Optimization of CNN model with hyper parameter tuning for enhancing sturdiness in classification of histopathological images

Anil Johny, Dr. K. N. Madhusoodanan, Dr. Tom J Nallikuzhy
Research Scholar, Department of Instrumentation, CUSAT, Cochin, Kerala, India
Professor, Department of Instrumentation, CUSAT, Cochin, Kerala, India
Dept. of Anatomy, SN Institute of Medical Science(SNIMS), Ernakulam, India
aniljohny@gmail.com

ABSTRACT: The field of pathology has advanced so rapidly that it is now possible to produce whole slide images (WSI) from glass slides with digital scanners producing high-quality images. Image analysis algorithms applied to such digitized images facilitate automatic diagnostic tasks whilst assisting a medical expert. Successful detection of malignancy in histopathological images largely depends on the expertise of radiologists, though they sometimes disagree with their decisions. Computer-aided diagnosis provides a platform for a second opinion in diagnosis, which can improve the reliability of an expert's opinion. Deep learning provides promising results compared to the conventional approach that relies on manual extraction of features which is time-consuming and labor-intense. Due to the huge size, whole slide images are converted into patches and trained using a Convolutional Neural Network (CNN), a variant of the deep learning model for images. Experimental results show that the proposed native model achieved patch wise classification accuracy of 92.8% and area under ROC curve 0.97 which is close to the values while comparing with the existing pre-trained models.

Keywords - Computer Aided Diagnosis (CAD), CNN, WSI, Deep Learning, Patch Classification, Histopathology images.

1. INTRODUCTION

Breast cancer is the most common invasive cancer in women and the second main cause of cancer death in women, after lung cancer [1]. Ultrasound, low-dose X-ray (mammogram), CT, MRI are non-invasive imaging techniques capable of producing 2D and 3D images of the breast. Histopathology which is an invasive method involves different surgical steps (biopsy) for microscopical investigation of tissue. Images obtained from pathological tissue provide histopathological images commonly referred to as Whole Slide Images (WSI). Histopathology image analysis is a gold standard for cancer recognition and diagnosis. Digital histopathology image analysis can help pathologists diagnose tumour subtypes, alleviate the workload of pathologists, and also improves the overall efficiency of routine diagnostic workflow [2]. The diagnosis and treatment in the early stages are essential to prevent the proliferation of the disease and reduce morbidity. Over the past decade, a dramatic increase in computational power and improvement in deep learning, especially Convolutional Neural Network (CNN)[3], has allowed the development of computer-assisted analytical approaches to the medical image analysis field, including histology images. CNN is a state-of-the-art technique for classification problems when the input consists of high-dimensional data such as WSI. Microscopically, cancer cells have distinguishing histological features. The nucleus is often large and irregular, and the cytoplasm may also display Atypia which shows clear structural differences between diseased tissues and normal tissues. There are many previous attempts to extract handcrafted feature representations, involving the labor-intense process. Due to the heterogeneous nature of breast cancer cells pathologist inspects a large number of tumour tissue slides which introduces different types of error in analysis. Contrarily to the hand-crafted feature extraction methods, CNNs learn features directly from the histopathology images. Moreover, global feature extraction allows the CNN model to extract more hidden features from the images [4], and classify them into a different class. Deep learning can be utilized to train a model and learn from labeled images, subsequently use the model to predict unlabeled histology images.

The methods based on handcrafted features[5] which are based on segmentation from selected areas(ROI) subsequently feeds different sets of features into traditional classifiers for classifying into benign or malignant class. A baseline method in [5] suggests a dataset that consists of histopathology images stained with Hematoxylin and Eosin (H&E) and achieved accuracy ranges from 80% to 85% for different magnification factors. Patch-based training and classification were used in [6] where the patch sizes are (32x32) and (64x64) using a sliding window strategy with a 50% overlap. The reported accuracies are 83.3% and 82.8% for patient-level and image-level respectively for the 200x magnification factor. In [7] authors suggest another method, in-between hand-crafted and task-specific CNN methods which reuses the pre-trained model to extract features and achieved an accuracy of 86.3% at the patient level for 200x magnification factor. A single task CNN model with a prediction on malignancy and image magnification was suggested by [8] and achieved an average recognition rate of 83.25% for the classification task. In [9] Patch-based sampling from WSI for the detection of invasive ductal carcinoma (IDC) the authors was able to achieve a balanced accuracy of 84.23%.

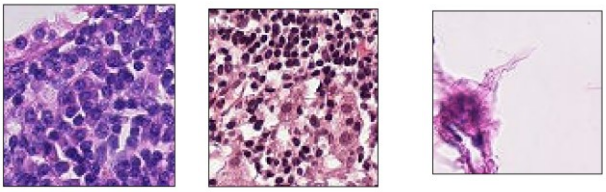
Whole slide images (WSI) based classification techniques encompass several roadblocks such as high implementation cost and insufficient productivity for high-volume clinical inspection and difficulty in retrieving very high-resolution images. Training CNN model using WSI requires a large memory footprint for computation while training using downsampled WSI may result in losing some distinctive features. Benign and malignant cells are differentiated by the various structural information such as the shape and size of nuclei and other information depending on different stages of metastases.

1.1. Characteristic of Benign neoplasms

A benign tissue appears like normal cells from which it originated, and has a slow rate of growth. Benign neoplasms do not invade encompassing tissues and they do not metastasize. The benign characteristics include slow growth, similarity to the tissue of origin, circumscription, lack of invasion, and absence of metastases.

1.2. Characteristics of Malignant neoplasms

A malignancy consists of cells that appear less like the normal cells of origin. It has a high rate of proliferation and can invade and metastasize rapidly. Malignant neoplasms derived from epithelial cells and those derived from mesenchymal cells are known as carcinomas and sarcomas respectively. The characteristics of malignant neoplasms include a rapid increase in size, less differentiation, a tendency to invade close tissues, and the ability to metastasize to distant tissues. Upon inspection, a medical expert looks for certain traits of atypia in WSI which are characterized by certain abnormalities that distinguish healthy and unhealthy tissues. Training a CNN model using whole slide images (WSI) directly from any database using CNN encounters the following limitations. Firstly the size of WSI is large (1024x1024) which cannot be fed to deep learning models directly. Second, WSI contains regions that are neither malignant nor benign, referred to as parenchymal tissues, which must be removed from images or it can be included in benign class optionally. Third, the variation in depth of staining between different slides affects training performances largely. In this work, a native CNN model is proposed and trained using the patches from the standard database PCam [10], which is task-specific and provides competitive results. A trade-off between training time and better accuracy is achieved by fine-tuning model hyperparameters.



(a) Benign patch; (b) Malignant patch; (c) Adipose patch

Fig. 1 – Different patches from single WSI image (best viewed in color)

Fig. 1 represents benign, malignant, and adipose patches from a single WSI image whose label is known, and patch level labels are not available. Considering the above constraints, a patch-based dataset PCam is used for our experiments to train all the models.

The remaining sections of the article is organized as follows: Section II portrays dataset and evaluation metrics. Section III describes the methodology followed in the work. Section IV discusses the experimental results. Section V concludes with various insights to further research.

2. DATASET AND EVALUATION METRICS

Evaluation metrics used in the work are listed in Table 1. AUC value of the ROC curve is used for evaluating a binary classification model where True Positive Rates (TPR) is plotted against False-Positive Rates (FPR) for different thresholds. It is also used here to find the best model which has good prediction performance.

Table 1 – Evaluation metrics used

Metrics	Definition	Range
Accuracy	$\mathrm{Acc} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$	(0,1)
Precision	$\mathrm{Pr} = \mathrm{TP/TP + FP}$	(0,1)
Recall	$\mathrm{R} = \mathrm{T/TP + TP + FN}$	(0,1)
F1-score	$\mathrm{f1} = 2 \times \frac{\mathrm{Pr} \times \mathrm{Re}}{\mathrm{Pr} \pm \mathrm{Re}}$	(0,1)

Often it is cumbersome to separate the background from whole slide images as it is time-consuming and requires expert annotations. For example, when a whole slide image labelled as malignant is divided into patches, it consists of images of both the malignant portion, which is our area of interest, as well as image patches of background and adipose tissues which are now under the malignant label. This creates error while training and classification unless it is separated, as the whole image is categorized into one main class, i.e. malignant. The background images which are either normal or adipose tissues if included in the same class where the image belongs inherently affects model training and loss will be very high. Removing the background patches manually requires careful expert annotation of the Area Of Interest (AOI) in WSI and removal by cropping or any segmentation algorithm. This is inevitable to train and optimize the patch classifier model through hyper-parameter tuning, standard database is crucial as the impact of false positives will get eliminated in this step. PCam repository contains 400 H&E stained WSIs of sentinel lymph node sections. The slides in the database were acquired and digitized at 2 different centers using a 40x objective (pixel resolution of 0.243 microns) which is then under-sampled at 10x to increase the field of view [10]. Train / test split from the Camelyon-16 challenge is also followed in the database, and further hold-out 20% of the train WSIs for the validation set. A novel method is proposed to train a patch classifier (PCF) model using PCam repository which can separate background information from labeled WSI, thus separating patches containing malignancy and normal patches.

3. METHODOLOGY

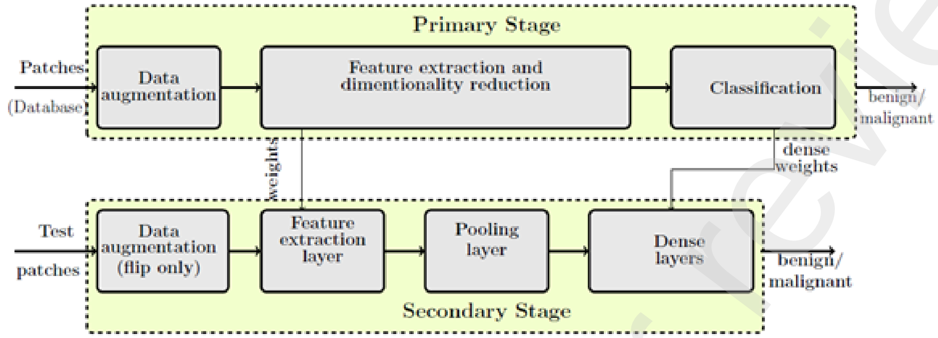


Fig.2. Block diagram of the proposed method

Figure 2 represents the block diagram of the proposed patch-based training network. Model parameters such as the number of hidden layers, network activation function are defined beforehand by heuristics. Since the available pre-trained models require a large memory footprint and computational complexity, a native CNN model is proposed for the work. The proposed work has two stages. The primary stage focus is to replace the random weights and initializes in which the results in vanishing or exploding the gradient. In this stage, the range of hyper-parameter values and consistency among several models are also estimated as a prerequisite for the next stage which cannot be monitored together for optimal settings. This stage eliminates the need for parameter initialization and estimation by trial and error method which in-turn hastens subsequent training stages by converging to local minima within a reasonable time. In the secondary stage, training is performed by selecting the learning rate, batch size and several patches by analyzing the results in the first stage of training from different models and well within the range. This supports the fine-tuning process of training by setting the hyper-parameters till the accuracy is improved. The optimal learning rate for this stage is found to be between 0.01 and 0.001 from Fig. 3c. This stage surpasses the limitations of gradients that vanishes and is trapped at local minima during training which is the effects of very high learning rates. Mini-batch gradient descent with default values is selected in both stages since it divides the training set into smaller batches and updates the model parameters for each iteration. The optimizer Stochastic Gradient Descent(SGD) [11] – [13] is also implemented as it estimates the error gradient for the model from the training dataset and updates the weights of the model using a back-propagation algorithm with momentum set to the default value. The two stages are performed separately with the same loss functions.

3.1 CNN model architecture

The input layer of the CNN model consists of 96x96 RGB square patches which are preprocessed and augmented before training using data-generator in Keras[14], an open-source neural-network library written in python. There are three main layers in the proposed CNN architecture namely convolutional layer, pooling layer, and fully connected layer. Table 2 shows the detailed architecture of various models considered. There are three convolution layers in the model which extract the local and global features of images in the training set. The kernel scans through the input data and extracts the features in strides which is the number of steps a kernel takes each time it hovers over the input data. The activation function used for all convolutional layers are ReLU [15] with $f(x) = \max(0, x)$ and Softmax for top layers[16]. For all the pooling layers the stride is set to 1x1 and it is found that it works better than 2x2 in terms of performance. Max-pooling layers are introduced to reduce dimensionality and computational complexity using downsampling which affects the model performance otherwise. Flattening is applied before dense layers which are fully connected layers used to process the information available from the pooling layers using a soft-sigmoid function to provide the classifier output. Two fully connected dense layers with one output perform the classification whereas convolutional layers perform feature extraction. Training and testing of patches are implemented in python3 and the model is saved as '$saved model.h5$' format so that it can be loaded in a system with a lower configuration and optimized for classification of patches which in turn facilitates whole image classification. It is worth mentioning that the size of saved model weights is well below 10 MB. Data augmentation [16] is done, which generates new data from original data, to suffice the size of input training dataset samples and thereby increasing the performance of deep neural networks. Native model performance improves with various combinations of hyper-parameters for task-specific implementations.

Table 2 – Detailed architecture of models tested

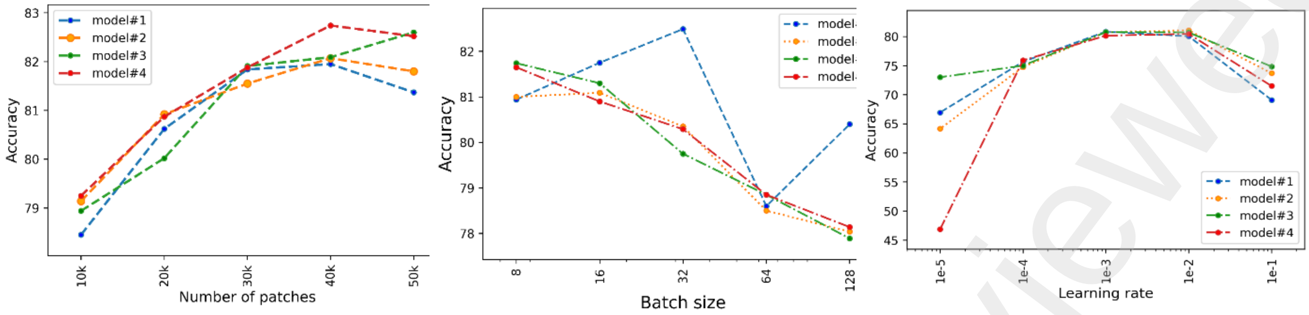
Models	Input shape	Conv1	Conv2	Conv3	Dense 1	Dense2	Output	Parameters
Model#1	96x96x3	9x9,30,3x3	4x4,100,1x1	3x3,100,1x1	200	200	1	451791
Model#2	96x96x3	7x7,30,3x3	3x3,100,2x2	3x3,100,2x2	100	100	1	376391
Model#3	96x96x3	10x10,30,4x4	4x4,100,1x1	2x2,100,1x1	300	200	1	511891
Model#4	96x96x3	8x8,30,2x2	5x5,100,2x2	5x5,100,2x2	200	50	1	421491

3.2 Model parameter selection

Optimal model hyper-parameters are obtained by varying different model parameters and analyzing its impact on the accuracy of each model. This main step is performed in the primary phase. Large batch size is often constrained by GPU memory availability for computation. For a batch size of 32, the model performs an update for every mini-batch training example thereby reducing the variation of parameter updates and leads to convergence.

3.3 Training of CNN model

In the primary stage, the model is trained from scratch based on the prefixed model parameters with a test-train split of 80:20 and the same ratio is followed in the second stage also for consistency. The model weights obtained from the primary phase is reused in the second phase to reduce training time. Training of the native model is performed by setting the obtained hyper-parameters mentioned in Section 3.2 to optimize the model. The optimized model will be well suited for the classification of benign and malignant patches as well as background patches from whole slide images. The model is trained for various epochs and the changes in the accuracies are plotted for different architectures keeping other parameters such as the number of patches, learning rate, and batch size fixed. Figure 4 and Fig. 3c show the variation in accuracy for different architectures while changing epochs and learning rates respectively, keeping other parameters unchanged.



(a) Accuracy changes for varying number of patches; (b) Accuracy comparison for different batch sizes; (c) Accuracy comparison for different learning rates

Fig. 3 – Performance analysis of various models

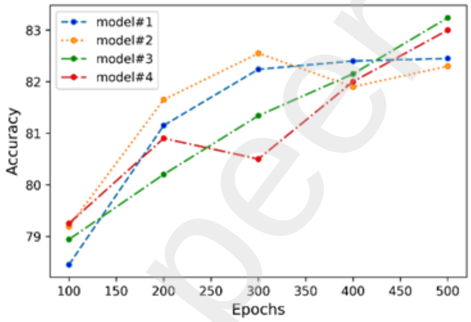
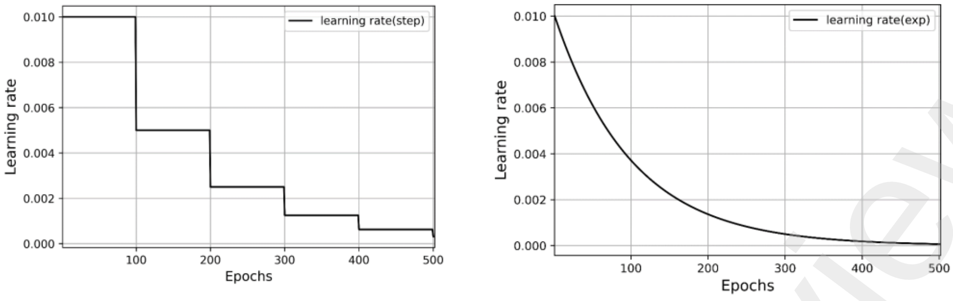


Fig. 4 – Model performances at different epochs

Increasing the number of patches do not contribute much to the accuracy since the augmentation creates several images while training undergoes, by flipping and rotating. Reduction in batch size improves accuracy but the computation time for the training process increases drastically. By setting these hyper-parameters a trade-off can be reached between accuracy and time required for training. The learning rate is varied for improving the performance of the model at different stages of training in different models whilst it updates weights after each step during training. The high learning rate employed in CNN model training may cause over-fitting. The various learning rate are implemented in this work termed as the constant type, time-based, step change, and exponential learning rates as shown in Fig.5.



(a) Step decay; (b) Exponential decay

Fig. 5 – Different learning rate strategies used

The step size is varied for every 100 epochs in both learning strategies and the upper and lower limits of learning rates are 0.1 and 0.001 respectively.

$$\mathrm{lr} = \mathrm{lr}_{(\mathrm{init})} \times 1/(1 + (\mathrm{decay} \times \mathrm{iterations})) \qquad (1)$$

Eqn (1) represents the variation in the learning rate with a change in decay. $\mathrm{lr}_{(\mathrm{init})}$ is the initial learning rate and decay is the amount by which the learning rate, lr is reduced after each epoch.

$$\mathrm{lr} = \mathrm{lr}_{(\mathrm{init})}) \times \mathrm{F}^{(1+\mathrm{E})/\mathrm{D}} \qquad (2)$$

In the above Eqn (2), F is the factor by which the rate is controlled for every time epochs drop by D and E is the current epoch. For a low value of F, learning rate decay will be fast.

$$\mathrm{lr} = \mathrm{lr}_{(\mathrm{init})}) \times \mathrm{e}^{-\mathrm{kt}} \qquad (3)$$

In Eqn (3) the hyper-parameters 'k' and 't' are varied for applying different learning rates during training. The time-based strategy follows a built-in function in Keras where it decreases the learning rate from the previous epoch by a fixed amount and depends on decay. Step-based decay reduces the learning rate after a fixed number of epochs and exponential decay reduces the learning rate exponentially as epochs increase.

Different learning strategies are implemented based on Eqn (1), Eqn (2), and Eqn (3). Learning rate strategy and learning rates are case-specific and are cautiously selected as higher learning rates causes instability in model training whereas more training time is required for lower learning rates. From the graph, the final accuracy value for the fixed learning rate is substantially higher than other learning rate modalities which show suitability for subsequent training of models. The same learning strategy is followed in the second stage also for better comparison.

4. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

Training and validation accuracy of different models are compared initially by varying epochs, batch sizes, learning rates, and several patches. Note that only one parameter is varied at a time while the remaining are fixed. Estimation of different ranges of hyper-parameters and model parameters is obtained from this step which eliminates exhaustive methods like random search [17]. Figure 4 shows the variation of accuracy for different models as the epochs are changed. A linear relation between the number of epochs and accuracy is observed in a few models which lack in other models due to architectural dissimilarities. Figure 3(a) shows the variation of accuracy for different models for a varying number of patches. The accuracy of models does not improve by increasing the number of patches beyond 40,000. Change in accuracy for different models as batch size changes are shown in Fig. 3(b). Model performance for different learning rates is shown in Fig.3(c) which infers instabilities while the learning rate is high or too low. The response of the model to different learning strategies are evaluated and identified as the best learning modality for the experiment. Figure 6 shows that for a fixed learning rate, the accuracy improves while for exponential and step-change learning rate, the curve stagnates as training epochs progress. The obtained values of accuracy, precision, recall, f1 score, and AUC before and after tuning are listed in Table 3. An increase in accuracy of 3.5% is obtained for various models under test as shown in Fig. 7. Similarly, an increase up to 3.6% for precision, 4.5% for recall, and 4.7% for F1-score are observed. The plots in Fig.8 show the variation in accuracy, precision, recall, f1-score, and AUC after tuning, for all the models for visual comparison. Figure 9 shows the ROC curve for all the models.

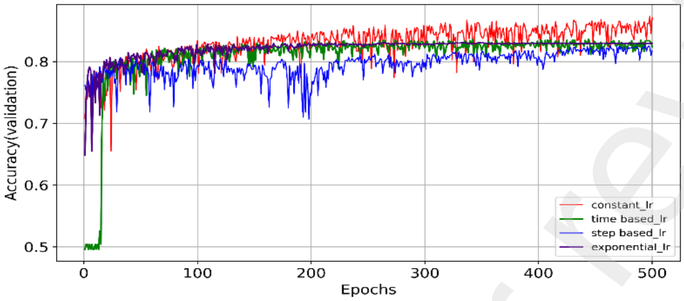


Fig. 6. Accuracy curves of single model for different learning strategies

The performance of the models is analyzed by plotting the ROC curve which shows the AUC of the model which has the highest accuracy. Here variation in FPR is plotted against TPR for different thresholds. The highest AUC value obtained is 0.98 which shows the efficiency of the proposed model. The graph shows the capability of proposed models to differentiate benign from malignant patches. The selection of the best model is done by analyzing the different parameters aforementioned.

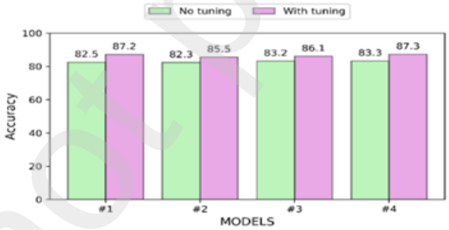
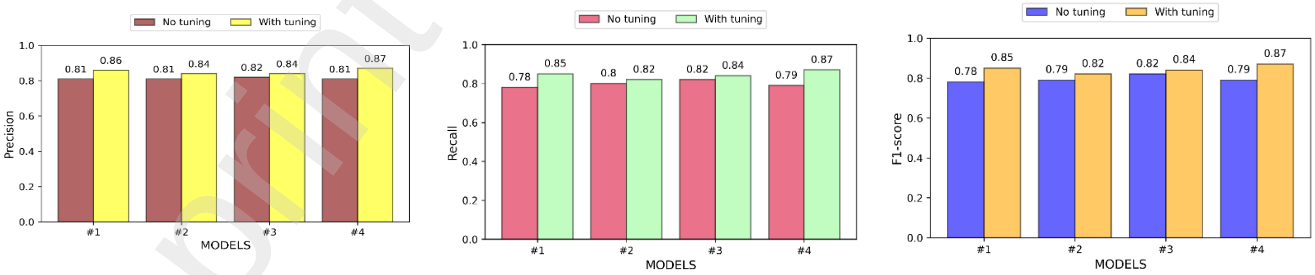


Fig. 7 – Accuracy of various models before and after tuning



(a) Comparison of precision; (b) Comparison of recall; (c) Comparison of F1-score.

Fig. 8 – Accuracy of various models before and after tuning

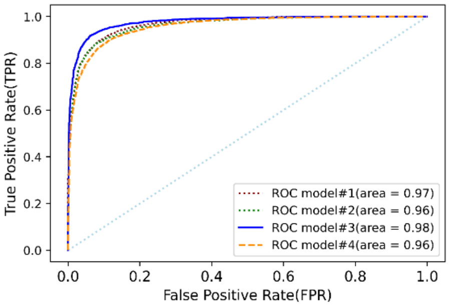


Fig. 9. Comparison of ROC curve and AUC for different models

Table 3 – Performance results of various models before and after tuning

Model	**Before tuning**					**After tuning**				
Model number	**Accuracy**	**Precision**	**Recall**	**f1 score**	**AUC**	**Accuracy**	**Precision**	**Recall**	**f1 score**	**AUC**
Model#1	0.8245	0.809	0.821	0.7847	0.9	0.8718	0.8561	0.8445	0.8535	0.94
Model#2	0.823	0.7865	0.8155	0.7995	0.9	0.8546	0.8538	0.8445	0.8228	0.92
Model#3	0.8324	0.8119	0.8089	0.8156	0.9	0.8611	0.837	0.872	0.8445	0.92
Model#4	0.833	0.8	0.7995	0.7992	0.9	0.8733	0.8246	0.8713	0.8713	0.94

Table 4– Best performance results of all models

Model	**Accuracy**	**Precision**	**Recall**	**f1 score**	**AUC**
#1	**0.9286**	0.8789	0.8557	0.8543	0.97
#2	0.9228	0.8928	0.8818	0.880	0.96
#3	**0.9247**	**0.9224**	**0.9213**	**0.9212**	**0.98**
#4	0.9039	0.8918	0.8917	0.8918	0.96

The best results of all the models after tuning hyper-parameters are summarized in Table 4 for 1k epochs. This shows that the performance of all models has improved after tuning with batch normalization applied at the intermediate phase of the secondary stage except fully connected (dense) layers without dropout. Normalizing input layers reduces internal covariate shift [18] while training as mini-batches [19] and reduces over-fitting. The normalization of each scalar feature is performed independently instead of whitening the features in layer inputs and outputs together to make zero mean and the unity variance. This shows that increasing the epochs after tuning the model makes a predominant impact on performance measures. The proposed model is implemented in python3 using Tensorflow[20] and Keras library on a GPU based system with an Intel Core-i7 processor with 32GB RAM.

5. CONCLUSION

A novel native model is proposed for the classification of histopathology images. Four CNN models with different architectures are selected to identify the dependency of hyper-parameters in performance optimization. Training of the model is performed using two stages with hyperparameters and different learning strategies. The primary stage is for model parameter selection and the secondary stage is for tuning the model hyperparameters. The performance of the model is analyzed after tuning the hyper-parameters. The obtained values of accuracy, precision, recall, f1-score, and AUC after tuning show the improvement in the performance of the native model. The selection of the best model is also performed by analyzing different performance metrics. Accuracy of the model for different learning rates is obtained and plotted which shows constant mode with low learning rate as the best strategy. The ROC curve for different models after training is also obtained. The highest AUC value yielded is 0.98 which shows the efficiency of the proposed native model as patch classifier in differentiating benign and malignant patches. From the results, the model#3 outperforms other models when all the performance metrics are considered. A significant difference in performance is obtained after tuning all the models under consideration. The saved model with optimized parameters in '.h5' format can perform prediction on the histopathologicalimage using any low-end computing device with minimal hardware complexity.

REFERENCES

[1] F. BRAY, J. FERLAY, I. SOERJOMATARAM, R. L. SIEGEL, L. A. TORRE, AND A. JEMAL, "GLOBAL CANCER STATISTICS 2018: GLOBOCAN ESTIMATES OF INCIDENCE AND MORTALITY WORLDWIDE FOR 36 CANCERS IN 185 COUNTRIES," CA: A CANCER JOURNAL FOR CLINICIANS, VOL. 68, NO. 6, PP. 394–424, SEP. 2018. [ONLINE]. AVAILABLE: HTTPS://DOI.ORG/10.3322/CAAC.21492

[2] A. Cruz-Roa, H. Gilmore, A. Basavanhally, M. Feldman, S. Ganesan, N. N. Shih, J. Tomaszewski, F. A. Gonz´alez, and A. Madabhushi,"Accurate and reproducible invasive breast cancer detection in wholeslide images: A deep learning approach for quantifying tumor extent,"Scientific Reports, vol. 7, no. 1, Apr. 2017.[Online].Available: https://doi.org/10.1038/srep46450

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. https://doi.org/10.1038/nature14539

[4] A. Nahid and Y. Kong, "Involvement of machine learning for breast cancer image classification: A survey," Computational and Mathematical Methods in Medicine, vol. 2017, pp.

[5] F. A. Spanhol and L. S. Oliveira and C. Petitjean and L. Heutte, "A dataset for breast cancer histopathological image classification," IEEE Transactions on Biomedical Engineering, vol. 63, no. 7, pp. 1455–1462, July 2016.

[6] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "Breast cancer histopathological image classification using convolutional neural networks," in 2016 International Joint Conference on Neural Networks (IJCNN), July 2016, pp. 2560–2567.

[7] F. A. Spanhol, L. S. Oliveira, P. R. Cavalin, C. Petitjean, and L. Heutte, "Deep features for breast cancer histopathological image classification," in 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Oct 2017, pp. 1868–1873.

[8] N. Bayramoglu, J. Kannala, and J. Heikkil¨a, "Deep learning for magnification independent breast cancer histopathology image classification," in 2016 23rd International Conference on Pattern Recognition (ICPR), Dec 2016, pp. 2440–2445.

[9] A. Cruz-Roa, A. Basavanhally, F. Gonz´alez, H. Gilmore, M. Feldman, S. Ganesan, N. Shih, J. Tomaszewski, and A. Madabhushi, "Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks," in Medical Imaging 2014: Digital Pathology, M. N. Gurcan and A. Madabhushi, Eds. SPIE, Mar. 2014.[Online]. Available: https://doi.org/10.1117/12.2043872

[10] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling,"Rotation equivariant CNNs for digital pathology," Jun. 2018.

[11] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in COMPSTAT, 2010.

[12] H. Robbins, "A stochastic approximation method," Annals of Mathematical Statistics, vol. 22, pp. 400–407, 2007.

[13] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," 1952.

[14] F. Chollet et al., "Keras," https://keras.io, 2015.

[15] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel, J. F¨urnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 807–814. [Online]. Available: https://icml.cc/Conferences/2010/papers/432.pdf

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, p. 84–90, May 2017. [Online]. Available: https://doi.org/10.1145/3065386

[17] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," J. Mach. Learn. Res., vol. 13, pp. 281–305, 2012.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[19] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," ArXiv, vol. abs/1702.03275, 2017.

[20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zhang, "Tensorflow: A system for large-scale machine learning," in OSDI, 2016.