Machine Learning for Data Quality Monitoring

Thomas Britton

Goal

- **Goal: Replace myself with an AI**
  - Further goals: Create a system which can monitor output from the detectors and flag problems early. Potentially pick out correctable issues (through calibration) and perform adjustments as needed

Tools and tactics

- Chose Keras with tensorflow backend

- Start by looking at the histograms, as .pngs (just like how I do it)

  - Use convolutional NN to do image classification

  - Take each classification/confidence and use an aggregator to classify the entire run/take action on flagged issues
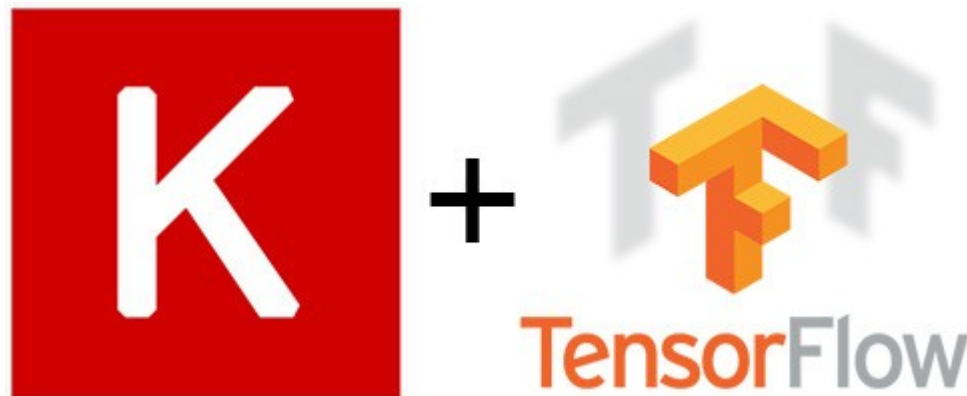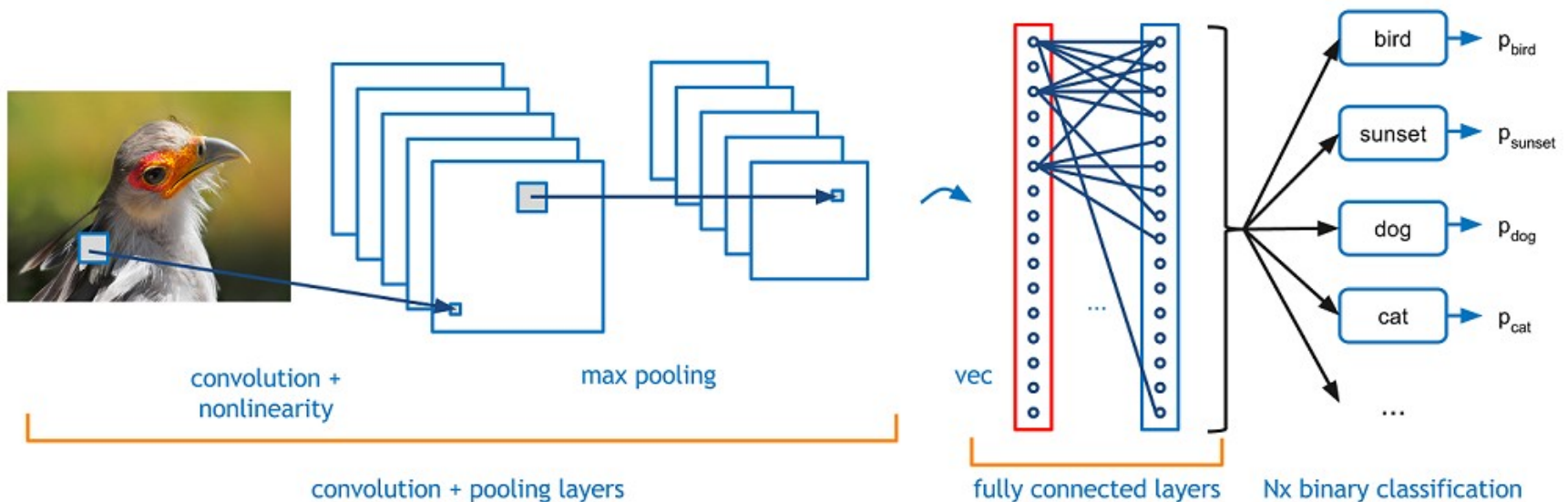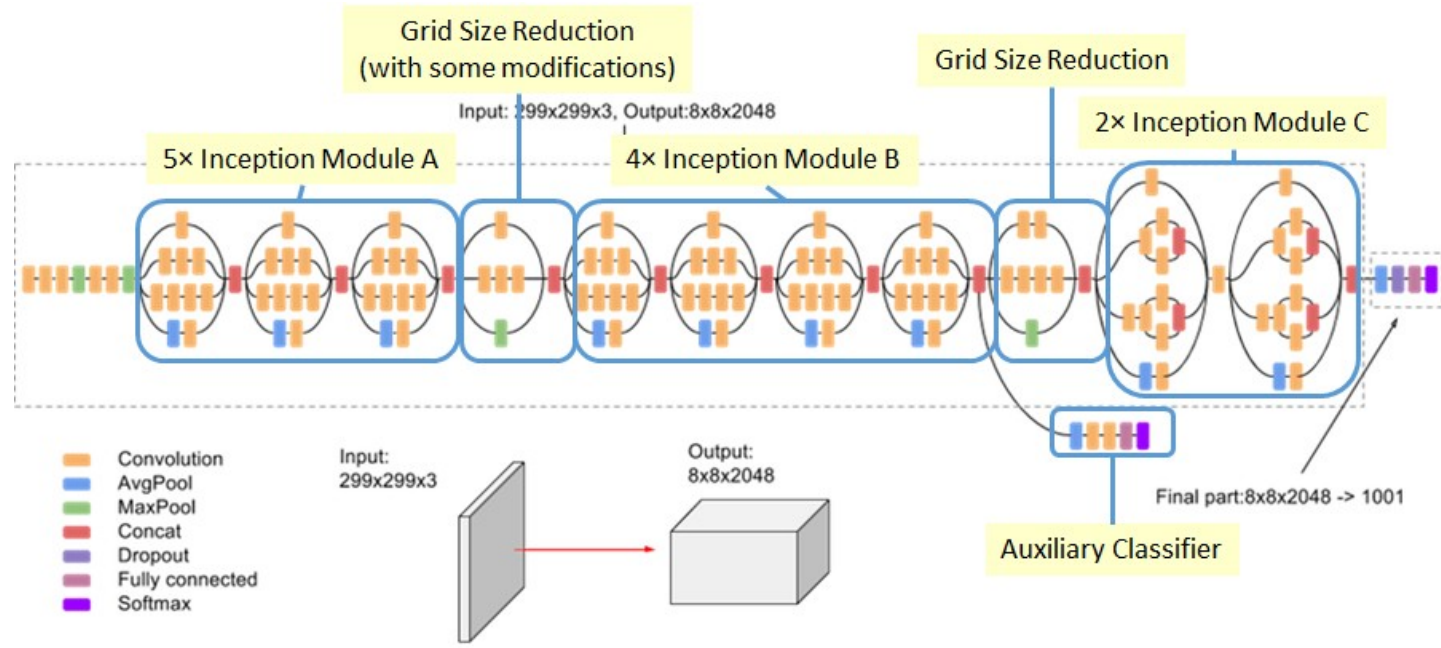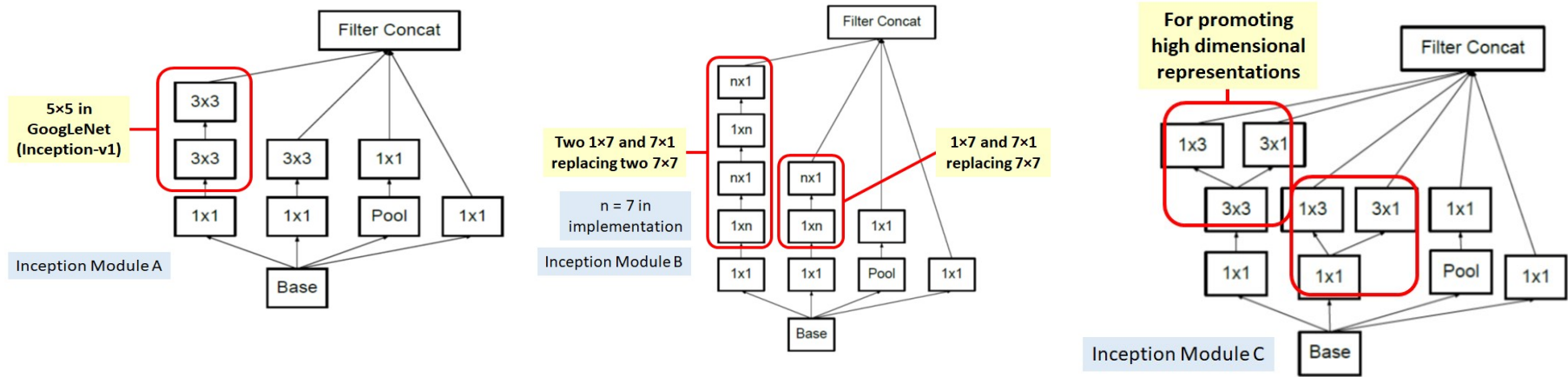
Early results

- Started with one image (BCAL_occupancy)
- Low sample training more okay in this case
  - **No** need for image augmentation!
- Training set: <u>75%</u> (randomly selected of the *2018-08 runs*)
- Validation: the other <u>25%</u>
- Grow the set by testing and labeling on other run periods
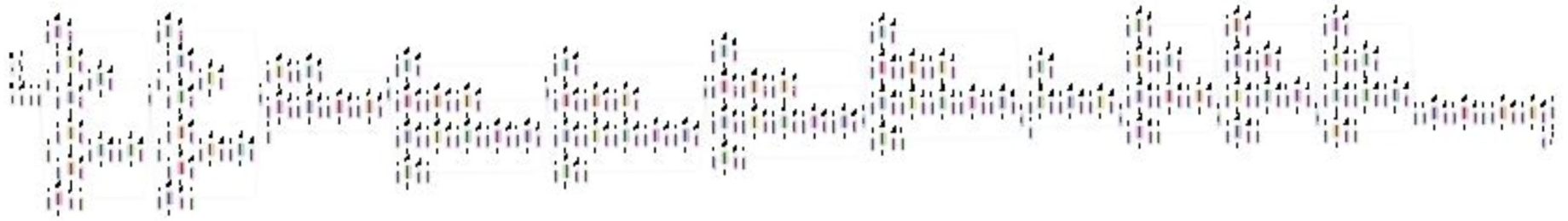
The Network

- My own: achieved **~96%** accuracy.

- Needed to use a more sophisticated network
  - Introducing **inceptionV3**
    - Think of it like a network trying various convolutions and figuring out which is best

The Network

Inception Module A

5×5 in GoogLeNet (Inception-v1)

Inception Module B

Two 1×7 and 7×1 replacing two 7×7

n = 7 in implementation

1×7 and 7×1 replacing 7×7

Inception Module C

For promoting high dimensional representations

Input: 299x299x3, Output:8x8x2048

Grid Size Reduction (with some modifications)

5× Inception Module A

4× Inception Module B

Grid Size Reduction

2× Inception Module C

Convolution
AvgPool
MaxPool
Concat
Dropout
Fully connected
Softmax

Input: 299x299x3

Output: 8x8x2048

Auxiliary Classifier

Final part:8x8x2048 -> 1001

Jefferson Lab

The Network

- "Good" accuracy of **99.4%**
  - False positive rate of **1.8%**
- "NoData" accuracy of **100.0%**
  - No false positives/ negatives
- "bad" accuracy of **93.3%**

Prediction v "Truth"

Towards Crowd sourcing labeling
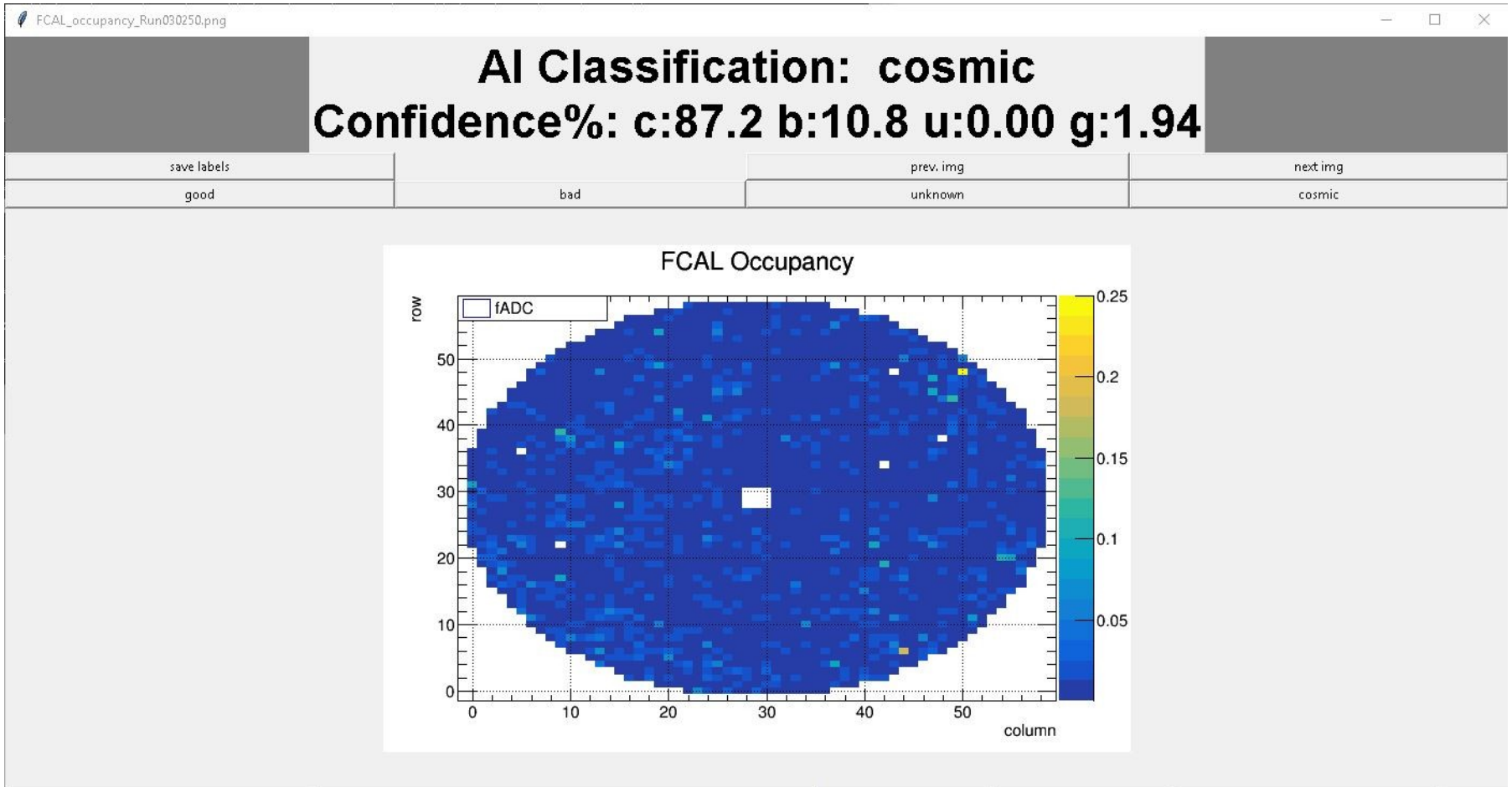
Future

- <u>Expert</u> labeling
  - Different/more varied classifications
- Build and train a few models
  - Connect them together (LSTM/Recurrent NN?)
- Integrate with data taking
  - Incremental training pipeline