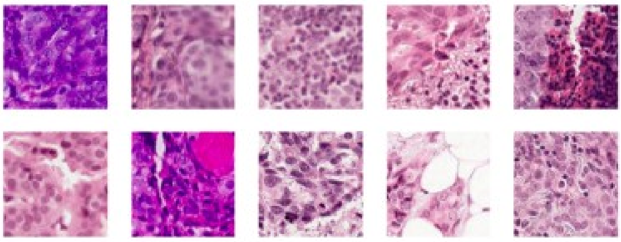
Histopathologic Cancer Detection

Abu Shahid *Anirudh Srikanth* *Atharva Pandey*
B20CS003 *B20CS006* *B20CS007*

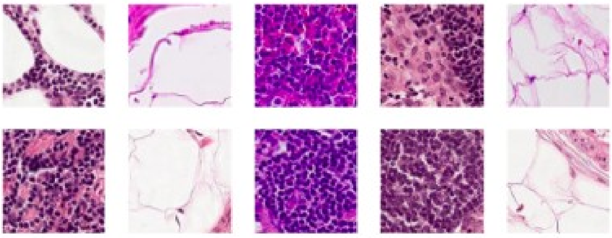
Abstract- Modern medical image processing techniques work on histopathology images captured by a microscope, and then analyze them by using different algorithms and methods. Machine learning algorithms are now being used for processing medical imagery and pathological tools. Manual detection of a cancer cell is a tiresome task and involves human error, and hence computer-aided mechanisms are applied to obtain better results as compared with manual pathological detection systems. In this paper, we share our experience with the exploratory analysis on the data and trying out different models to give the best results.

I. Introduction

Advancing the fight against cancer requires early detection which can only be possible with an efficient detection system. Images are acquired by histopathology, which generally includes biopsy of the affected tissue. These microscopic images can be collected and used for developing computer-aided detection systems. Manual detection is a tedious, tiring task and most likely to comprise human error, as most parts of the cell are frequently part of irregular random and arbitrary visual angles. The goal is to identify whether a tumor is benign or of a malignant in nature, as malignant tumors are cancerous and should be treated as soon as possible to reduce and prevent further complications. In short, it is a binary classification problem and can be resolved by various machine learning methods

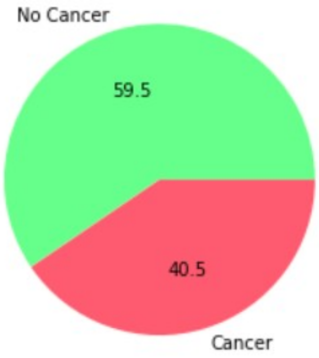


Cancer negative



Cancer positive

Dataset



histopathologic-cancer-detection.zip :The zip file contains 2 folders and a csv file

- train: 220k images named with unique id
- test: 55k images
- train_labels.csv: containing ground truth of training set
  - Later a Random_Sample dataset was generated containing 5500 images.

Training dataset has 59.5 cancer positive images and 40.5 cancer negative ones.

II. Methodology

i. Big Data Processing

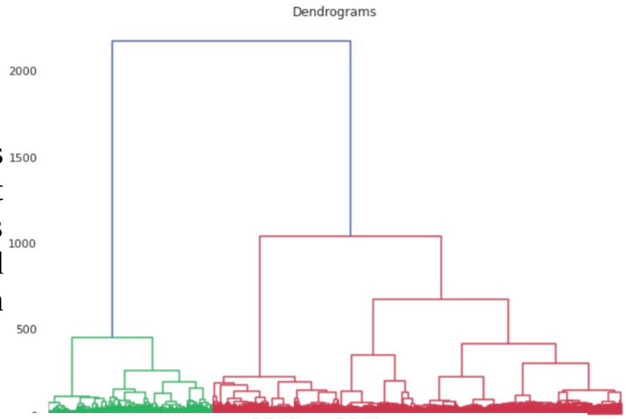
For our task, we had to deal with image data that was enormous in size. Since, processing this huge amount of data is a challenge for learning algorithms of Artificial Intelligence, we target the Sampling of data in big data paradigm. Increasing the amount of data doesn't necessarily increase the information it has. Sampling or Re-sampling is used to summarize or to reduce the amount of data into a smaller data set while preserving its semantics and structure.

[Sampling](#) is "the process of selecting units from a population of interest so that by studying the sample we may fairly generalize our results back to the population from which they were chosen"

ii. Exploratory Data Analysis

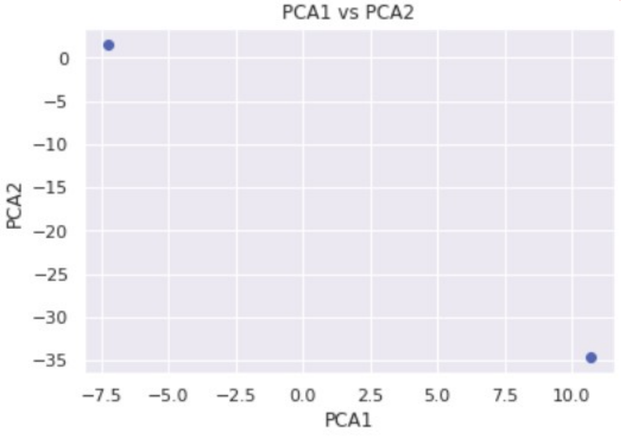
a. Clustering

On our reduced dataset, we performed grouping to inspect if there exists clear groups in an unsupervized fashion. For this, we plotted a dendogram which gave us the existence to two clearly separable clusters.



b. Dimensionality reduction

Further, we attempted dimensionality reduction on our images to see if there exists a lower dimension subspace that preserves the clusters to a satisfactory extent. To explore this we implement various PCA (to reduce dimensionalioty) and LDA (to compute discriminatory features) models and fit it in various models to see how it performs.



Variance Captured	No. of features
1.00	9216
0.98	3400
0.95	2432

- same can be summarized using table below.
- **Note: the training for the same is being done on reduced resampled dataset.**
- **Note: Models without PCA and LDA also put the benchmark in place.**

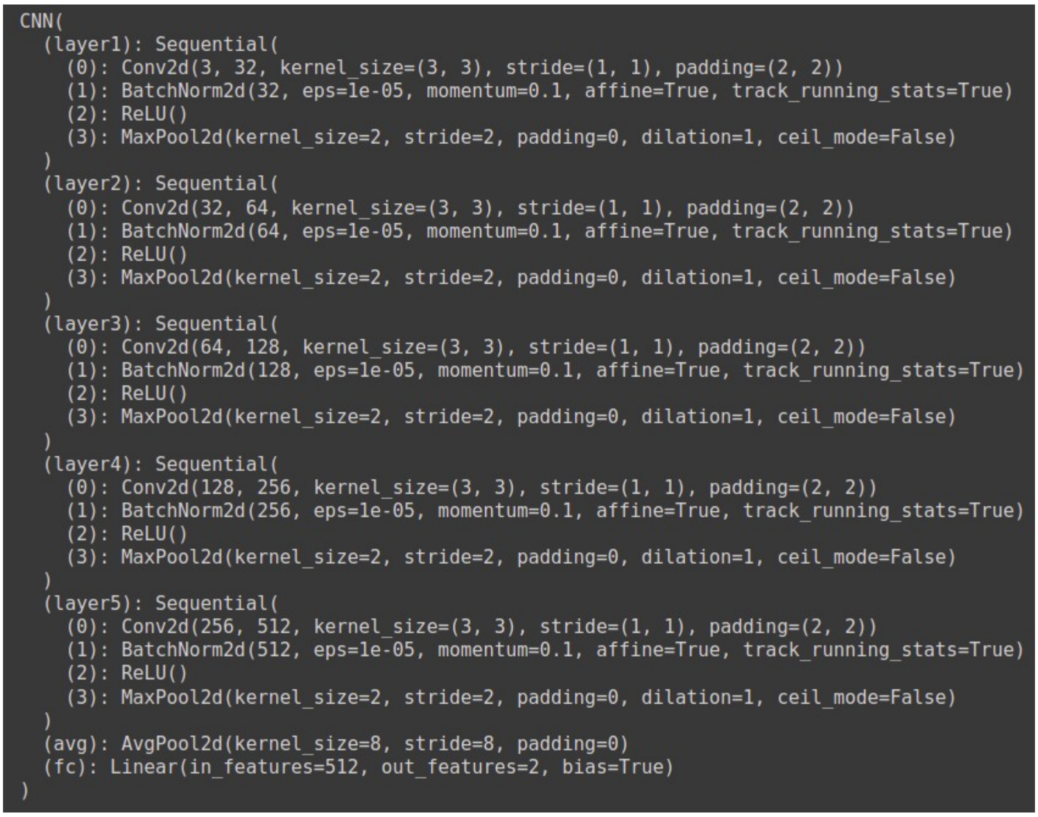
Model	Precision	Recall	F1-score
Without PCA and LDA			
Linear SVC	0.60	0.61	0.60
KNN	0.65	0.65	0.62
RBF SVC	0.78	0.77	0.78
LightGBM	0.79	0.77	0.77
PCA with 2 components			
Linear SVC	0.63	0.61	0.61
KNN	0.65	0.65	0.65
RBF SVC	0.68	0.68	0.68
LightGBM	0.67	0.66	0.67
PCA with 0.8 variance preservation			
Linear SVC	0.63	0.63	0.63
KNN	0.64	0.65	0.61
RBF SVC	0.74	0.70	0.71
LightGBM	0.78	0.77	0.78
PCA with 0.8 variance preserved + LDA			

Linear SVC	0.76	0.75	0.76
KNN	0.73	0.73	0.73
RBF SVC	0.77	0.77	0.77
LightGBM	0.76	0.75	0.75
PCA with 0.9 variance preserved + LDA			
Linear SVC	0.84	0.83	0.83

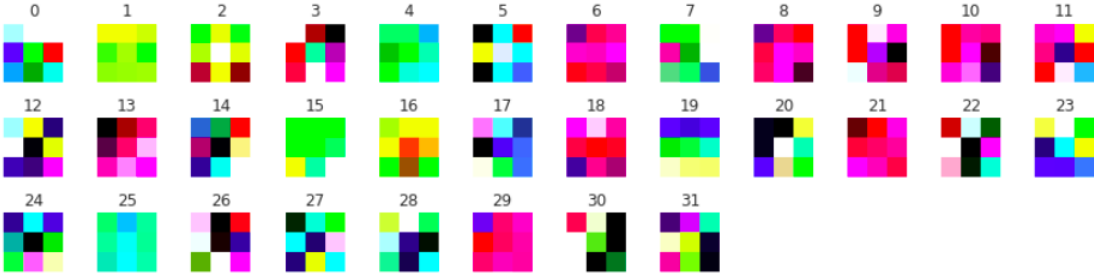
- It became clear from the above exercise that dimensionality reduction increases the efficiency of models generally. The places where a drop is seen, the loss is small and we should not forget that the reduced mathematical complexity more or less can compensate for the reduction in the f1-scores. This lays foundation to try out CNN.
- CNN's are really effective for image classification as the concept of dimensionality reduction suits the huge number of parameters in an image. CNNs are fully connected feed forward neural networks. CNNs are very effective in reducing the number of parameters without losing on the quality of models. [2]

iii_Model Training

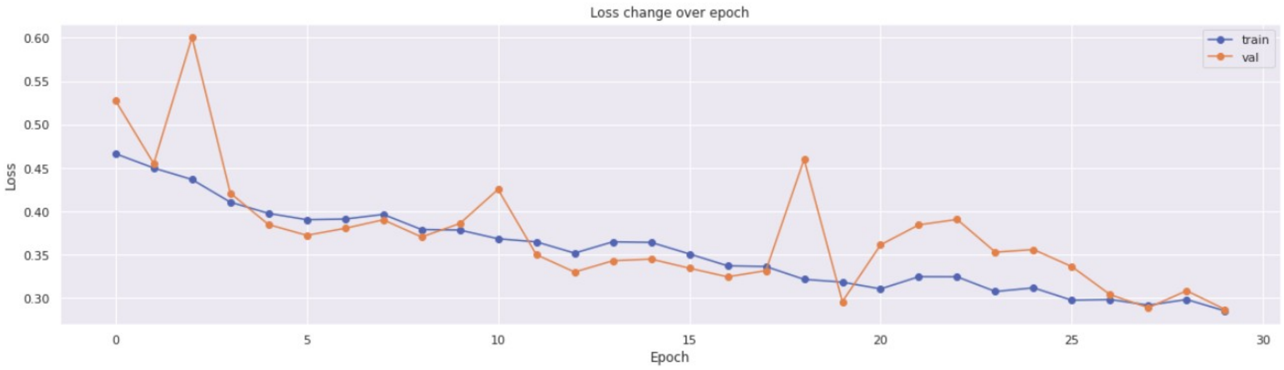
For our training, we implemented the following CNN architecture in pyTorch.



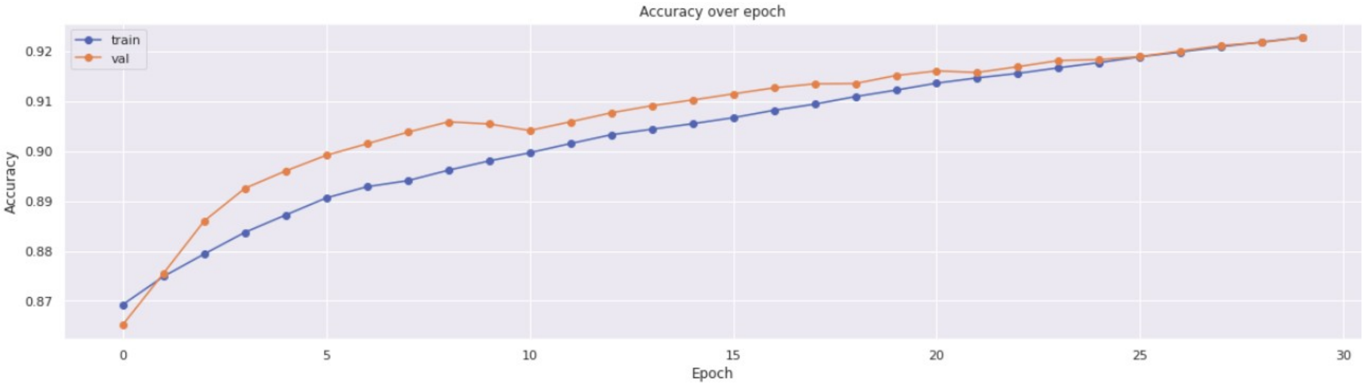
```
CNN(
  (layer1): Sequential(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(2, 2))
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (layer2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(2, 2))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (layer3): Sequential(
    (0): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(2, 2))
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (layer4): Sequential(
    (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(2, 2))
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (layer5): Sequential(
    (0): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(2, 2))
    (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avg): AvgPool2d(kernel_size=8, stride=8, padding=0)
  (fc): Linear(in_features=512, out_features=2, bias=True)
)
```



Weight tensors for first layer



Loss vs Epoch



Accuracy vs Epoch

Result and Analysis

Training CNN on our sampled dataset gave best validation accuracy of 92.27%, which is better than any of the models we tried previously, hence validating our choice of implementing CNN. Next we trained the same architecture on original complete dataset for 10 epochs, which reported a testing accuracy of 94.64%. Hyperparameter tuning and model refining could not be attempted due to resource constraints, demanding enormous training runtimes.

CNN goes through two major transformations. Convolution is the first method, in which pixels are convolved with a filter or kernel. Subsampling is a second significant transformation that can be of several forms (max pooling, min pooling, and average pooling) and employed as needed. The pooling layer is responsible for reducing the data's dimensionality, and it's quite useful for reducing overfitting. The output can be passed to a fully connected layer for efficient classification after employing a combination of convolution and pooling layers.

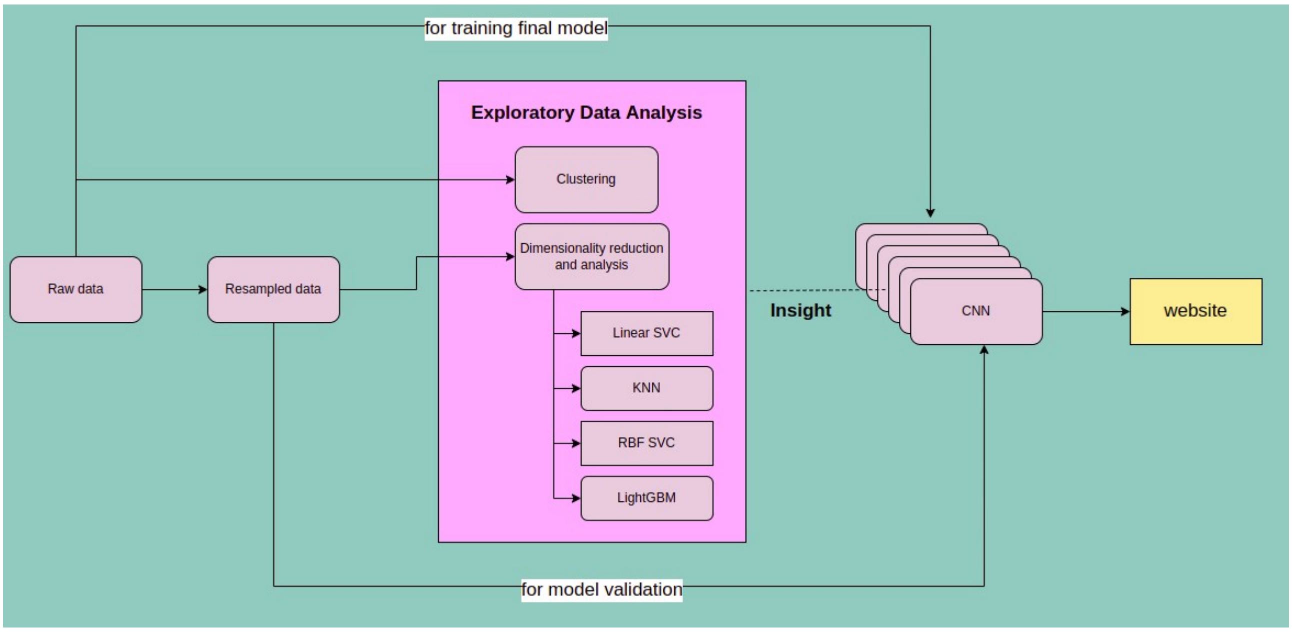
The model was deployed on web and can be accessed here.
Github repo for the code files can be found here.

Contributions

Learning, planning and implementation was done as a team. Individual contributions can be mapped as follows:

- **Abu Shahid** : Report, Readme, CNN, Website
- **Anirudh Srikanth** : Website, PCA, LDA, SVC, KNN, LightGBM
- **Atharva Pandey** : CNN, Resampling, Clustering, BenchMarking

Overview



References

- Cancer diagnosis in histopathological image: CNN based approach
- A Survey on Sampling and Profiling over Big Data
- Why are Convolutional Neural Networks good for image classification?