Project Report for CIS 419/519- Applied Machine Learning

Roshan Santhosh [1] **Rutuja Moharil** [2] **Siddharth Singh** [3]

Abstract

The proposed project aims to develop a robust framework for detection of cancerous lymph nodes cancer pathology whole slide images (WSI). The images are taken from the histopathological scans of lymph node sections and provide tumor visualizations of tumor tissues. The proposed methods include Deep Learning based approaches to create classifier frameworks which perform end to end data set manipulation, training, test augmentation and detection. We aim to deal with the various discrepancies in the images and the dataset by taking various data pre processing and augmentation methods. The aim is to perform the classification with high degree of confidence.

1. Introduction

The Health Industry is witnessing a revolutionary change with the advent of Data driven learning techniques. Using various Machine Learning and Deep Learning based approaches the Health Personnel are now able to take better decisions.
The proposed project aims at a very specific problem statement in the health industry. The proposed idea is to supplement the decision making process of Physicians in detecting cancerous tissue and ultimately making the diagnosis for the patient.
Given the high risk nature of this diagnosis it becomes extremely important that the classification pipeline being built is extremely robust and is able to handle various discrepancies in the training data and the test data as well.
To target this, a thorough analysis of the data was done to understand the discrepancies such as missing images and images with too much noise. Moreover, various data preprocessing and data augmentation techniques were used to bring more randomness and diversity to the data set. This helps in generalizing the classifier to a large extent and ultimately leads to better results. With the augmented and pre-processed data, various learning techniques were used and compared to make sure that the best classifier is chosen. Moreover, data augmentation and processing was also performed on the test data (and unseen data if and when to be classified) in order to have even more robust classification performance.

2. Data Description

The dataset consists of 220,026 color images (96 x 96px) extracted from histopathologic scans of lymph node sections. Each image is annotated with a binary label indicating presence of metastatic tissue.
The dataset is divided into a training set of 198,022 examples, and a validation and test set both of 11,002 examples.
A positive label indicates that the center 32x32px region of a patch contains at least one pixel of tumor tissue.

3. Pre-processing and Data Augmentation

The very first stage of data processing was to analysis the data and look for missing images. During that phase, it was noticed that along with missing images, there were also images which were of very low resolution and of very bad compositions. Such images turned out to be too dark to provide any information and were thus removed from the dataset. Failing to do would lead to faulty training of the model. .

Extensive Data augmentation techniques have been used during the training process increase the dataset size and to avoid over-fitting on training data. Following are the techniques used on the original images to create additional training data :

1. Flip, Scale, Shear, Rotate
2. SuperPixels, SimplexNoiseAlpha, ElasticTransformation
3. Sharpen, Emboss, Dropout
4. Hue Saturation Value conversation
5. Gaussian, Average, Median blur

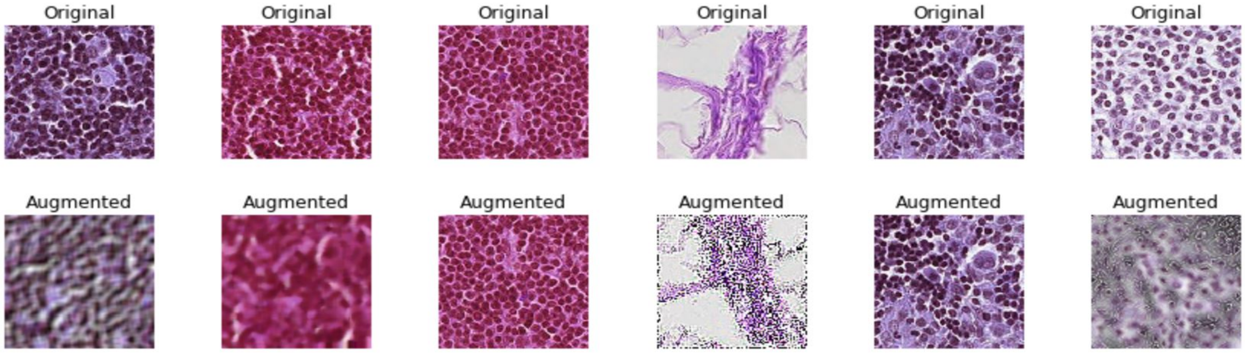


Figure 1. Data Augmentation

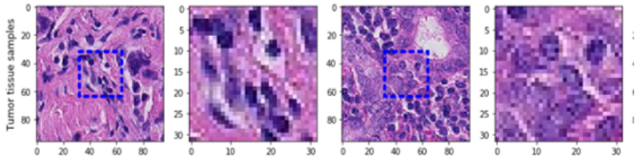


Figure 2. ROI Samples of Positive Images

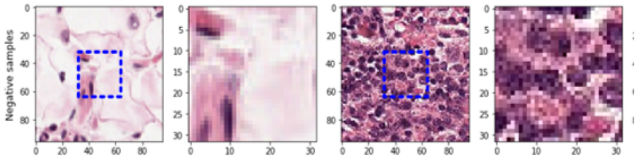


Figure 3. ROI Samples of Negative Images

4. Modeling

We have used various pre-built models for comparisons and have also developed a baseline CNN model to serve as a baseline model. The architecture of the model is depicted in *Figure 3*.

Moreover, we have employed Transfer learning over a ResNet Model to make use of both architectures.

The Split has been performed in a 80-10-10 manner for the training, validation and testing data set. The Validation accuracy provided insights for hyperparameter tuning along with the generalization error derived from the test accuracy.

For getting a more stable accuracy estimate, the modeling process was repeated 10 times for each model architecture and the scores were averaged from all runs.

Tumor tissue in the outer region of the patch does not influence the label.We randomly extract size patches from each WSI image , 1k normal 1k tumor patches from each Tumor slides and 1k normal patches from each Normal slides.The best test accuracy hyper-parameters were selected for the final model.

Layer	**Parameters**
Conv2D	64 channels, kernelSize = 3x3, stride = 1
BatchNorm2D	activation = ReLU
Conv2D	64 channels, kernelSize = 3x3, stride = 1
BatchNorm2D	activation = ReLU
MaxPool2D	poolSize = 2, stride = 2
Conv2D	128 channels, kernelSize = 3x3, stride = 1
BatchNorm2D	activation = ReLU
Conv2D	256 channels, kernelSize = 3x3, stride = 1
BatchNorm2D	activation = ReLU
MaxPool2D	poolSize = 2, stride = 2
Conv2D	256 channels, kernelSize = 3x3, stride = 1
BatchNorm2D	activation = ReLU
Conv2D	512 channels, kernelSize = 3x3, stride = 1
BatchNorm2D	activation = ReLU
GlobalAveragePooling2D	
Dropout2D	p = 0.3
Dense	units = 256, activation = ReLU
Dense	units = 1, activation = sigmoid

Figure 4. Base CNN Architecture

5. Training and Analysis

5.1. Extraction of ROI

Initial modeling pipelines used the entire 96x96 images for training the model. We also experimented with training alternate pipelines where instead of the entire image, only the ROI of 36x36 was used for training. Extraction of ROI gave great computational benefits but didnt show any significant performance benefits. For our final model, we chose to retain the entire image to make sure that in case there are certain features and image information which might be outside of the ROI but could help provide contextual information, then the information should not be lost.

5.2. Transfer Learning

In addition to building CNNs from scratch, we also utilised Transfer Learning approaches using ResNet and DenseNet architectures. One concern regarding using the TL approach is that the model were originally trained on ImageNet data, which has very different properties from the cancer slide images. Specifically, ResNet50, ResNet152 and DenseNet169 models were tried.

5.3. Ensemble Techniques

We also tried using model ensembling by combining the outputs of the 5 best performing DenseNet169 models and taking the maximum vote from each model. Vanilla ensemble with no weighting failed to provide a performance gain over the best performing single DenseNet169 model. However, assigning weights based on performance of the individual models, gave the best performing model, outperforming the best performing DenseNet169 model by 0.10.

5.4. Test Time Augmentation

During model testing, all images were passed through the same preprocessing steps to maintain similarity with the training images. In addition to this, we also utilised the Test Time Augmentation technique, which are meant to make model predictions more robust. For TTA, we use the same augmentation techniques used on our training data and create multiple copies of the same test image. The model is then used to predict the multiple images and a consensus of the label with the maximum votes from each augmented image is taken. This helps in improving the discrepancies that might arise during the prediction phase due to any sort of noise in the test images.

5.5. Adaptive learning rate

Initially we have used the reduceLRonPlatuea callback of Keras to implement an adaptive learning rate. If the loss over 1 epoch does not decrease, we decrease the learning rate by a factor of 0.1. It proved to be effective in improving the convergence and accuracy to some extent.

We also explored the use of the 1-cycle learning rate policy as suggested in [1].
In this method we use one cycle smaller than the total number of iterations of the epochs and in doing so we decrease the learning rate for the remaining iterations by some order.
Based on experimentation, we ccouldn't observe any significant improvements in performance using the 1-cycle learning rate policy.

6. Results and Performance Comparisons

Based on the various experimentations, we can see that the DenseNet169 is the best performing architecture, while the best overall performance is given by the ensemble of DenseNet169 models. One clear observation is the added benefit of using Data augmentations. Strong use of augmentations allowed the models to prevent overfitting as well as improve test data performance. Surprisingly though, Test Time Augmentation didnt provide the performance benefits we expected. This could likely be due to the use of the same strong augmentations that were used during training. Possibly the use of more subtle augmentations could have improved test data performance.For evaluation, we have used the Accuracy and AUC score metrics.

It is also visible that Data Augmentation is able to improve upon the validation accuracy and the test accuracy to support its usage.Despite more training epochs, the base CNN model could not outperform the Transfer Learning models (discounting data augmentations). We could also see that the deeper architectures gave better results since Resnet152 outperformed Resnet50.

The following table depicts the scores for the various models that have been tested and implemented.

Model	Train Acc.	Test Acc.	Test AUC
Base CNN	95.2	93.3	94.5
Base CNN(Aug)	94.8	95.8	96.7
ResNet50	96.3.4	93.1	94.8
ResNet152	97.4	94.3	95.6
ResNet152(Aug)	95.4	96.1	98.4
Densenet169	98.65	95.16	98.3
Densenet169(Aug)	96.9	96.79	99.42
Ensemble	-	96.89	-

The accuracy and loss plots for the frameworks show the general trend of the improvement in accuracy and decrements in the loss for both train and validation data. A notable trend is the overlapping and crossing of the loss and accuracy curves for the DenseNet model without augmentation.

The contrast in the learning curves for the models with and without data augmentations show the impact of data augmentations to control overfitting. Without augmentations, the model begins to overfit and hence we see the train accuracy crossing test accuracy and test loss going above train loss. However, with augmentations, the model is prevented from overfitting as it sees new images every batch. This prevents the training accuracy from going over test accuracy. Instead we see the test/train loss and

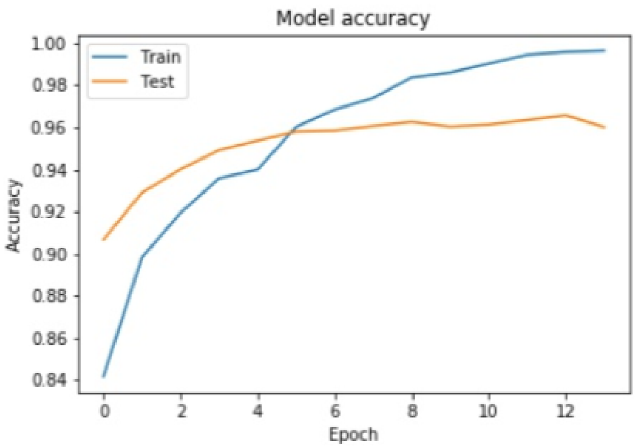


Figure 5. DenseNet Without Augmentation: Accuracy curve

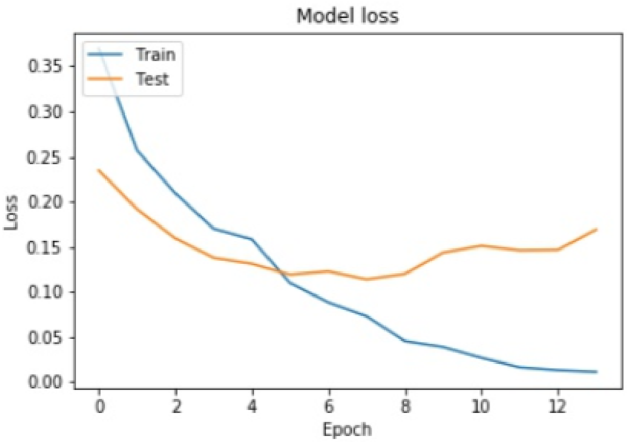


Figure 6. DenseNet Without Augmentation: Loss curve

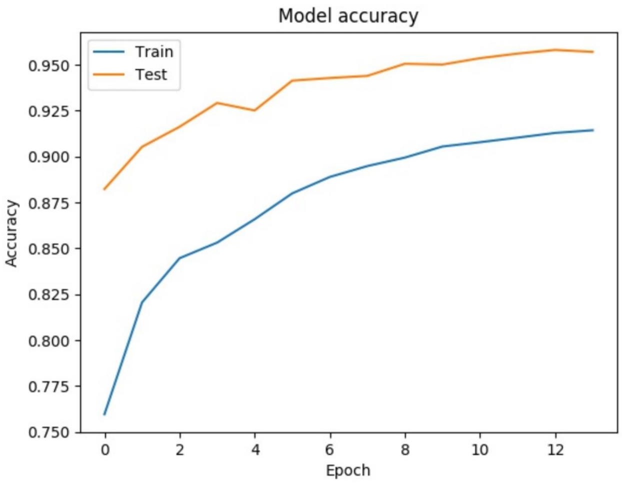


Figure 7. DenseNet With Augmentation: Accuracy curve

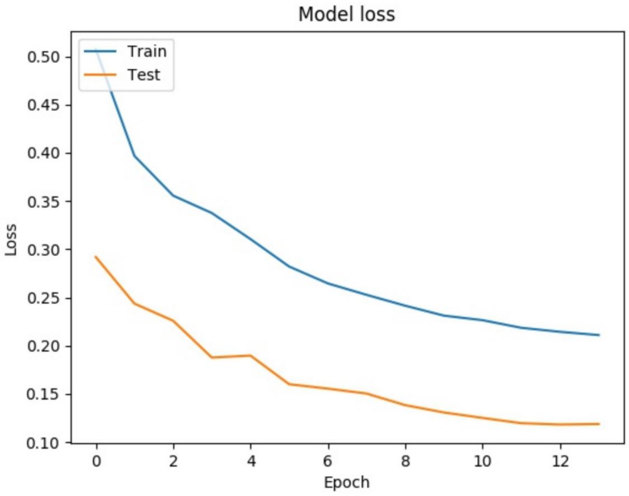


Figure 8. DenseNet With Augmentation : Loss Curve

accuracy curves plateau after certain epochs.

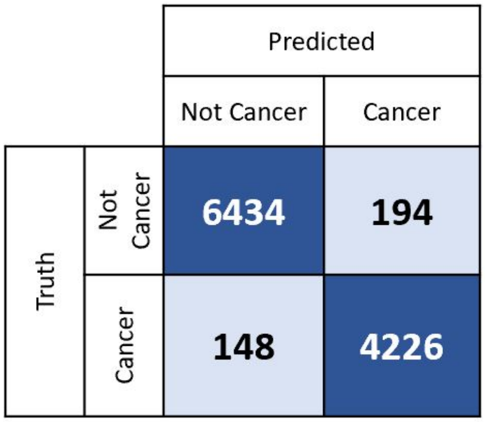
It can be seen in the Augmented data set that the trend of accuracy and loss is consistent.

7. Final Training Details

The details of the best performing model is given here. The complete model uses a DenseNet169 model as its main body and 1-layer fully connected layer as its classifier head. The output of the DenseNet169 model is passed through GlobalMaxPooling2D and GlobalAvgPooling2D layers and the output from these layers are concatenated before feeding to the fully-connected layer. The final model output is passed through a sigmoid activation layer. Model uses a Adam optimizer with learning rate of 0.0001 for the first 5 epochs and 0.00001 for next 10 epochs. Binary CrossEntropy is used as the loss metric with accuracy as the evaluation metric. Batch Size of 64 is used with each epoch taking 2000 steps for training and 200 steps for validation. A ReduceLROnPlateau callback is used to reduce learning rate with increasing epochs. For TL model, corresponding preprocessing functions provided by Keras are used to preprocess the images.

8. Conclusions

In its entirety, the given approach can be used as a complete pipeline to use raw image as an input and perform a highly confident classification. There is still scope for improvement as there is a 4% error rate. We would have to inspect the incorrect classifications to further understand the source of error as well as measures to circumvent that.



		Predicted	
		Not Cancer	Cancer
Truth	Not Cancer	6434	194
	Cancer	148	4226

Figure 9. Confusion Matrix of Ensemble model

9. References

[1] Smith, L.N., 2018. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv:1803.09820..

[2] Ehteshami Bejnordi B, Veta M, Johannes van Diest P, et al. Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. JAMA. 2017;318(22):2199–2210. doi:10.1001/jama.2017.14585

[3] Pan, S.J. and Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), pp.1345-1359.

[4] Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T. and Keutzer, K., 2014. Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869.

[5] Wu, Z., Shen, C. and Van Den Hengel, A., 2019. Wider or deeper: Revisiting the resnet model for visual recognition. Pattern Recognition, 90, pp.119-133.