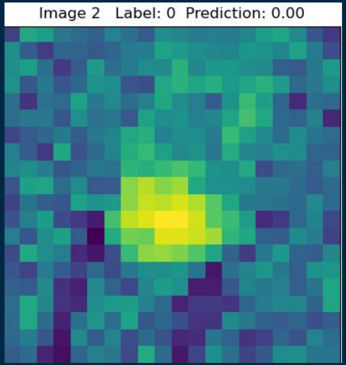
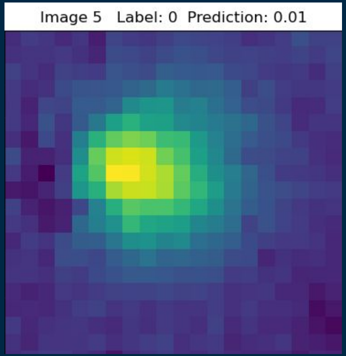
# Supernova Hunters

Chris Chang

# Goal

Improve (decrease) the Missed Detection Rate of supernovae while keeping a False Positive Rate of 1%



# Previous Models

## Flat

Had a ~21% MDR for the PS1 Medium Deep Survey, but didn't generalize well to PSST data because more artifacts were present in the dataset

## Convolutional Neural Net

Has a ~21% MDR on the PSST dataset @ 1% FPR and 5.2% MDR @ 5% FPR

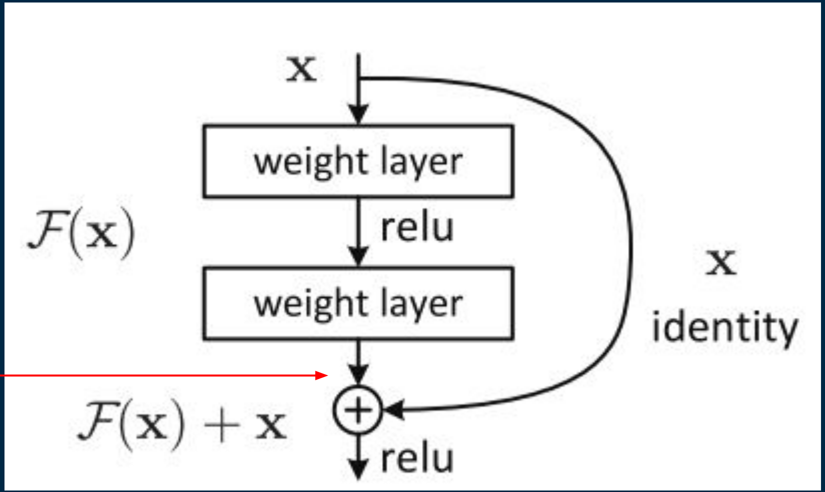
# ResNet Architecture

## Advantages

Can be large without overfitting

## How?

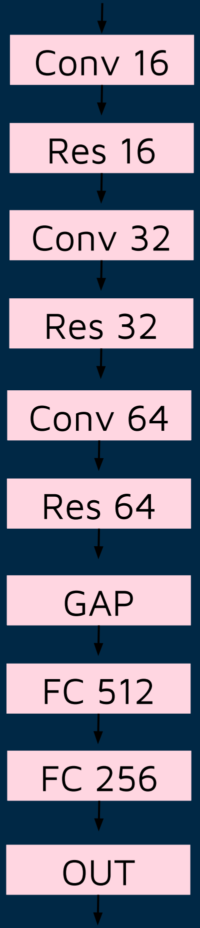
A res block can easily learn the identity function



# ResNet Architecture

## Alternating blocks

- Conv to increase channels, then Res
- Might not be the proper implementation



# Datasets

Same images

### 3π 50x50 3ch

~10,000 samples
Expert labels

### 3π 20x20 1ch

~10,000 samples
Expert labels

### Psst 20x20

~300,000 samples
? labels on train
Expert on test

# Models

### ResNet (50x50)

(Outlined in previous slides)

### CNN (Baseline)

Original CNN model but input layer is 3 channel now

### ResNet20

Architecture is identical to ResNet but input layer is 20x20x1

### Flat (Baseline)

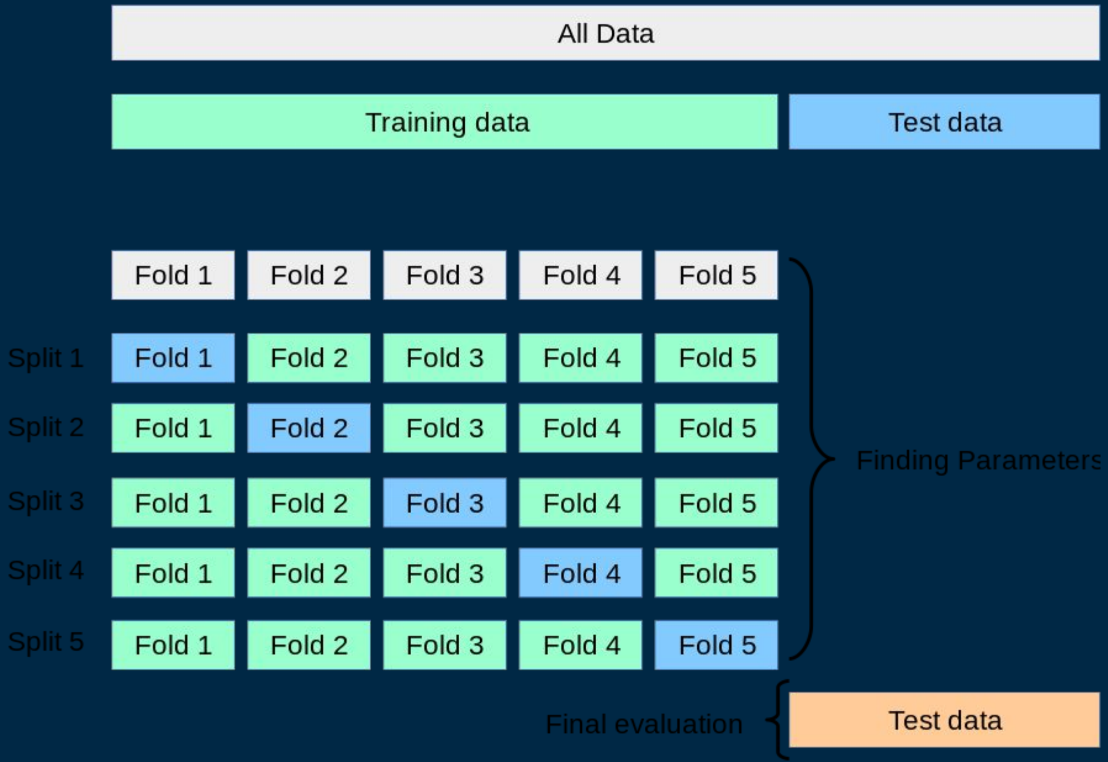
3 FC layers of size 500, 300, 10

# Data methods

## K-fold cross validation

K models are trained on (1-1/k)% of the training data

Predictions are averaged (Ensembling)



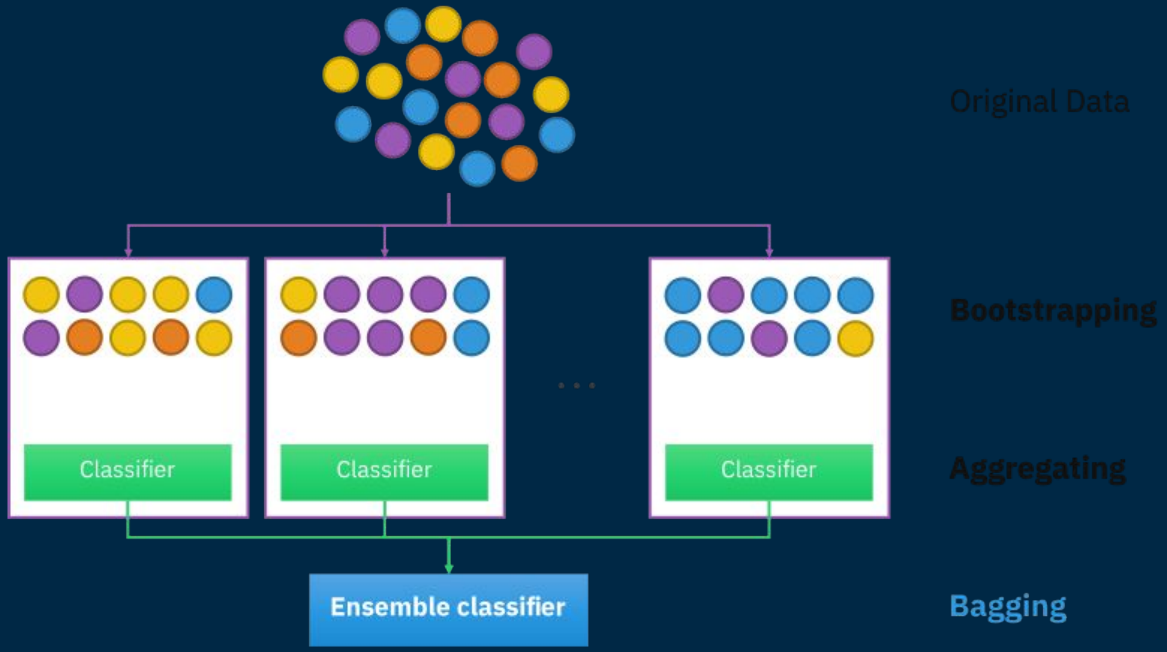
# Data methods

## Bootstrap Aggregation (Bagging)

K-bags are created

Each bag is composed of m training examples Which contains ~0.66m unique training examples

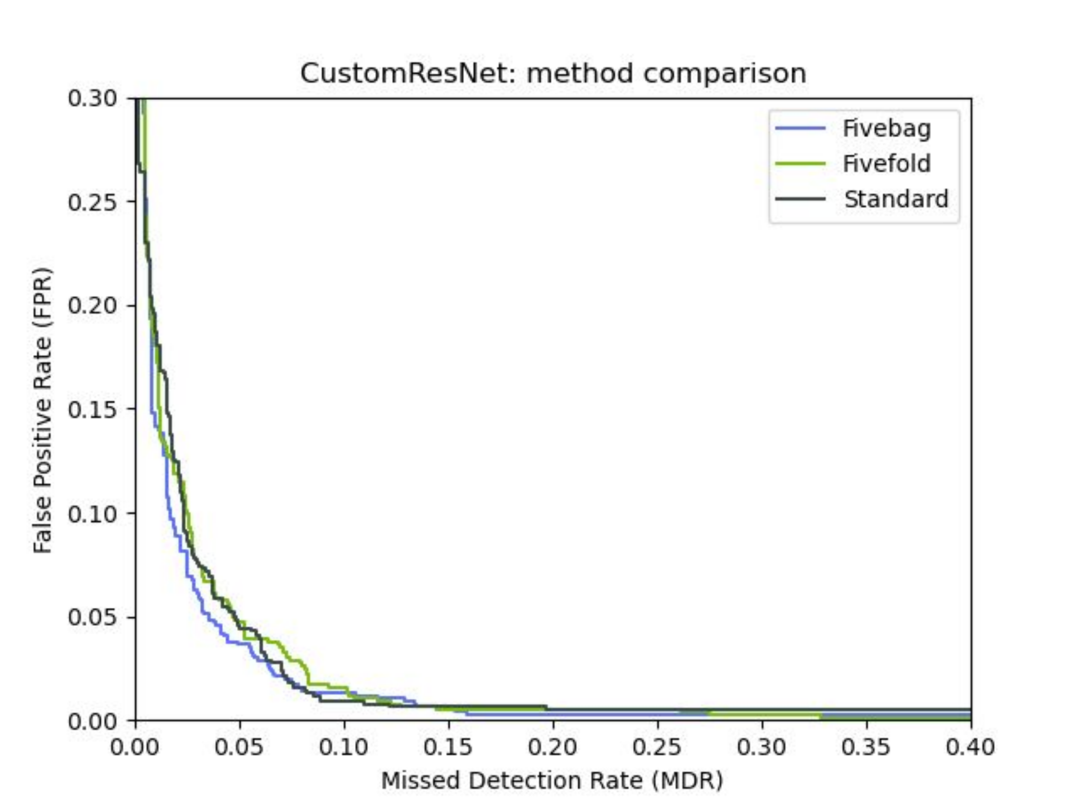
Ensembling is used



# Performance

ResNet
3 channel

5-bag is best

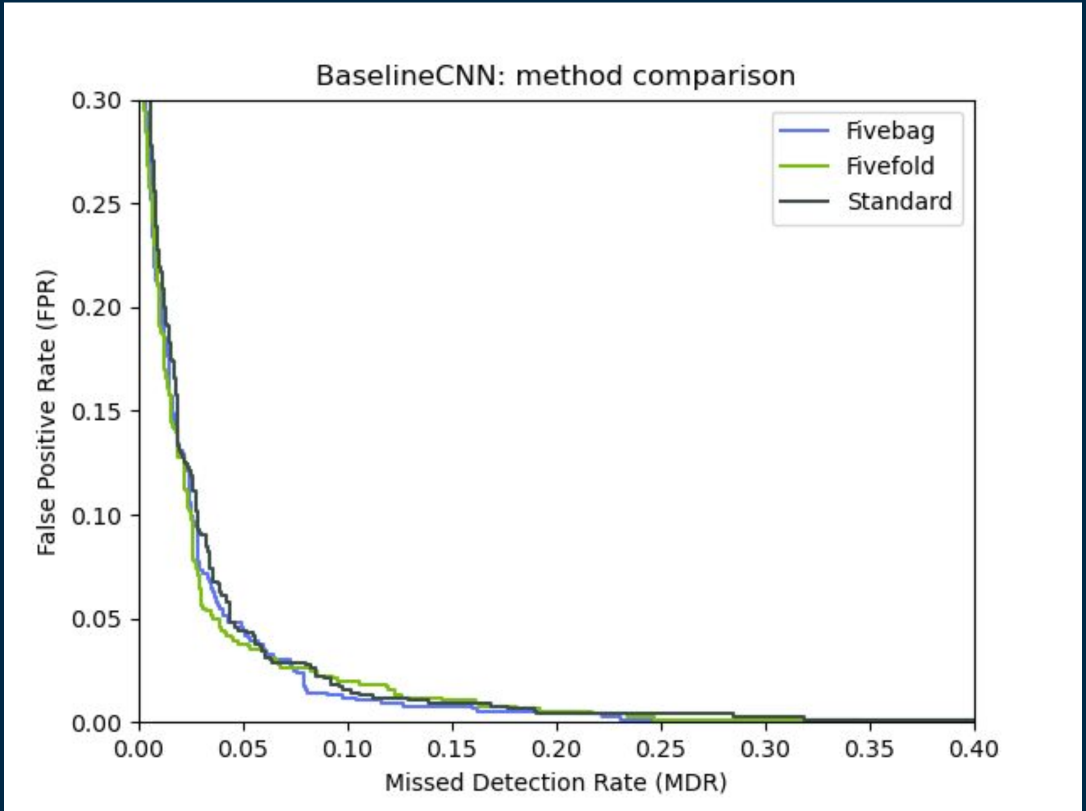


# Performance

## CNN (Baseline)

3 channel

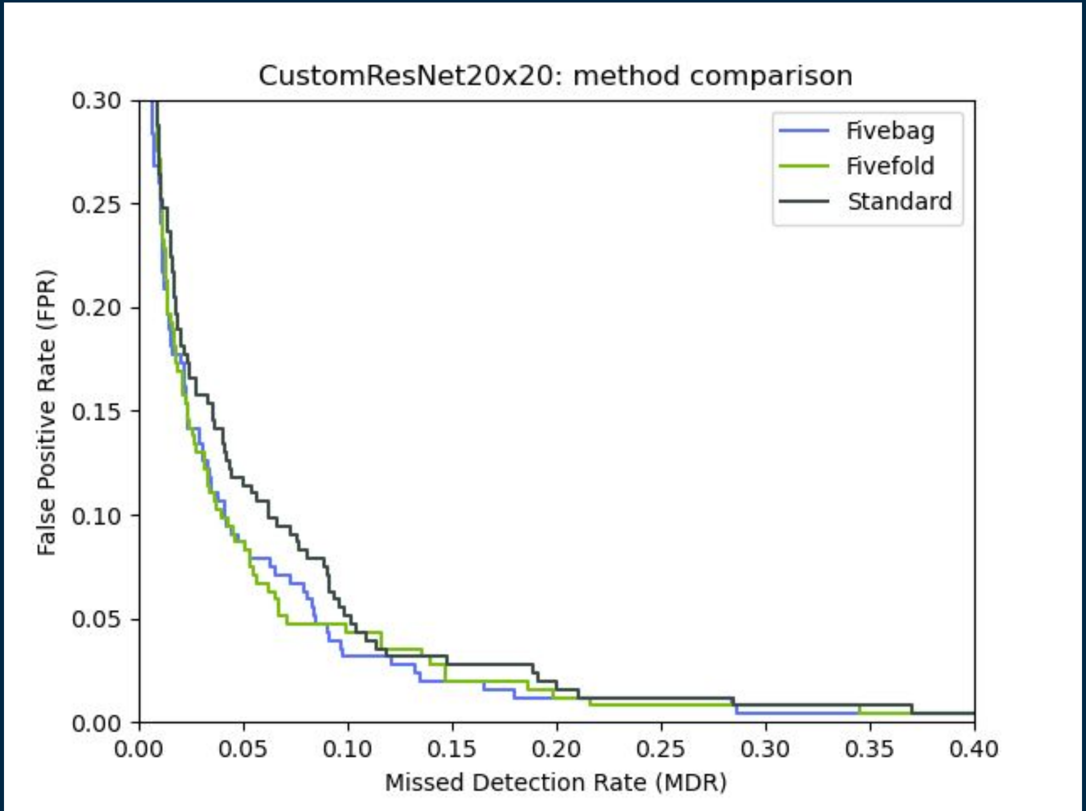
5-bag is best



# Performance

## ResNet20
1 channel

5-bag is best

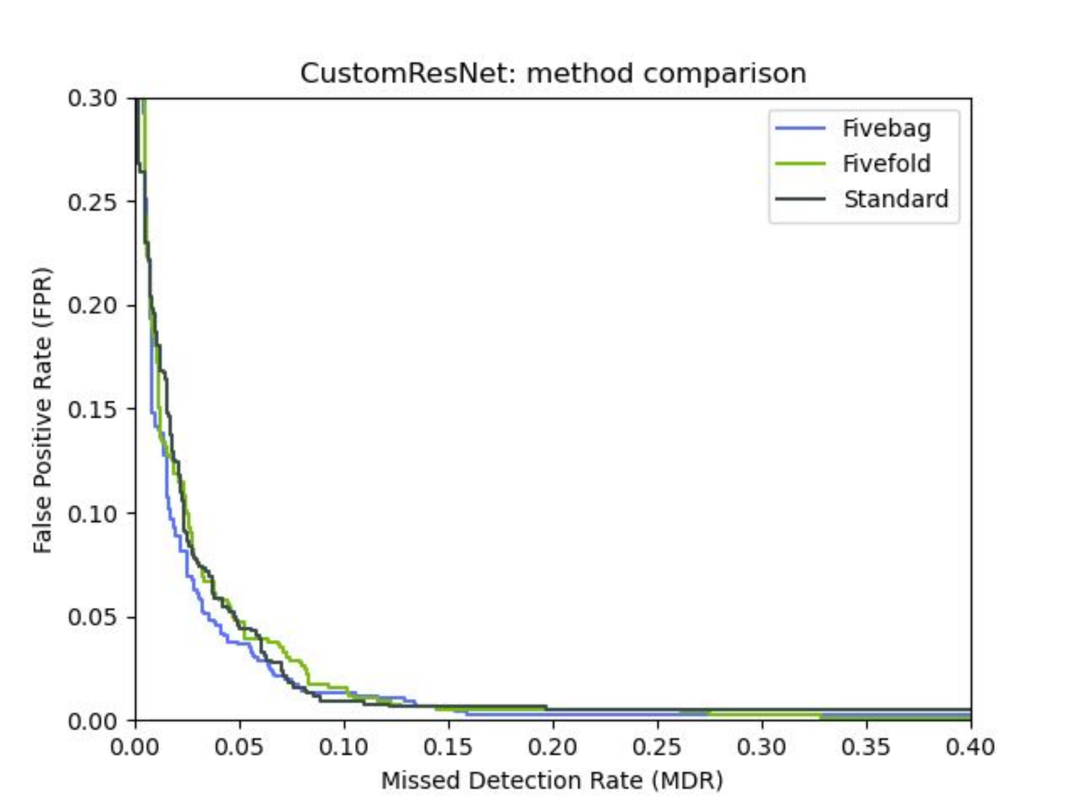


# Performance

## Flat (Baseline)

1 channel

5-fold is best



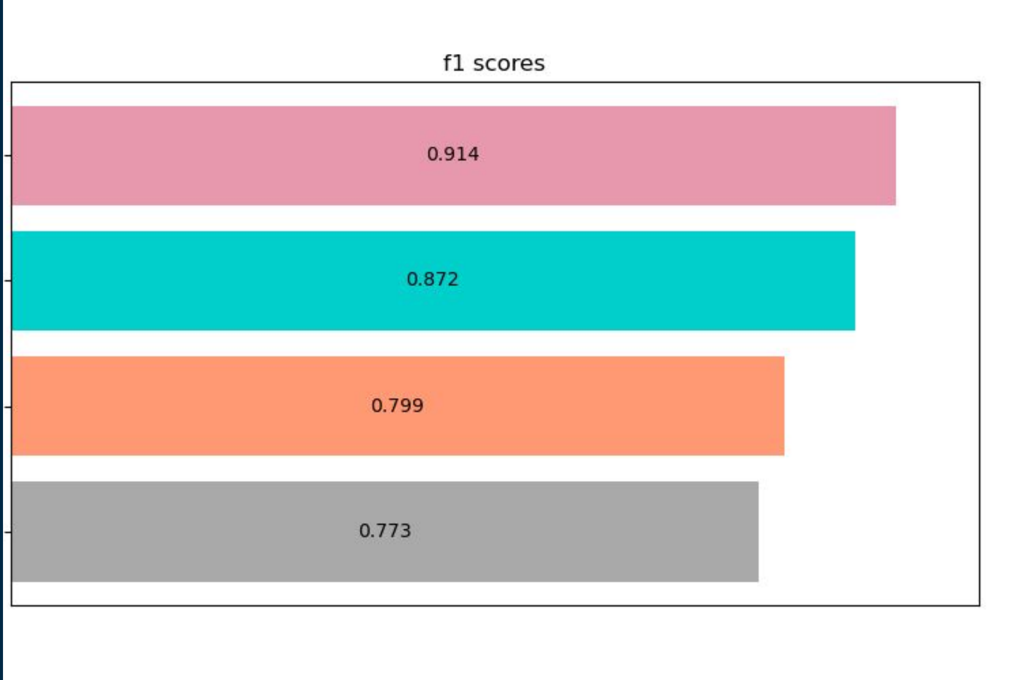
# Performance

(50x50) ResNet
14%

CNN (Baseline)
21%

*ResNet20
26%

Flat (Baseline)
30%



# New Material

# Datasets

## 3π data

~10,000 samples
Expert labels
100x100x3

## Psst Large

~300,000 samples
? labels on train
Test set is 3π data
20x20x1

3π data was rescaled for
50x50x3 and 20x20x1 models

# Models

### ResNet50x50
(Outlined in previous slides)

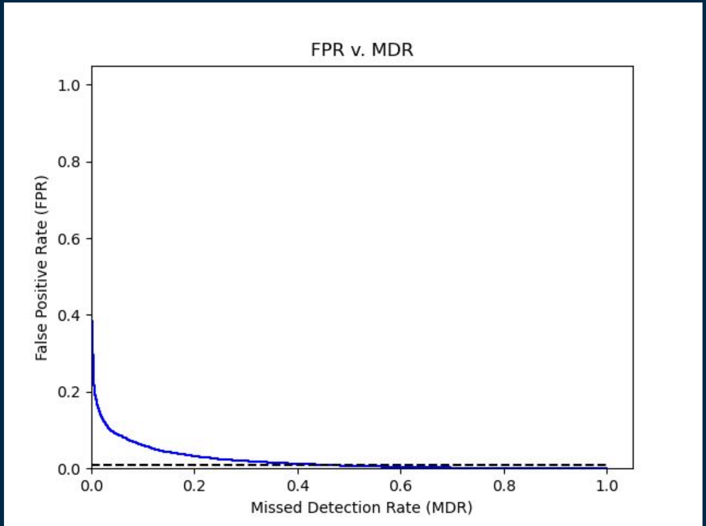
### ResNet100x100
Architecture is identical to ResNet but input layer is 100x100x3

### ResNet20x20
Architecture is identical to ResNet but input layer is 20x20x1

### Flat (Baseline)
3 FC layers of size 500, 300, 10

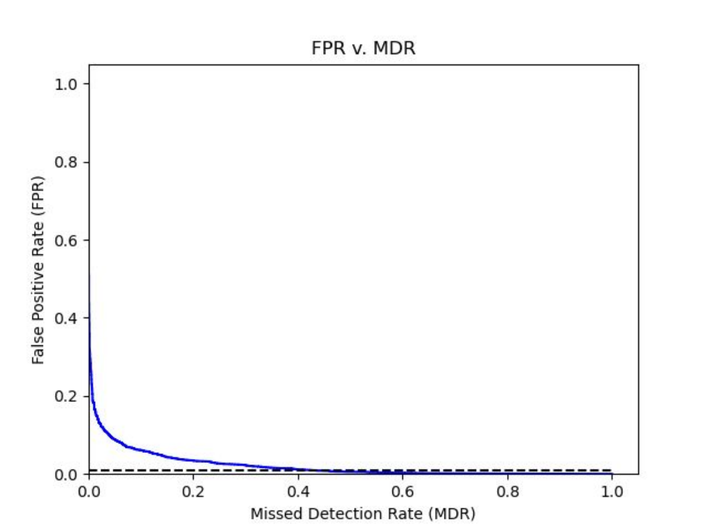
# Psst Large dataset (Not used due to poor performance)

## Method 1: full dataset




MDR 44%

## Method 2: half real detections



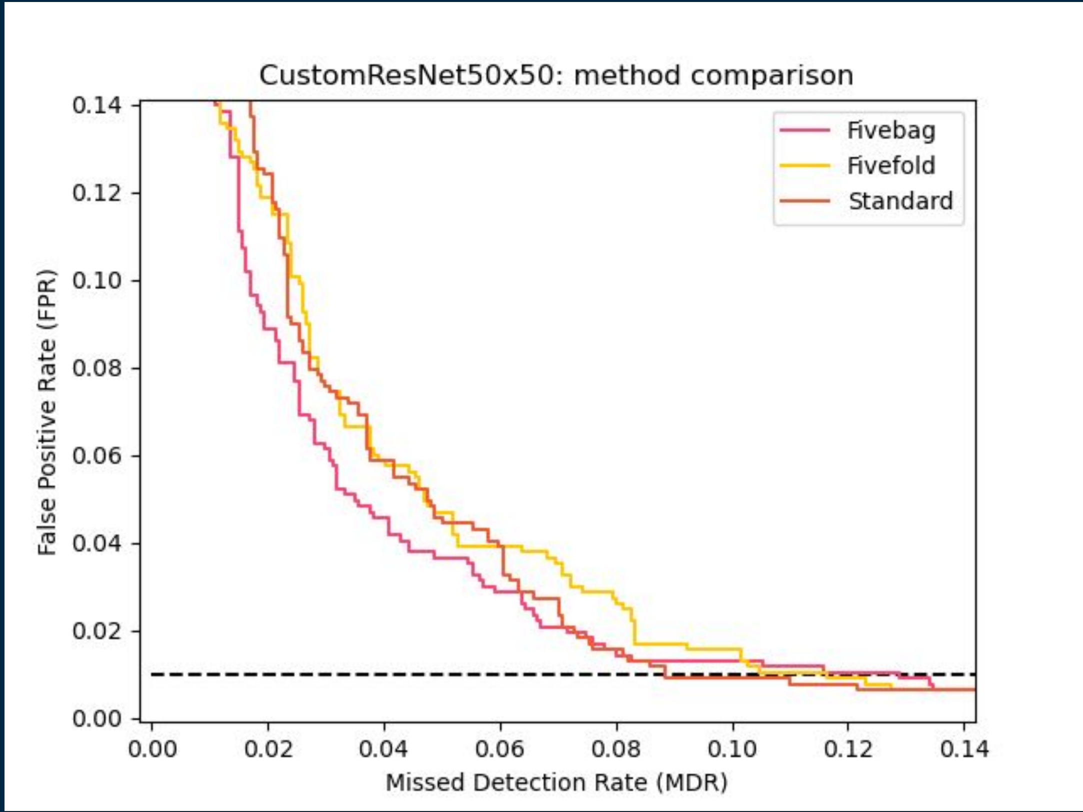

MDR 42%

# Performance

## ResNet 50x50

3 channel

5-bag is best

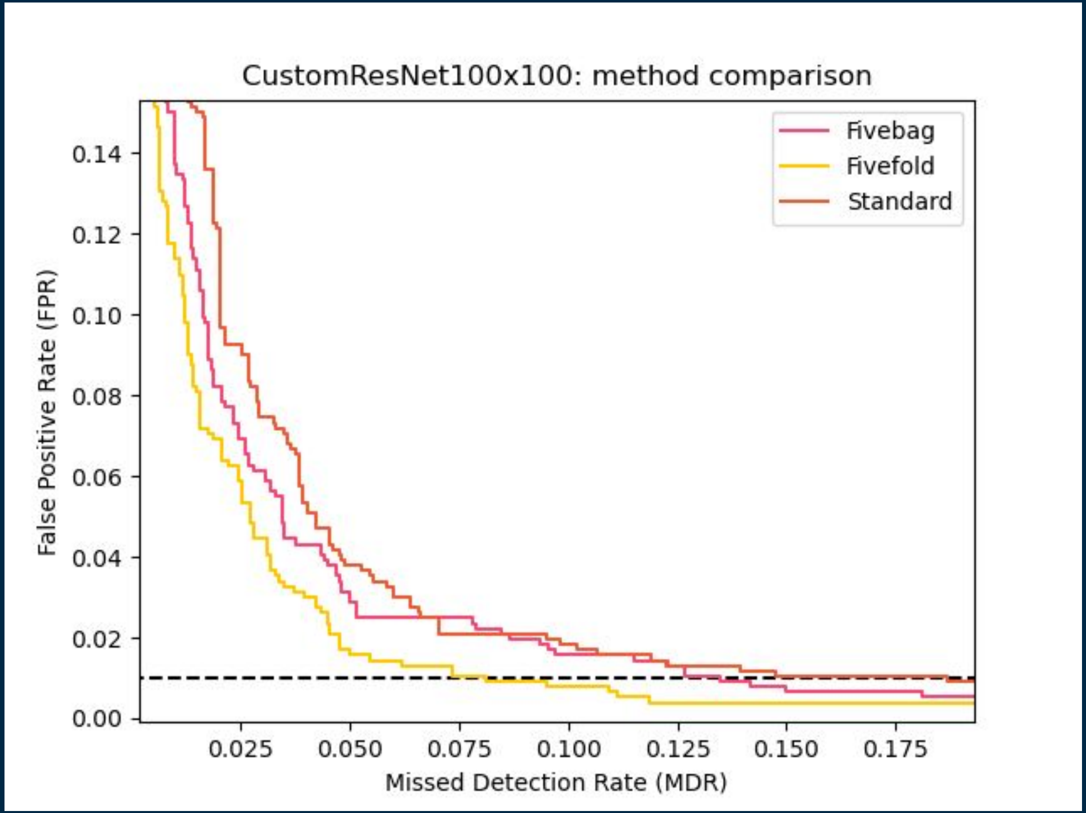


# Performance

## ResNet 100x100

3 channel

5-fold is best

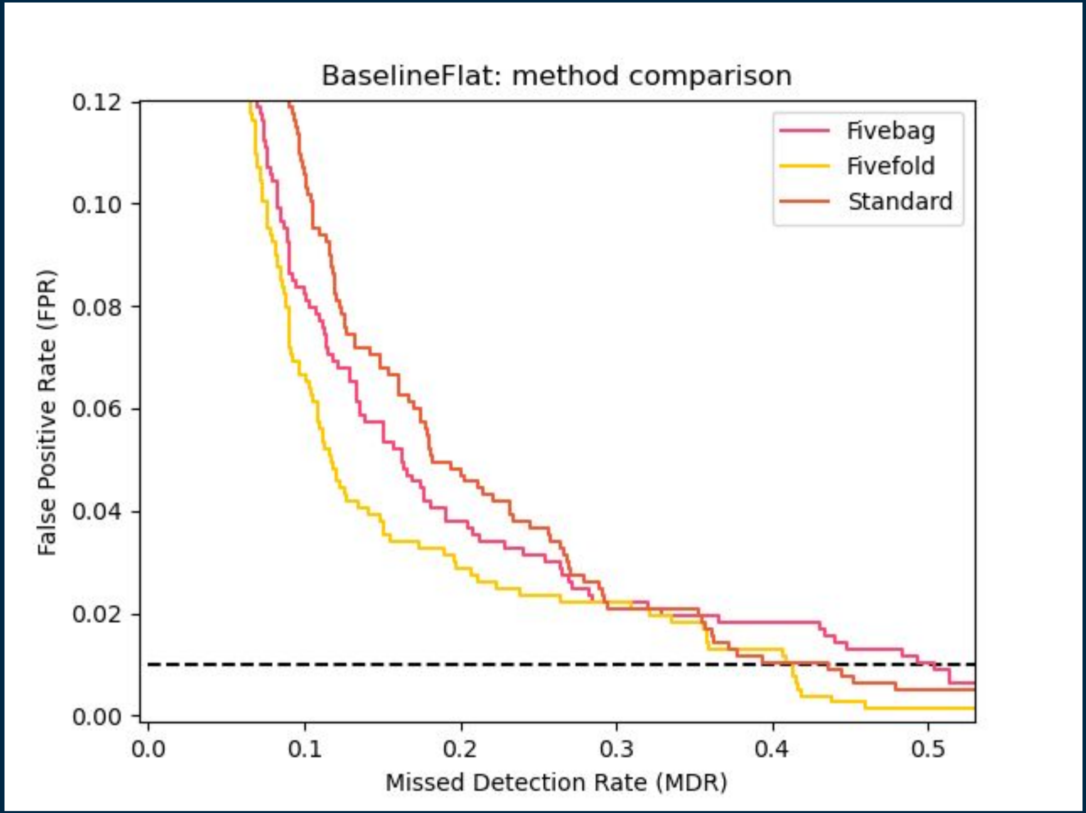


# Performance

## Flat (Baseline)

1 channel

5-fold is best



# Performance

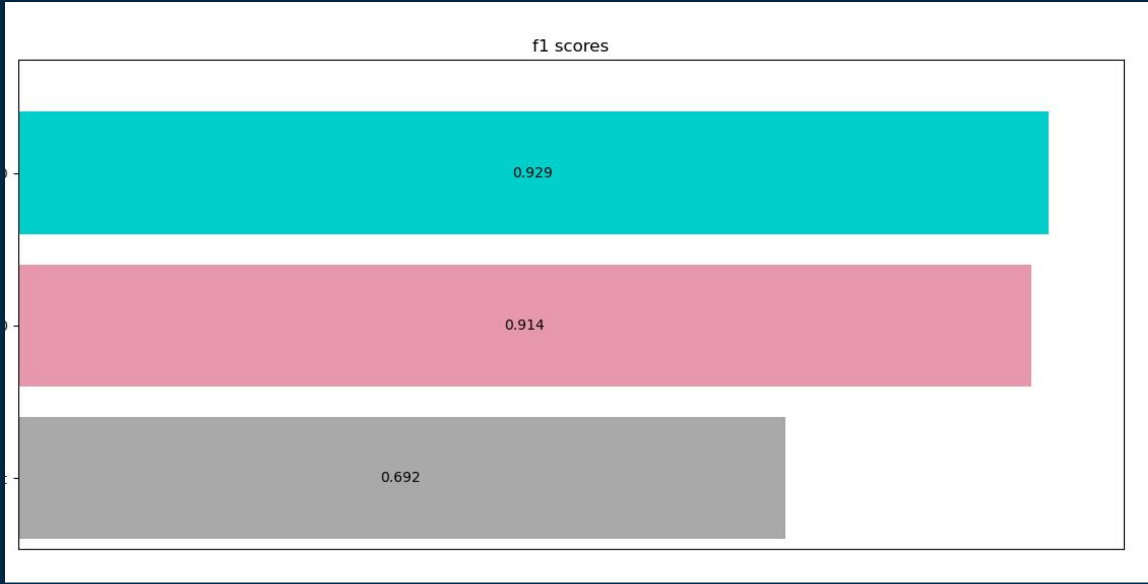
| Model | method | MDR |
|---|---|---|
| ResNet100x100 (120 epochs) | Standard | 19 |
| | **Five-fold** | **11** |
| | Five-bag | 14 |
| ResNet50x50 (35 epochs) | Standard | 19 |
| | Five-fold | 18 |
| | **Five-bag** | **14** |
| Flat (Baseline) (75 epochs) | Standard | 52 |
| | Five-fold | 46 |
| | **Five-bag** | **42** |

# Best Performance

(100x100) ResNet
11%

(50x50) ResNet
14%

Flat (Baseline)
42%

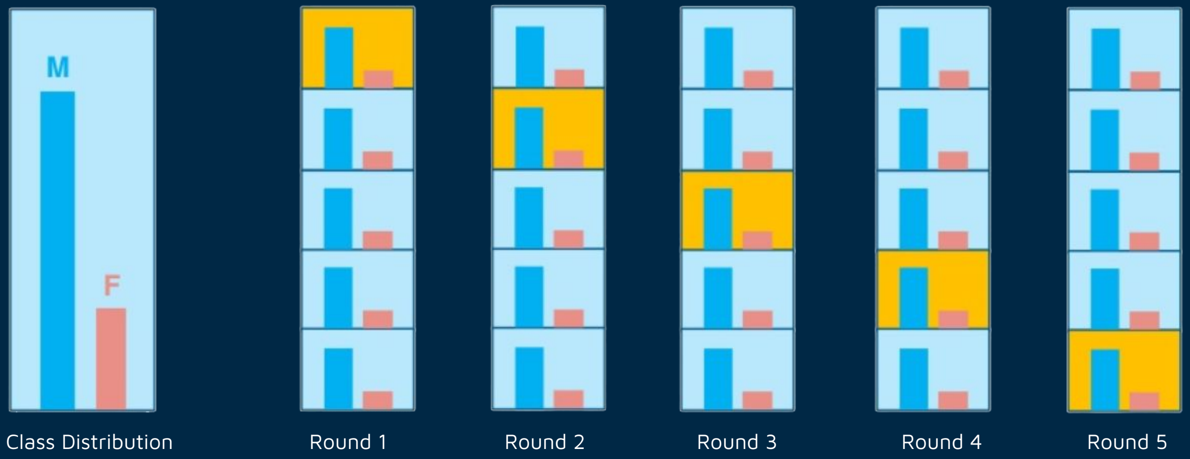


# Other

## *Stratified* K-fold Cross validation

Each fold is guaranteed a consistent proportion of each class

Helps with stability




Was out performed by basic K-fold in my tests, however this method may be better on average across many test runs. Worth more experimentation

# Other

## Soft f1 loss

Loss function is the "f1 score" without thresholding

Ex: For image i, a prediction 0.6 and a label 1.0 yields +0.6 to True Positives

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{loss} = 2 * \frac{(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})}$$

Ideally this causes training to optimize for f1 score. Was outperformed by Binary Cross Entropy and took much longer to train. Not worth using.

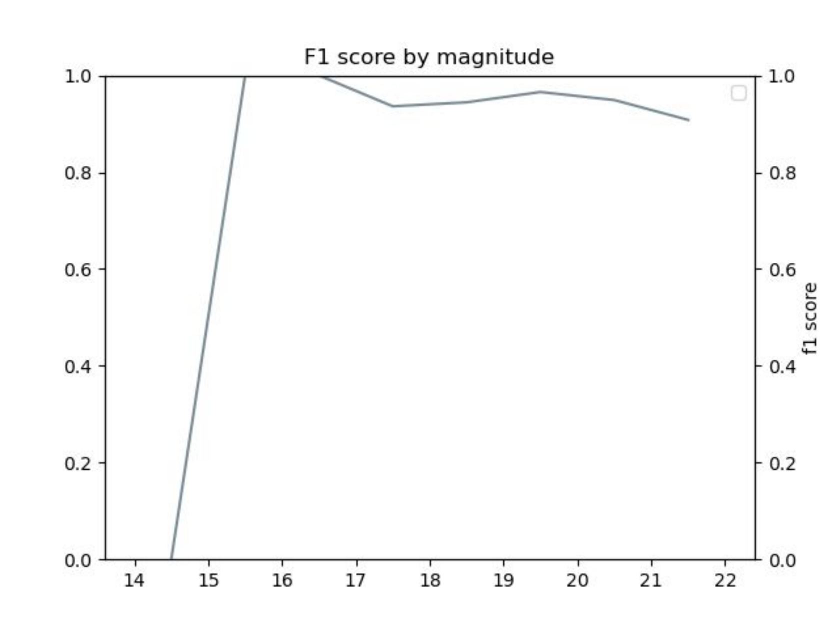
# Other

## Magnitude

F1 score tends to drop off slightly as magnitude increases

Very low sample size for low magnitudes

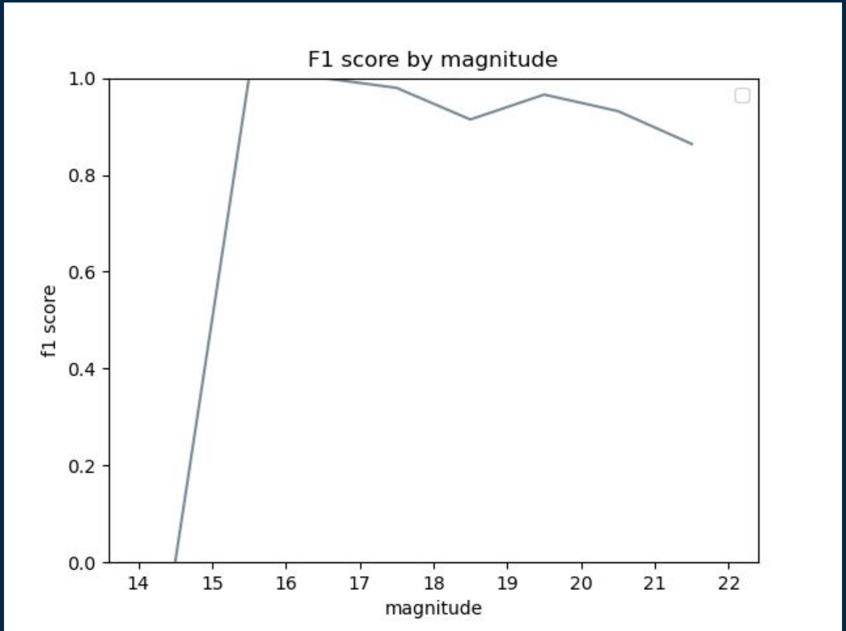
Larger magnitude usually means more noise




Earlier I had considered splitting data by magnitude or using magnitude as an input into one of the model's FC layers. 100x100 model seems to handle magnitude better than 50x50, so this might not be worth pursuing anymore

# Other

## Magnitude




(50x50) plot for comparison

# Future

## Data augmentation

Upscale/Downscale

Adding noise

Use model to generate labels for psst then train on labeled psst

Expert labeled dataset is ~10,000 examples which is fairly large. Model may benefit from training on more data w/o labelling more by hand.

# Notes for reproduction

## Datasets

I re-formatted data for easier use

Located on github in /datasets/
https://github.com/ctchang-png/Supernova-Hunters

3pi_20x20_small.mat
3pi_100x100_channels3_formatted.mat
3pi_50x50_channels3_formatted.mat

3pi_20x20_xpert_testset.mat
Test set is 3pi_20x20_small

# Notes for reproduction

## Training/saving

Ensembles are saved to /saved_models/

If satisfied, copy /saved_models/ to /Ensembles/ and name folder {architecture}_{data_method}_MDR{mdr}

Tensorboard was used to watch metrics after training, logs must be cleared then sftp'd from msi to view b/c tensorboard does localhosting

I realize these things are sort of finicky so if you'd like me to change them during the next week or so I will