APPLIED MACHINE LEARNING

ICEBERG CLASSIFICATION

DANA LÜDEMANN
ANNA PUGGAARD
ELLOISE FANGEL-LLOYD
HELENE PEHRSSON

Outline

- Introduction
- Data
- Chosen Models (Training)
- Comparison (Testing)
- Conclusion

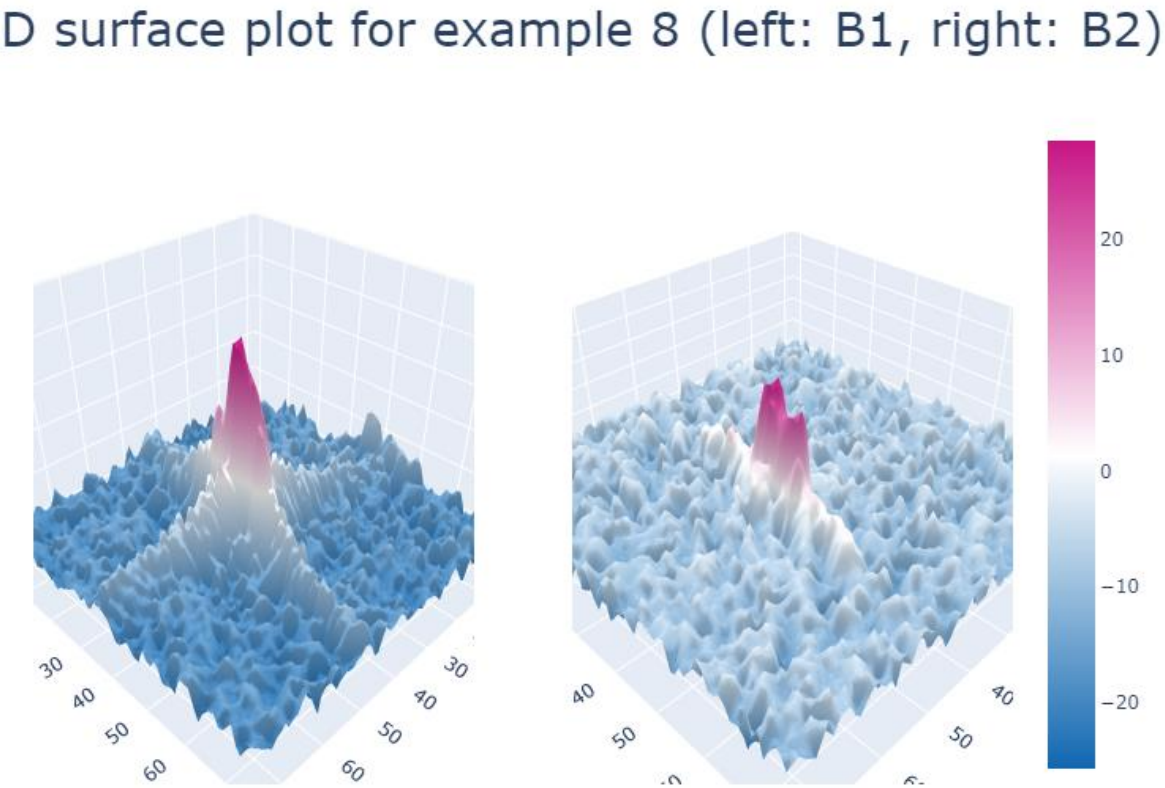
Introduction

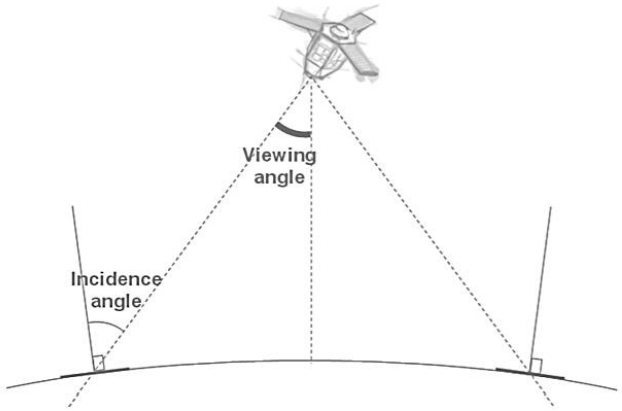
- Kaggle competition: Statoil/C-CORE Iceberg Classifier Challenge
- Motivation: aid navigation and asses risks from icebergs in remote offshore areas

Goal: identifying if a remotely sensed target is a ship or iceberg using ML

Data

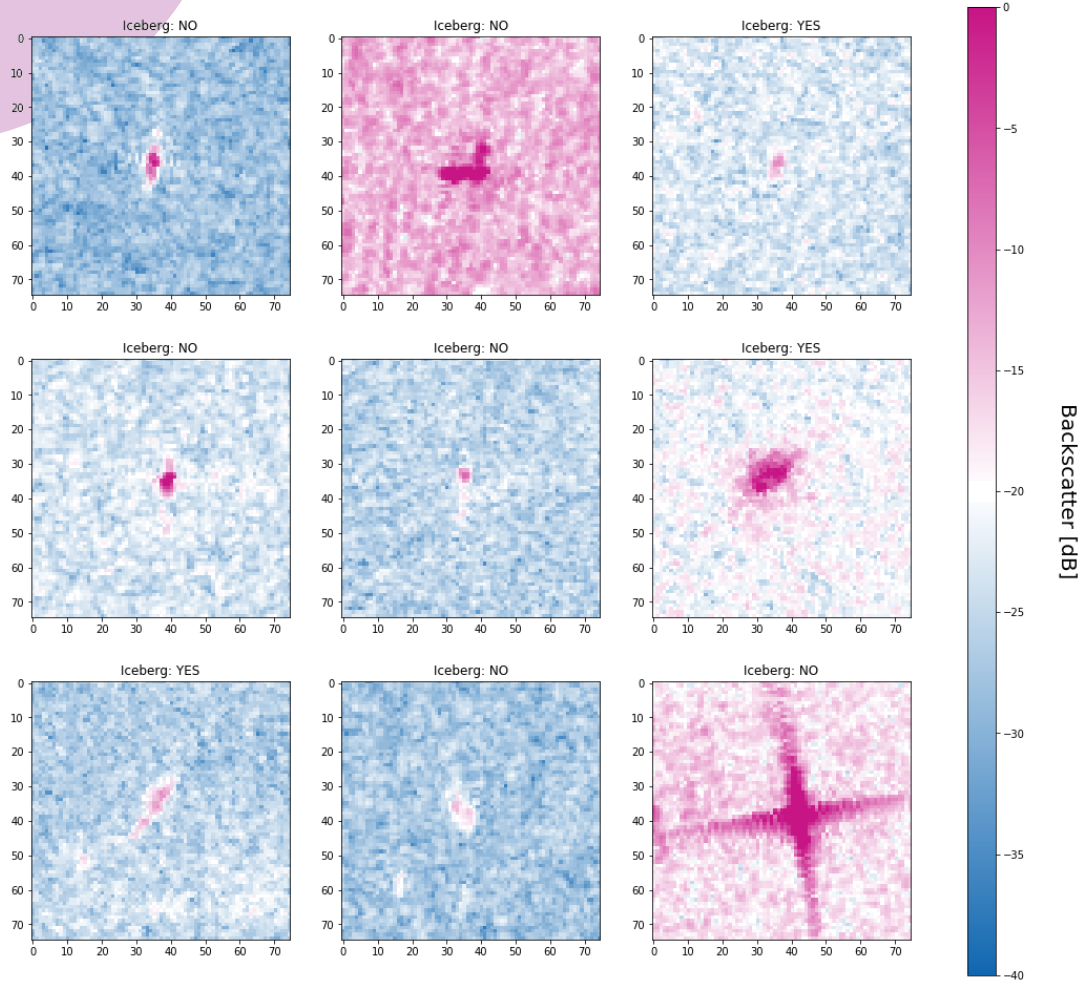
- C-band SAR images from Sentinel-1 satellite
- Only HH and HV polarization
- Incidence angle
- Train dataset: 1604 images
- Test dataset: 8424 images



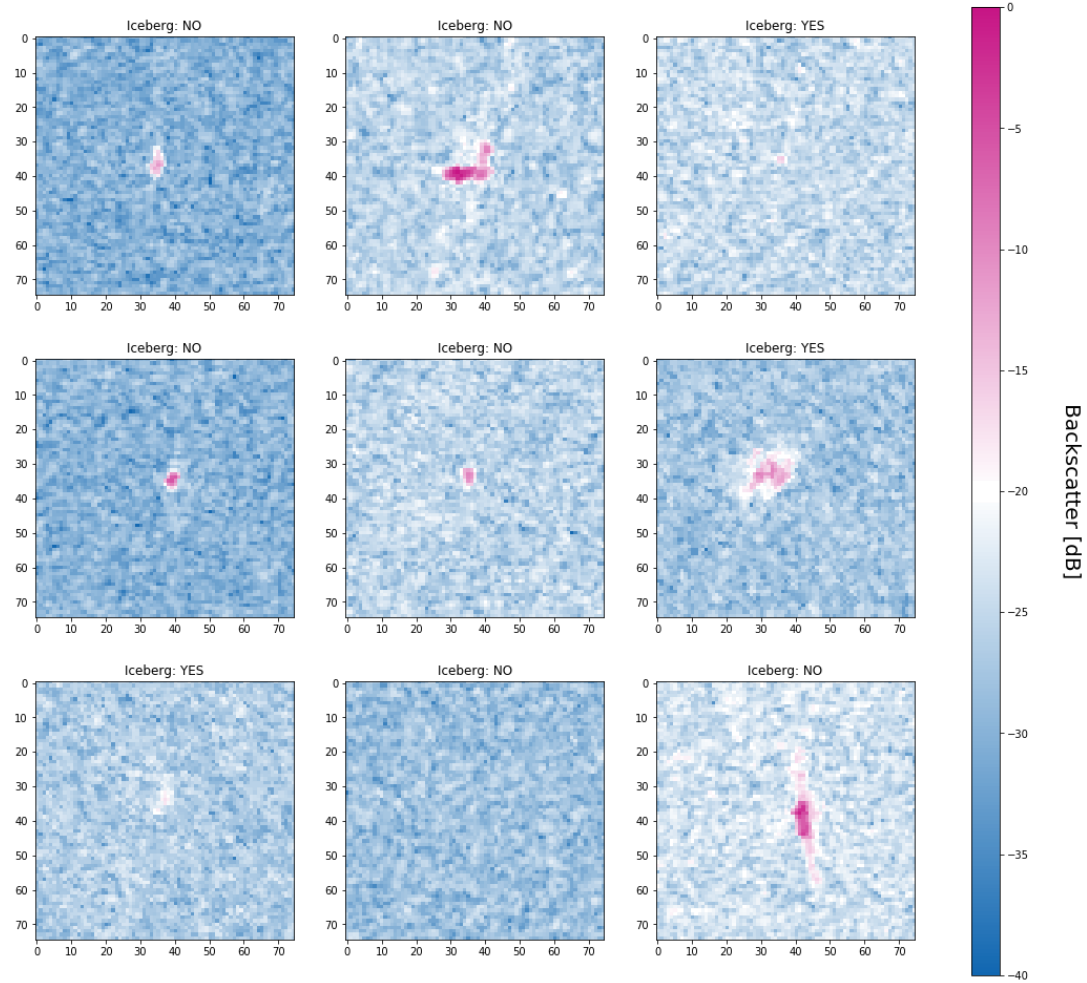


Data

Band 1 (HH polarization)



Band 2 (HV polarization)



Acessing the data

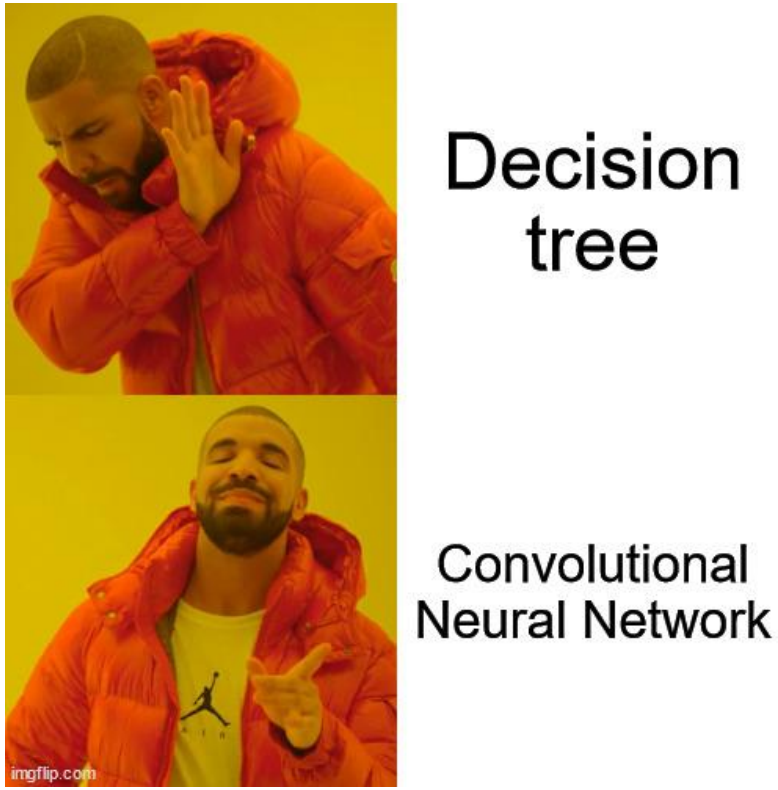
- Missing angles in the data
- Using the full dataset

```
In [3]: train['inc_angle'].value_counts()

Out[3]: na         133
        34.4721     23
        42.5591     16
        33.6352     15
        36.1061     15
                   ...
        41.7479      1
        33.1518      1
        36.4845      1
        34.79        1
        34.7608      1
        Name: inc_angle, Length: 879, dtype: int64
```

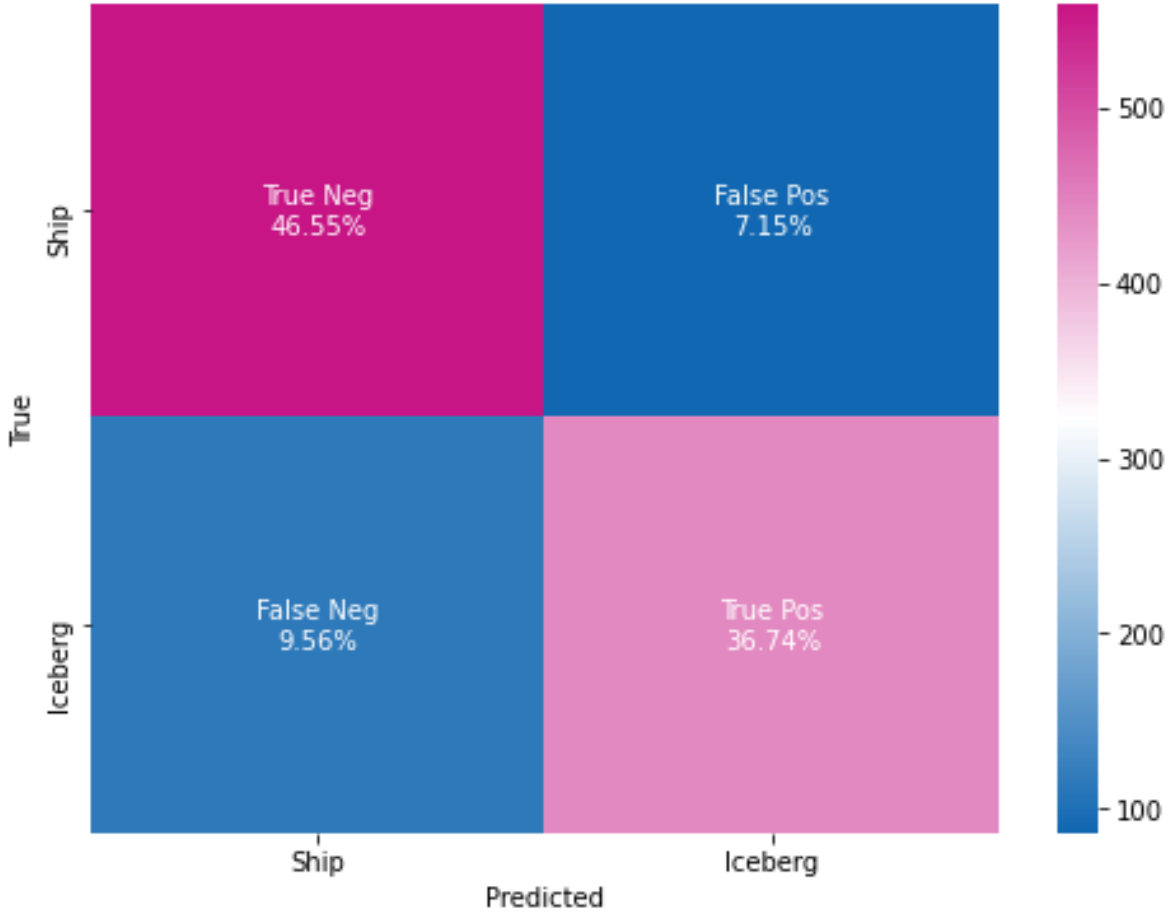
Chosen Models

- Decision tree
  - SKlearn
  - LightGBM
- Neural Network
  - Tensorflow
  - MLP classifier - SKlearn
- CNN - Tensorflow



Decision tree – SKlearn

- Training on HH and HV polarization and satellite angle
- Validating on 25% of the data
- 30 leaves
- Accuracy: 0.928
- Loss: 0.167

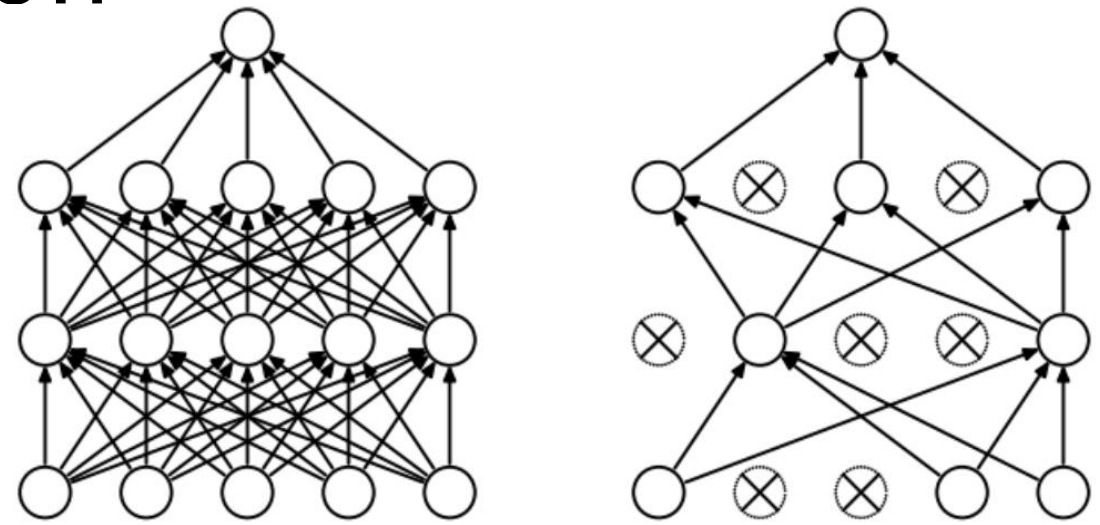


LightGBM

```
[Warning] No further splits with positive gain
```

Dropout regularization

- Overfitting is prevalent, especially with small datasets
- Randomly drop node in a given layer

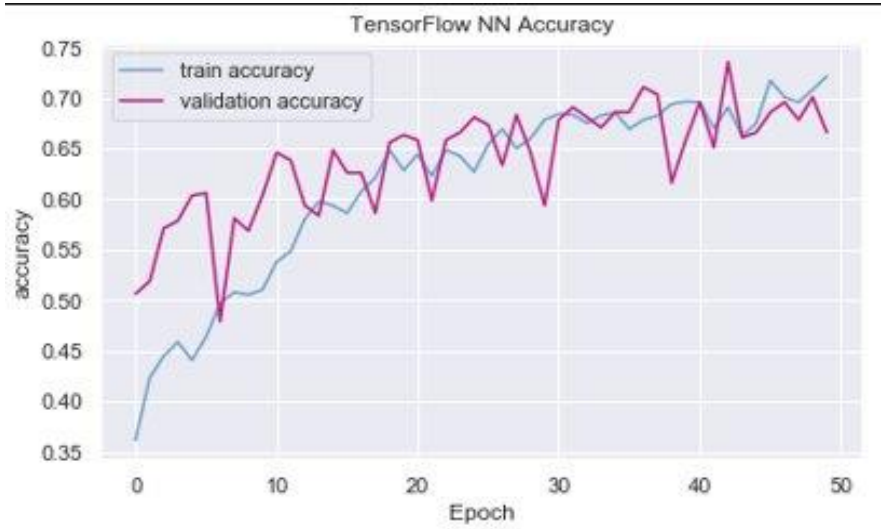


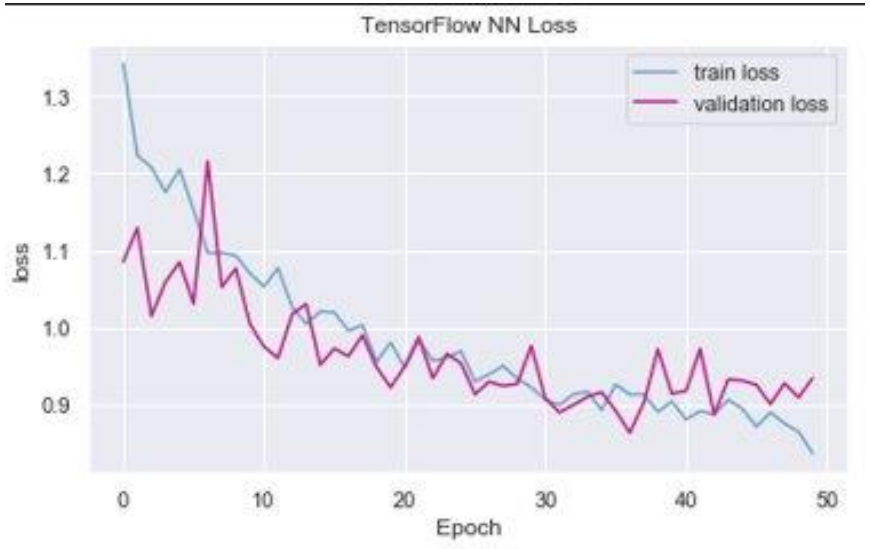
(a) Standard Neural Net (b) After applying dropout.

Srivastava, Nitish, et al. ”Dropout: a simple way to prevent neural networks from overfitting”, JMLR 2014

Neural Network - TensorFlow

- Training on both polarization bands and the incoming angle
- Three hidden layers
- Hyperparameters, activation functions and optimiser varied with trial and error
- Dropout to minimise overtraining





CNN - Data Preprocessing

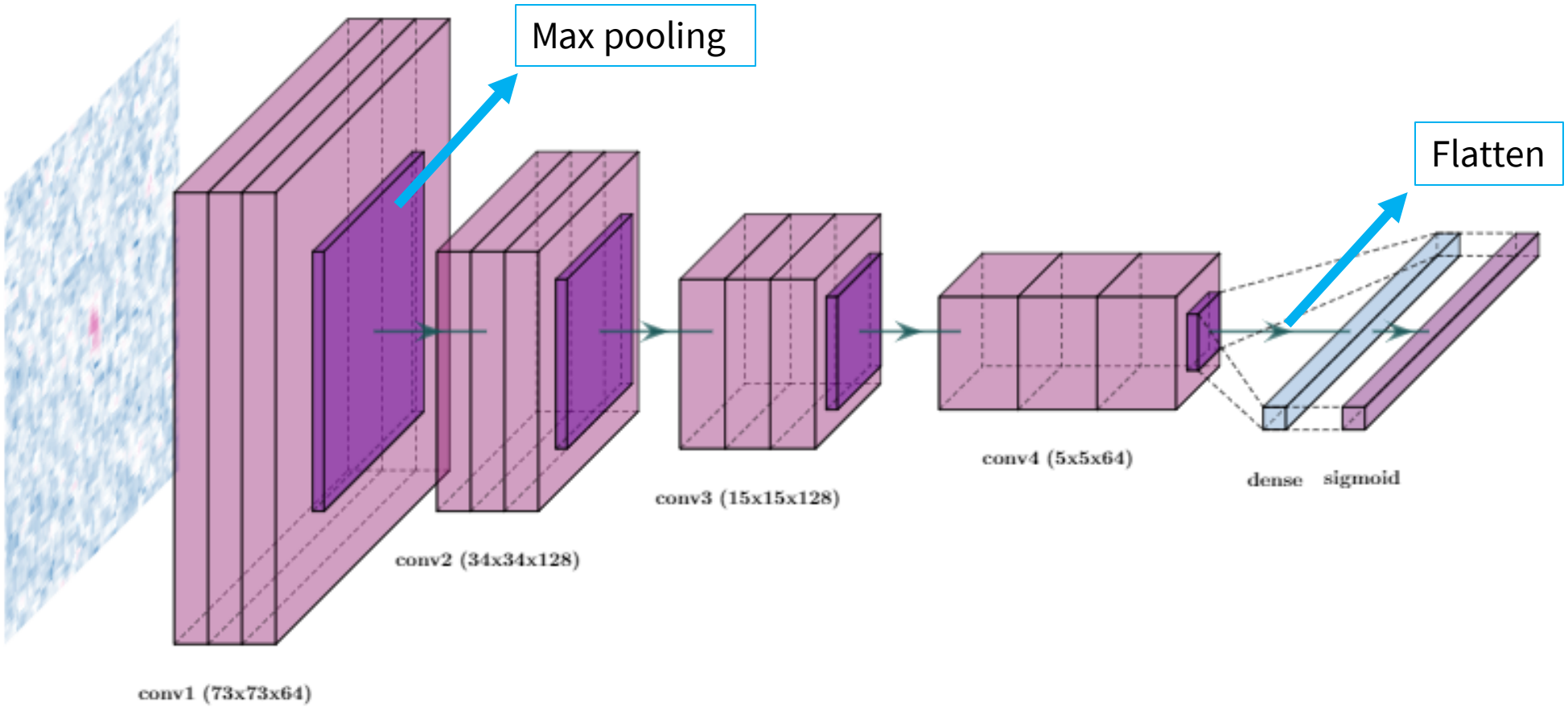
- Generated the training data by creating 3 reshaped bands: HH, HV, avg of both

- Increase train data by including horizontally and vertically flipped data

```
print(Xtr_more.shape)
print(X_train.shape)
```

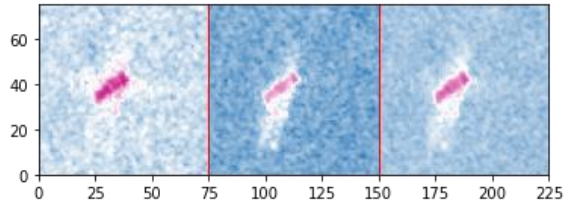
```
(4812, 75, 75, 3)
(1604, 75, 75, 3)
```

CNN – architecture

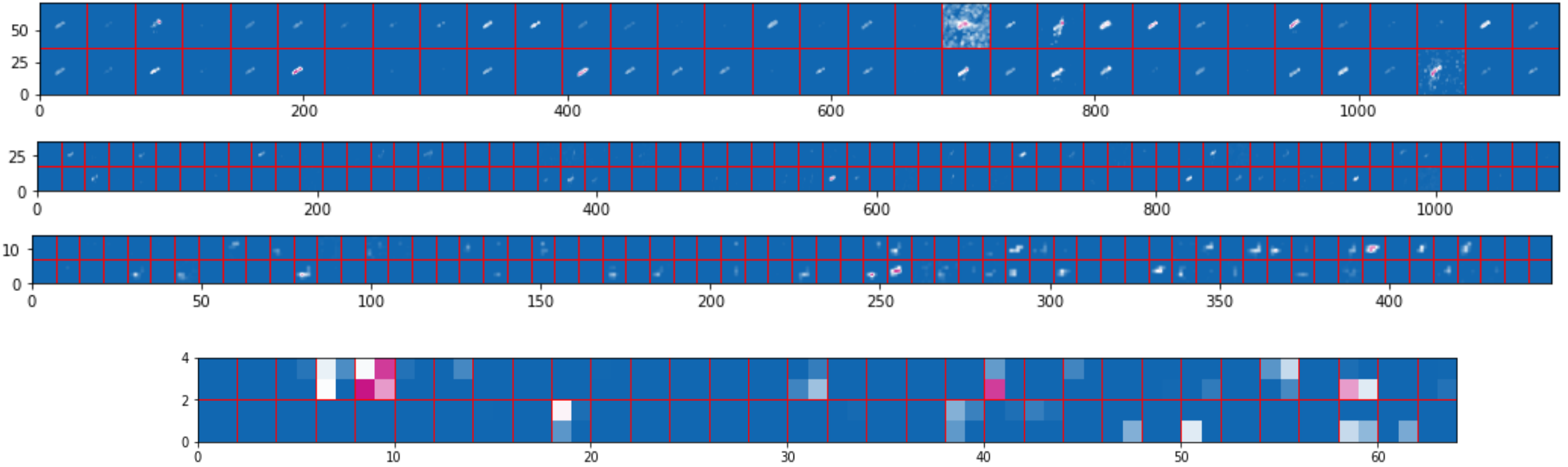


CNN – Convolutional layers

Input picture nr. 8:

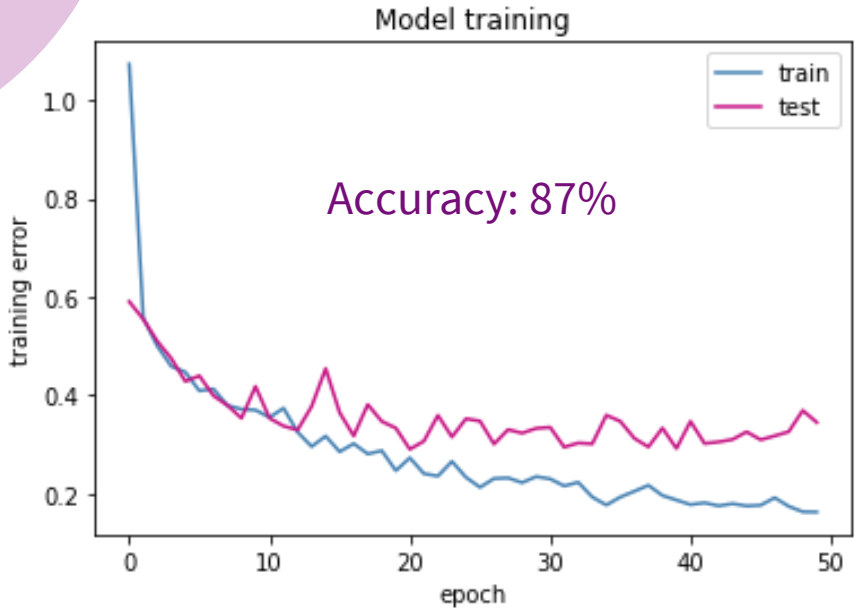


Convolutional layers after Max pooling:



CNN - training

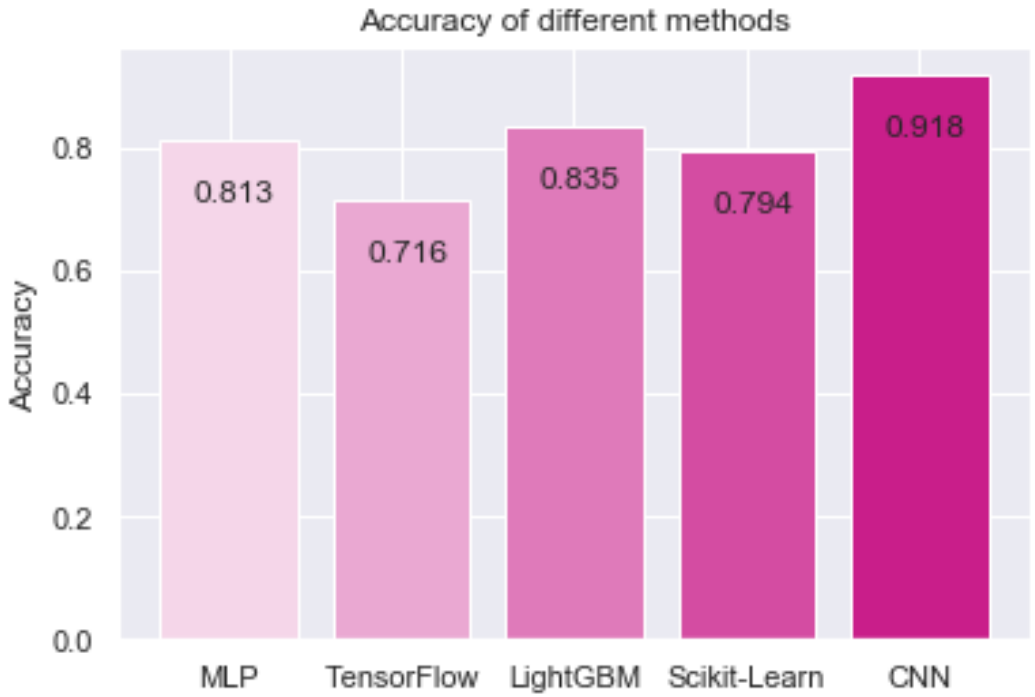
CNN - original dataset

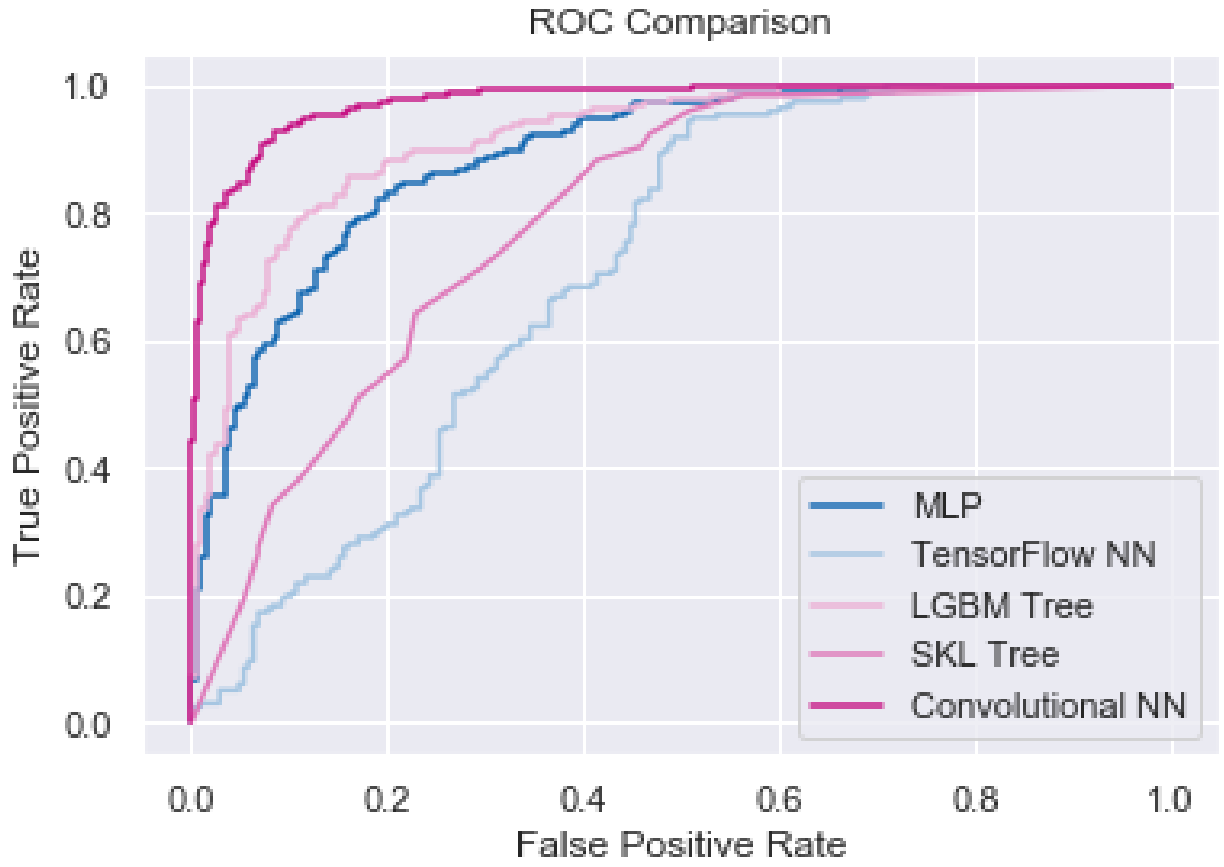


CNN – expanded dataset

- Architectural optimisation had little effect
- Optimisation:
  - ReduceLROnPlateau (lr =0.002)
  - Used early stopping and loaded the optimal weights for evaluation
  - Dropout between 0.1-0.25

Comparison – all models

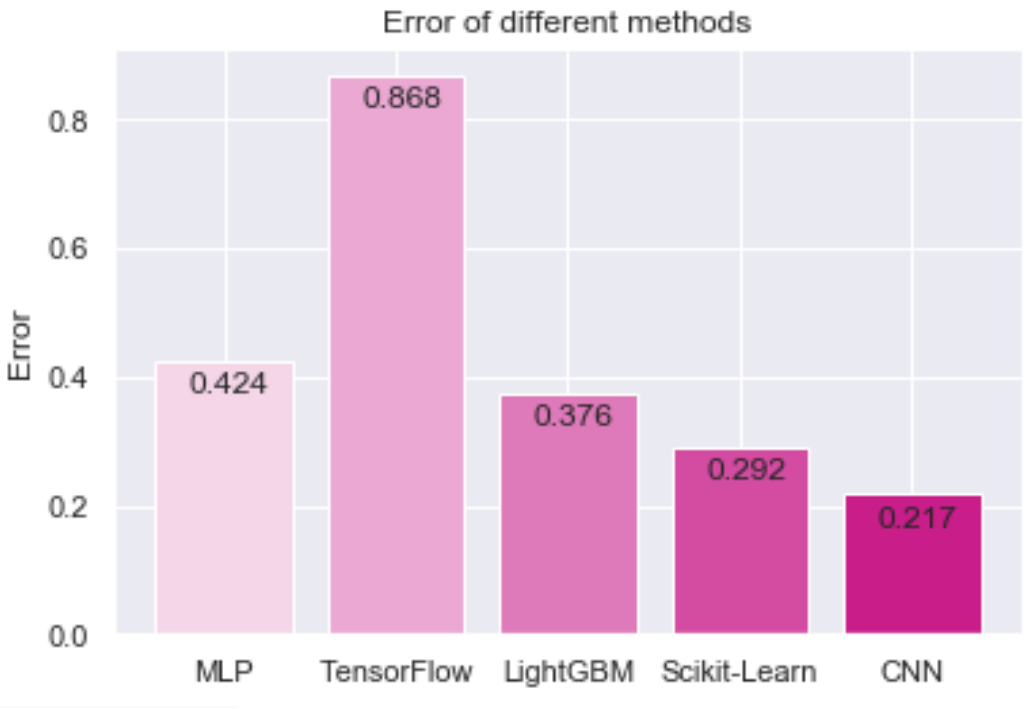




Comparison- Kaggle

In the money | Gold | Silver | Bronze

#	△p...	Team Name	Notebook	Team Members	Score
1	—	David & Weimin			0.08227
2	▲ 3	beluga			0.08555
3	▲ 3	Evgeny Nekrasov			0.08579
4	—	Tarslabs			0.08687
5	▼ 3	Kohei and Medrr			0.08883
6	▲ 3	AzAkhtyamov			0.09102
7	▲ 7	Juan Zhai 卷宅			0.09305
8	▲ 3	alijs			0.09817



Unfortunately we didn't win $ 50,000 😭

Conclusion & Outlook

- We did reasonably well! :)

- Original Dataset is too small leading to overfitting
- Truncating training data might increase accuracy
- Decisiontrees did surprisingly well, but CNN performs best

THANK YOU FOR YOUR ATTENTION!

References

Official Hompage for Kaggle Iceber classification challenge

https://www.kaggle.com/c/statoil-iceberg-classifier-challenge/overview/description

Learning, C., 2021. *ANN vs CNN vs RNN* | *Types of Neural Networks*. [online] Analytics Vidhya. Available at: <*https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/*>

Brownlee, J., 2021. **A Gentle Introduction to Dropout for Regularizing Deep Neural Networks.** Machine Learning Mastery. Available at: *https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/*

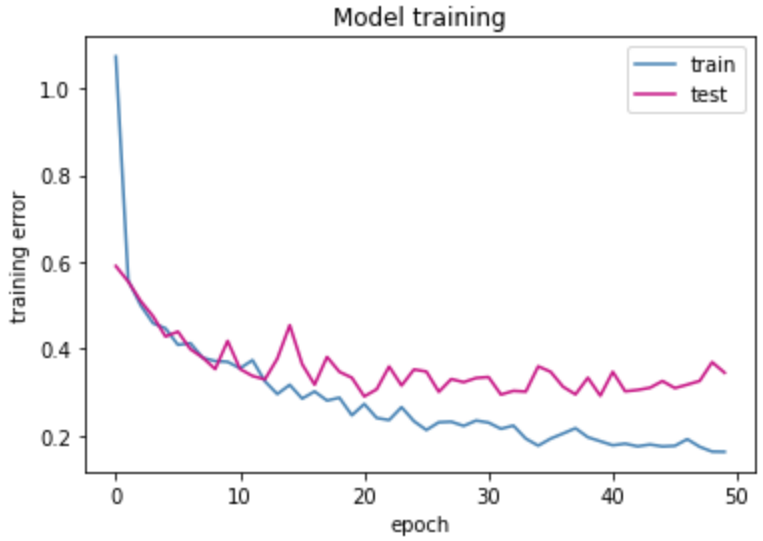
APPENDIX

CNN – original dataset

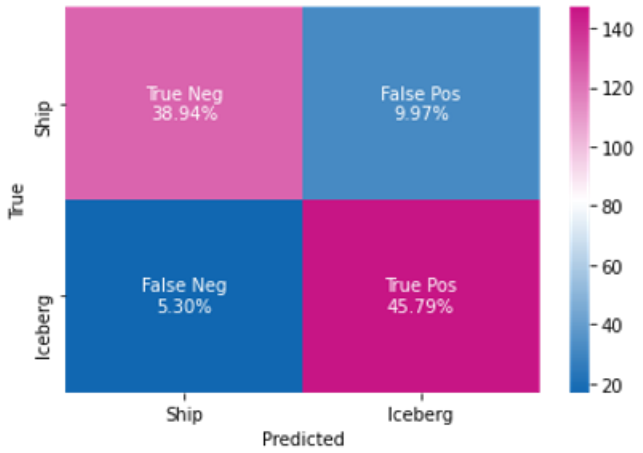
This is all the information on the original CNN with only optimizing the learning and using the smaller, original dataset of 1604 pictures.

As one can see, it was overtraining.

CNN training:



CNN validation:



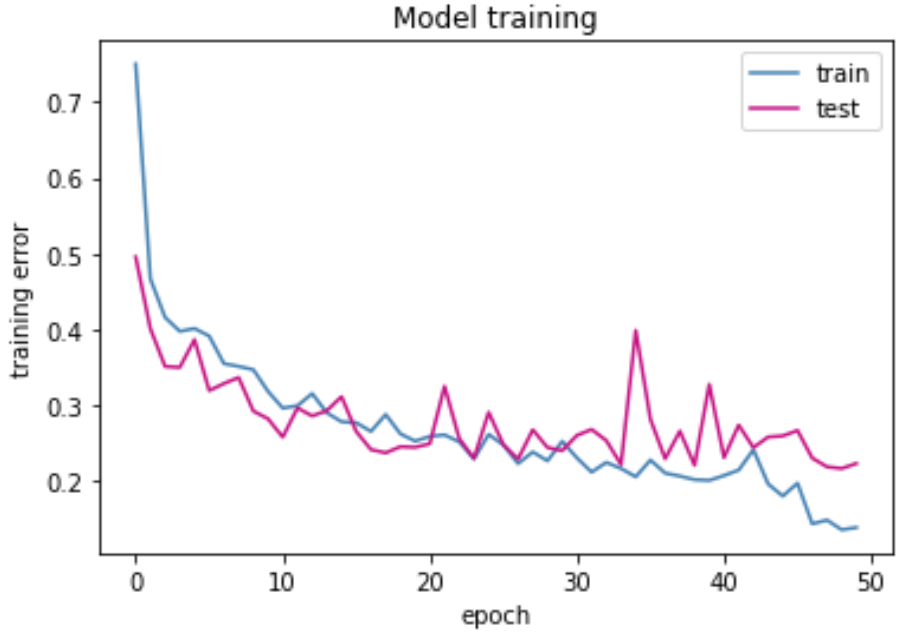

```
Test loss:
0.3452695906162262
Test accuracy:
0.8753893971443176
```

CNN- best model

This slide contains the information on the best CNN we obtained. We optimized it using a learning rate decay on plateaus, saved all the weights and used dropout regularisation. Further, we expanded the training dataset.

CNN validation:

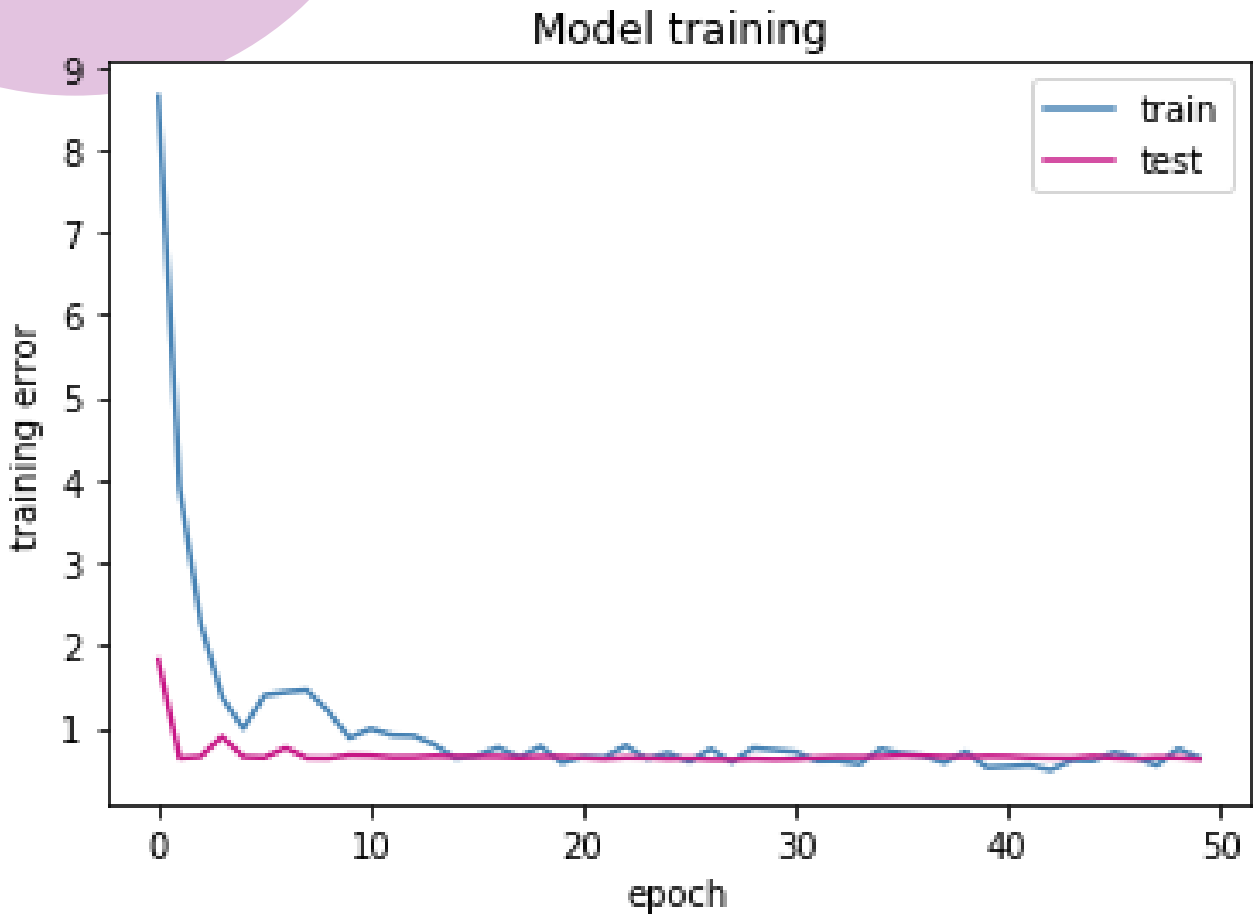
CNN training:




```
Train score: 0.1052284762263298
Train accuracy: 0.9579111337661743

Test loss: 0.21691828966140747
Test accuracy: 0.9179646968841553
```

CNN on false positive and false negatives



Train another CNN on only false positive and false negatives.

The dataset is too small to get a meaningful result, but is poissble to train a CNN on only the false positive and false negative.

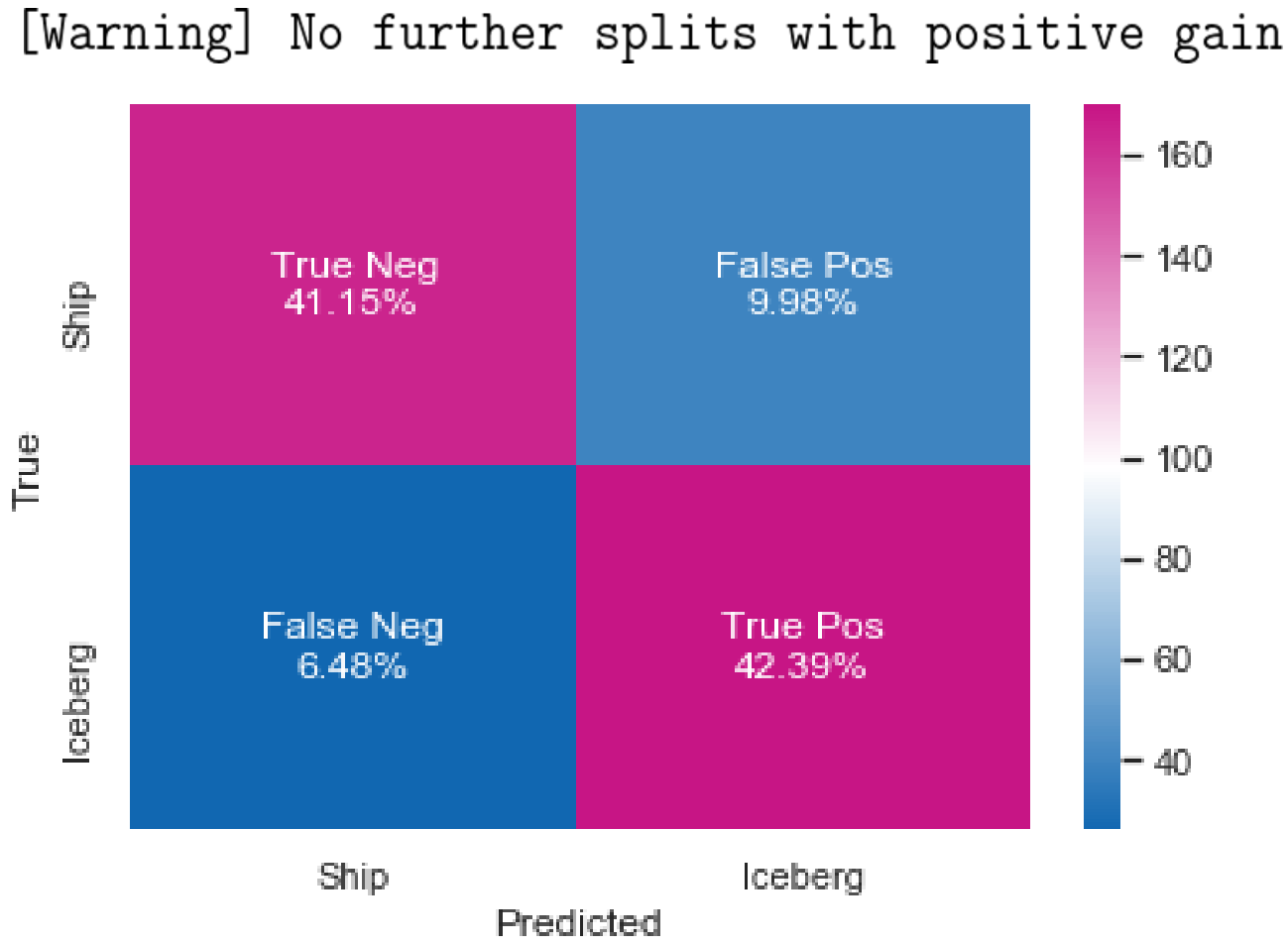
A better result would be obtained if the whole dataset was bigger.

```
Train score: 0.58954
Train accuracy: 0.7307692

Test loss: 0.62392
Test accuracy: 0.750
```

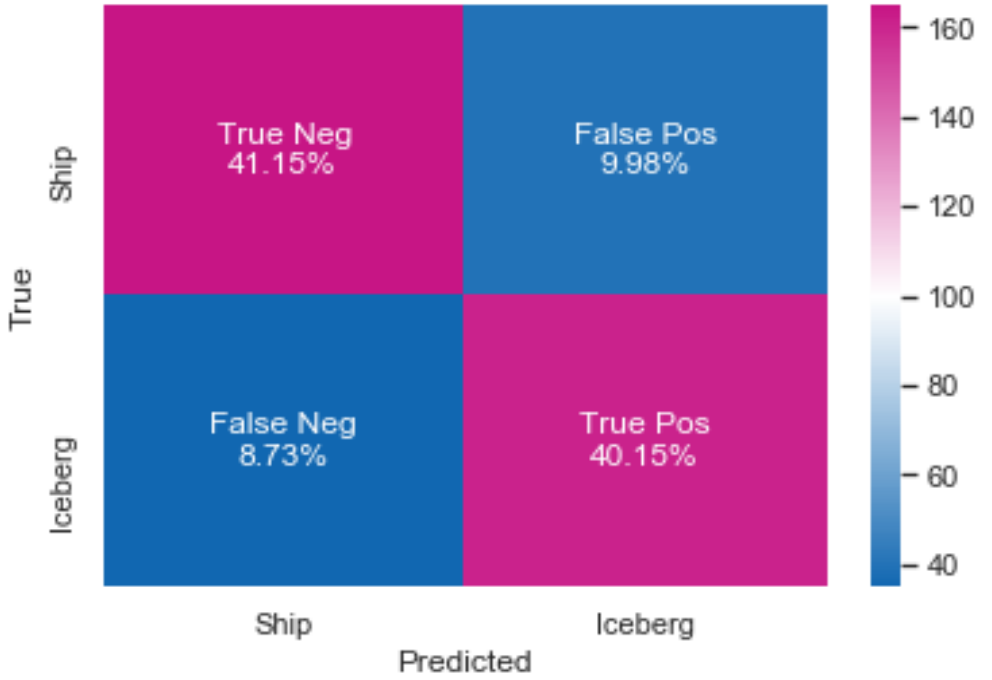
Decision Tree – LightGBM

- Train on both polarisation bands and incident angle
- Test on 25% of data, 300 boosted rounds
- 50 leaves

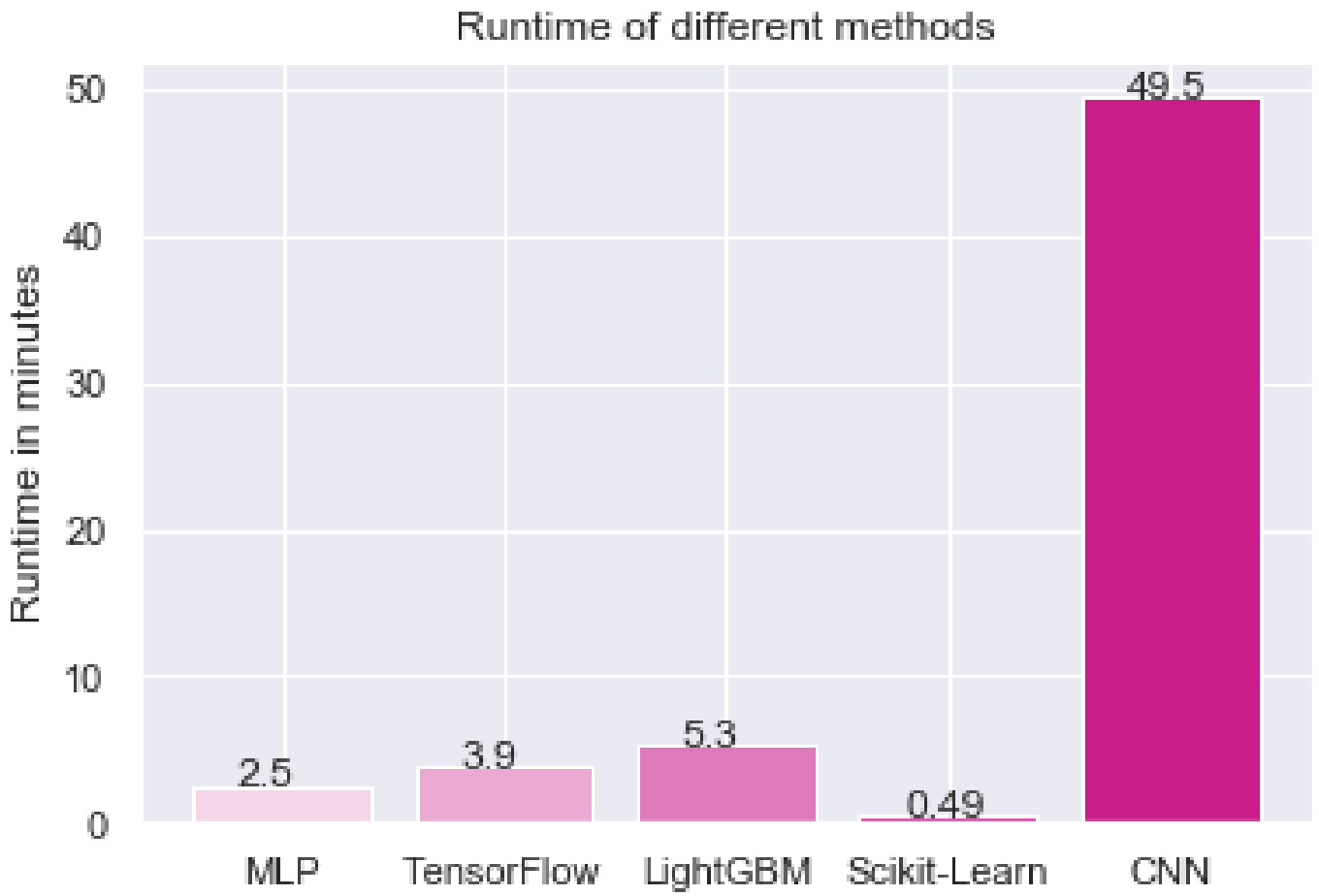


Neural Network – MLP Classifier

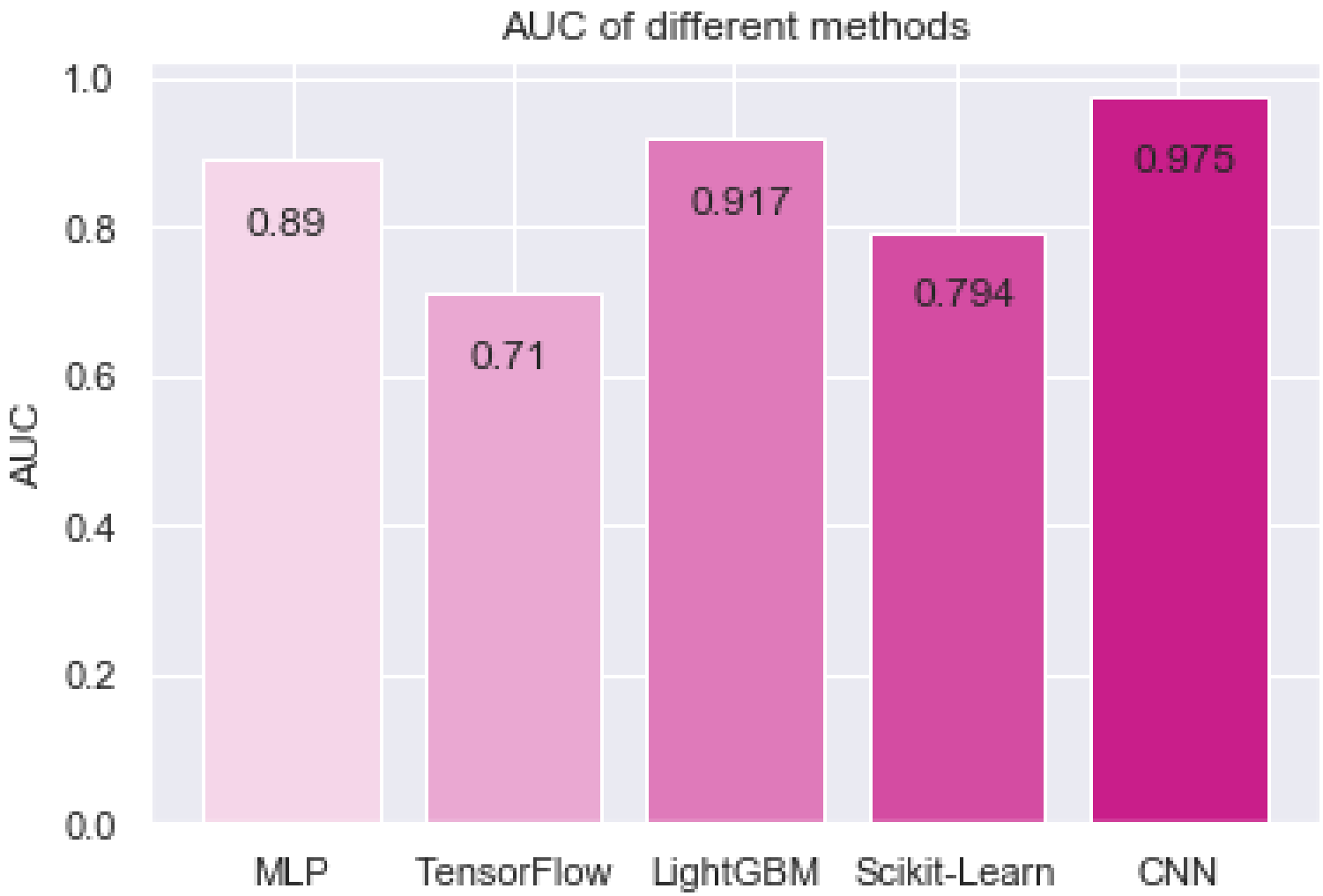
- Activation function and learning rate varied by trial and error
- Three hidden layers
- Training on both bands and incoming angle
- MLP accepts tabular data rather than image data



Runtime


Runtime of different methods
Runtime in minutes
50
40
30
20
10
0
2.5
3.9
5.3
0.49
49.5
MLP
TensorFlow
LightGBM
Scikit-Learn
CNN

AUC


AUC of different methods
AUC
1.0
0.8
0.6
0.4
0.2
0.0
0.89
0.71
0.917
0.794
0.975
MLP
TensorFlow
LightGBM
Scikit-Learn
CNN

THE END!

Give me the truth of the data!

no...