MJSAT

Malaysian Journal of Science and Advanced Technology

journal homepage: https://mjsat.com.my/

Hyperparameter Tuning in Deep Learning Approach for Classification of Classical Myeloproliferative Neoplasm

Umi Kalsom Mohamad Yusof[1], Syamsiah Mashohor[1,*], Marsyita Hanafi[1], Sabariah Md Noor[2], and Norsafina Zainal[3]

[1]*Department of Computer and Communication Systems Engineering, Faculty of Engineering Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia*

[2]*Department of Pathology, Faculty of Medicine and Health Sciences, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia*

[3]*Pathology Department, Hospital Serdang, Jalan Puchong, 43000 Kajang, Selangor, Malaysia*

KEYWORDS

Deep learning
Artificial intelligence
Histopathology images
Classical MPN

ARTICLE HISTORY

Received 25 July 2022
Received in revised form 3 August 2022
Accepted 4 August 2022
Available online 5 August 2022

ABSTRACT

Histopathology images are an essential resource for defining biological compositions or examining the composition of cells and tissues. The analysis of histopathology images is also crucial in supporting different class of disease including for rare disease like Myeloproliferative Neoplasms (MPN). Despite technological advancement in diagnostic tools to boost procedure in classification of MPN, morphological assessment from histopathology images acquired by bone marrow trephine (BMT) is remained critical to confirm MPN subtypes. However, the outcome of assessment at a present is profoundly challenging due to subjective, poorly reproducible criteria and highly dependent on pathologist where it caused interobserver variability in the interpretation. To address, this study developed a classification of classical MPN namely polycythemia vera (PV), essential thrombocythemia (ET) and primary myelofibrosis (MF) using deep learning approach. Data collection was undergoing several image augmentations processes to increase features variability and expand the dataset. The augmented images were then fed into CNN classifier followed by implementation of cross validation method. Finally, the best classification model was performed 95.3% of accuracy by using Adamax optimizer. High accuracy and best output given by proposed model shows significant potential in the deployment of the classification of MPN and hence facilitates the interpretation and monitoring of samples beyond conventional approaches.

1. INTRODUCTION

MPN are characterized by the abnormal proliferation of mature blood cell lineages. This group of disorders shares many overlapping characteristics in clinical presentations and laboratory findings of both reactive and neoplastic conditions. In the latest WHO classifications 2016 version, bone marrow morphology is fundamental importance for distinguishing between MPN subtypes [1]. This includes an assessment of the cellularity, the extent of differentiation in different blood lineages, any atypical cell morphology, blast percentage, degree of fibrosis and osteosclerosis [2-3].

Substantial scientific advances have been achieved regarding MPNs over the last twenty years, which included higher accuracy in diagnosis, new risk classifications, and therapeutic approach updates. 'The Classification of Hematopoietic and Lymphoid Tissue Tumours' of the WHO demonstrated significant importance in this purpose. The first document to incorporate genetic information into the diagnostic algorithms of MPN was published in 2001. The classification

*Corresponding author:
E-mail address: Syamsiah Mashohor < syamsiah@upm.edu.my >.

had its second publication in 2008 and a third one recently, in 2017[4]. In this latest edition, both genetic and molecular aspects and histopathological features from BMT have been reviewed and emphasized. Such approaches comprise a crucial step forward for the diagnostic standardization of MPNs worldwide.

The developing blood cells in the bone marrow are called hematopoietic cells. There are three main types of hematopoietic cells and each produces a different group of blood cells [5]. Cells from all three lineages, namely erythroid, granulocytic and megakaryocytic are found in normal BM biopsy as shown in Figure 1.

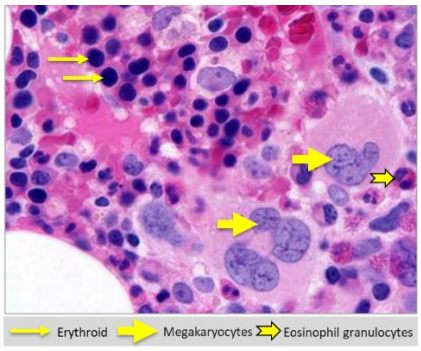

Fig. 1. Normal bone marrow biopsy [5].

The relative incidence of discrimination features according to WHO morphological criteria generating the histological pattern in performing BM biopsy specimens. Figure 2 represent the BM biopsy for ET, PV and MF with the example of morphology criteria to discriminate the type of diseases.

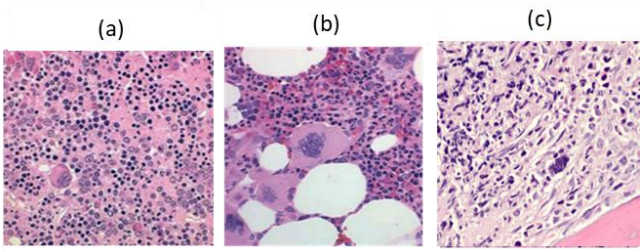

Fig. 2. (a) PV patient show a hypercellular marrow as a result of an increase in myeloid, erythroid, and megakaryocytic elements [6]. (b) Megakaryocytes in ET increased in size and number [1]. (c) The fibrotic stage of MF is marked by the extensive replacement of the normal marrow stroma by fibrous tissue [7].

AI approach in the classification of MPN was found the earliest in 2002 using data-mining to extract new decision rules in the diagnosis of PV [8]. More approaches using AI especially machine learning to identify MPN subtypes persisted for twenty years, including feature extraction [8], feature selection [9] and segmentation of BMT [10]. In more recent study, autoencoder neural network was applied to identify a feature set of megakaryocytes cytomorphology to distinguish MPN subtype with outcome of area under the curve metrics shows the result of 0.95 [11]. In the following year, study by [12] was implemented morphological feature extraction to discriminate the size of blood cells in pre-processing stage for early detection of MPN. The backpropagation network architecture was used as a classifier with a learning rate variation to produce the best accuracy rate of 91.82%.

Recently, deep learning was brought high impact in AI applications with the promising outcome and has been widely adopted in various fields [13]. The major advantage of deep learning is the algorithm is able to automatically extract meaningful information from the data source in the feature learning process and classify the disease accordingly [14-16]. Hence, the implementation of deep learning has the advantage of overcoming traditional handcrafted feature-based methods that could possibly missed out extraction of important features in the classification of MPN. To this extent, deep learning was presented in the classification of 24 types of cancers including MPN using whole genome sequencing samples in year 2020 [17]. Despite of that, the classification of MPN using BM morphology as per WHO guidelines are still lacking in execution. Therefore, a new study can be done using deep learning to improve the existing method for the classification of MPN.

2. MATERIALS AND METHODS

The whole study consists of a few stages: data collection from online resources with labelling validated by the hematologist, data cleaning stage, and data augmentation to increase the variety of datasets. Later, learning-based classification is performed between the control and reconstructed image complemented by cross validation. Finally, model evaluation was done to select the best MPN classification model. Figure 3 was prepared to show the organizational flow of the study.

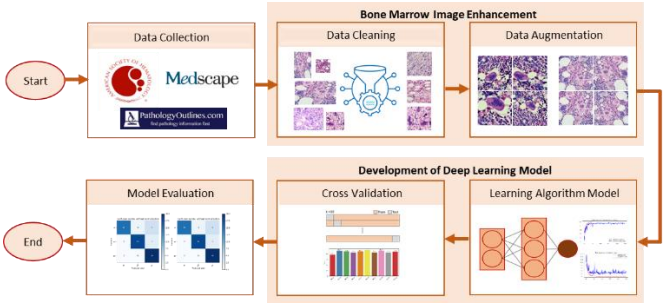

Fig. 3. The flow of the study.

2.1 Data Collection

In the advancement of the digitization era nowadays, the researcher can share a dataset that is foreseen to benefit the academic community. Every discovery and new finding can be shared globally through the digital platform. On top of that, anyone interested in the subject domain can access the information and contribute back through the platform. Similar circumstances are observed in the medical field, including in the haematology speciality. Three platforms that provide shared information and a costless downloadable image dataset with peer review from an expert are chosen as the data source for data collection. Those three platforms are The American Society of Hematology Image Bank, PathologyOutlines.com and Medscape.

Exploratory data analysis (EDA) is an approach to analyzing datasets to guide the modelling process. For that purpose, the entire data collected is then reviewed thoroughly to filter for similar images. A high possibility of the duplicated image can occur for a similar type of disease due to data sources coming from an open sharing dataset. Together, the description

of the image from the data sources also rechecked to confirm the inclusion and exclusion criteria. Dataset of images are excluded for evolved MPN and the output was not attributed to time, period or geographic location.

2.2 Bone Marrow Image Enhancement

At the early stage of model development, bone marrow image enhancement in the dataset was implemented as an intention to improve the quality of the original dataset and identify information content before further processing. Hence, exploratory data analysis and data cleaning, data augmentation as well as image reconstruction will be presented in this section.

After data collection and data cleaning process, data augmentation was applied to the whole dataset as a strategy to expand the size of a training dataset and enhance the variability of the images. Transforms include a range of operations from the field of image manipulation such as shifting, zooms, rotating flipping and shearing. Augmentation using bone marrow from ET class as shown in Figure 4 is demonstrated below to see the effect and suitability of value setting for each operation. Normal cellular bone marrow with higher number of hyperlobulated megakaryocytes are common features for ET was observed. These morphological features to differentiate subtype of MPN is important to retain in the augmentation process along with the target to increase the number of the dataset.

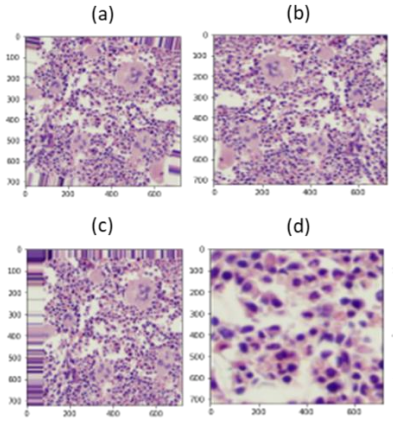

Fig. 4. Sample BMT image augmentation output using (a) rotate between 10 to 30 degrees, (b) flip horizontal, (c) shifting between value 0.1 and 0.2, and (d) zoom value between 0.3 to 0.5.

Following by final data checking, data split was applied by ratio 70:20:10 for training, validation and test accordingly. Finally, after data preparation was done, the training process in the development of a learning algorithm using a deep learning approach can be initiated in the next stage.

2.3 Development of CNN for MPN Classification

Moving forward to development of deep learning model, hardware and software part need to be prepared carefully since the architecture is well known desired to high computational power. Program will be run in Colab notebook provided by Google Colaboratory (Colab). The model is scripted in Python programming language (Python 3 version) using backend framework of Tensorflow and Keras. Colab is Jupyter notebook environment providing a runtime fully configured for deep learning and free of charge access to a robust virtual GPU. 12GB RAM are available with types of GPU vary over time. GPU often available include Nvidia K80, T4, P4 and P100.

CNN shows promising result in computer vision task for instance in classification of the image. Therefore, this deep learning model was proposed to classify the MPN subtypes. The model is separated into two parts which is feature learning and classification. Figure 5 shows the complete architecture of CNN and details of each phase are described below.

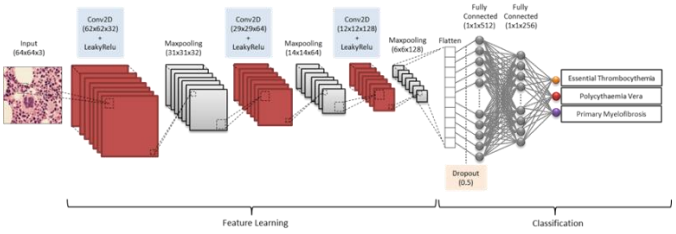

Fig. 5. CNN architecture for classification of MPN disease.

In feature learning, convolution layers were designed to extract important features from input images. The bone marrow image from pre-processing output becoming input for the CNN model with three channels to signified RGB. The normalization of the pixel value is applied after parsing the input image as float numbers. All the numbers are divided by 255 and make the pixel range between 0 to 1. This early process is important to produce lower numeric values and after all ease complexity in the computation of deep learning.

The first few convolutional layers often detect low-level features such as edges and geometries in the image. In this model, the first layer of the convolutional layer has a depth of 32 with 3x3 kernel size. The kernel is a filter that is composed of many weighted matrices defined by their width, height, and depth. The size of the receptive field of vision is determined by the dimension of width and height. While the depth is equal to the number of kernels in the convolutional layer.

Kernels will slide across the image from left-to-right, taking as input only a subarea of the image. Activation of filters occurs when the elementwise multiplication results in high, positive values. This informs the network of the presence of something in the image, such as an edge or blotch of colour. As an image passes through more convolutional layers, more precise details activate the layer's filters. At the end of convolution, the input image has been transformed into a stack of feature maps and the size of feature stack size is equal to the number of filters in the convolutional layer.

As the number of filters in the convolutional layers increase, the complexity of a dataset is also increase. So does the possibility of overfitting. To account for this, CNN have pooling layers after the convolutional layers. Pooling layers take the stack of feature maps as an input and perform down-sampling. Down sampling can be achieved with convolutional layers by changing the stride of the convolution across the image. Stride is the number of pixels that shifts over the input matrix and dictates how the pooling kernel moves across the feature map.

Subsequently, MaxPooling approach is specified in pooling layer. In MaxPooling, feature map dimensions are reduced by selecting the maximum value in the kernel window. The layers will take two input arguments: kernel width and height, and stride. The kernel starts at the top left corner of a

feature map, then passes over the pixels to the right at the defined stride. The pixel with the highest value in the kernel window will be used as the value for the corresponding node in the pooling layer. Kernel size 2x2 was specified in the pooling layer model with stride of 1 and returned the maximum value by 31x31x32. Later in convolutional blocks, the second convolution with dimension of 29x29x64 was added. Again, followed by pooling layer and dimensional was reduced to 14x14x64. As for the third convolutional layer and the network gets deeper, the architecture adapts to the high-level features. For instance, the kernel activations could be targeting pixel intensity and different splotches of colour within the image. This final convolution layer of 12x12 then flows to Maxpooling layer to produced 6x6 matrix with similar depth size of 128.

The fully connected layer in the classification part was set to have a complete connection to all the activations from the previous layers to get probabilities of the input in a particular image classification. The last 2-dimensional max-pooled matrix from the feature learning process was transformed into one dimensional array in a process called as flatten. Dropout layer with a value of 0.5 was added after flattening layer to randomly deactivates some neurons. Dropout is a mask that nullifies the contribution of some neurons towards the next layer and leaves unmodified the rest of the neurons. This layer is important to prevent overfitting the training data. Passing through two fully connected layers after dropout with depth 512 and 256 respectively, the final layer uses the Softmax activation function to get probabilities of MPN subtypes.

The hyperparameter tuning is a process to achieve a balance between underfitting and overfitting to strive for optimal development of a classification model. The input size and hyperparameter setting in the architecture of CNN is tune as variables in Table 1 and the performance of classification output is recorded for each variable tested. In the deep learning model, optimizer are algorithms or methods used to change network attributes such as weights and learning rate to reduce the losses in the difference between the predicted output and the actual output. Loss function will act as guide to the terrain, telling the optimizer if it is moving in the right direction to reach the global minimum. Five types of optimizers namely Adaptive Moment Optimization (Adam), Adamax. RMSprop, Stochastic Gradient Descent (SGD), and Adaptive Gradient (Adagrad) is tested to find the best suit to the classification model. Other than that, time taken for training to testing data set will also be consideration in evaluation of the model.

Table 1. Variables in hyperparameter tuning

Parameter	Variables
Input size	[32 x 32], [64 x 64], [128 x128]
Batch	30, 50, 80, 100, 120
Optimizer	Adam, Adamax, Adagrad, RMSprop, SGD

Each of model developed using variables in Table 1 were went through cross-validation process. Cross-validation is a resampling procedure used to evaluate classification models. In k-fold cross validation, a single parameter called k refers to the number of groups that a given data sample is to be split into as shown in Figure 6. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as in this study k=10 and becoming 10-fold cross-validation.

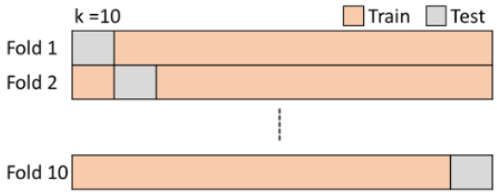

Fig.6. k-fold cross validation, k=10.

2.4 Evaluation of Models using Performance Metrics

The performance of the selected models from cross validation is investigate further by using evaluation metrics. This model evaluation aims to estimate the generalization accuracy on future unseen or real application data. Type of evaluation model includes accuracy, visualization of confusion metrics, precision, recall, F1-score, macro and micro average, Receiver-Operating Characteristic (ROC) and area under the ROC curve (AUC).

3. RESULTS AND DISCUSSIONS

In the development of the classification model, deep learning based using CNN architecture was constructed. Therefore, output of implementation of hyperparameter selection and cross validation output will be discussed in this section. Following the observation of model performance, the best classification model was chosen and evaluated further using performance metrics.

The number of batch sizes for observation are 30, 50, 80, 100 and 120. By taking 30 sizes of batch as example, 30 samples of images will be passing at a time to learning algorithm until eventually all the training data was transited to complete one single epoch. The process repeated until the end of number of epochs was setup in the model. Subsequently, 10-fold cross validation was applied and the output was recorded. As similar in tuning number of epochs, observation output from average of accuracy and loss together with time taken was tabulated for further analysis.

In input size 32x32, highest accuracy was given by Adam with 90.3% and batch size equal to 100 while for lowest accuracy was yielded by Adagrad with 44.8% and batch size equal to 120 as shown in Table 2. The best performance in tuning batch size was performed by Adamax in input size 64x64 with 94.6% of accuracy by applying batch size equal to 100. Lowest accuracy is still given by Adagrad using similar input size with 57.6% of accuracy and 120 of batch size. Lastly for input size 128x128, Adamax produced 91.7% of accuracy using 100 as batch size for maximum accuracy achieved in contrast with Adagrad to formed 61.4% of accuracy for the minimum using batch size of 120. In overall, Adam shows the best performance in smallest input size while Adamax outperformed the other optimizer when larger input size with similar batch size of 100 was applied. The lowest output consistently given by Adagrad for all input size at highest batch size of 120. Meanwhile, RMSprop and SGD shows average performance regardless input size and batch size tuned in training the model

Table 2. Average accuracy output from implementation of k-fold cross validation

Batch size	Input size	Average accuracy from cross validation (k=10)				
		Adam	RMSprop	SGD	Adamax	Adagrad
30	32x32	87.2%	79.0%	74.3%	87.2%	58.2%
	64x64	87.0%	87.6%	91.8%	94.6%	62.5%
	128x128	83.3%	78.9%	85.6%	90.6%	70.3%
50	32x32	87.5%	82.3%	79.9%	87.4%	56.9%
	64x64	88.8%	88.1%	91.2%	94.3%	62.4%
	128x128	83.4%	79.7%	85.9%	91.1%	69.6%
80	32x32	86.7%	83.9%	67.5%	86.5%	50.9%
	64x64	90.5%	88.9%	85.8%	94.4%	60.6%
	128x128	74.7%	78.8%	83.5%	90.0%	66.8%
100	32x32	90.3%	82.8%	62.9%	85.9%	46.3%
	64x64	91.1%	89.5%	82.0%	94.6%	60.4%
	128x128	86.4%	81.2%	83.8%	91.7%	63.6%
120	32x32	87.8%	84.0%	62.8%	86.4%	44.3%
	64x64	89.7%	85.9%	76.6%	93.8%	57.6%
	128x128	87.6%	79.6%	78.9%	89.2%	61.4%

Further analysis to compare performance of the optimizer based on accuracy and size of batch was presented in radar plot in Figure 7. Adagrad and SGD was plotted towards inside of the radar reflected by low accuracy output. Contrarily, Adamax spotted towards most outside of the radar given by high accuracy output. RMSprop and Adam alternately plotted in between as the accuracy shows an average performance. From the plot, clearly best performance from tuning batch size was given by Adamax for producing the best output among the other optimizers.

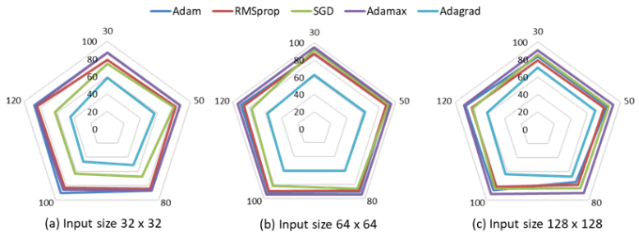

Fig. 7. Radar plot for tuning batch size in training model

Decreasing pattern of time taken was observed when batch size and input size getting larger as observed in Table 3. The reason is when number of batch size getting higher, more images will be processed in a batch and hence lesser time will be taken to complete one epoch. However, RMSprop was consumed highest time for almost parameter applied unless for 120 of batch size in 128x128 input size. SGD was train faster despite low accuracy output to indicate the model was not well converged. The rest optimizer for Adam, Adagrad and Adamax took almost similar time to train however different approach of formula in the model cause different performance output.

From the result of hyperparameter tuning and cross validation, the best output was shown by optimizer Adamax with input size of 64x64 and batch size is 100. The model was achieved 94.6% average accuracy from cross validation with 19.6 minutes time taken for training. Consequently, this model was selected as proposed classification model of MPN. Finally, the proposed model was evaluated further using performance metrics as presented in the next section.

Table 3. Time taken by tuning batch size

Batch size	Input size	Time taken (minutes)				
		Adam	RMSprop	SGD	Adamax	Adagrad
30	32x32	18.1	19.72	17.34	17.93	17.46
	64x64	29.49	32.94	23.87	29.73	28.65
	128x128	88.72	100.25	84.85	90.45	88.02
50	32x32	12.62	14.09	12.32	12.74	12.4
	64x64	23.82	25.94	23.52	24.36	23.86
	128x128	79.61	88.27	75.79	80.4	81.79
80	32x32	8.97	9.59	8.77	9.1	8.9
	64x64	21.24	22.53	20.62	21.46	20.95
	128x128	77.77	82.89	75.98	76.64	75.34
100	32x32	8.14	8.57	7.88	8.13	7.98
	64x64	19.52	20.57	19.14	19.6	19.88
	128x128	74.73	75.72	71.9	73.72	72.78
120	32x32	7.73	7.9	7.28	7.51	7.36
	64x64	18.88	19.88	18.74	19.01	18.79
	128x128	74.65	76.33	76.96	74.29	71.77

3.1 Evaluation of Classification Model

Referring to the classification report in Table 5, the accuracy achieved by the proposed model is 95.3% and slightly higher than average accuracy from cross validation output with execution training 2.33 minutes. Further evaluation, the highest precision was recorded for MF followed by PV and ET. Inversely, recall output was the best for ET as implied in the confusion matrix for the highest class correctly classified in the model. Then, expect for MF as second highest and the least is PV. Meanwhile, F1-score for this model shows the highest for MF for the most harmonic output between precision and recall. Consecutively, ET was found to have higher F1-score compared to PV. Further evaluation on macro average and weighted average, identical output for both parameters was observed and expected for balance dataset. In overall, slightly higher precision was found compared to recall. However, F1-score output shows high score to indicate good fit of the model to be deployed.

Table 5. Classification report for proposed model

	Precision	Recall	F1-Score	Total Images
ET	0.9317	0.9646	0.9479	198
MF	0.9895	0.9495	0.9691	198
PV	0.9397	0.9444	0.9421	198
Accuracy	-	-	0.9529	594
Macro Avg	0.9536	0.9529	0.9530	594
Weighted Avg	0.9536	0.9529	0.9530	594

Confusion matrix was plotted as shown in Figure 8 to evaluate performance of the model to classify each class. The matrix shows that classifier was the best to classify ET class. However, higher possibility for the model to confuse ET with PV compared to MF. Second best class for the classifier to identify is MF with misclassification occurred for PV is higher than ET. The least correctly classified is PV class. More than 90% of misclassification from PV was wrongly recognize as ET. This higher percentage could indicate the model has better separation to differentiate between ET and MF but low ability to split between ET and PV.

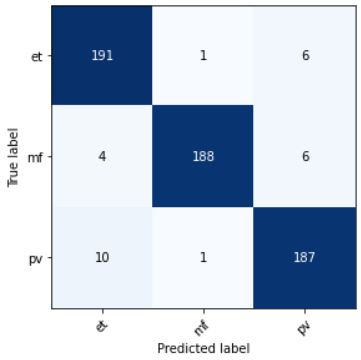

Fig. 8. The output of confusion matrix.

Based on ROC-AUC curve plot as presented in Figure 9, optimal output for AUC reflected by ROC curve with good value of probability to distinguish classes available in the model. The ROC curves for micro average and micro average as well as the other three classes shows bend towards top left corner of the graph with similar AUC value 0.99 for class 0 represent for ET and class 2 for PV. Meanwhile, class 1 which refer to MF class was achieved the best for AUC value 1.00.

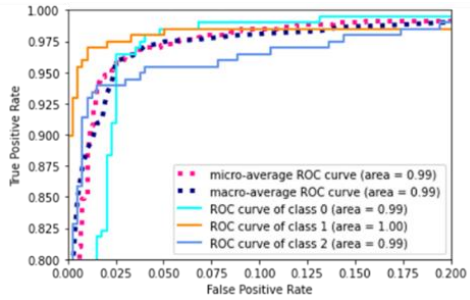

Fig. 9. ROC-AUC curve plot for tuning of batch size.

The proposed model yield highest accuracy to classify ET was found in parallel with incidence rate reported [18]. However, slightly lowered accuracy for MF and PV was performed from the study has become limitation on performance of the proposed model. Due to that, combination of clinical finding such as age, history of thrombosis, presence of constitutional symptoms, splenomegaly and laboratory investigations for instance complete blood count beside the bone marrow images examination are suggested to increase number of features that feed to the learning algorithm and reduce limitation on data availability. Therefore, more information will lead the model development to become more accurate and robust to classify MPN.

4. CONCLUSION

The proposed model with high accuracy output as 95.3% of accuracy on test data with 2.33 minutes of execution time for training. With less than a minute to predict the output, this advantage gives huge potential to deep learning for deployment compared to common clinical practice that require multiple procedures and high time consumption. Apart of successfully developed using deep learning approach, accomplishment of this study was also contributed to classification of MPN using bone marrow image as gold practice based on WHO guideline which was less reported in previous study.

REFERENCES

[1] A. A. and K. T. Michael Keng, "Cleveland Clinic Myeloproliferative Neoplasms - Virtual Medical School," *Cleveland Clinic Myeloproliferative Neoplasms*, Dec. 2013. https://teachmemedicine.org/cleveland-clinic-myeloproliferative-neoplasms/ (accessed May 17, 2021).

[2] W. J. Wong and O. Pozdnyakova, "Myeloproliferative neoplasms: Diagnostic workup of the cythemic patient," *Int. J. Lab. Hematol.*, vol. 41, no. S1, pp. 142–150, 2019, doi: 10.1111/ijlh.13005.

[3] H. Fujiwara, "Histological evaluation of myeloproliferative neoplasms," *J. Clin. Exp. Hematop.*, vol. 58, no. 2, pp. 45–50, 2018, doi: 10.3960/jslrt.18006.

[4] Swerdlow SH *et al.*, "WHO Classification of Tumours of Haematopoietic and Lymphoid Tissues 2017 - Revised 4th Ed, Volume 2 (IARC WHO Classification of Tumours) | Lymphoma | Leukemia," *IARC Publications*, 2017. https://www.scribd.com/document/376677186/WHO-Classification-of-Tumours-of-Haematopoietic-and-Lymphoid-Tissues-2017-Revised-4th-Ed-Volume-2-IARC-WHO-Classification-of-Tumours (accessed May 18, 2021).

[5] "Histology images of Bone marrow by PathPedia.com: Pathology e-Atlas." https://www.pathpedia.com/education/eatlas/histology/bone_marrow/Images.aspx?1 (accessed Sep. 12, 2021).

[6] Aaron T. Gerds, "Myeloproliferative Neoplasms," 2016. https://www.clevelandclinicmeded.com/medicalpubs/diseasemanagement/hematology-oncology/chronic-myeloproliferative-disorders/ (accessed May 17, 2021).

[7] "Latest Medical News, Clinical Trials, Guidelines - Today on Medscape." https://www.medscape.com/ (accessed Oct. 02, 2021).

[8] M. Kantardzic, B. Djulbegovic, and H. Hamdan, "A data-mining approach to improving Polycythemia Vera diagnosis," *Comput. Ind. Eng.*, vol. 43, no. 4, pp. 765–773, 2002, doi: 10.1016/S0360-8352(02)00138-9.

[9] V. C. Korfiatis, P. A. Asvestas, K. K. Delibasis, and G. K. Matsopoulos, "A classification system based on a new wrapper feature selection algorithm for the diagnosis of primary and secondary polycythemia," *Comput. Biol. Med.*, vol. 43, no. 12, pp. 2118–2126, 2013, doi: 10.1016/j.compbiomed.2013.09.016.

[10] T. H. Song, V. Sanchez, H. Eidaly, and N. M. Rajpoot, "Dual-Channel Active Contour Model for Megakaryocytic Cell Segmentation in Bone Marrow Trephine Histology Images," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 12, pp. 2913–2923, 2017, doi: 10.1109/TBME.2017.2690863.

[11] K. Sirinukunwattana *et al.*, "Artificial intelligence-based morphological fingerprinting of megakaryocytes: A new tool for assessing disease in MPN patients," *Blood Adv.*, vol. 4, no. 14, pp. 3284–3294, 2020, doi: 10.1182/BLOODADVANCES.2020002230.

[12] Z. E. Fitri and A. M. N. Imron, "Classification of White Blood Cell Abnormalities for Early Detection of Myeloproliferative Neoplasms Syndrome Using Backpropagation," 2021, doi: 10.1007/978-981-33-6926-9_43.

[13] L. Alzubaidi *et al.*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, vol. 8, no. 53. Springer International Publishing, 2021.

[14] J. Wang, H. Zhu, S. H. Wang, and Y. D. Zhang, "A Review of Deep Learning on Medical Image Analysis," *Mob. Networks Appl.*, 2021, doi: 10.1007/s11036-020-01672-7.

[15] M. Puttagunta and S. Ravi, "Medical image analysis based on deep learning approach," *Multimed. Tools Appl.*, 2021, doi: 10.1007/s11042-021-10707-4.

[16] J. Xu, C. Zhou, B. Lang, and Q. Liu, *Deep learning for histopathological image analysis: Towards computerized diagnosis on cancers*, no. 9783319429. 2017.

[17] W. Jiao *et al.*, "A deep learning system accurately classifies primary and metastatic cancers using passenger mutation patterns," *Nat. Commun.*, vol. 11, no. 1, pp. 1–12, 2020, doi: 10.1038/s41467-019-13825-8.

[18] C. Roaldsnes, R. Holst, H. Frederiksen, and W. Ghanima, "Myeloproliferative neoplasms: trends in incidence, prevalence and survival in Norway," *Eur. J. Haematol.*, vol. 98, no. 1, pp. 85–93, 2017, doi: 10.1111/ejh.12788.