FAST RADIO BURSTS CLASSIFICATION

EXP-4

API mistake

metrics

Accuracy: 0.92

Recall: 0.00

```python
@app.post("/eval/")
async def process_image(arr: ArrayRequest):
    arr = np.array(json.loads(arr.arr))
    print(arr.shape)
    return PlainTextResponse('000')
```

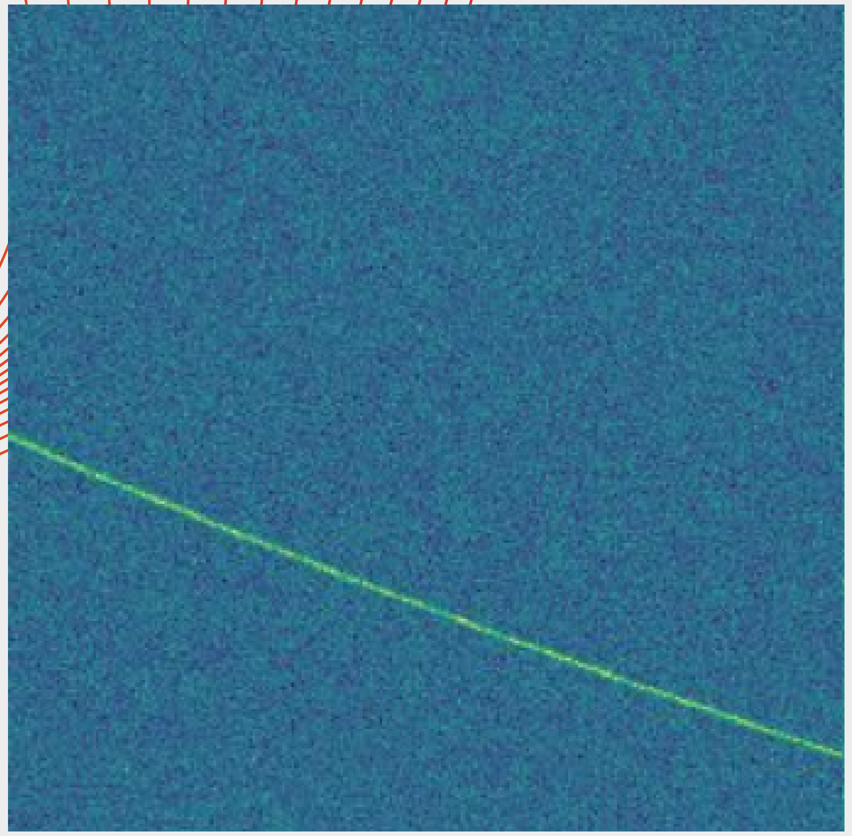
API responses all **None** data!

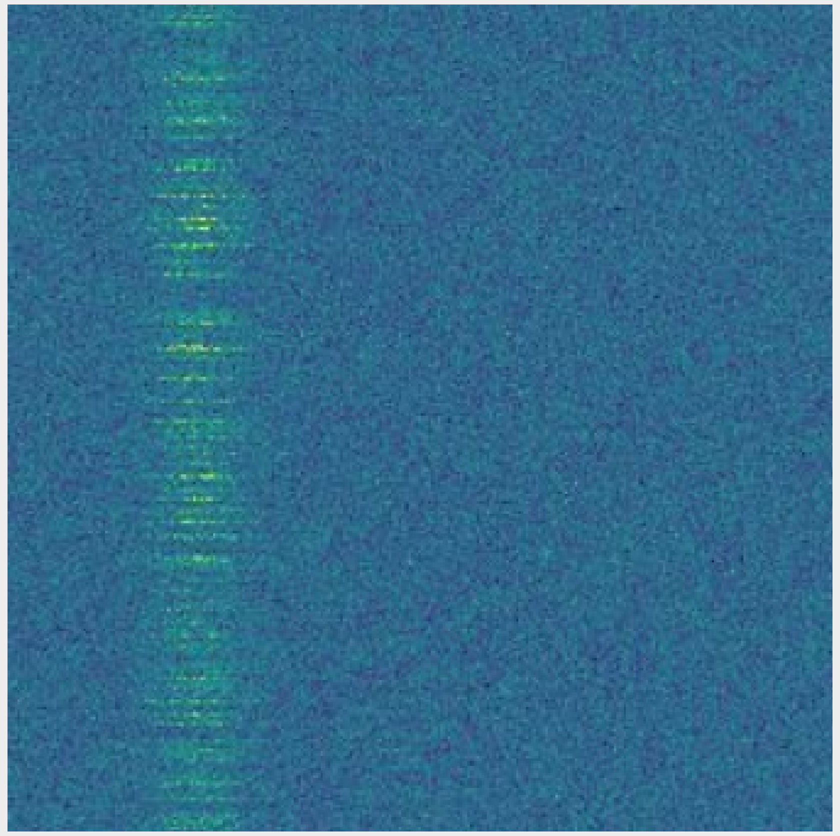
Introduction

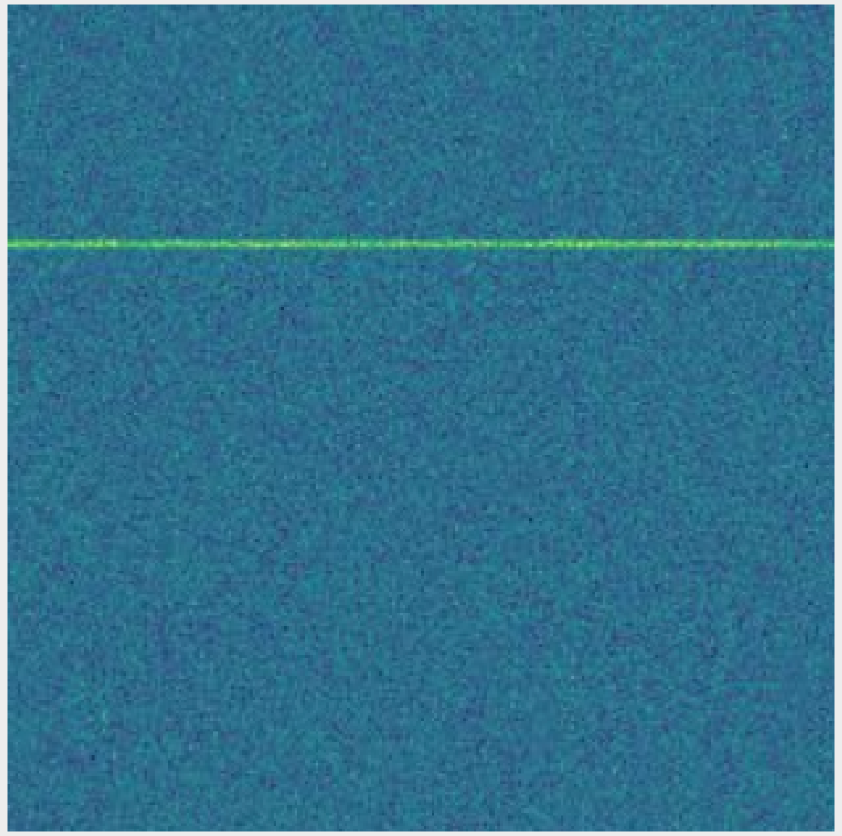
- Data are from Effelsberg Radio Telescope
- Discriminate fast burst from broadband and narrowband
- The solution should classify exactly the Pulses from pulsars
- Need to be faster best model (3ms)

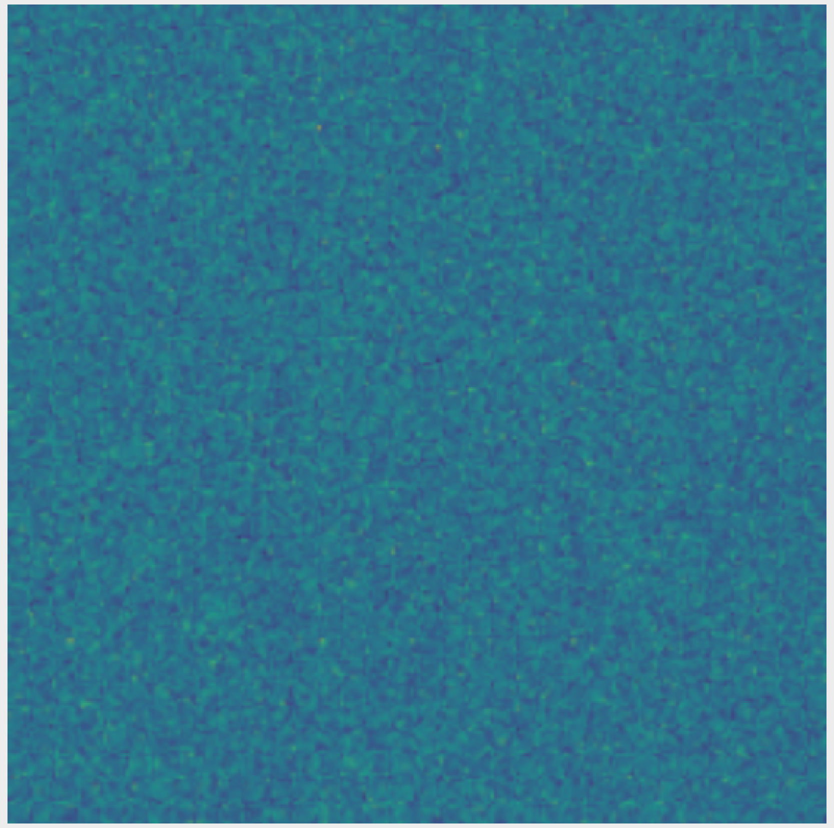
Ref : https://en.wikipedia.org/wiki/Effelsberg_100-m_Radio_Telescope

Class


Pulse

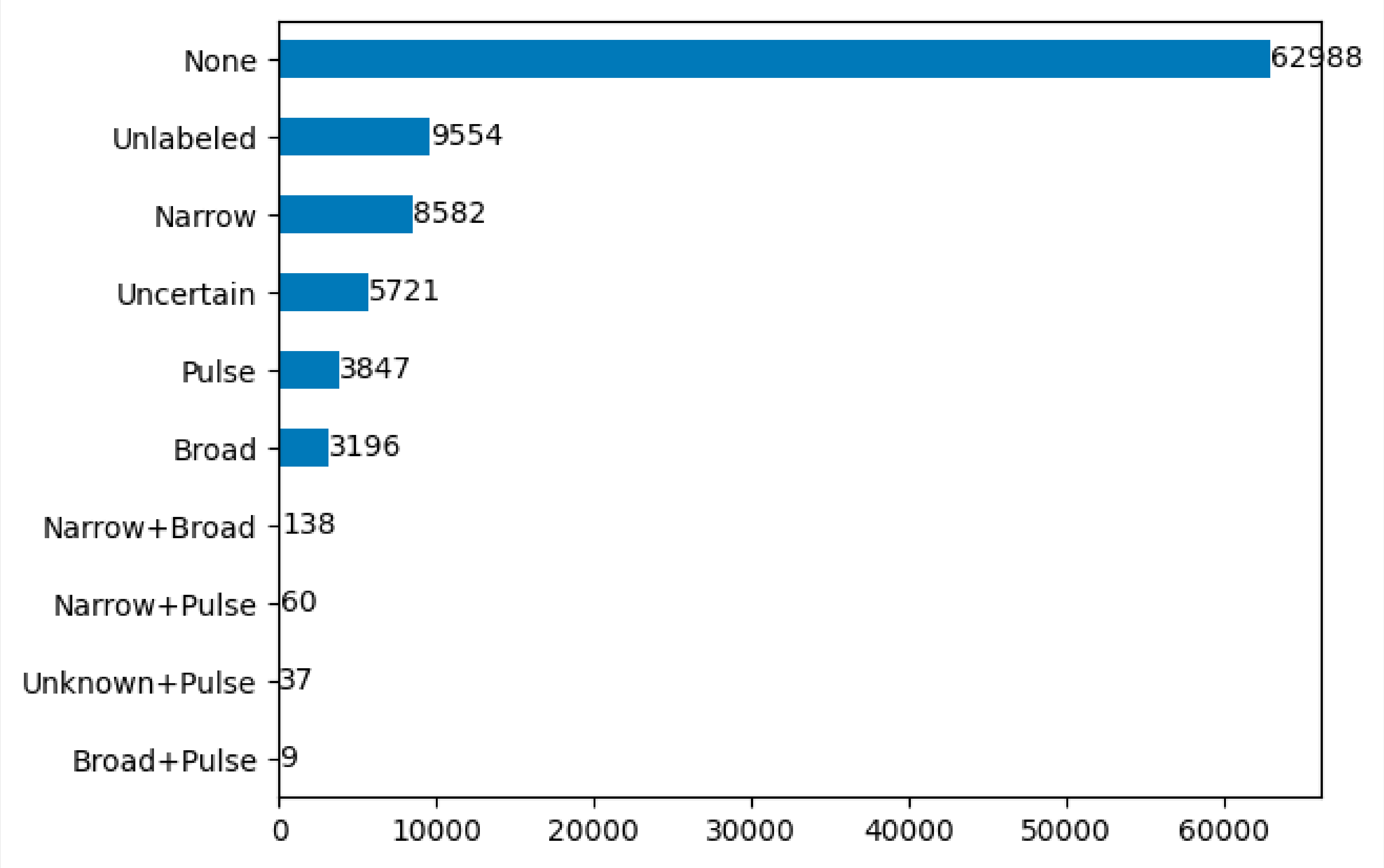

Boardband RFI


Narrowband RFI


None

Problem

Class imbalance



Preprocessing

Unbalance

Before

	path	idx	pulse	broad	narrow	label
0	/lustrefs/disk/project/lt900011-ai2310/dataset...	41	True	False	False	100
1	/lustrefs/disk/project/lt900011-ai2310/dataset...	232	True	False	False	100
2	/lustrefs/disk/project/lt900011-ai2310/dataset...	233	True	False	False	100
3	/lustrefs/disk/project/lt900011-ai2310/dataset...	244	True	False	False	100
4	/lustrefs/disk/project/lt900011-ai2310/dataset...	245	True	False	False	100
...	...	...	...	...	...	...
78852	/lustrefs/disk/project/lt900011-ai2310/dataset...	1019	False	False	False	000
78853	/lustrefs/disk/project/lt900011-ai2310/dataset...	1020	False	False	False	000
78854	/lustrefs/disk/project/lt900011-ai2310/dataset...	1021	False	False	False	000
78855	/lustrefs/disk/project/lt900011-ai2310/dataset...	1022	False	False	False	000
78856	/lustrefs/disk/project/lt900011-ai2310/dataset...	1023	False	False	False	000

78857 rows × 6 columns

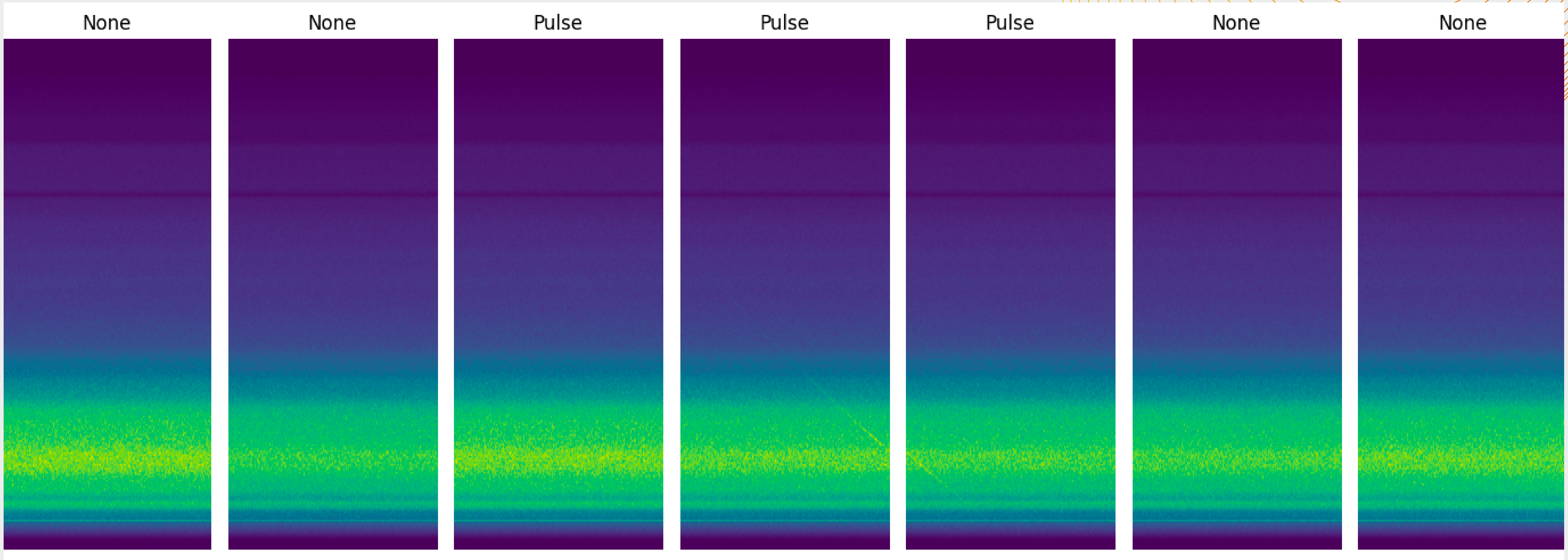
After

	path	idx	pulse	broad	narrow	label
0	/lustrefs/disk/project/lt900011-ai2310/dataset...	41	True	False	False	100
1	/lustrefs/disk/project/lt900011-ai2310/dataset...	232	True	False	False	100
2	/lustrefs/disk/project/lt900011-ai2310/dataset...	233	True	False	False	100
3	/lustrefs/disk/project/lt900011-ai2310/dataset...	244	True	False	False	100
4	/lustrefs/disk/project/lt900011-ai2310/dataset...	245	True	False	False	100
...	...	...	...	...	...	...
31733	/lustrefs/disk/project/lt900011-ai2310/dataset...	1014	True	False	False	100
31734	/lustrefs/disk/project/lt900011-ai2310/dataset...	1015	True	False	False	100
31735	/lustrefs/disk/project/lt900011-ai2310/dataset...	1016	True	False	False	100
31736	/lustrefs/disk/project/lt900011-ai2310/dataset...	1021	False	False	False	000
31737	/lustrefs/disk/project/lt900011-ai2310/dataset...	1023	False	False	False	000

31738 rows × 6 columns

Preprocessing

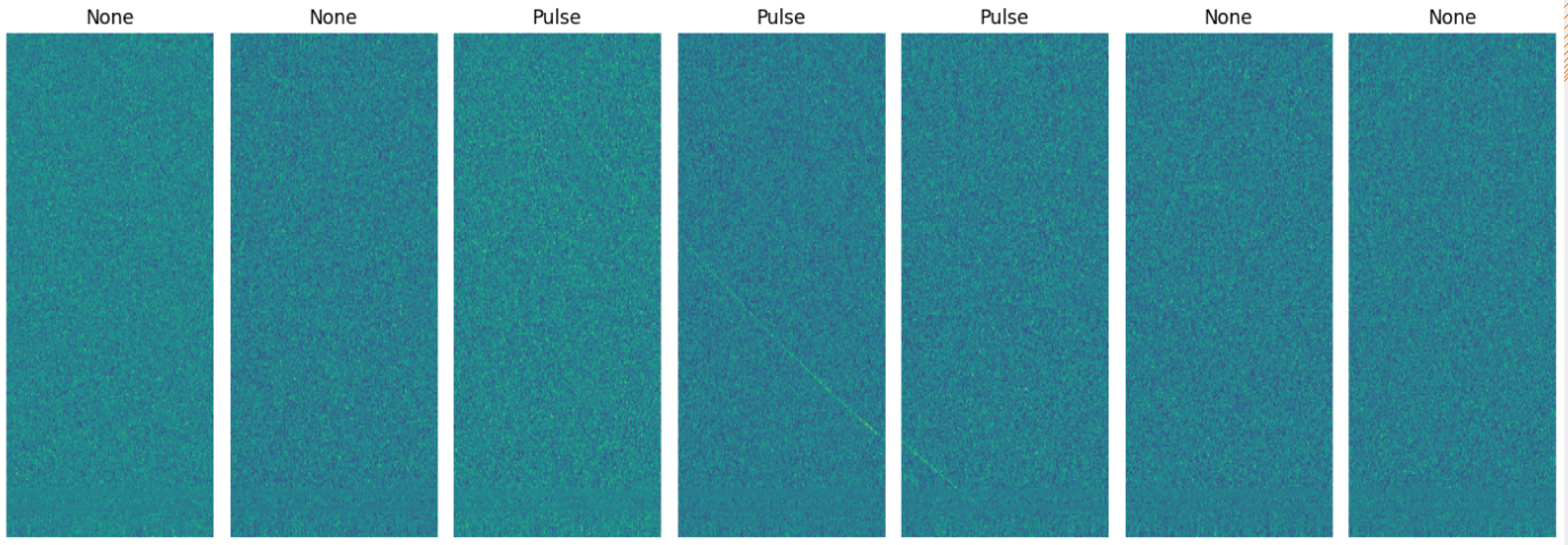
Training data



Preprocessing

Filter by mean of current signal

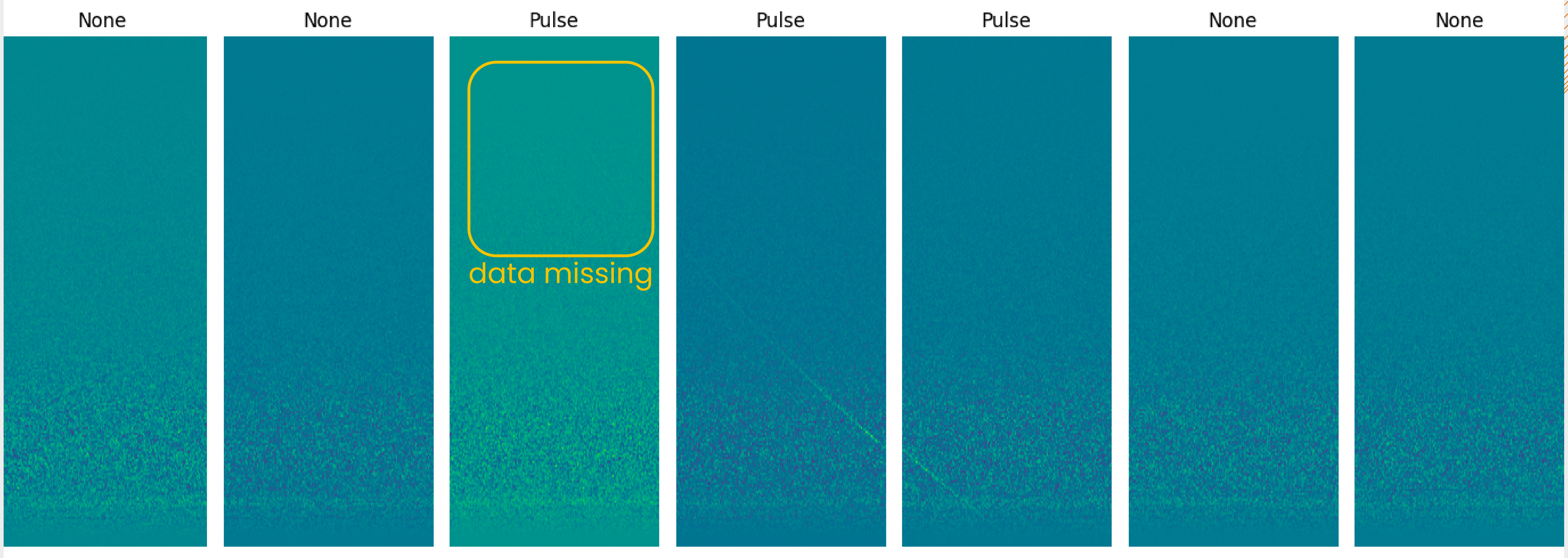
subSignal - subSignal.median(axis=0)



Preprocessing

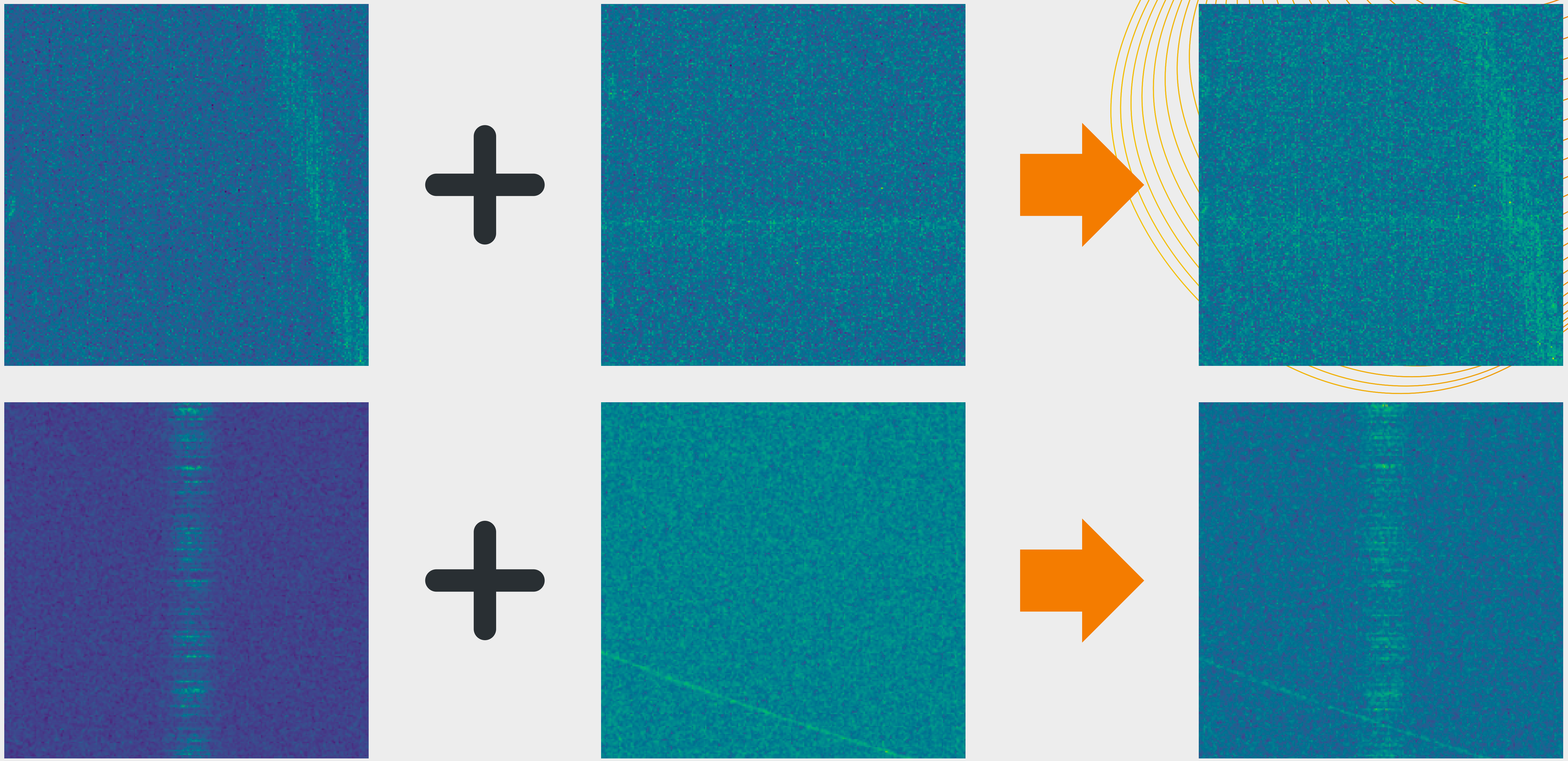
Filter by mean of all None signal

subSignal -noneSignals.median(axis=0)



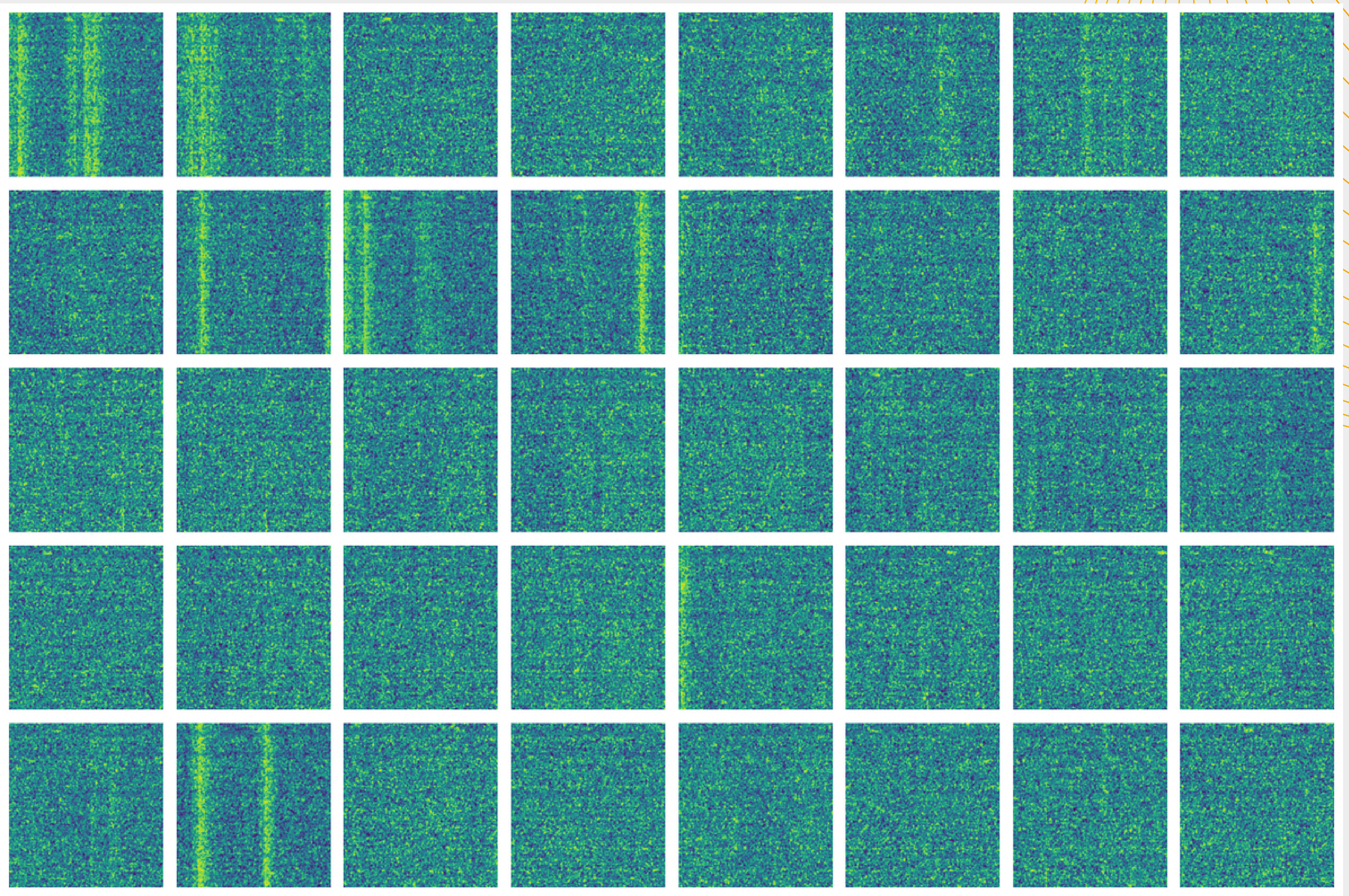
Preprocessing

Merging data



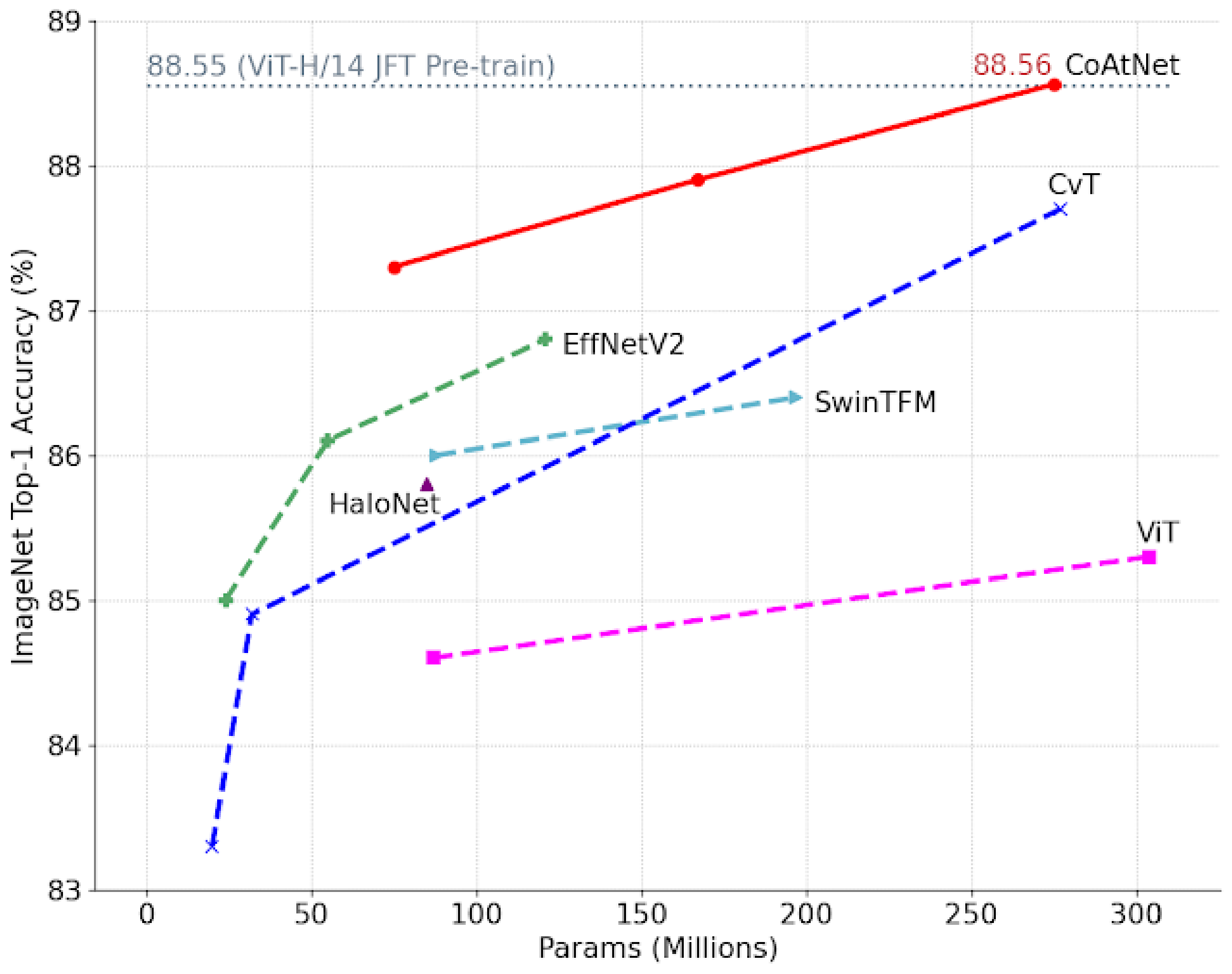
Preprocessing

normalize_data , gaussian_filter and enhance_features



Model

CoATNets_2_RW_224



CoAtNet: Fast and Accurate Models for Large-Scale Image Recognition

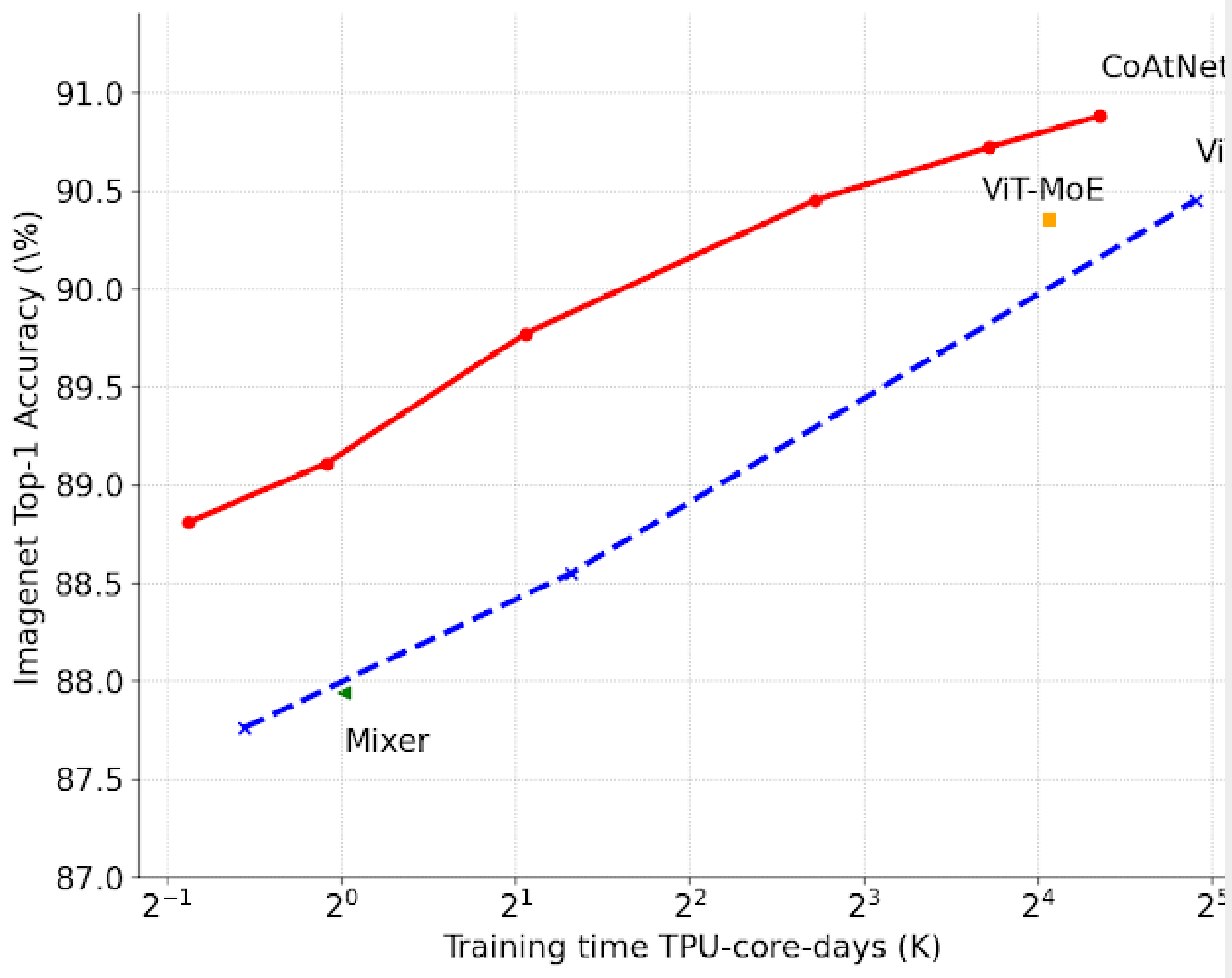
While EfficientNetV2 is still a typical convolutional neural network, recent studies on Vision Transformer (ViT) have shown that attention-based transformer models could perform better than convolutional neural networks on large-scale datasets like JFT-300M. Inspired by this observation, we further expand our study beyond convolutional neural networks with the aim of finding faster and more accurate vision models.

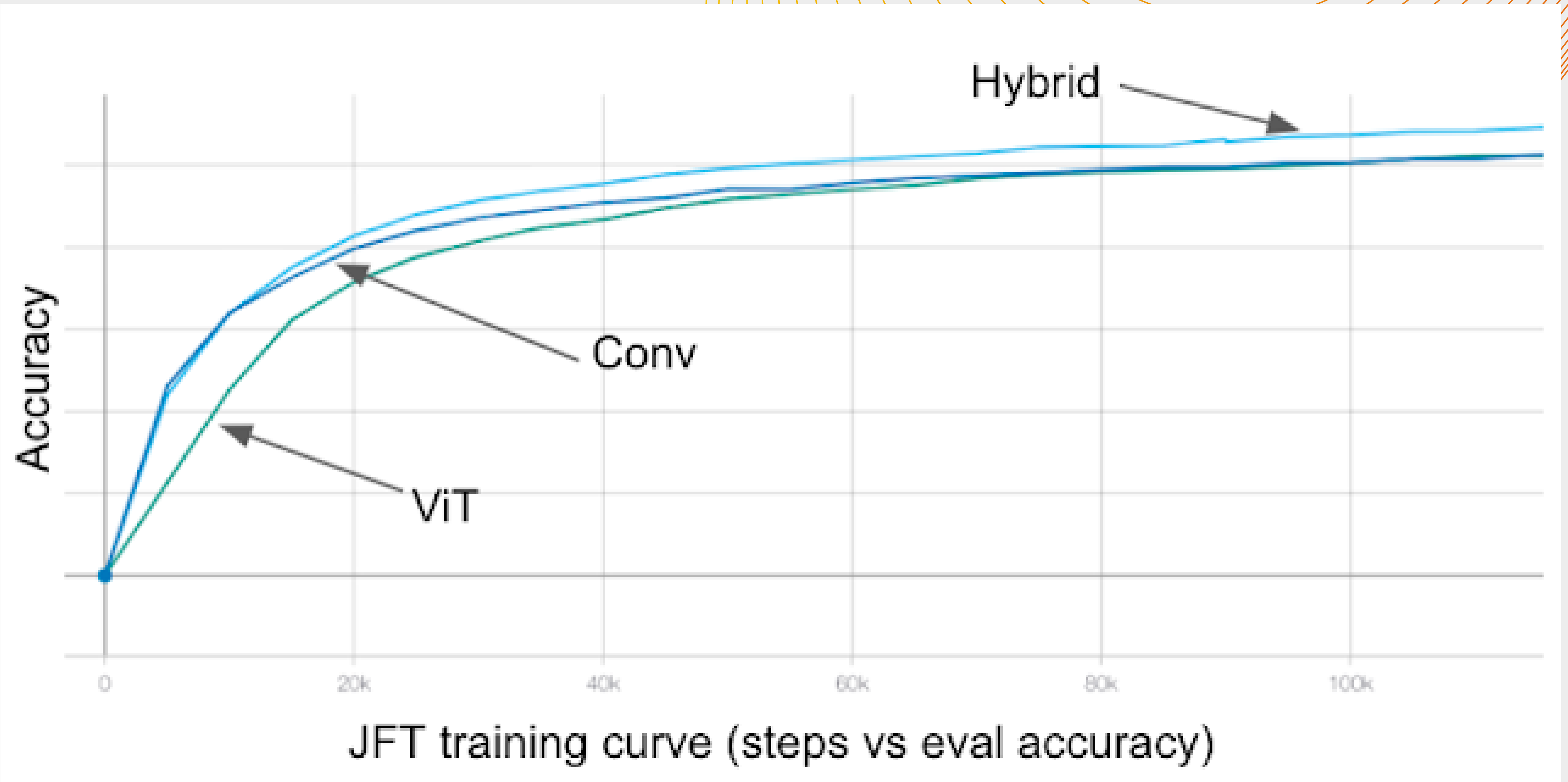
In “CoAtNet: Marrying Convolution and Attention for All Data Sizes”, we systematically study how to combine convolution and self-attention to develop fast and accurate neural networks for large-scale image recognition. Our work is based on an observation that convolution often has better generalization (i.e., the performance gap between training and evaluation) due to its inductive bias, while self-attention tends to have greater capacity (i.e., the ability to fit large-scale training data) thanks to its global receptive field. By combining convolution and self-attention, our hybrid models can achieve both better generalization and greater capacity.

https://ai.googleblog.com/2021/09/toward-fast-and-accurate-neural.html

Model

CoATNets_2_RW_224

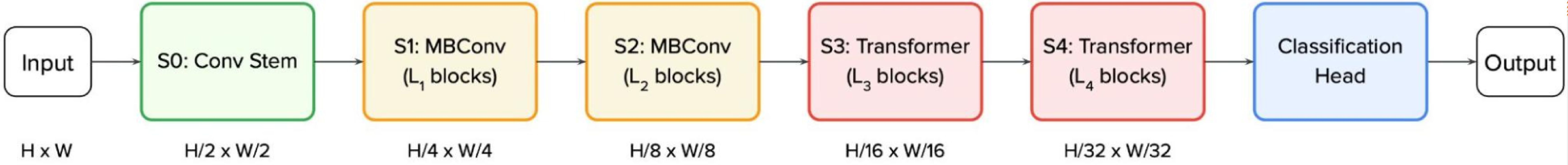




https://ai.googleblog.com/2021/09/toward-fast-and-accurate-neural.html

Model

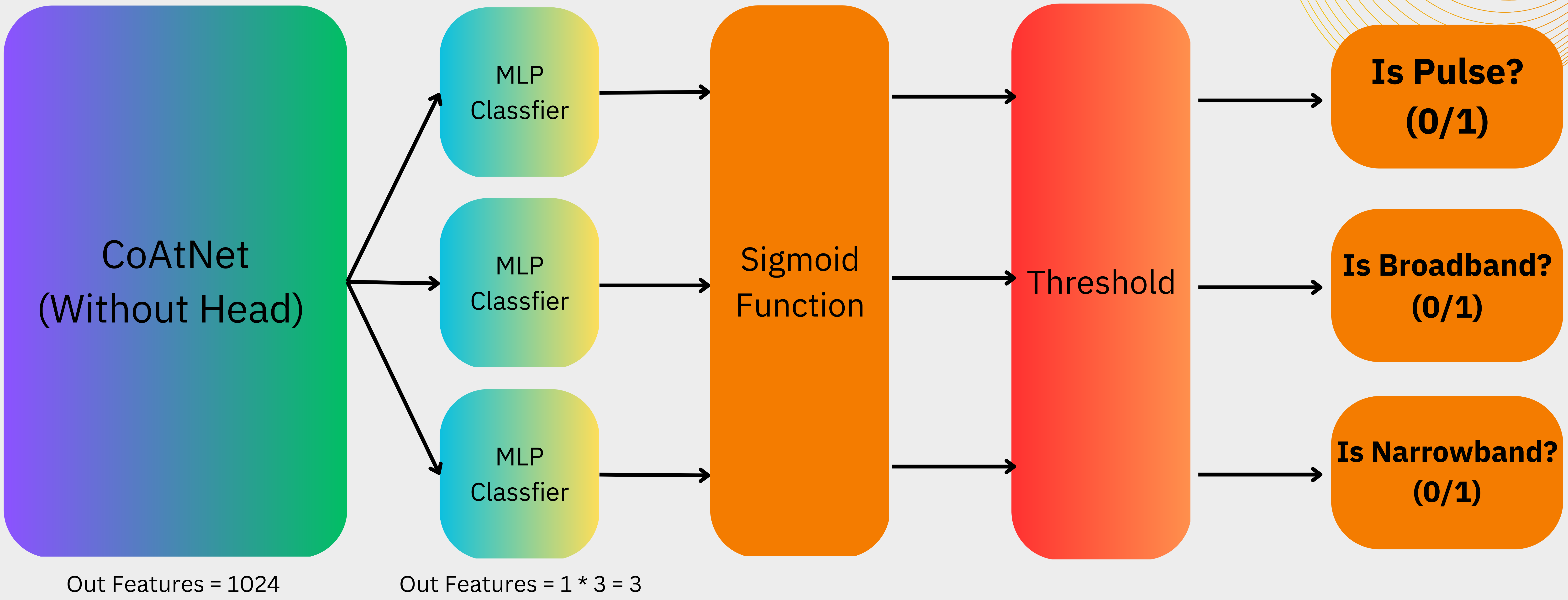
CoATNets_2_RW_224



*Overall CoAtNet architecture. Given an input image with size HxW, we first apply **convolutions** in the first stem stage (S0) and reduce the size to H/2 x W/2. The size continues to reduce with each stage. L_n refers to the number of layers. Then, the early two stages (S1 and S2) mainly adopt **MBConv** building blocks consisting of **depthwise convolution**. The later two stages (S3 and S4) mainly adopt **Transformer** blocks with relative **self-attention**. Unlike the previous Transformer blocks in **ViT**, here we use **pooling** between stages, similar to **Funnel Transformer**. Finally, we apply a classification head to generate class prediction.*

https://ai.googleblog.com/2021/09/toward-fast-and-accurate-neural.html

Model



https://ai.googleblog.com/2021/09/toward-fast-and-accurate-neural.html

Evaluation

Validation Metrics	Score
Accuracy	0.827
Pulse Recall	0.939
Pulse Recall * Accuracy	0.777
Time	0.0397s.

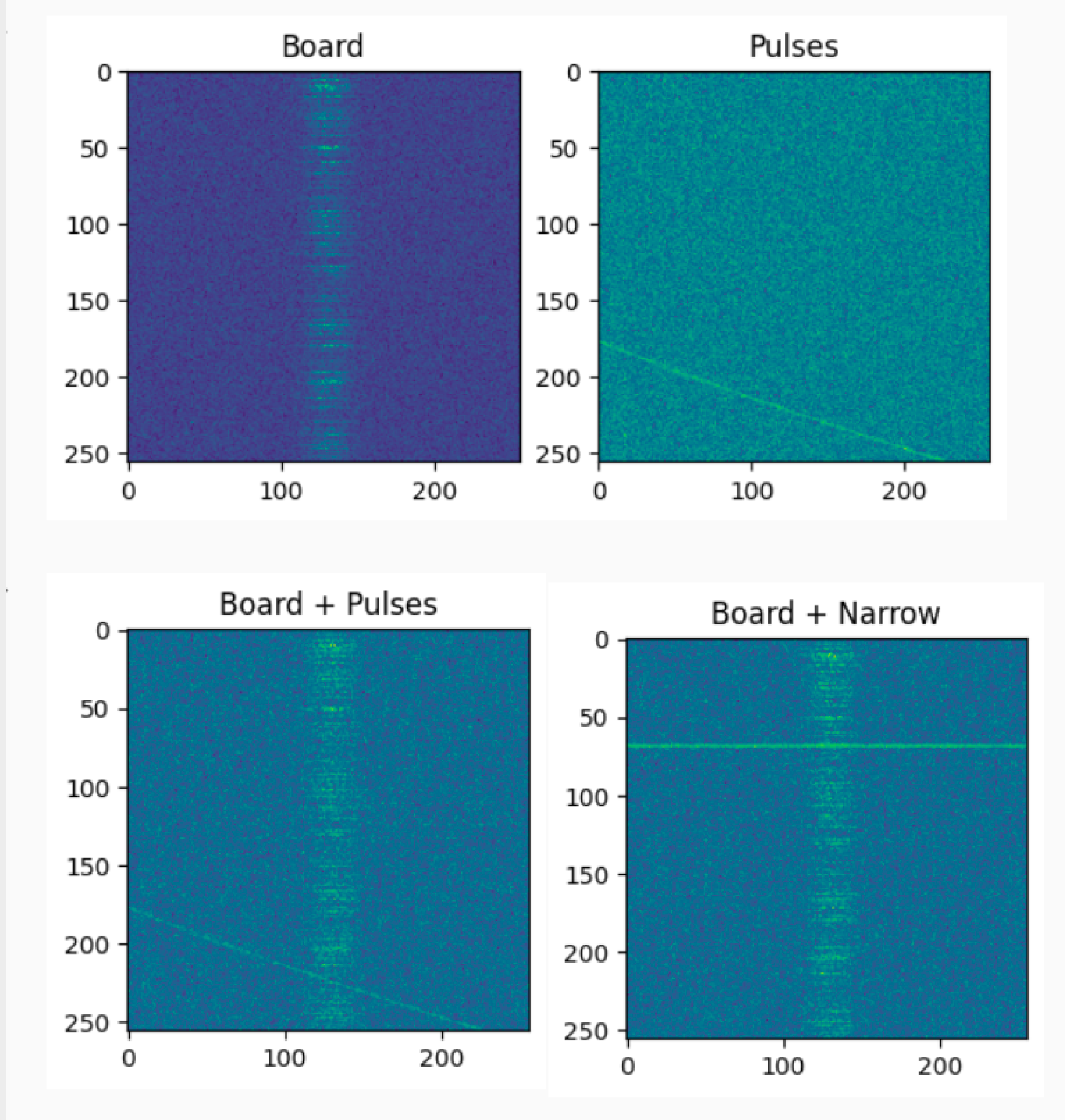
CoatNet_2_RW_224

Summary

- **Downsampling** to balance data
- **CoATNet Model** High Accuracy High Speed
- Uses **Dynamic Threshold**:
  - Find suitable threshold for each class
  - But threshold overfitted on validation

Future work

- Increase Recall for Pulse class
- Training data for mix class Merging data   Narrow+Broad, Narrow+Pulse, Broad+Pulse for training.
- Making simulation for API submission.
- Remove thresholding by outputting multiple neurons.
- Improve performance by using TensorRT



Thank You

EXP TEAM