Deep Learning with Keras Lab

Challenge results

INFIERI19, Wuhan, 25 May 2019

Emmy Noether-Programm
DFG Deutsche Forschungsgemeinschaft

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

DESY.

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

Challenge results

Best scores

|4| Celia_Shreya_v1 | **0.96784** |

|5| resultLMF_v4 | **0.96739** |

|6| infieriLMF | **0.96631** |

|7| YuboHan_and_DanZhang_v1 | **0.96507** |

|8| Zhou-Huang | **0.96496** |

|9| Celia_Shreya_v3 | **0.96495** |

|10| danzhang_yubohan_v2 | **0.96436** |

|11| Lorenzo_ML | **0.96427** |

|12| Three_Of_Kingkong | **0.96390** |

|13| Team_Yi_Pablo | **0.96174** |

|14| ODJGPF_v0 | **0.95972** |

|15| RAMON | **0.95965** |

|16| Md_RASEDUJJAMAN | **0.95634** |

|17| anyi_longlong_v1 | **0.95564** |

|18| ODJGPF_v1 | **0.95327** |

|19| Emanuele | **0.95140** |

|20| anyi_longlong_v2 | **0.94559** |

|21| ROCstefano | **0.87252** |

|22| Lorenzo_Tau | **0.86926** |

Challenge results

Best scores

|3| infieriLMF_v3 | 0.96809 |

|4| Celia_Shreya_v1 | **0.96784** |

|5| resultLMF_v4 | **0.96739** |

|6| infieriLMF | **0.96631** |

|7| YuboHan_and_DanZhang_v1 | **0.96507** |

|8| Zhou-Huang | **0.96496** |

|9| Celia_Shreya_v3 | **0.96495** |

|10| danzhang_yubohan_v2 | **0.96436** |

|11| Lorenzo_ML | **0.96427** |

|12| Three_Of_Kingkong | **0.96390** |

|13| Team_Yi_Pablo | **0.96174** |

|14| ODJGPF_v0 | **0.95972** |

|15| RAMON | **0.95965** |

|16| Md_RASEDUJJAMAN | **0.95634** |

|17| anyi_longlong_v1 | **0.95564** |

|18| ODJGPF_v1 | **0.95327** |

|19| Emanuele | **0.95140** |

|20| anyi_longlong_v2 | **0.94559** |

|21| ROCstefano | **0.87252** |

|22| Lorenzo_Tau | **0.86926** |

Challenge results

Best scores

|2| Celia_Shreya_v2 | 0.96818 |

|3| infieriLMF_v3 | **0.96809** |

|4| Celia_Shreya_v1 | **0.96784** |

|5| resultLMF_v4 | **0.96739** |

|6| infieriLMF | **0.96631** |

|7| YuboHan_and_DanZhang_v1 | **0.96507** |

|8| Zhou-Huang | **0.96496** |

|9| Celia_Shreya_v3 | **0.96495** |

|10| danzhang_yubohan_v2 | **0.96436** |

|11| Lorenzo_ML | **0.96427** |

|12| Three_Of_Kingkong | **0.96390** |

|13| Team_Yi_Pablo | **0.96174** |

|14| ODJGPF_v0 | **0.95972** |

|15| RAMON | **0.95965** |

|16| Md_RASEDUJJAMAN | **0.95634** |

|17| anyi_longlong_v1 | **0.95564** |

|18| ODJGPF_v1 | **0.95327** |

|19| Emanuele | **0.95140** |

|20| anyi_longlong_v2 | **0.94559** |

|21| ROCstefano | **0.87252** |

|22| Lorenzo_Tau | **0.86926** |

Challenge results

Best scores

|1| TheThreeStooges | 0.96903 |

|2| Celia_Shreya_v2 | **0.96818** |

|3| infieriLMF_v3 | **0.96809** |

|4| Celia_Shreya_v1 | **0.96784** |

|5| resultLMF_v4 | **0.96739** |

|6| infieriLMF | **0.96631** |

|7| YuboHan_and_DanZhang_v1 | **0.96507** |

|8| Zhou-Huang | **0.96496** |

|9| Celia_Shreya_v3 | **0.96495** |

|10| danzhang_yubohan_v2 | **0.96436** |

|11| Lorenzo_ML | **0.96427** |

|12| Three_Of_Kingkong | **0.96390** |

|13| Team_Yi_Pablo | **0.96174** |

|14| ODJGPF_v0 | **0.95972** |

|15| RAMON | **0.95965** |

|16| Md_RASEDUJJAMAN | **0.95634** |

|17| anyi_longlong_v1 | **0.95564** |

|18| ODJGPF_v1 | **0.95327** |

|19| Emanuele | **0.95140** |

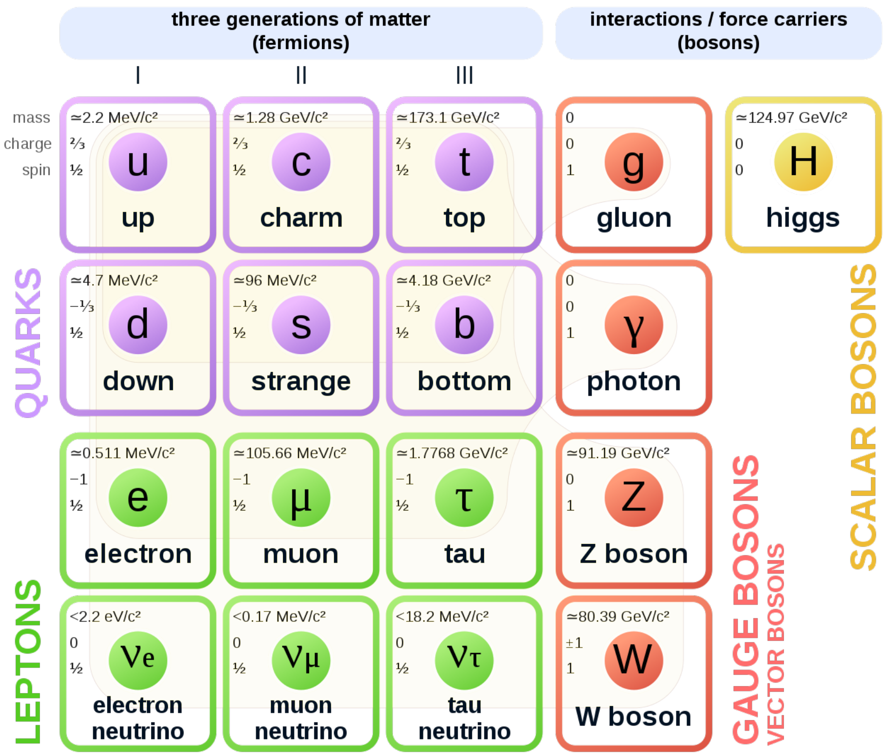
|20| anyi_longlong_v2 | **0.94559** |

|21| ROCstefano | **0.87252** |

|22| Lorenzo_Tau | **0.86926** |

Top-quark in the Standard Model

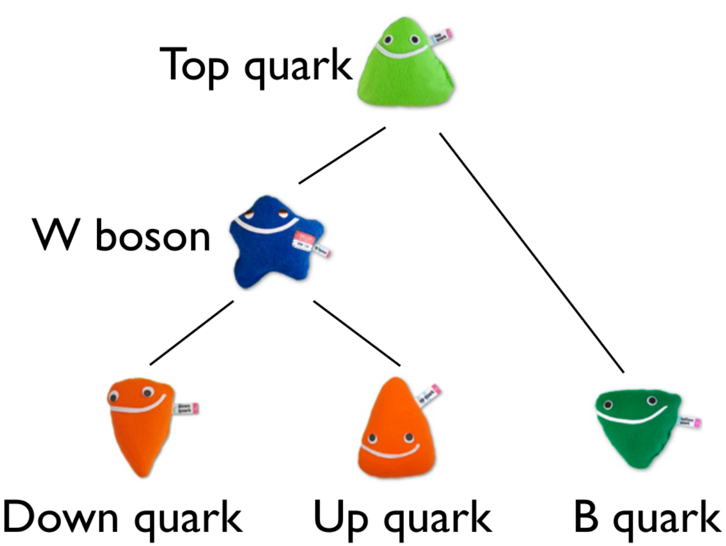
Standard Model of Elementary Particles



https://en.wikipedia.org/wiki/Standard_Model

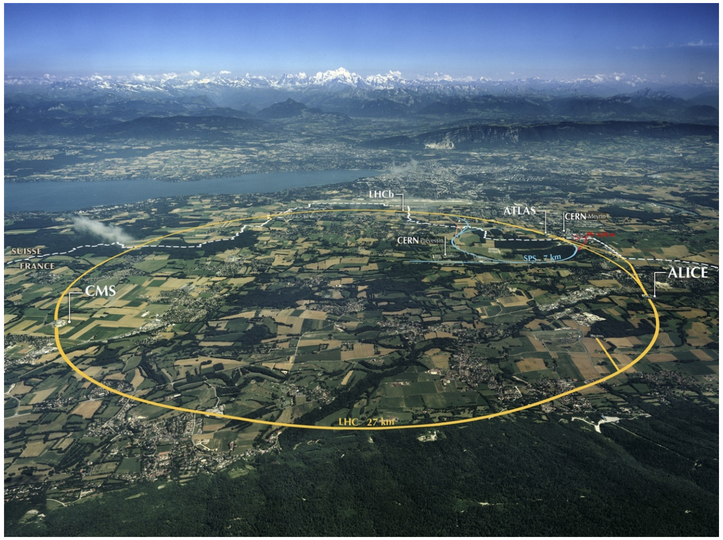
top quark: key particle to search for new physics and for precision measurements

- heaviest known elementary particle (m ≈ 172.5 GeV)
- very short lifetime (10^{-25} s) → only decay products detectable
- study hadronically decaying top quarks:



https://www.particlezoo.net

Producing and detecting top quarks

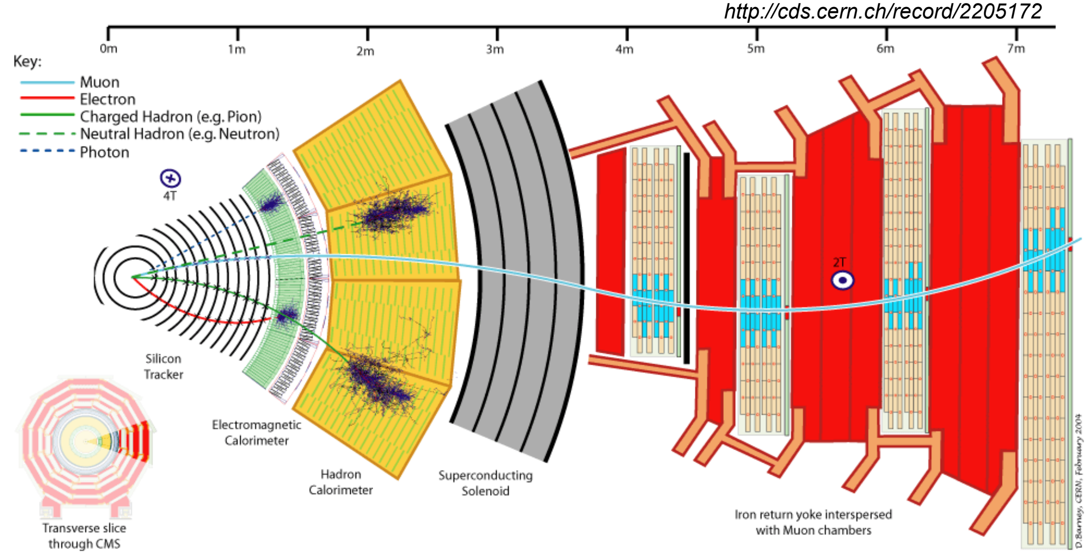


https://home.cern/news/news/cern/25-years-large-hadron-collider-experimental-programme

- **L**arge **H**adron **C**ollider (LHC)
- pp collider at CERN
- 27 km circumference
- centre-of-mass energy: 13 TeV

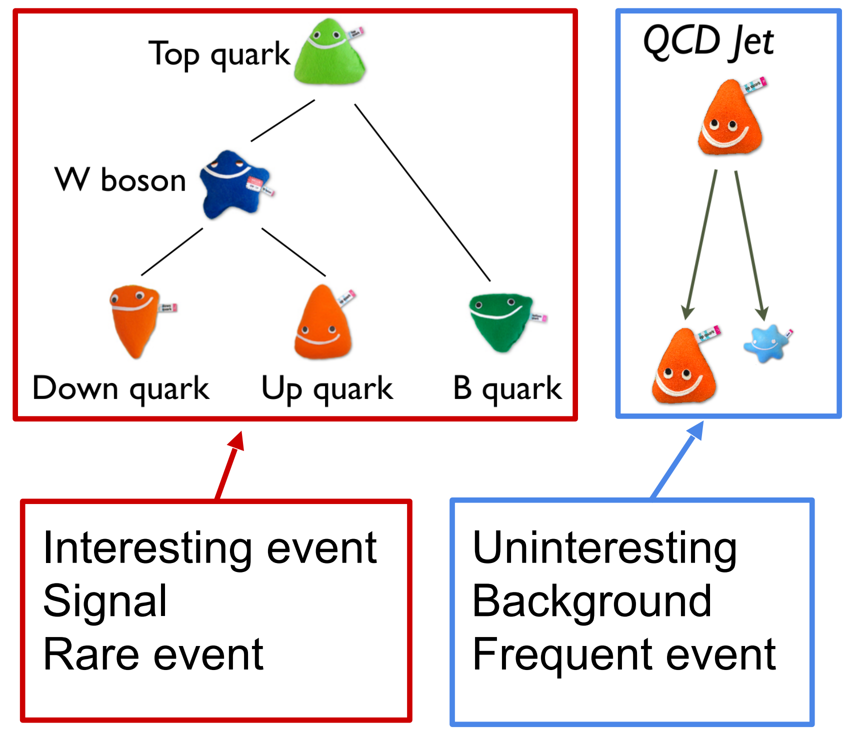
- **C**ompact **M**uon **S**olenoid (CMS)
- silicon tracker, ECAL, HCAL, solenoid, muon system
- reconstructs four-momenta of all visible final-state particles

http://cds.cern.ch/record/2205172



Why is top-quark tagging complicated?

- Top quark production is a “rare” phenomenon
- Other processes initiated by strong interaction (*QCD*) occur way more often
- They produce lighter quarks (up, down, strange, ...)
- They look similar to top quarks and they happen enormously more often
- Fighting against this background is a huge challenge!

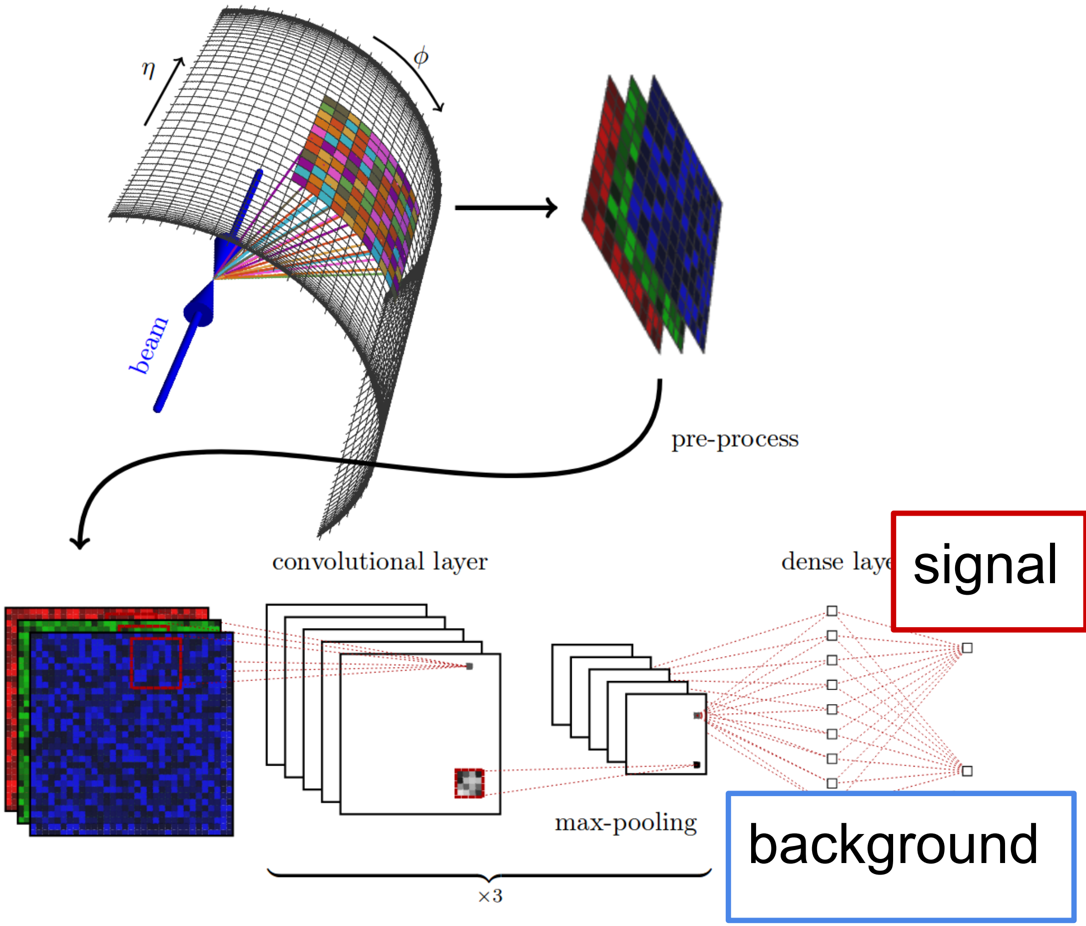


Machine learning formulation:

- Solve a binary classification problem
  - class 0: background (QCD)
  - class 1: signal (top)
- We must build a good architecture

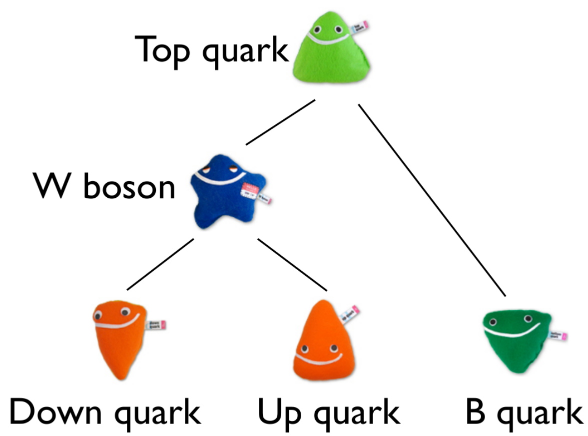
From jets to images

- In particle detectors, quarks are reconstructed as **jets** of particles
- Cylindrical surface of detector can be unrolled along longitudinal and radial coordinates and divided in **pixels**
- The energy deposits of the jets constituents are transformed into "intensities" of a 2D **black and white 40x40 pixelated image**
- Image recognition algorithms can be applied to a high-energy physics problem!

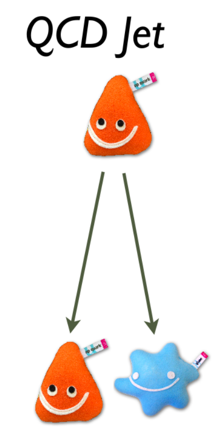


Machine learning approaches to top-quark tagging

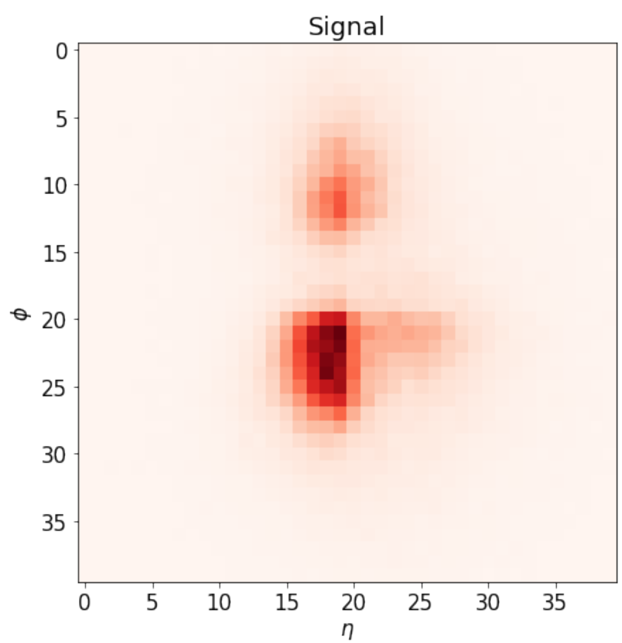
signal event (rare):

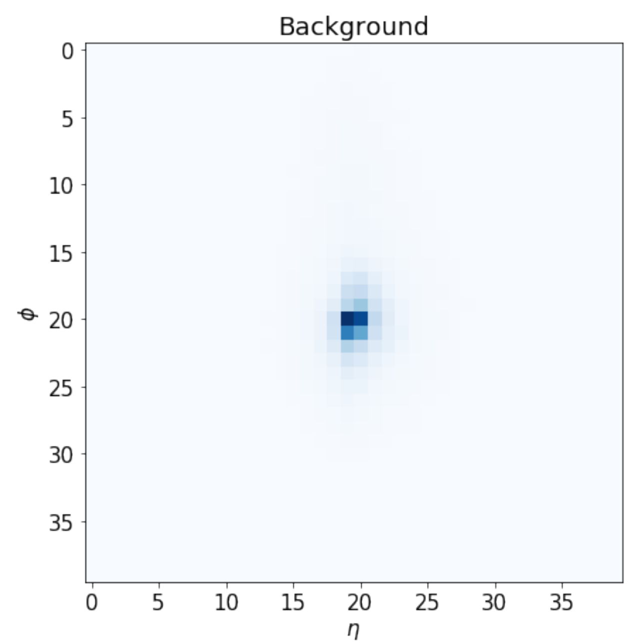


background event (frequent):



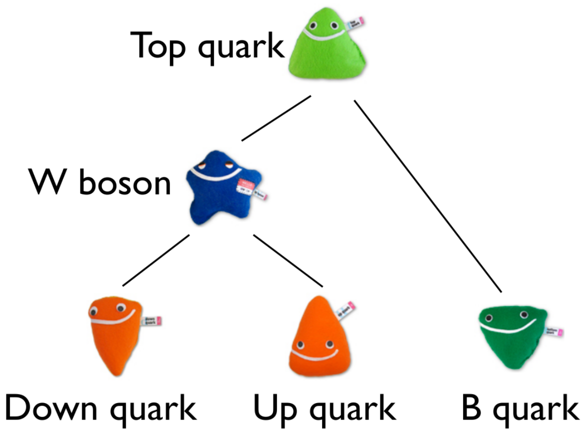
in the detector:
1 M image average



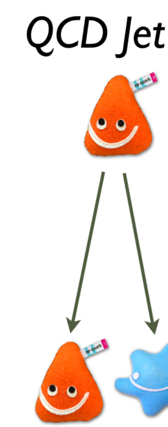


Machine learning approaches to top-quark tagging

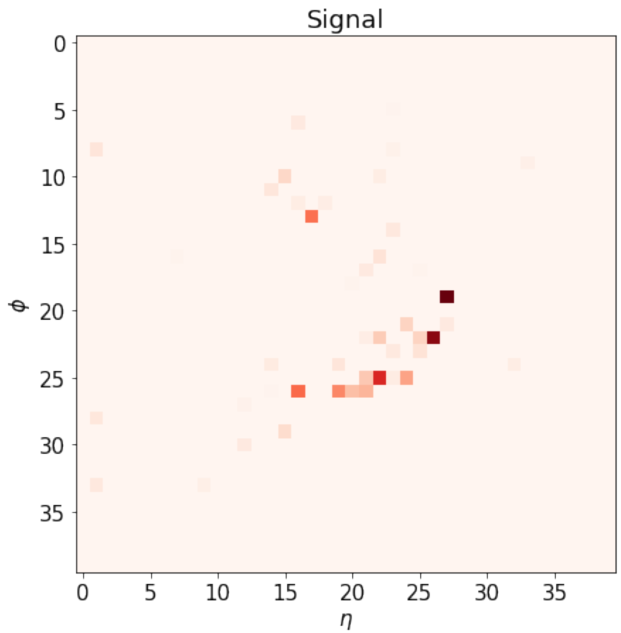
signal event (rare):

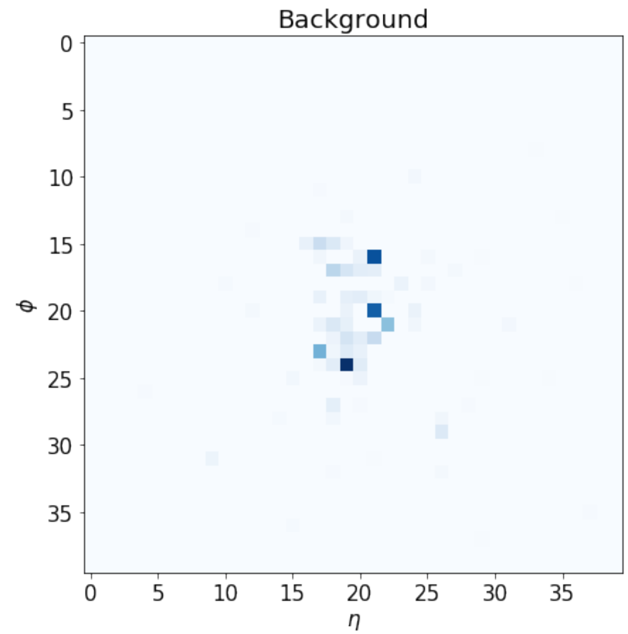


background event (frequent):



in the detector:
1 image

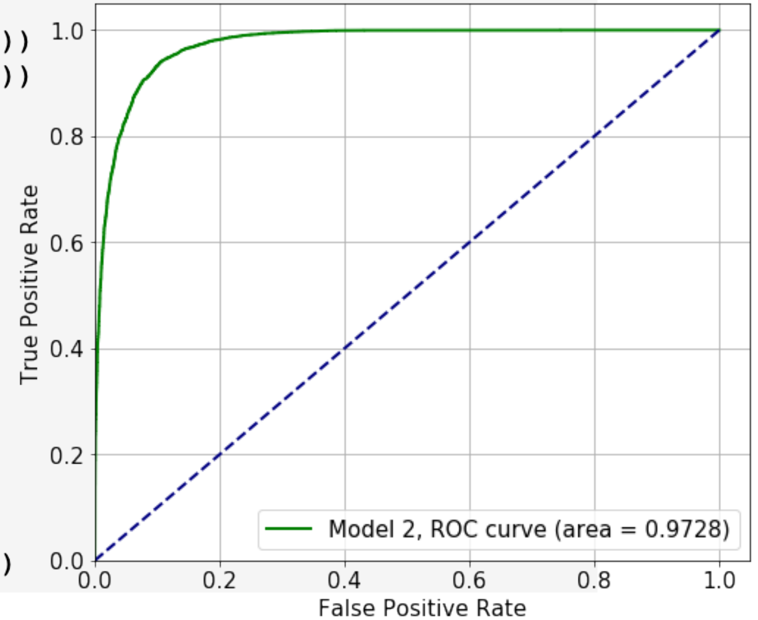




Winning architecture

- Convolutional Neural Networks are the best ML techniques for image recognition
- Input: training with 280k images, validation with 20k images, 7 epochs

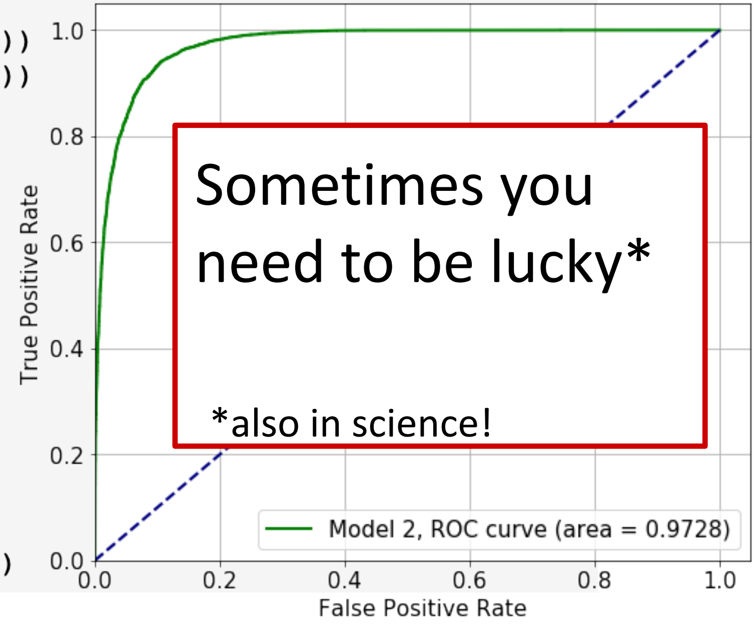
```
model2 = keras.models.Sequential()

model.add(keras.layers.GaussianNoise(0.1,input_shape=(40, 40,1)))  #added gaussian noise

model2.add(keras.layers.Conv2D(32,(3,3),padding='same',input_shape=(40,40,1)))
model2.add(keras.layers.Conv2D(32,(3,3),padding='same'))
model2.add(keras.layers.MaxPooling2D(pool_size=(3,3)))

model2.add(keras.layers.Conv2D(32,(3,3),padding='same'))
model2.add(keras.layers.Conv2D(64,(3,3),padding='same'))
model2.add(keras.layers.MaxPooling2D(pool_size=(3,3)))



model2.add(keras.layers.Flatten())

model2.add(keras.layers.Dense(100, activation='relu'))
model2.add(keras.layers.Dense(200, activation='relu'))
model.add(keras.layers.Dropout(0.15))

model2.add(keras.layers.Dense(200, activation='relu'))
model.add(keras.layers.Dropout(0.3))

model2.add(keras.layers.Dense(2, activation='softmax'))
```



Winning architecture

- Convolutional Neural Networks are the best ML techniques for image recognition
- Input: training with 280k images, validation with 20k images, 7 epochs

```
model2 = keras.models.Sequential()

model.add(keras.layers.GaussianNoise(0.1,input_shape=(40, 40,1)))  #added gaussian noise

model2.add(keras.layers.Conv2D(32,(3,3),padding='same',input_shape=(40,40,1)))
model2.add(keras.layers.Conv2D(32,(3,3),padding='same'))
model2.add(keras.layers.MaxPooling2D(pool_size=(3,3)))

model2.add(keras.layers.Conv2D(32,(3,3),padding='same'))
model2.add(keras.layers.Conv2D(64,(3,3),padding='same'))
model2.add(keras.layers.MaxPooling2D(pool_size=(3,3)))


model2.add(keras.layers.Flatten())

model2.add(keras.layers.Dense(100, activation='relu'))
model2.add(keras.layers.Dense(200, activation='relu'))
model.add(keras.layers.Dropout(0.15))

model2.add(keras.layers.Dense(200, activation='relu'))
model.add(keras.layers.Dropout(0.3))

model2.add(keras.layers.Dense(2, activation='softmax'))
```



Sometimes you need to be lucky*

*also in science!

Thanks for your attention

&

have a safe return journey!