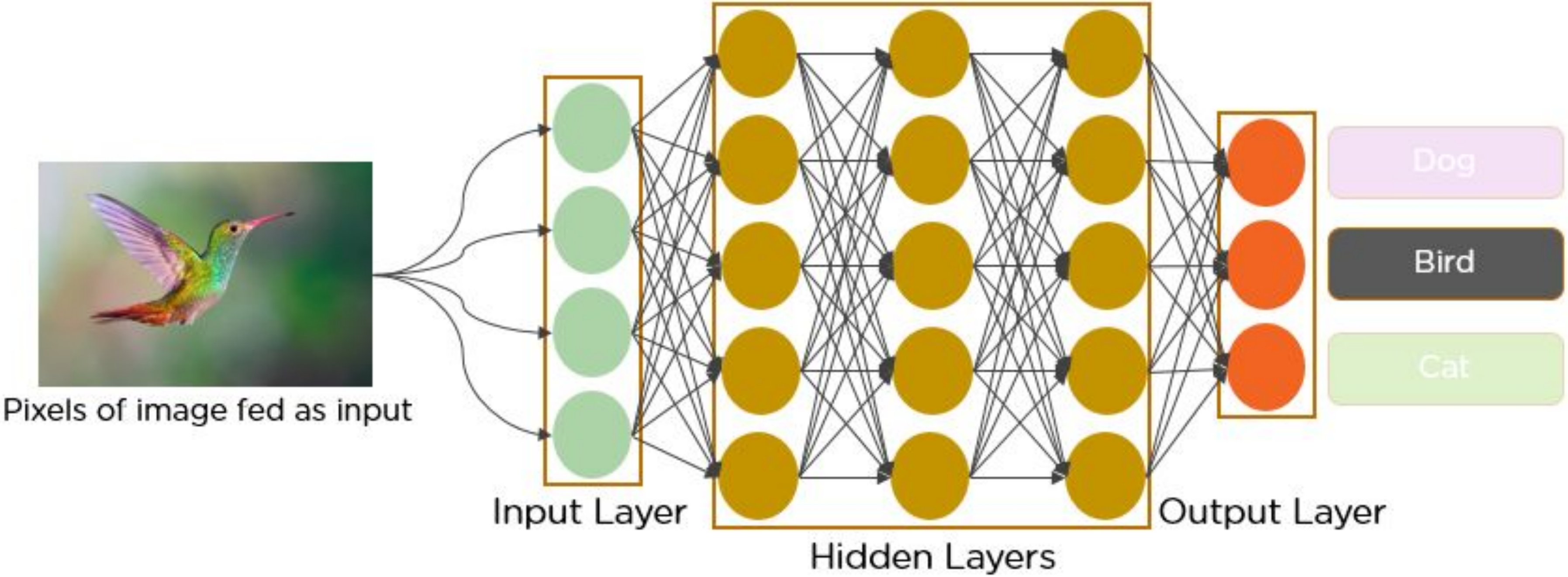
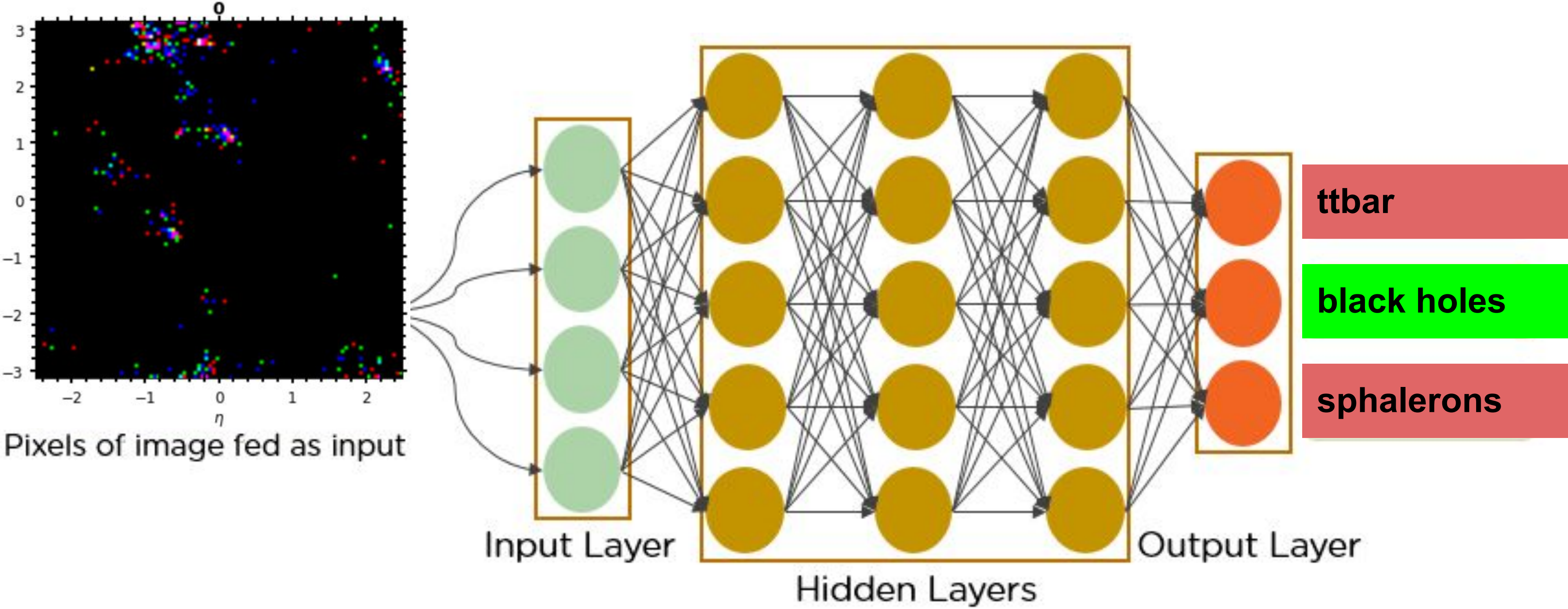
Event generation progress

AKA what Aurora did this summer

What is my goal?

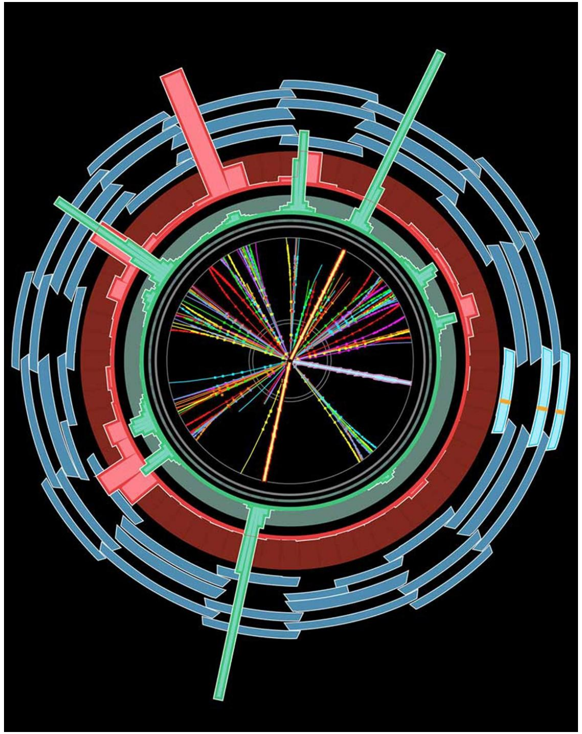
Apply ML computer vision techniques on LHC data.

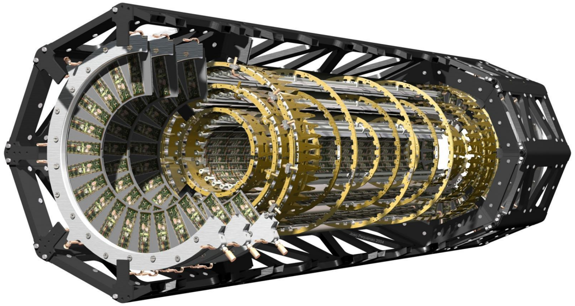



0
η
Pixels of image fed as input
Input Layer
Hidden Layers
Output Layer
ttbar
black holes
sphalerons

End-to-end classifier

- Takes in low-level matrix data:
    - Calorimeter towers
    - Tracks
- Does not use reconstructed particle objects (like met, photons, jets etc)
- Matrix data corresponds to detector geometry: ATLAS is a BIG camera
- Images made from data (the whole event in one image)
- Uses neural networks for classification
- Outputs the label of the event





3 different event types

1. ttbar
   - top + antitop -> $W^+ b W^- b^-$
   - p_T > 1000 GeV (in event generation stage)
2. Microscopic black holes (hypothetical)
   - Requires extra spatial dimensions (4-6 should be explored)
   - Could be produced at LHC
   - Minimum mass can be defined. 8-12 TeV should be explored.
3. Sphalerons (hypothetical)
   - A “particle like” solution to the electroweak field equations
   - Could look similar to black hole event

Generation process (will be improved)

	Parton level	Hadronization	Detector response
ttbar	**PYTHIA** g g -> t tbar q qbar -> t tbar PhaseSpace:mHatMin = 1000 GeV PhaseSpace:pTHatMin = 1000 GeV	**Herwig7** PDFName MSTW2008lo68cl	**DELPHES** ATLAS card
Black holes	**BlackMax** Minimum_mass(GeV) 8-12000 Maximum_mass(GeV) 18000	**Herwig7** PDFName MSTW2008lo68cl	**DELPHES** ATLAS card
Sphalerons	**Herwig7** Not working	*Not working now* **Herwig7** Andreas Papaefstathiou	**DELPHES** ATLAS card

Data sets for **this** presentation

ttbar: min 1 TeV during event generation
black_holes: BH_n6_M8 (6 extra dimensions, min 8 TeV mass)
sphalerons: Andreas made this file

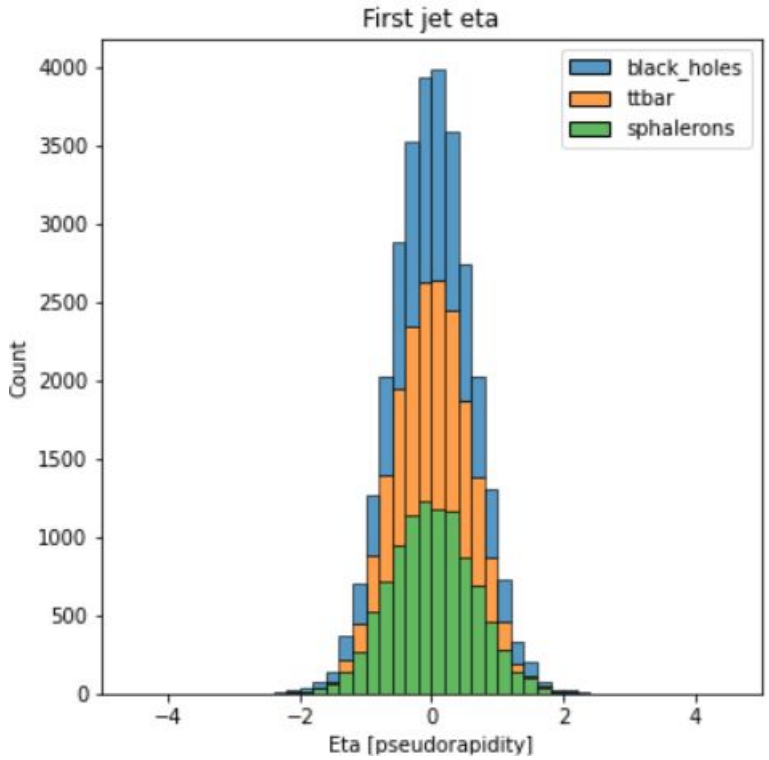
n_events = 10 000 per event type

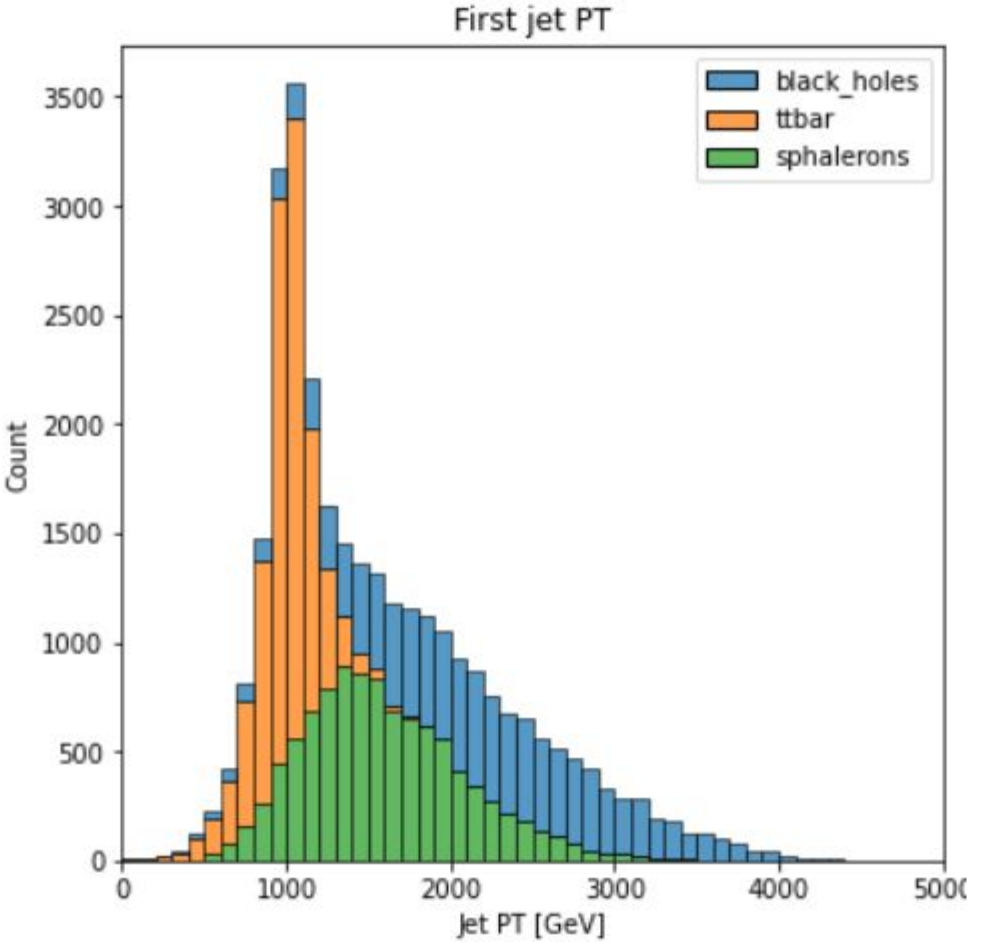
train/test = 0.8/0.2

ttbar is pretty easy to separate from the other two, but it is good to have such a sanity check. With this data I expect at least 80% accuracy on the ttbar classification. I will make a more relevant sample later.

Jets

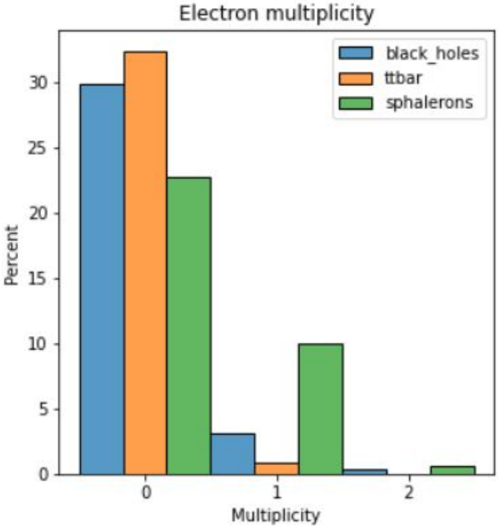
First jet = jet with highest p_T

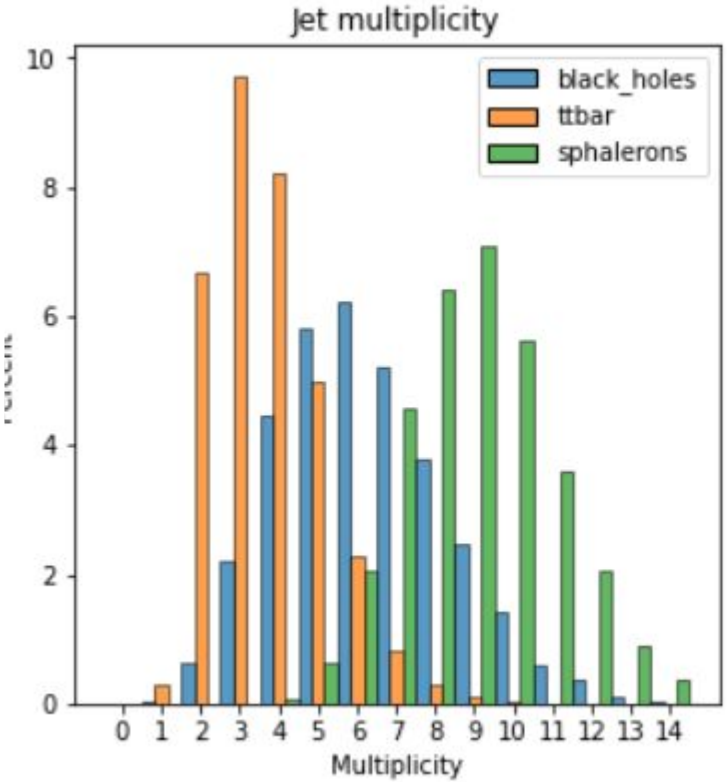


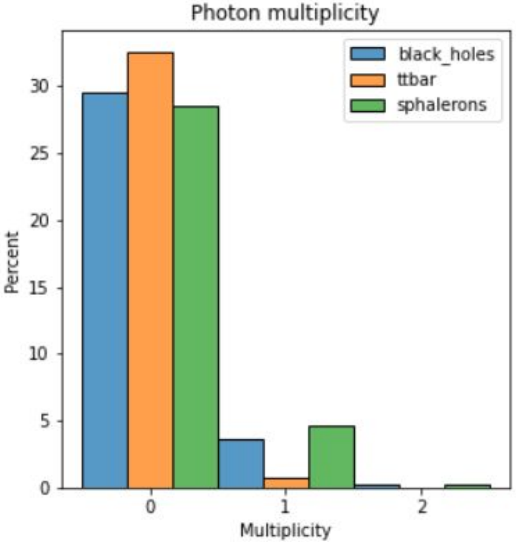


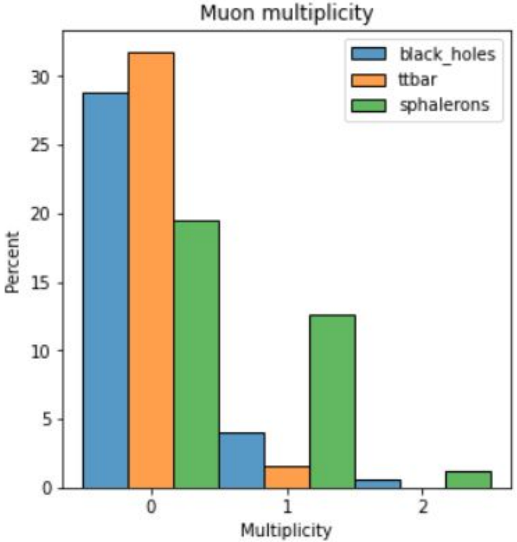
Object multiplicity

Sphalerons have significantly more jets, electrons and muons.



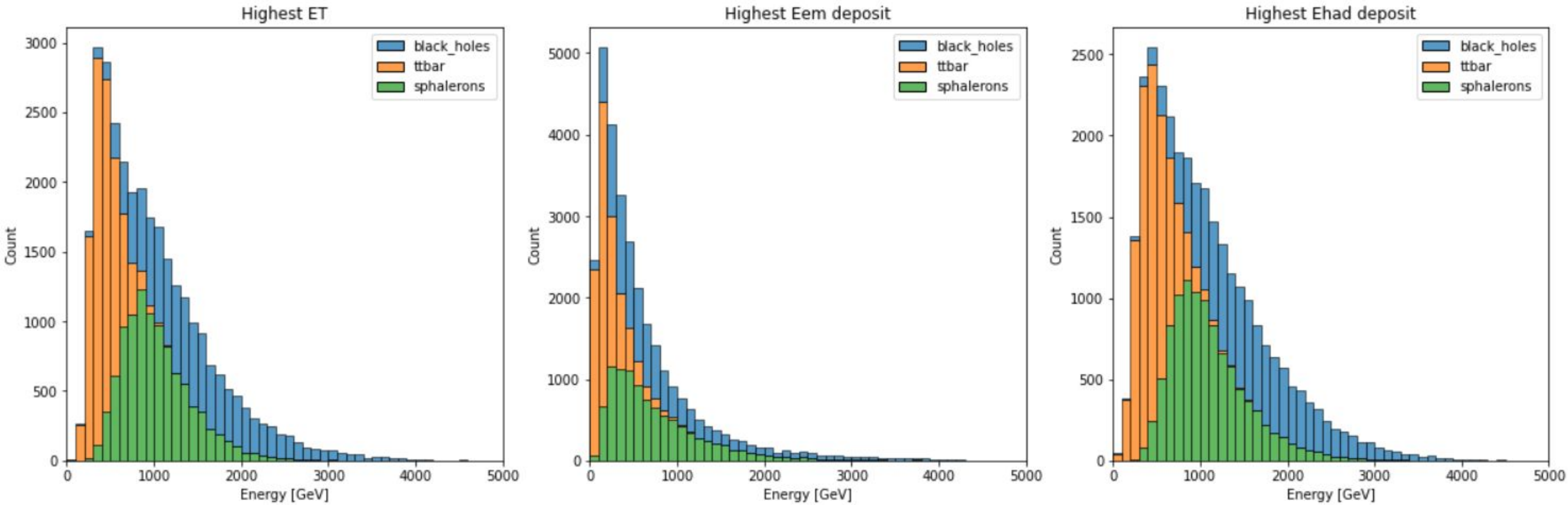






Tower deposits

ET = Total energy, Eem = electromagnetic calorimeter energy, Ehad = hadronic calorimeter energy.



Making the images

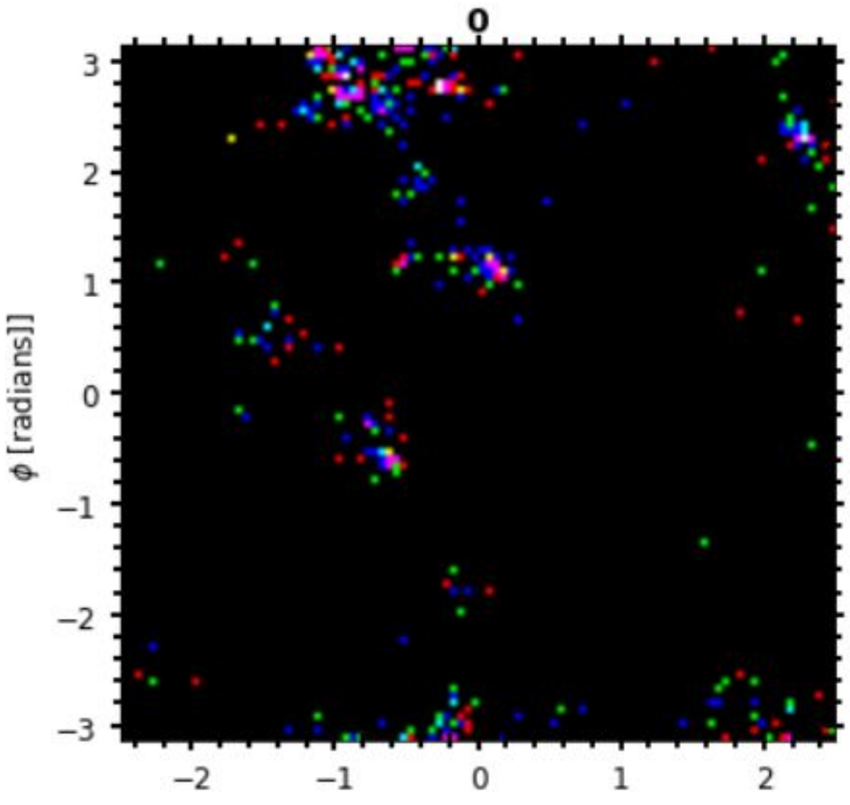
3 channel RGB, square RESxRES, normalised to 0-255

RES = 100 for now

Following procedure by Andrews2020 (CMS Open Data paper)

R = Eem
G = Ehad
B = Tracks

Energy[Energy>MAX_ENERGY] = MAX_ENERGY
Energy = 255*Energy/MAX_ENERGY



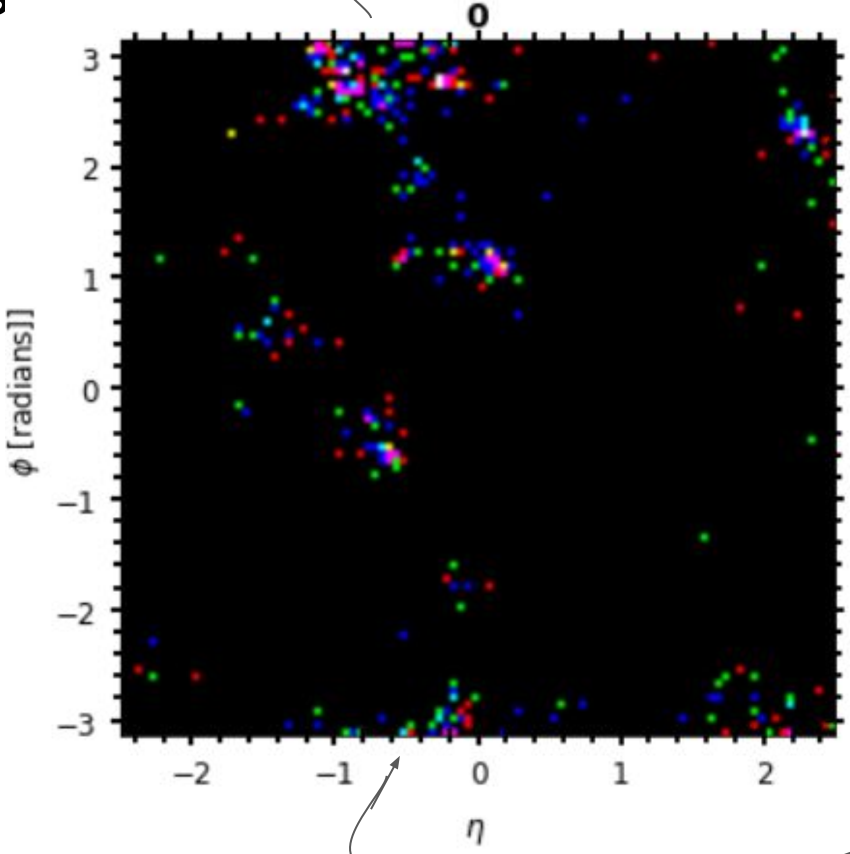
There are many possibilities

Data augmentation: create many more images by doing random phi rotations or flipping over phi axis (eta = -eta).

Resolution

Filters:
Different saturation level
Treat layers differently
Smearing

Same jet



Machine learning

- Very preliminary efforts
- Mostly proof of concept
- See notebook for details on implementation: https://github.com/choisant/imcalML/blob/main/notebooks/CNN_simple_classifier.ipynb
- Pytorch, from scratch models:
    - Simple CNN with 3 layers
    - Resnet with 18 layers

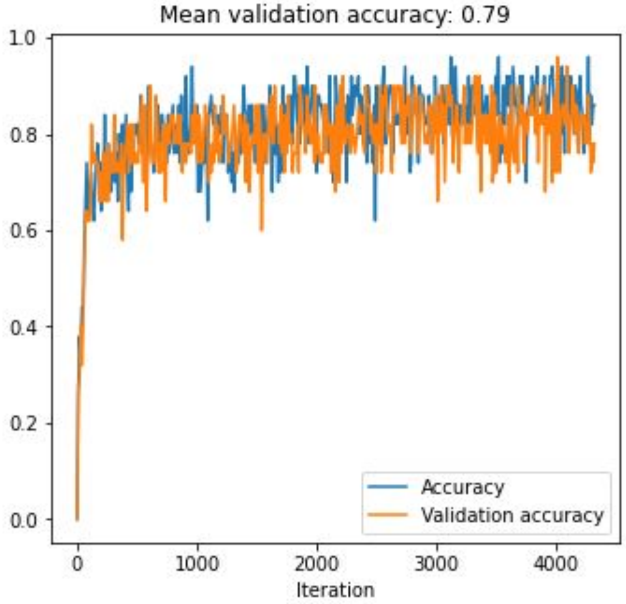
CNN (3 conv layers, 2 fc layers)

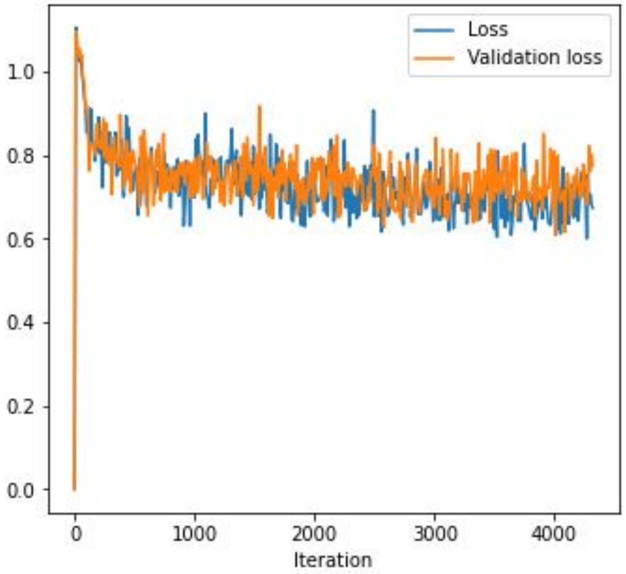
Optimizer: Adam
Loss function: Cross Entropy

Lr = 0.0001 constantly

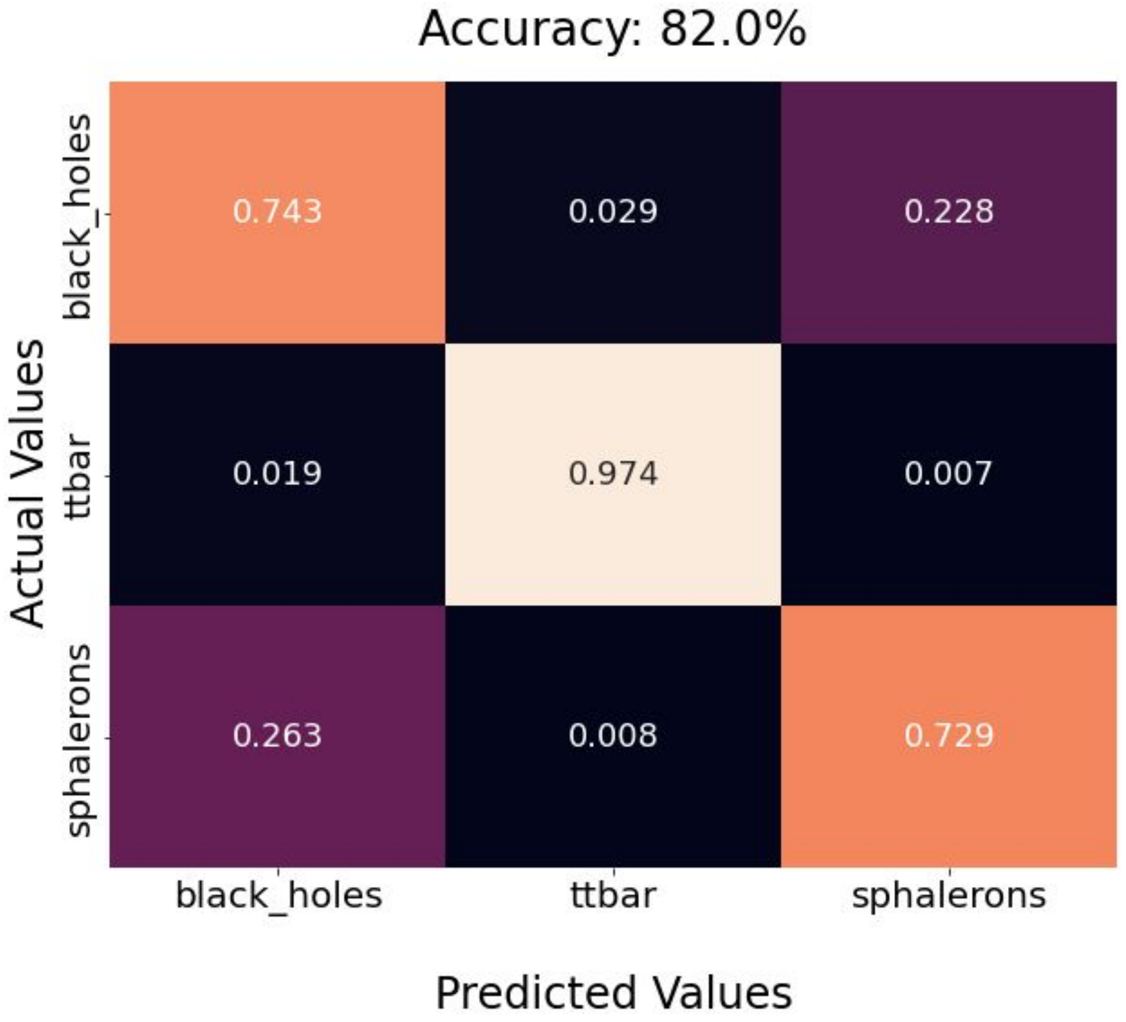
10 000 images, 10 epochs,
58 seconds (GPU)

Data aug:
Random flip horizontally and vertically





Easy to distinguish ttbar from the rest (expected with this data), harder to separate BH and sphalerons.



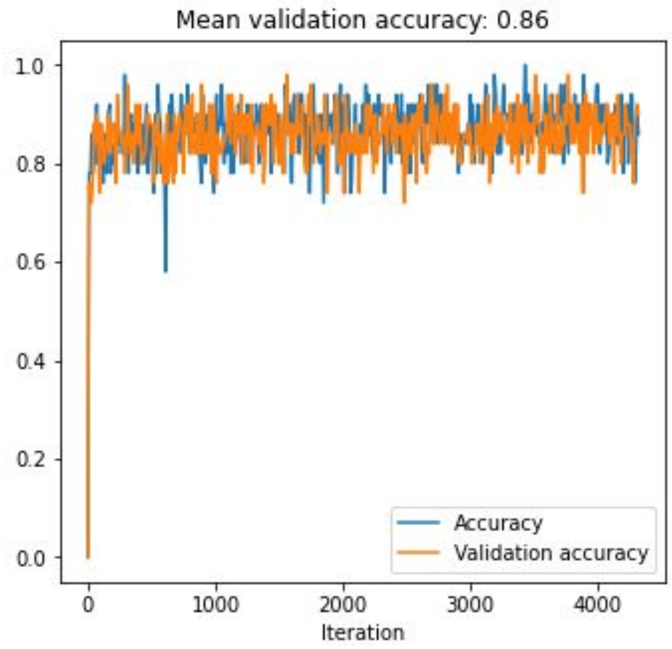
Resnet18 (18 layers)

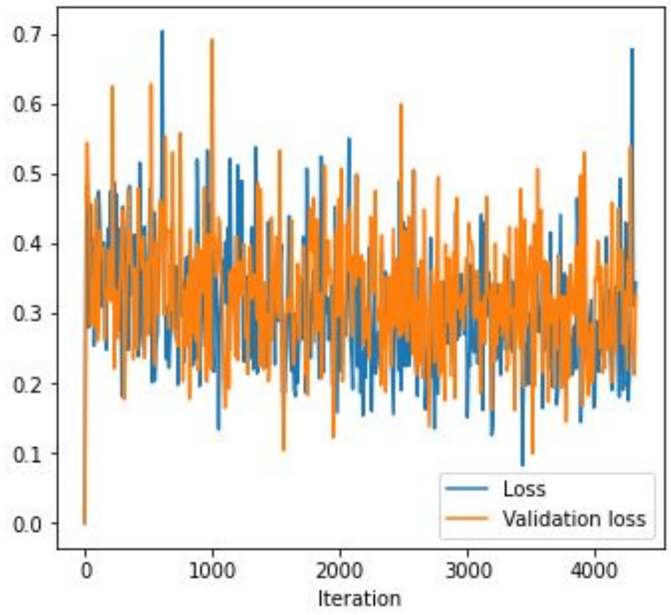
Optimizer: Adam
Loss function: Cross Entropy

10 000 images, 10 epochs,
1 min 58 seconds (GPU)

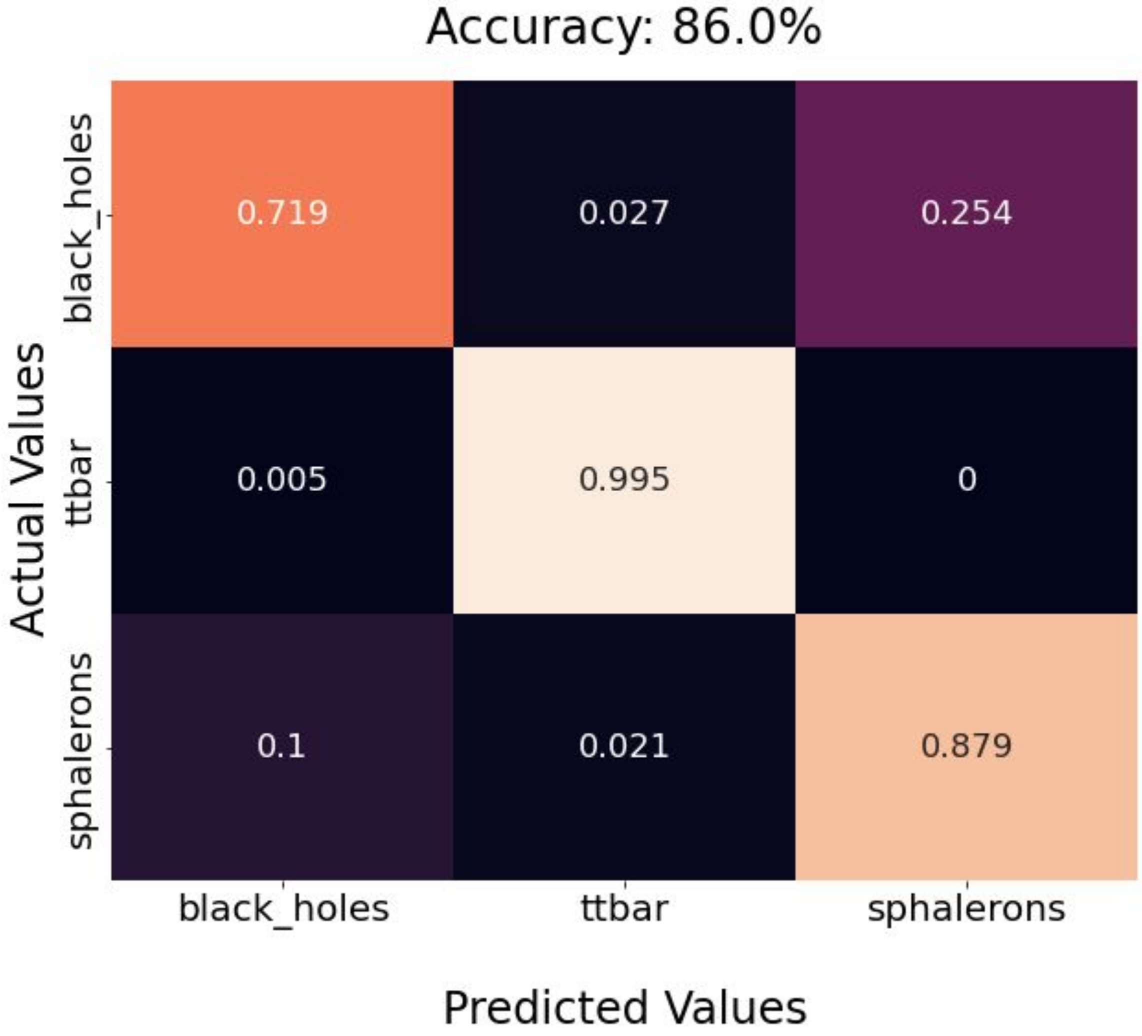
Lr = 0.0005 constantly

Data aug:
Random flip horizontally and vertically





Improvement by 4%



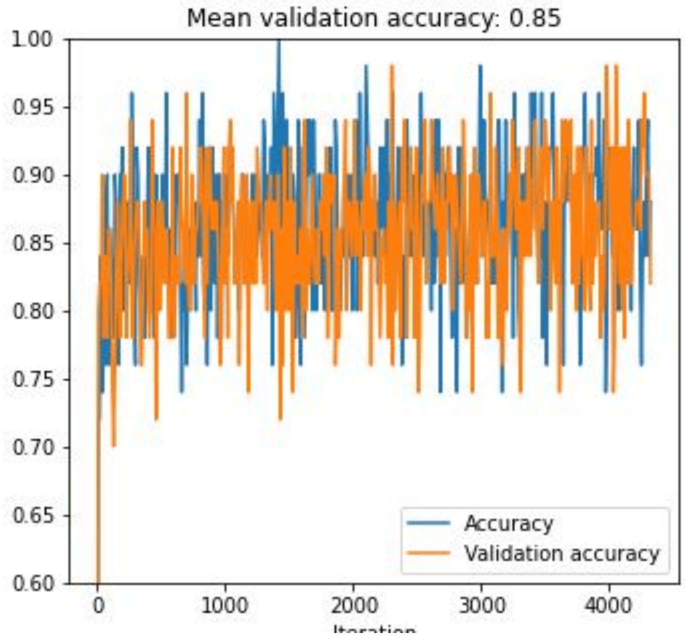
Resnet34 (34 layers)

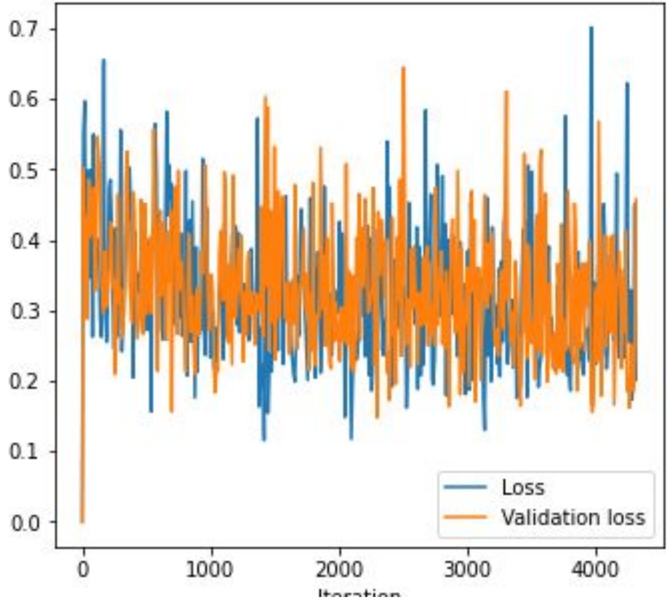
Optimizer: Adam
Loss function: Cross Entropy

10 000 images, 10 epochs,
3 min 36 seconds (GPU)

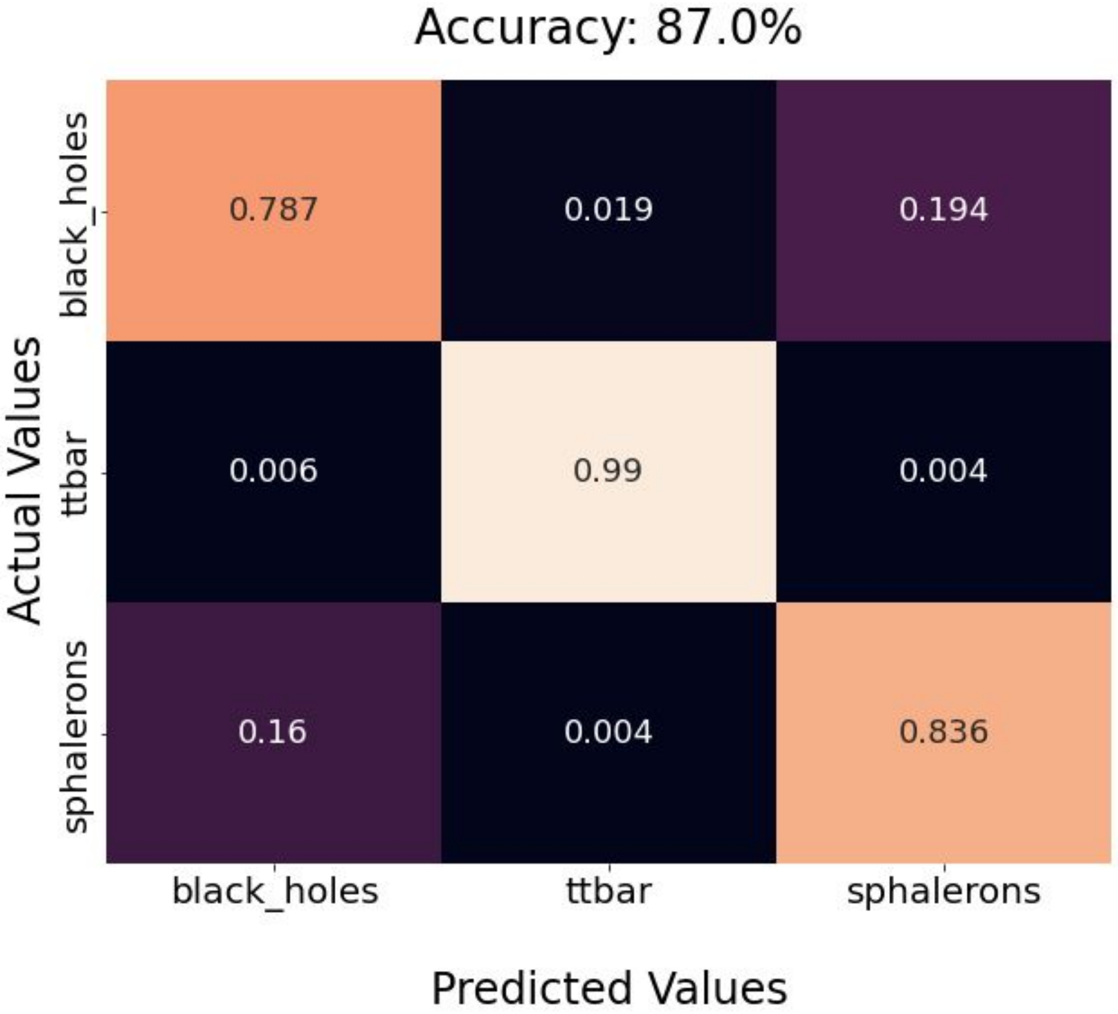
Lr = 0.0005 constantly

Data aug:
Random flip horizontally and vertically





Improvement by 1% from resnet18, but took twice the time to train.



To do

- Decide which data to use
- Implement random phi rotations for basically unlimited data augmentation
- Do some model analysis
    - Try to use the LUMIN framework to utilise FoldYielder functions (making ensembles of models)
- Figure out who is doing what for this paper
    - HVL contributions
    - GRIEG contributions
    - UiB contributions
    - Aurora contribution