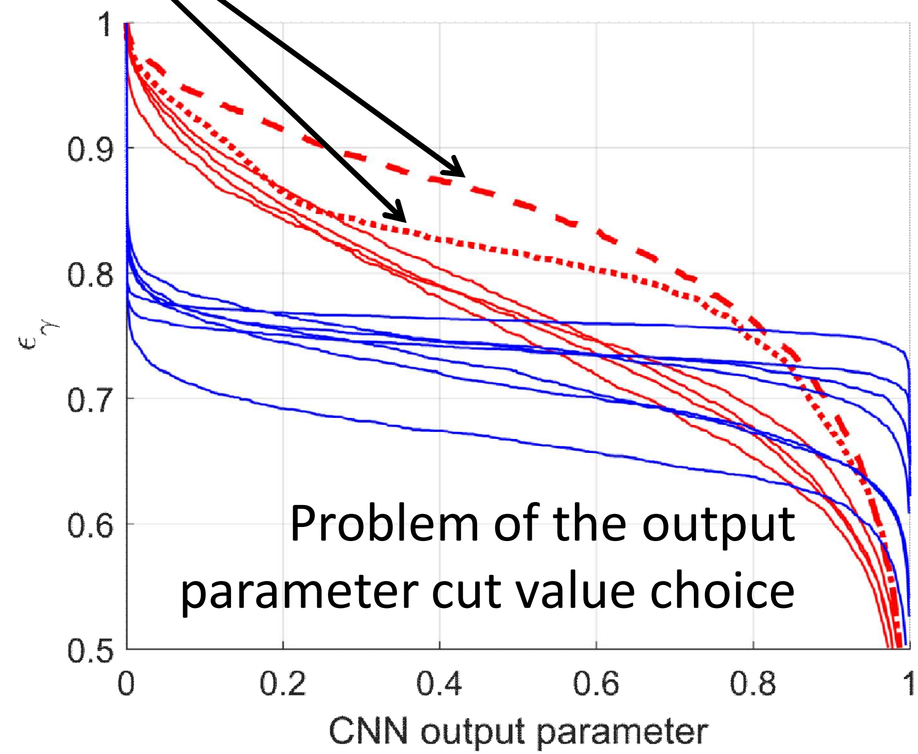
ϵ_γ is γ efficiency

	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

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

After additional rotations of learning sample by 60°, leading to a sample size increase from ~30 000 to ~180 000

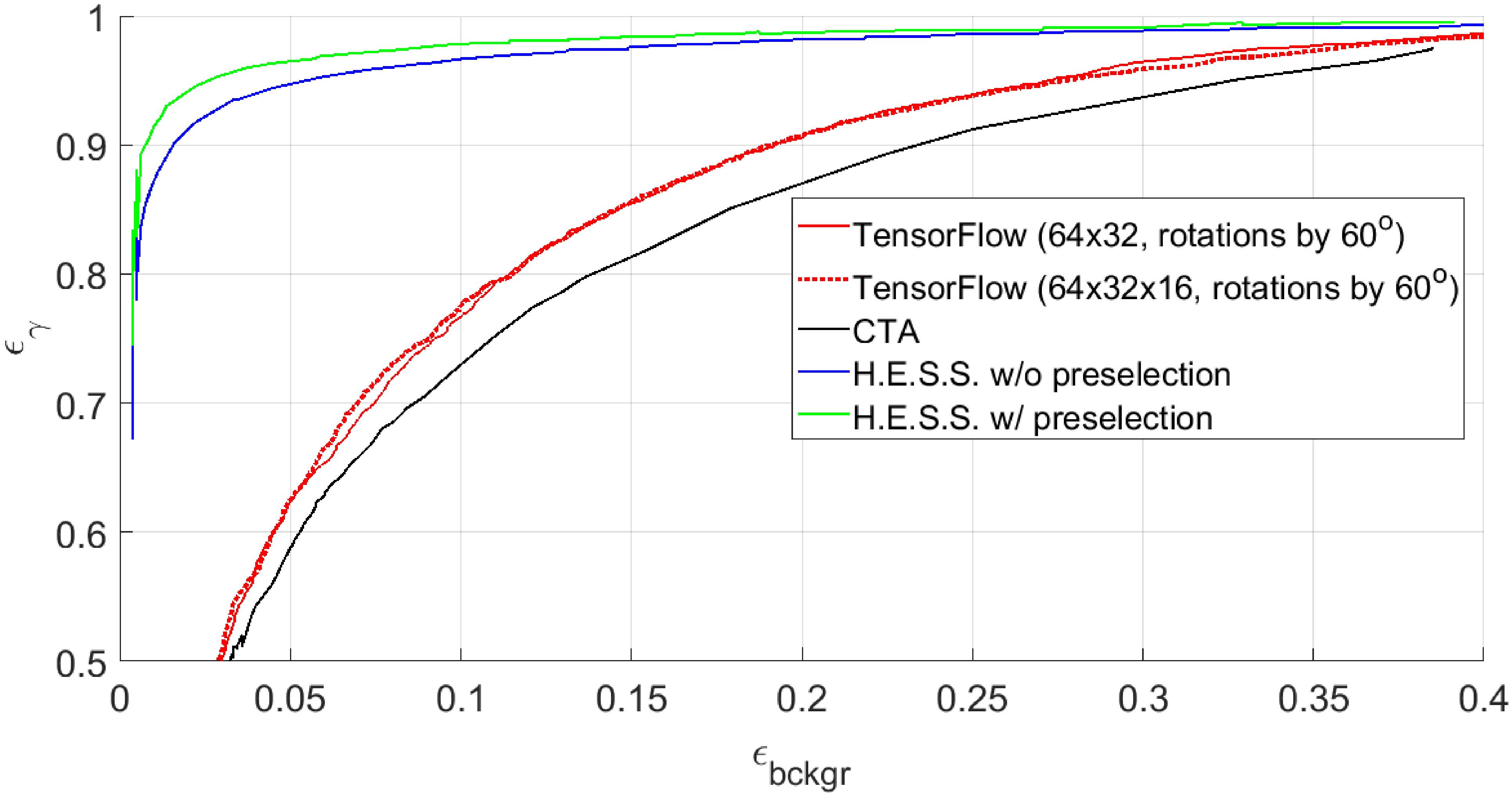




CNN for IACTs [VERITAS, H.E.S.S., CTA]

IACT name	Sample size, x10^3	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	—

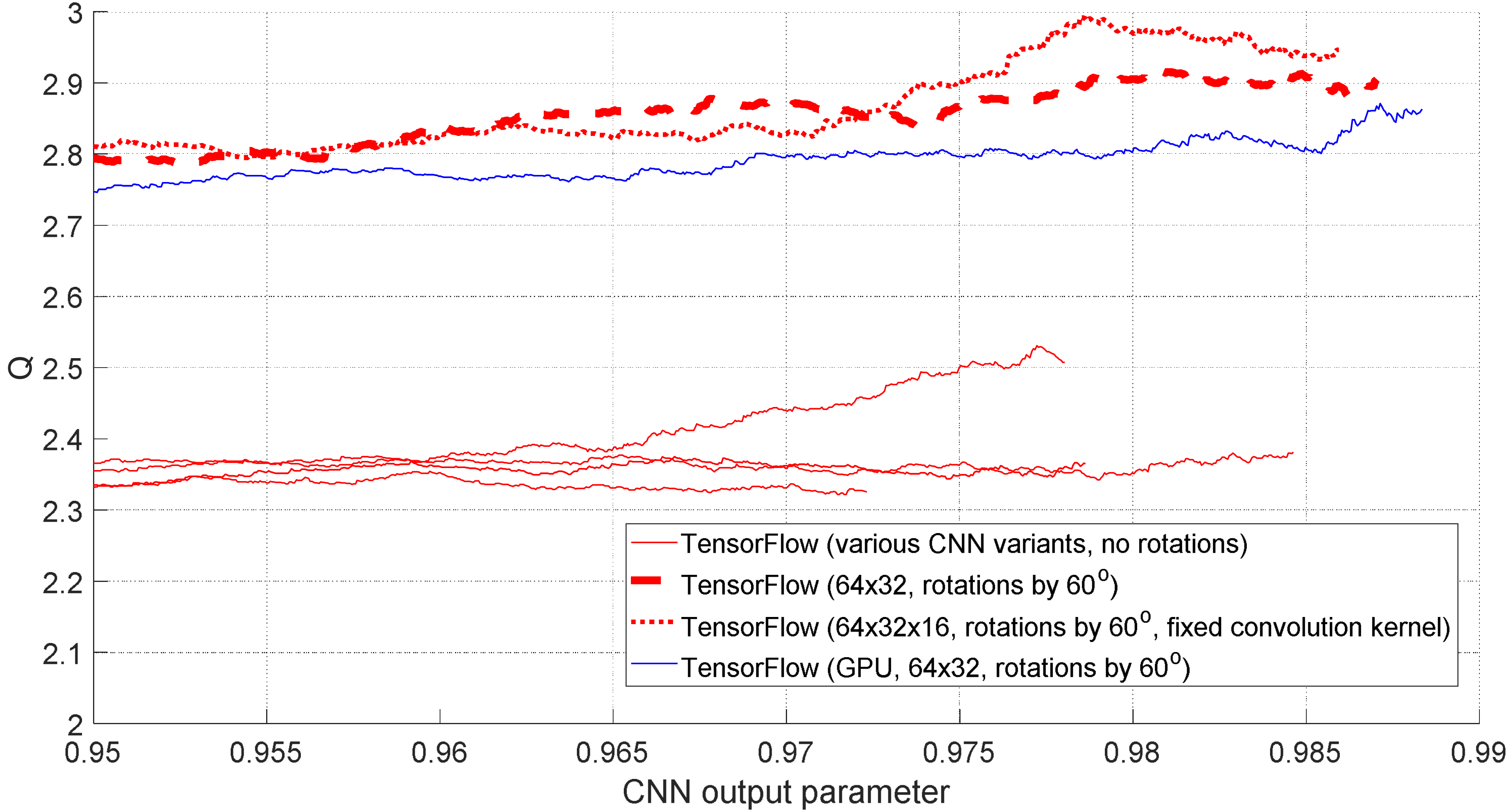
CNN for IACTs [H.E.S.S., CTA]



- Now, the same investigation was repeated using the graphics processing unit (GPU)

A significantly faster calculation
(for the TensorFlow package, 6 times faster)

Q factor vs CNN output parameter

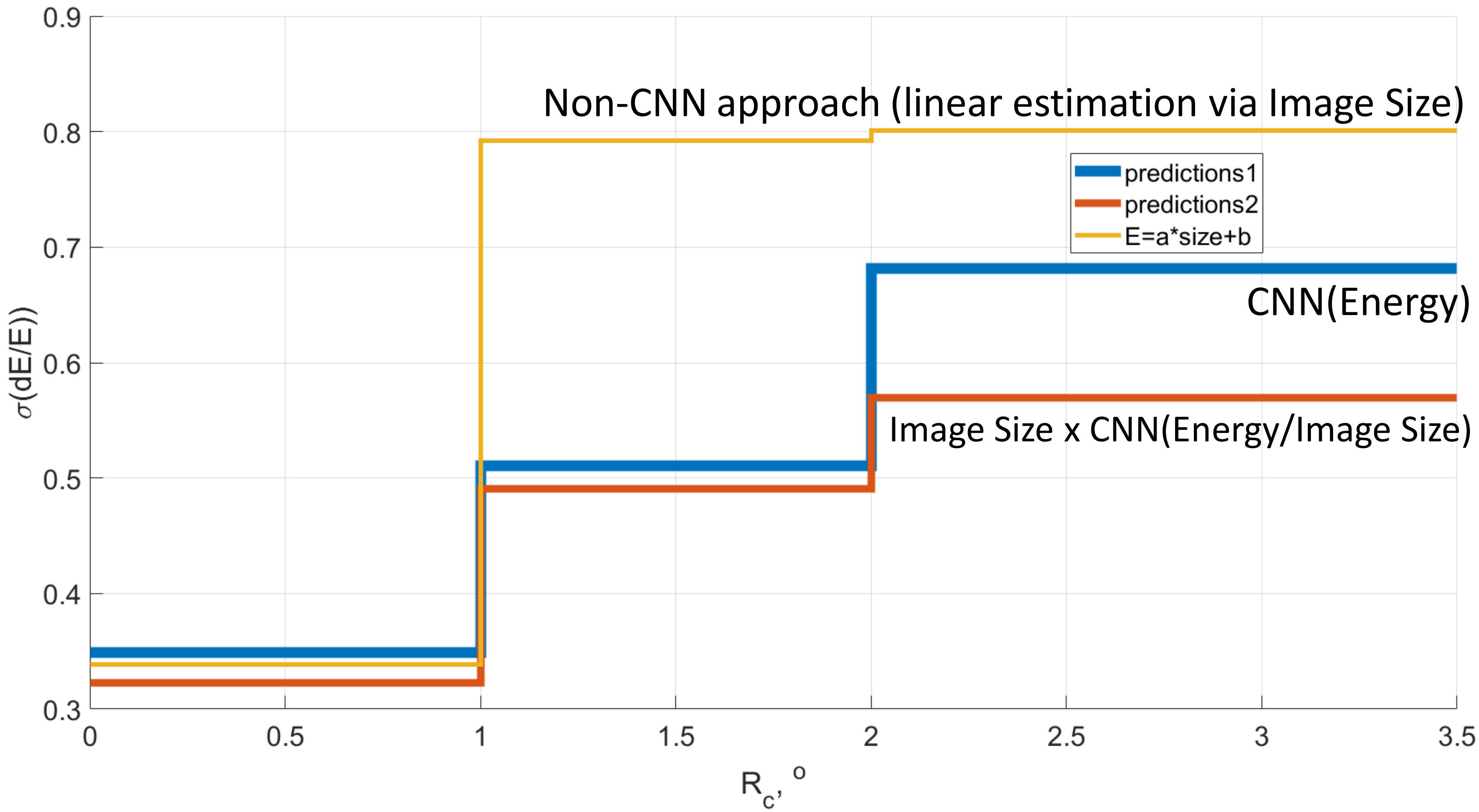


- Another new task of data analysis:
  gamma-ray candidates energy estimation

  Some improvement found in comparison
  with the conventional method

- Conventional method of energy prediction is based on linear correlation between the energy and the image size, which works only for gamma-rays incident very close to the telescope (up to 100-150 m on the ground, or equivalently up to ~1° on the camera)

Gamma-ray energy accuracy vs distance to the image



Preliminary conclusions

1. Deep learning has previously been developed to select gamma-ray events in the TAIGA experiment, having achieved a good quality of selection as compared with the conventional approach.
2. DL gamma-ray selection quality was also confirmed using the (GPU), which led to a significantly faster calculation.
3. Deep learning was developed to estimate gamma-ray energy in TAIGA, it has shown some improvement in comparison with the conventional method.
4. There is still strong potential to further improve the results: taking into account the hexagonal pixel shape, and increasing Monte Carlo sample size.
5. Verification using experiment data is then required.

Backup slides

Preliminary conclusions

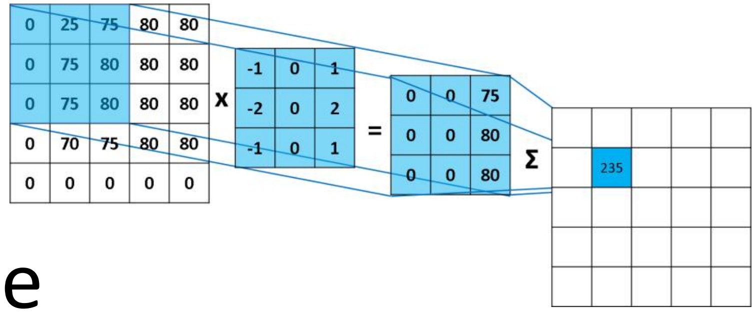
- The standard image cleaning procedure even in a very soft variant led to significant improvement of the Q-factor.
- Another yet improvement in quality of identification is due to the additional image rotation in learning sample, which allows increasing sample size.
- To get higher Q, problem of choosing 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.

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.

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’

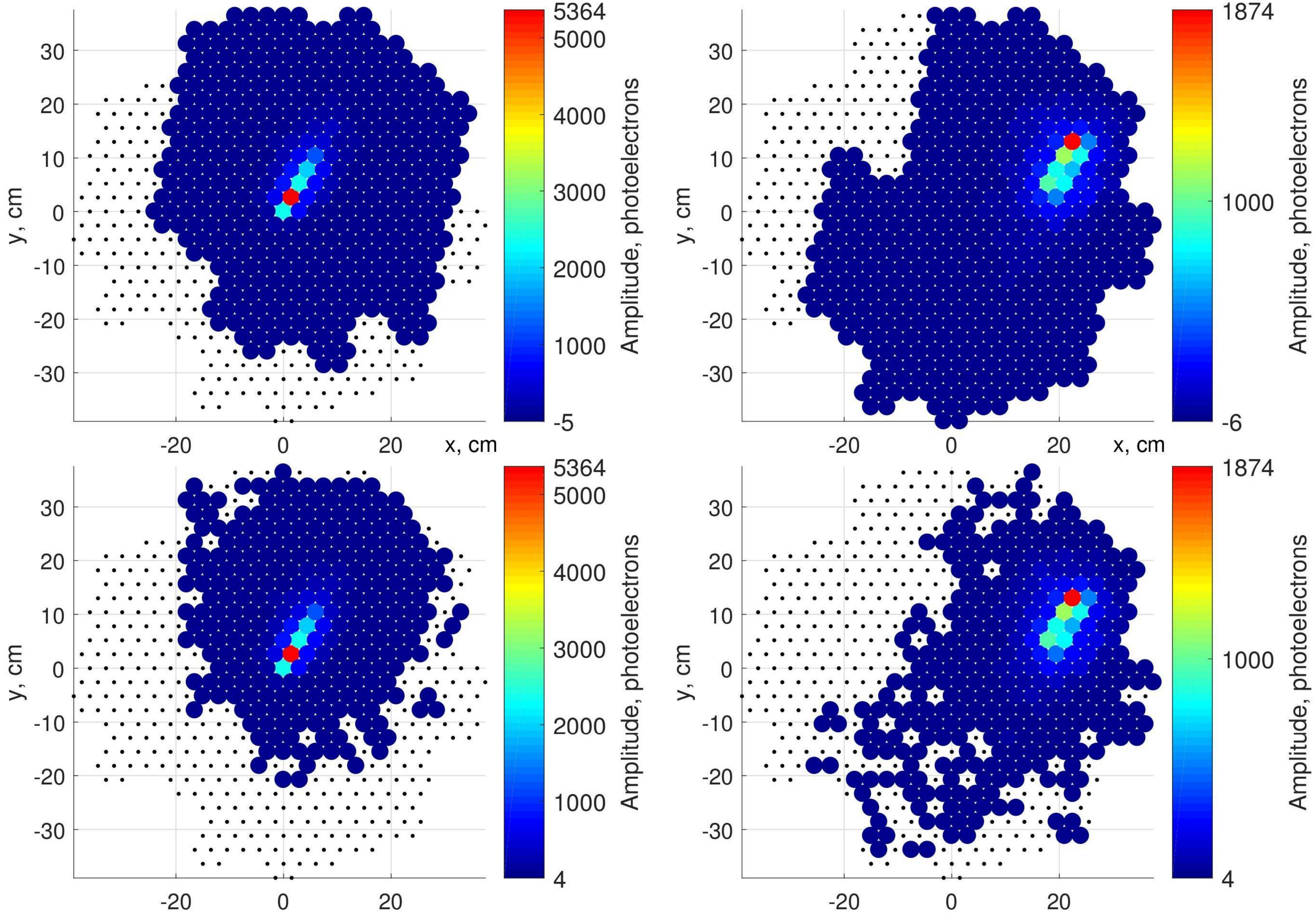
- Trying gamma-ray separation from proton background using Monte Carlo images without ‘image cleaning’ at all.
- For that purpose special Monte Carlo samples were prepared and given for analysis to both CNN packages (PyTorch, TensorFlow) as well as for a simple Hillas analysis using only two basic cuts.

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

- 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)

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.

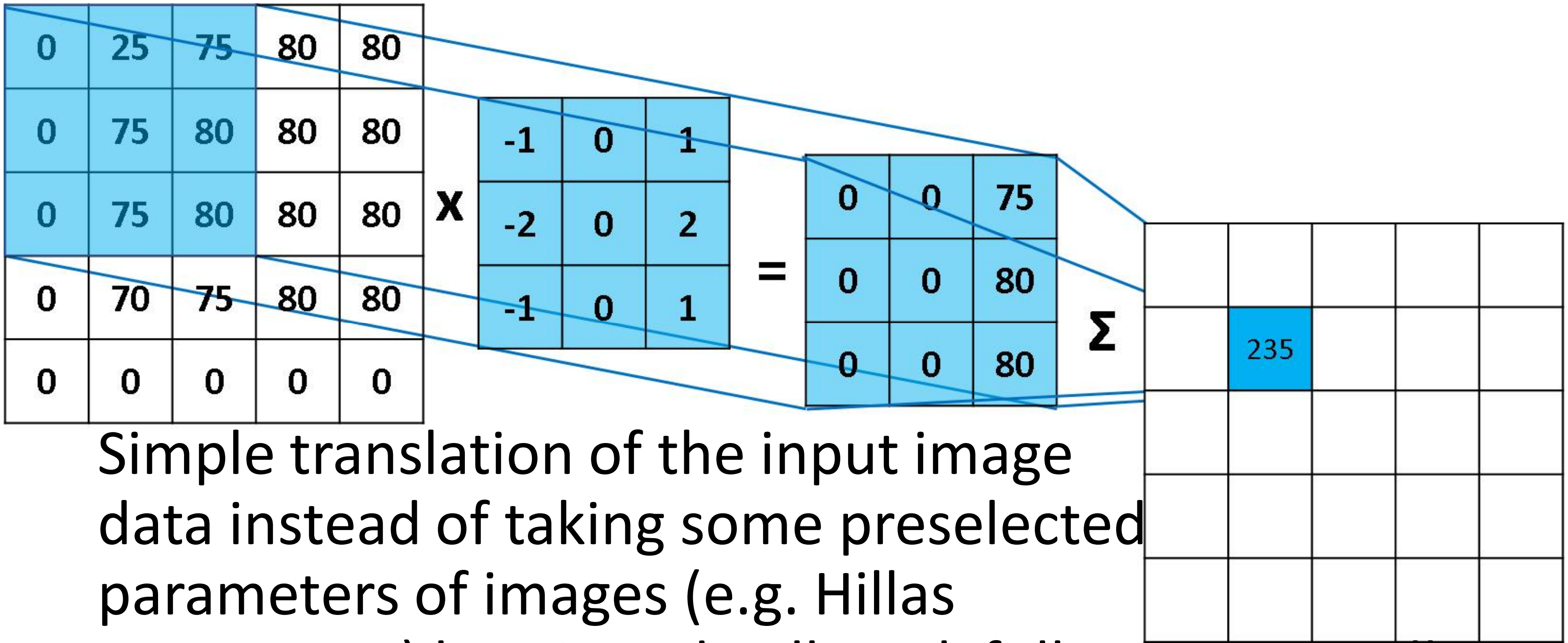
- CTA:
  - selection of gamma-ray events, standalone IACT, >11000 square pixels, PoS(ICRC2017)809.
  - energy, direction and impact point

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

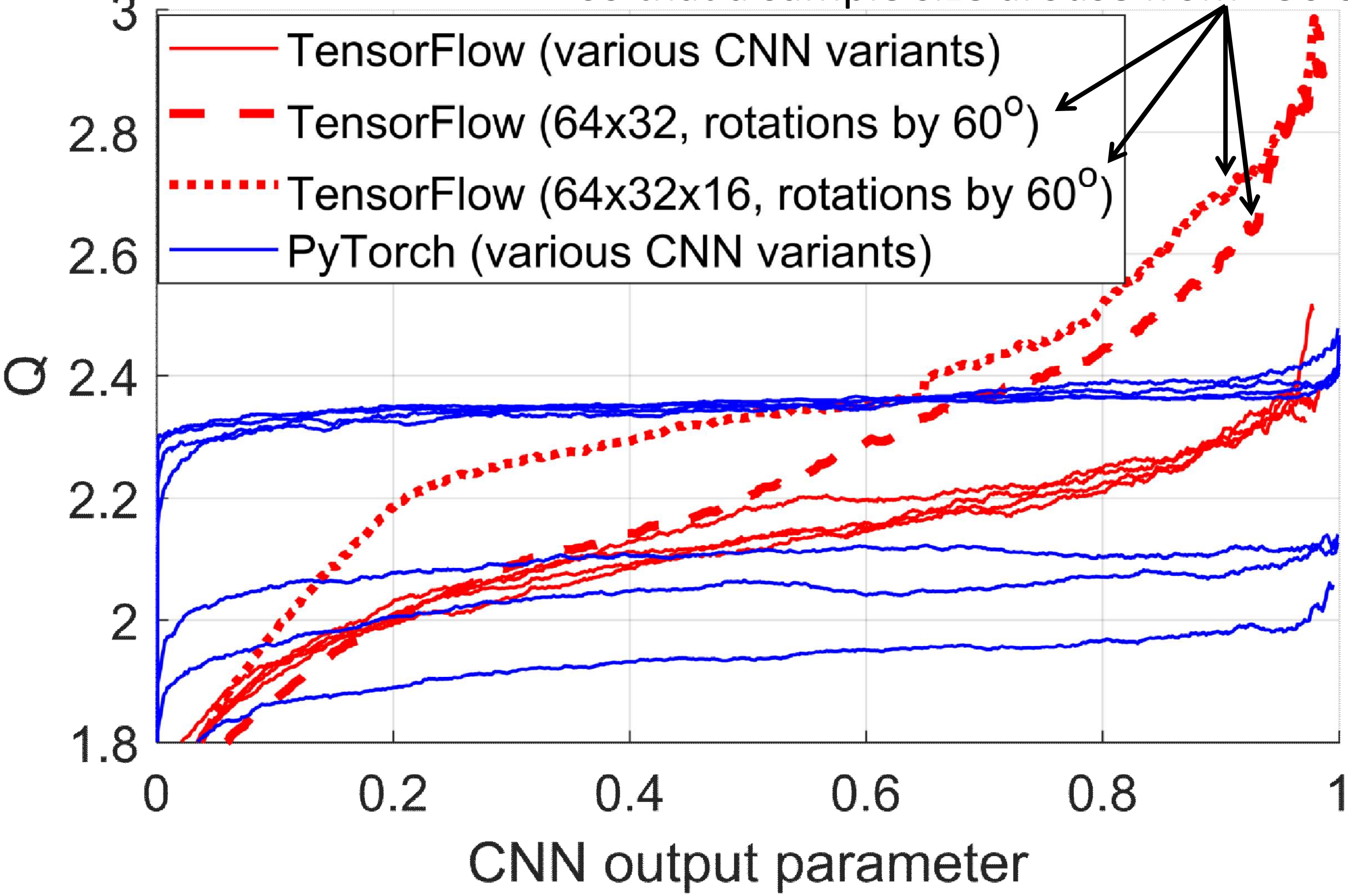
The idea of CNN is to behave in an invariant way across images.



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*)

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



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