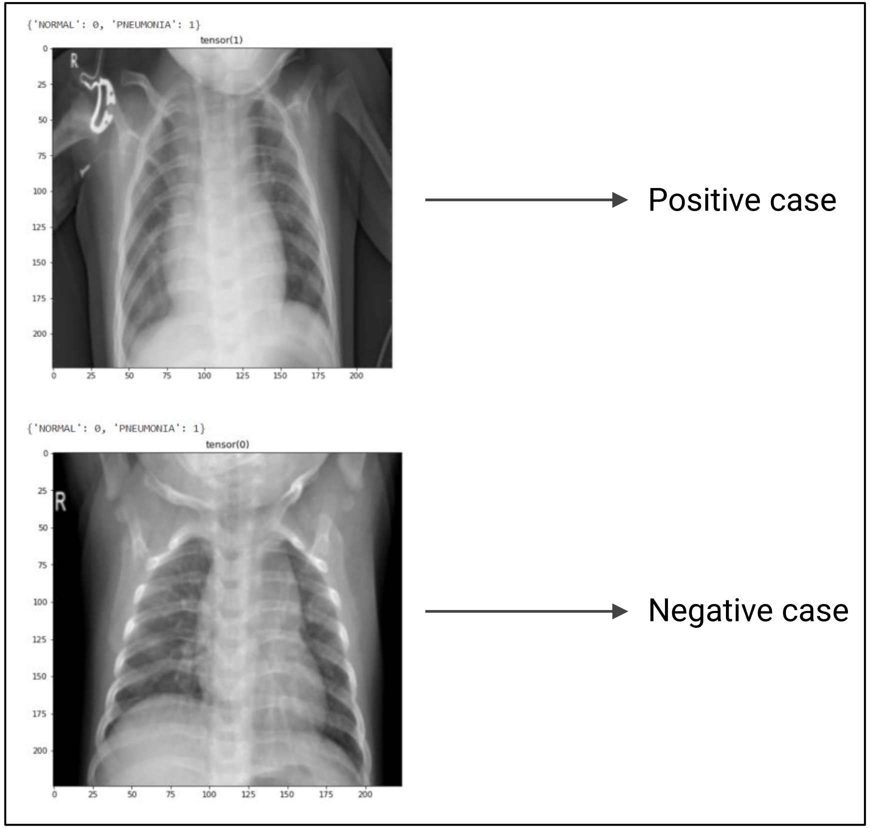
ML Final Project Report

Group 32
Aditya Saini
Lavanya Verma
Prabhat Soni

Problem Statement

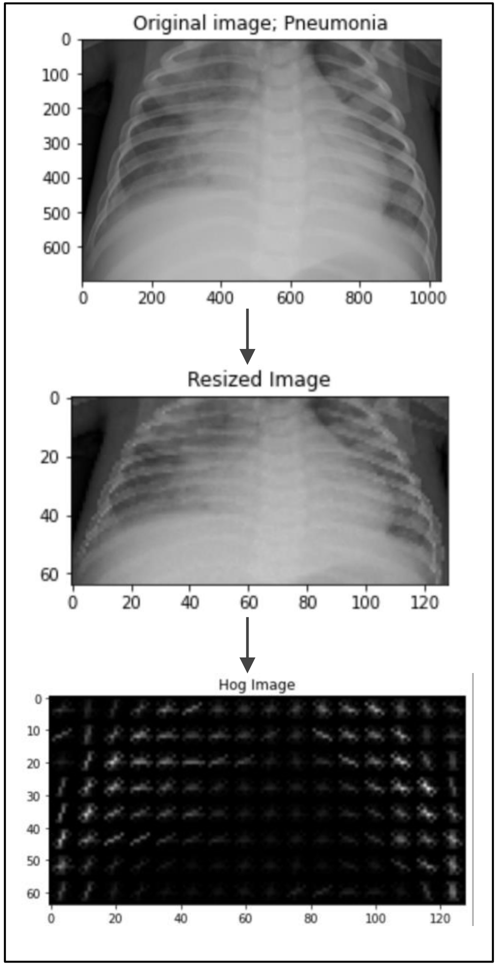
- On the basis of a chest X-ray scan image, determine whether the patient has pneumonia or not.

- By automating the diagnostic pipeline, we hope to make the process much quicker and accurate for both the doctors and the patients.



Approaches used - 1

- **<u>Baseline model</u>**
  - **Feature processing technique:** HoG features
  - **ML model:** SVM classifier
- **<u>Advanced technique 1</u> (Prabhat)**
  - **Feature processing technique:** HoG features
  - **ML model:** MLP classifier



<u>Feature preprocessing pipeline</u>

Approaches used - 2

- **<u>Advanced technique 2</u> (Lavanya)**
  - **ML model:** Convolutional Neural Network
- **<u>Advanced technique 3</u> (Aditya)**
  - **Feature processing technique:** Image transforms as per input layer of the model
  - **ML model:** Transfer learning with pre trained VGG-16 model

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 224, 224]           1,792
              ReLU-2         [-1, 64, 224, 224]               0
            Conv2d-3         [-1, 64, 224, 224]          36,928
              ReLU-4         [-1, 64, 224, 224]               0
         MaxPool2d-5         [-1, 64, 112, 112]               0
            Conv2d-6        [-1, 128, 112, 112]          73,856
              ReLU-7        [-1, 128, 112, 112]               0
            Conv2d-8        [-1, 128, 112, 112]         147,584
              ReLU-9        [-1, 128, 112, 112]               0
        MaxPool2d-10          [-1, 128, 56, 56]               0
           Conv2d-11          [-1, 256, 56, 56]         295,168
             ReLU-12          [-1, 256, 56, 56]               0
           Conv2d-13          [-1, 256, 56, 56]         590,080
             ReLU-14          [-1, 256, 56, 56]               0
           Conv2d-15          [-1, 256, 56, 56]         590,080
             ReLU-16          [-1, 256, 56, 56]               0
        MaxPool2d-17          [-1, 256, 28, 28]               0
           Conv2d-18          [-1, 512, 28, 28]       1,180,160
             ReLU-19          [-1, 512, 28, 28]               0
           Conv2d-20          [-1, 512, 28, 28]       2,359,808
             ReLU-21          [-1, 512, 28, 28]               0
           Conv2d-22          [-1, 512, 28, 28]       2,359,808
             ReLU-23          [-1, 512, 28, 28]               0
        MaxPool2d-24          [-1, 512, 14, 14]               0
           Conv2d-25          [-1, 512, 14, 14]       2,359,808
             ReLU-26          [-1, 512, 14, 14]               0
           Conv2d-27          [-1, 512, 14, 14]       2,359,808
             ReLU-28          [-1, 512, 14, 14]               0
           Conv2d-29          [-1, 512, 14, 14]       2,359,808
             ReLU-30          [-1, 512, 14, 14]               0
        MaxPool2d-31            [-1, 512, 7, 7]               0
AdaptiveAvgPool2d-32            [-1, 512, 7, 7]               0
           Linear-33                 [-1, 4096]     102,764,544
             ReLU-34                 [-1, 4096]               0
          Dropout-35                 [-1, 4096]               0
           Linear-36                 [-1, 4096]      16,781,312
             ReLU-37                 [-1, 4096]               0
          Dropout-38                 [-1, 4096]               0
           Linear-39                    [-1, 2]           8,194
       LogSoftmax-40                    [-1, 2]               0
================================================================
Total params: 134,268,738
Trainable params: 8,194
Non-trainable params: 134,260,544
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 218.77
Params size (MB): 512.19
Estimated Total Size (MB): 731.54
----------------------------------------------------------------
```

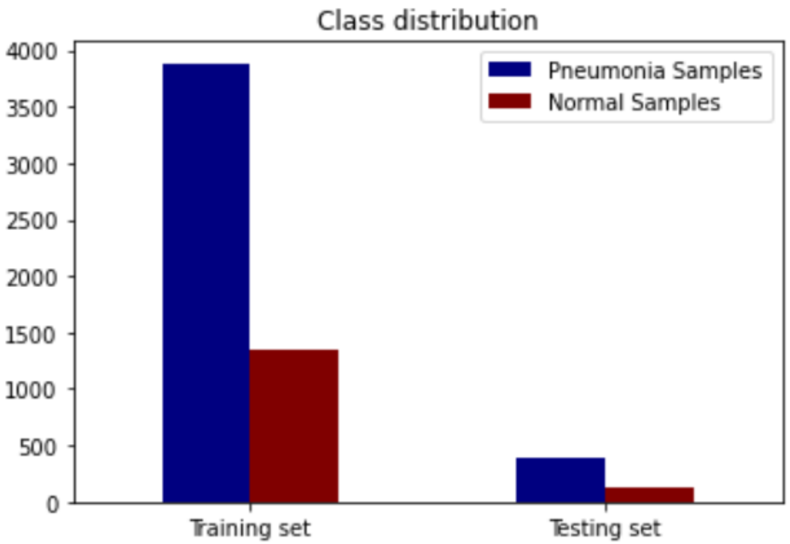
<u>Architecture used for transfer learning</u>

Dataset used and Evaluation metrics

- **<u>Dataset used</u>**

For training our models, we used the Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification dataset.

- **<u>Evaluation metrics</u>**
    - Accuracy
    - Confusion matrix
    - F1- Score (data is imbalanced)



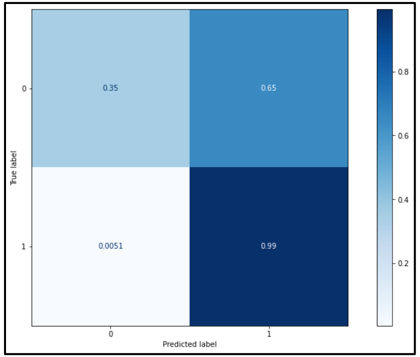
Class distribution in dataset

Summary of Results - 1

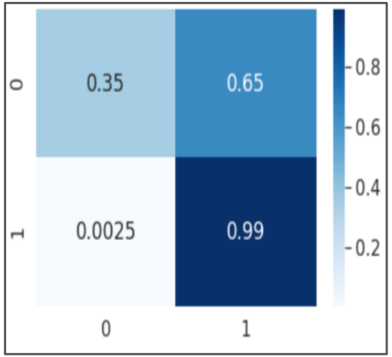
Technique	Test set accuracy	Class wise f1-score(0, 1)
Baseline: SVM + HoG	74%	0.51, 0.83
Advanced technique 1: ANN + HoG	77%	0.56, 0.84
Advanced technique 2: CNN	75.3%	0.51, 0.83
Advanced technique 3: Transfer Learning with VGG-16	**82.8%**	**0.72, 0.88**

Summary of Results - 2

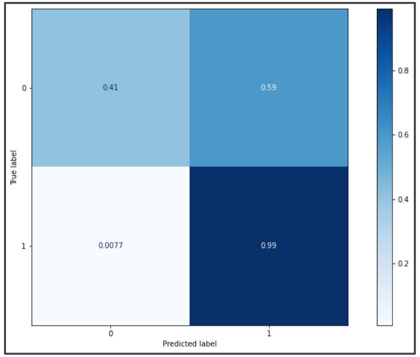
Baseline model



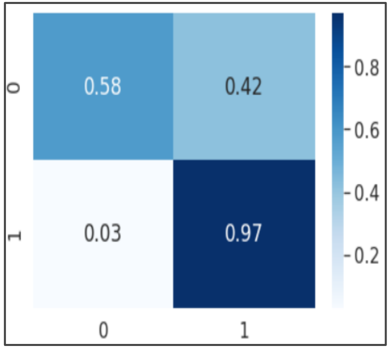
Vanilla CNN



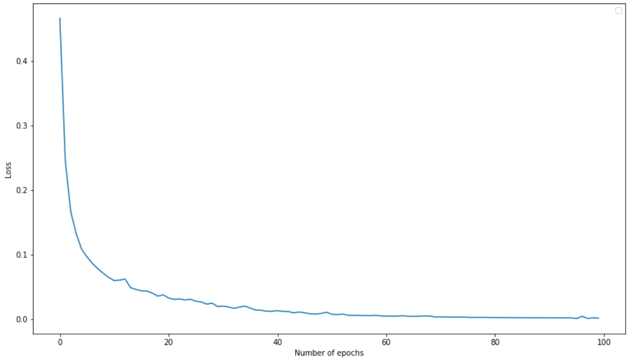
MLP + HoG model

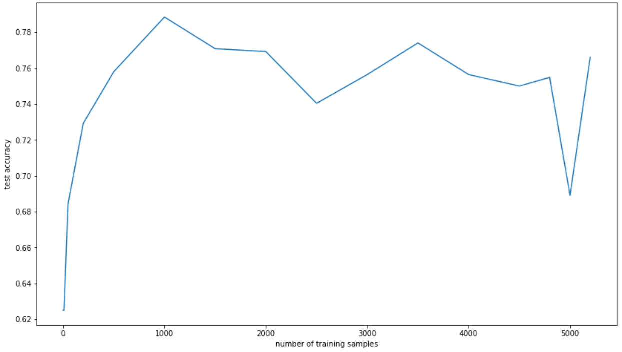


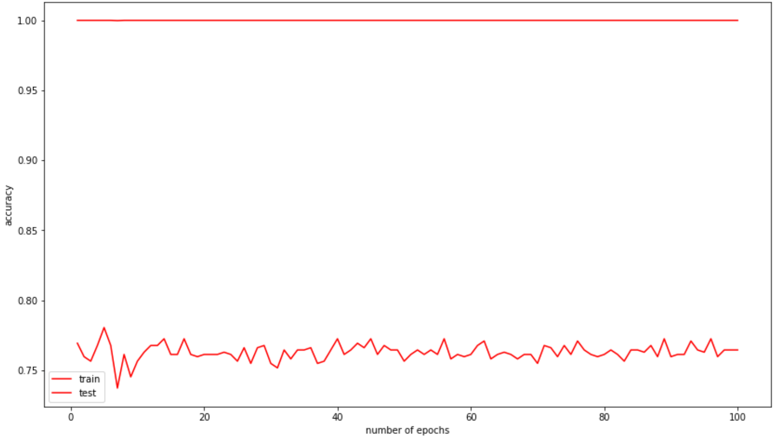
Transfer learning model



Analysis: ANN + HoG model





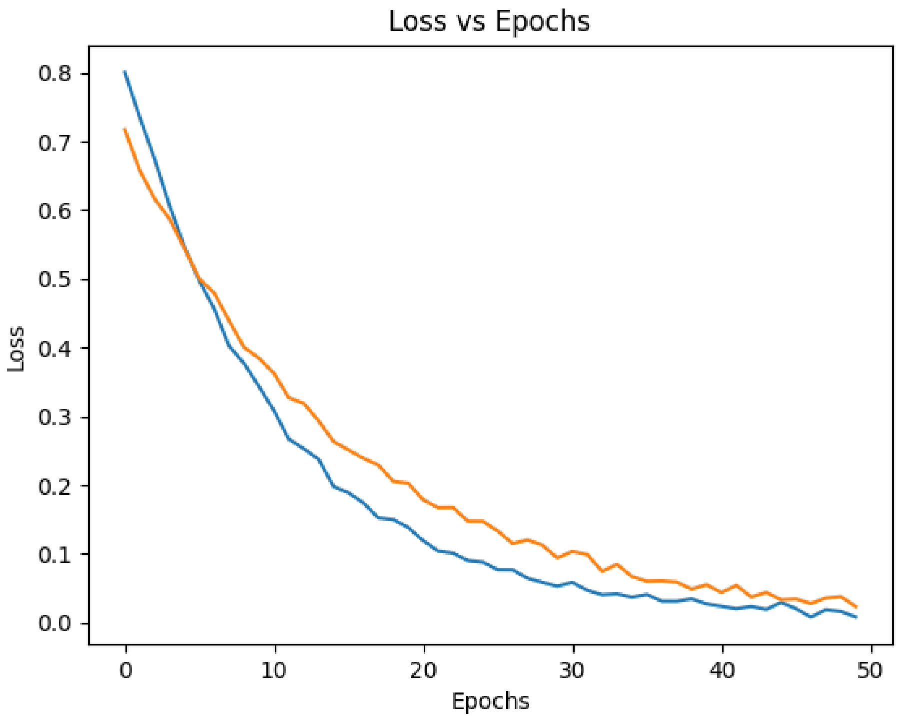


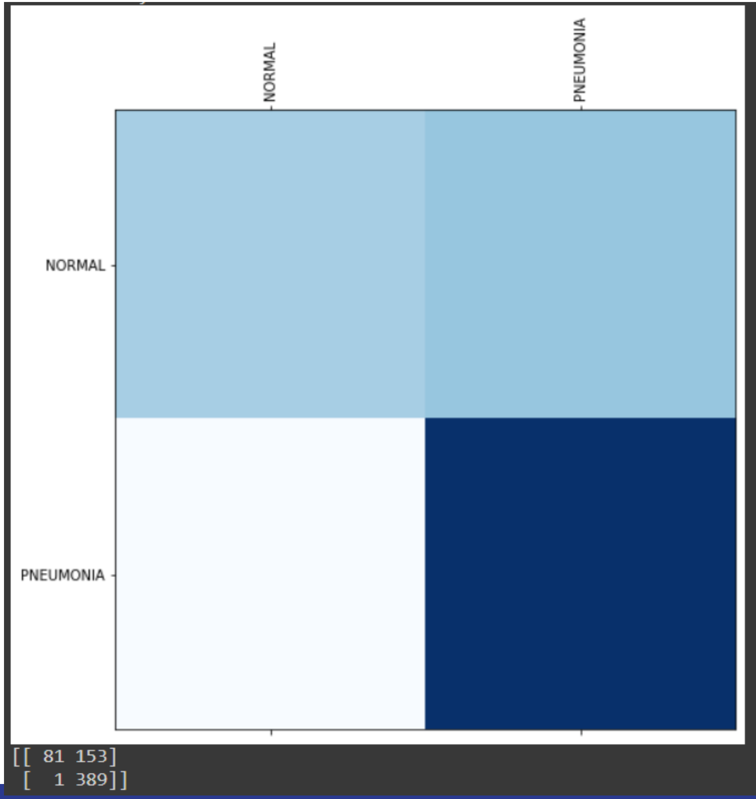
Evidence of overfitting / high bias. This led us to experiment with higher values of alpha (L2 regularization variable) to see if that would help

Inference: less features could improve the model. Early stopping makes no difference.

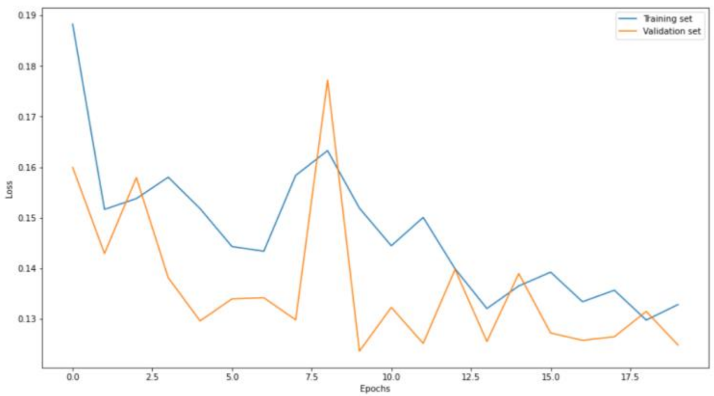
More difficult to detect overfitting/underfitting in this case because we are dealing with binary data.

Analysis: Loss curve + Confusion matrix for CNN model





Analysis: Transfer learning model



Analysis: Approaches used for improving over baseline

- **Advanced technique 1: HoG features + MLP classifier**
  - Used hyper parameter tuning for training, different model architectures
  - Tried different values of k in k-fold (best was k=3)

- **Advanced technique 2: CNN model**
  - Dropouts with p = 0.2
  - L2 regularization
  - Min-max scaling

- **Advanced technique 3: Transfer learning with VGG-16 model**
  - Appropriate Image transforms
  - Used a lr_scheduler.ReduceLROnPlateau scheduler for adaptive learning rate updation

Conclusion

- Automatic feature extraction models like CNN outperform baseline models.
- Pretrained models CNN models work the best as compared to CNN model
- Referring to the error analysis part of the MLY book, the train and test data distributions are different which directly hinders the accuracy of the results