

WBC-CNN: Efficient CNN-Based Models to Classify White Blood Cells Subtypes

<https://doi.org/10.3991/ijoe.v17i13.27373>

Najla Alofi, Wafa Alonezi, Wedad Alawad^(✉)
Qassim University, Buraydah, Saudi Arabia
wmaoad@qu.edu.sa

Abstract—Blood is essential to life. The number of blood cells plays a significant role in observing an individual's health status. Having a lower or higher number of blood cells than normal may be a sign of various diseases. Thus it is important to precisely classify blood cells and count them to diagnose different health conditions. In this paper, we focused on classifying white blood cells subtypes (WBC) which are the basic parts of the immune system. Classification of WBC subtypes is very useful for diagnosing diseases, infections, and disorders. Deep learning technologies have the potential to enhance the process and results of WBC classification. This study presented two fine-tuned CNN models and four hybrid CNN-based models to classify WBC. The VGG-16 and MobileNet are the CNN architectures used for both feature extraction and classification in fine-tuned models. The same CNN architectures are used for feature extraction in hybrid models; however, the Support Vector Machines (SVM) and the Quadratic Discriminant Analysis (QDA) are the classifiers used for classification. Among all models, the fine-tuned VGG-16 performs best, its classification accuracy is 99.81%. Our hybrid models are efficient in detecting WBC as well. 98.44% is the classification accuracy of the VGG-16+SVM model, and 98.19% is the accuracy of the MobileNet+SVM.

Keywords—deep learning, CNN, WBC subtypes, classification, transfer learning

1 Introduction

Blood is the life-sustaining liquid that courses through the whole body. The basic components of human blood are red blood cells (RBCs), white blood cells (WBCs), platelets, and plasma. The blood cells are produced in the bone marrow. Individuals might be influenced by a wide range of blood conditions and blood cancer. Common blood issues include anemia, oozing disorders such as: hemophilia, thrombus, and blood cancers such as: leukemia, lymphoma, and myeloma [1].

White blood cells play a significant role in the immune system of the human body. It is also referred to as a resistant cell. The WBC defends the body against epidemic diseases and extrinsic invaders. There are different types of WBC: eosinophils, lymphocytes, monocytes, and neutrophils, as appeared in Figure 1 [2]. All of the different types of WBCs have a part within the immune response.

A blood cell count is a frequently utilized routine in health examination, diagnosis, and determination of particular conditions of a patient. Depending upon which WBCs are lost, the patient is at risk for diverse sorts of infection. As a result, fast and accurate classification of WBC subtypes is critical for disease diagnosis.

The WBC classification has already been discussed in various studies. One of the most promising approaches to detect and classify the WBC is deep learning. To the best of our knowledge, we still need powerful WBC classification approaches. The aim of this study is to present efficient deep learning-based models to classify WBC subtypes. We have built six different models, two of them are fine-tuning CNN models while the rest are hybrid CNN-based models.

The rest of this paper is organized as follows: Section 2 presents a review of relevant studies. Section 3 discusses the methodologies used in detail. Section 4 presents the results of our models. Section 5 discusses the results of proposed models. Section 6 concludes the paper by mentioning the most important points. Finally, Section 7 proposed some ideas that could be implemented in future.

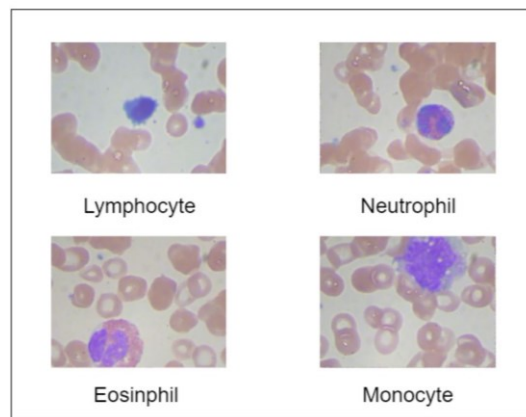


Fig. 1. Subtypes of White Blood Cells [2]

2 Related work

Many researchers have proposed different machine learning based models for blood cells classification tasks. These models differ in the algorithms used for feature extraction and classification. Some models are standalone CNN-based, in which the same CNN architecture is used for classification and feature extraction tasks. Tiwari et al, proposed a stand-alone CNN- based model to classify white blood cells. The accuracy of their approach is acceptable for the binary classification, but it is not the case for the multi-classification. The binary classification accuracy (polynuclear and mononuclear) is 94%, and the multi-classification (eosinophil, lymphocyte, neutrophil and monocyte) is 78% [3]. To enhance the multi-classification accuracy, Daouda Diouf et al, employed a deep convolutional neural network (CNN) model to classify four cell subtypes, 95.3% is their model's accuracy [4].

In [5], a fine-tuned model to classify blood images into health and Unhealthy was proposed. This model depends on the AlexNet architecture for both feature extraction and classification tasks. 100% is its classification accuracy. Furthermore, six CNN models were presented to classify malaria images as healthy and parasite in [6]. The Convolutional Neural Networks (CNN) architectures that were used to develop these models are ResNet50, AlexNet, GoogleNet, DenseNet201, VGG19, and Inceptionv3. Among these models, the best classification accuracy was obtained by the DenseNet201- base model, its classification performance is 97.83%. Additionally, Yusuf et al, used capsule networks to classify the WBCs into five categories, their model’s accuracy is 96.86% [7]. Table 1 illustrates Summary of CNN models that were presented for classifying Blood cells.

Table 1. Summary of CNN models for classifying blood cells

Ref.	Feature Extraction	Classification	Accuracy	Dataset size	Classes
[3]	Double Convolution Layer Neural Networks (DCLNN)		Polynuclear and Mononuclear: 94% Eosinophil, Lymphocyte, Neutrophil and Monocyte: 78%	13k	2 & 4
[4]	CNN		95.3%	1600	4
[6]	DanseNet 20		97.83%	6730	2
[7]	Capsule Network		96.86%	263	5
[8]	CNN		88.25% 81.74%	2478 & 3843	2 5

The other type of classification models that were presented in related studies is hybrid machine learning models, in which the machine learning algorithm used for classification tasks is different from that used for the feature extraction task. In [5], several different hybrid models were proposed to classify blood images into healthy or unhealthy, researchers used AlexNet for feature extraction and SVM, LDA, Decision Tree, and KNN for classification. The best classification accuracy, which is 99.79%, was obtained when the SVM was used for classification [8].

Another technique to build hybrid models is using CNN architectures for classification and different algorithms for feature extraction. One of the papers that apply that method is [9]. In this paper, authors used Region of Interest (ROI) for feature extraction and the Softmax architecture for classification to classify blood images into 6 classes. 99% is the classification accuracy of their model. Some authors presented non-CNN hybrid models to classify blood images. In [10], the blood images classification model was developed using SVM for feature extraction and KNN for classification, its accuracy is 93%. Another model was presented in [11], authors in this study used the Pseudo-Zernike (PZ) Moments as textural feature extraction from the images and SVM for the classification, 97% is the classification accuracy of this model. Table 2 shows Summary of hybrid models that were presented for classifying blood cells.

Table 2. Summary of hybrid models for classifying blood cells

Ref.	Feature Extraction	Classification	Accuracy	Dataset size	Classes
[12]	K-means	SVM	Best: 99.67%	1710	8
[11]	Pseudo-Zernike (PZ) Moments	SVM	97%	201	2
[13]	Domain-Invariant	Multi-target Domain Adaptation	98.09%	50k	5
[5]	SVM	CNN(AlexNet)	99.79%	2820	2
[10]	SVM	k-Nearest -Neighbours (kNN)	93%	121	4
[14]	CNN	RNN GRU LSTM	99.53% 98.86% 96.05%	1265 1182 1217	2
[9]	ROI	CNN(Softmax)	99%	1418	6

Tables 3 and 4 illustrate details about the models developed using the same dataset that we used to develop our models for WBC classification. In [15], researchers presented several standalone CNN models and hybrid machine learning models to classify the white blood cells. The AlexNet, LeNet, and VGG-16 architectures are used for feature extraction and classification. While in the hybrid models, AlexNet is the architecture that was used for feature extraction and several conventional machine learning algorithms were used for classification. Among their models, the hybrid model that consists of AlexNet and the quadratic discriminant analysis (QDA) classifier achieved the best accuracy, 97.78%.

Table 3. Summary of white blood cells classification models that were developed using the same dataset that we used [1]

Ref.	Feature Extraction	Classification	Accuracy
[16]	AlexNet GoogLeNet ResNet-50	QDA	95.79%
[15]	AlexNet	LetNet	70.62%
		AlexNet	84.47%
		VGG16	27.34%
		QDA	97.78%
		SVM	97.33%
		KNN	96.25%
		DT	77.35%
		Softmax	84.03%
[17]	CCA	CNNXception + RNNLSTM	95.89%
[2]	PatternNet-fused Ensemble of CNN	CNN(SoftMax)	99.09%

Table 4. Summary of white blood cells classification models that presented in [18] and were developed using the same dataset that we used in this study [2]

Feature Extraction	classification	Accuracy	Feature Extraction	classification	Accuracy
VGG-16	FCN	69.26%	VGG-19	FCN	63.42%
	Logistic Regr	95.80%		Logistic Regr	96.72%
	Decision Tree	72.23%		Decision Tree	67.73%
	Rand. Forest	92.42%		Rand. Forest	91.91%
	KNN	94.57%		KNN	93.44%
	LDA	27.05%		LDA	26.43%
	Naive Bayes	64.14%		Naive Bayes	63.83%
ResNet-50	FCN	27.36%	Xception	FCN	45.90%
	Logistic Regr	55.02%		Logistic Regr	91.39%
	Decision Tree	44.16%		Decision Tree	53.07%
	Rand. Forest	64.96%		Rand. Forest	75.1%
	KNN	67.73%		KNN	89.45%
	LDA	68.85%		LDA	90.06%
	Naive Bayes	38.83%		Naive Bayes	53.18%
DenseNet-121	FCN	76.84%	DenseNet-169	FCN	82.83%
	Logistic Regr	95.80%		Logistic Regr	96.62%
	Decision Tree	65.68%		Decision Tree	69.77%
	Rand. Forest	86.99%		Rand. Forest	88.01%
	KNN	94.77%		KNN	96.93%
	LDA	96.93%		LDA	96.52%
	Naive Bayes	67.73%		Naive Bayes	71.62%
InceptionV3	FCN	71.93%	Inception-ResNet-V2	FCN	69.57%
	Logistic Regr	92.42%		Logistic Regr	95.39%
	Decision Tree	55.12%		Decision Tree	60.35%
	Rand. Forest	78.38%		Rand. Forest	85.04%
	KNN	89.65%		KNN	87.09%
	LDA	89.96%		LDA	92.93%
	Naive Bayes	62.30%		Naive Bayes	59.02%
1.0 MobileNet224	FCN	73.26%	Mobile NASNet	FCN	69.47%
	Logistic Regr	97.03%		Logistic Regr	91.39%
	Decision Tree	61.68%		Decision Tree	54.82%
	Rand. Forest	86.07%		Rand. Forest	80.23%
	KNN	94.67%		KNN	90.06%
	LDA	92.42%		LDA	85.86%
	Naive Bayes	53.28%		Naive Bayes	46.62%

Furthermore, ten pre-trained deep learning architectures were used to extract features in order to build the classification models. These architectures are VGG-16, VGG- 19, ResNet-50, DenseNet-121, DenseNet-169, Inception-V3, Inception-ResNet-V2, Xception 1., MobileNet-224, and Mo- bile NASNet-A. For classifications, six machine

learning classifiers were used which are Logistic Regression, Decision Tree, Random Forest, Naive Bayes, KNN, LDA. The best model consists of the MobileNet-224 and the logistic regression classifier, its accuracy is 97.03% [18]. In [2], authors used a new method, PatternNet-fused Ensemble of Convolutional Neural Networks (PECNN) for feature extraction and softmax for classifying white blood cells, the accuracy of their method is 99.90%.

As a hybrid method, researchers in [17] used the canonical correlation analysis-based deep learning architecture to extract features from the blood cell images. For classification, they combined the CNN and LSTM, the classification accuracy of their method is 95.89%. One of the promising hybrid models for classifying white blood cells was presented in [16]. This model consists of AlexNet, GoogLeNet, and ResNet-50 as feature extractors, and quadratic discriminant analysis as a classifier, the overall accuracy of this model is 97.95%.

Among those models that were built using the same dataset that we used, the best classification accuracy was achieved by the model that uses AlexNet as a feature extractor and QDA as a classifier, its accuracy is 97.78 % [15]. While the worst accuracy (26.43 %) was obtained by using the VGG19 for features extraction and the LDA for classification [18].

3 Methodology

In this study, we have used a WBC dataset and different machine learning approaches to build several WBC classification models. This section illustrates and discusses our classification models and dataset used in detail.

The first four detection models are hybrid CNN-based models. The other two detection models are fine-tuned CNN models. Figure 2 illustrates a diagram of hybrid CNN-based models for classifying WBC and Figure 3 shows a diagram of the standalone fine-tuned CNN-based models for classifying WBC subtypes. The main steps used to implement our models consist of pre-processing and augmentation of dataset images, extracting features from images, training the model using extracted features, and testing the model's performance using unseen data. Even though the steps are almost the same in both approaches, the algorithms, techniques, and strategies used are different. The rest of this section discusses that in detail.

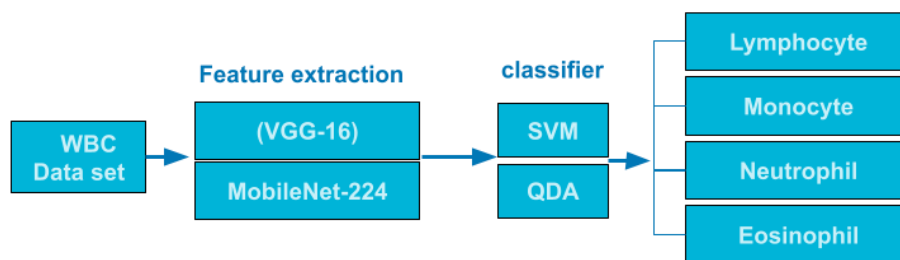


Fig. 2. Schematic diagram of hybrid CNN-based models for classifying WBC subtypes

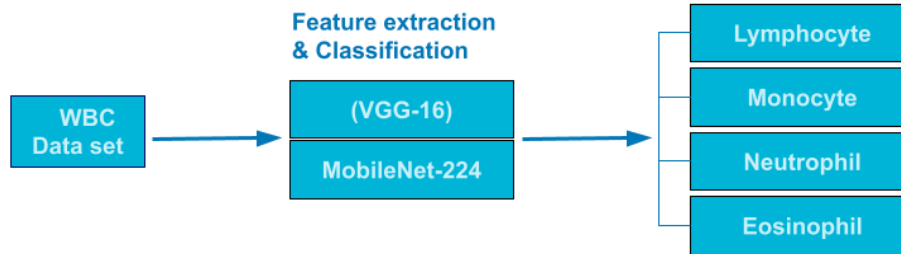


Fig. 3. Schematic diagram of standalone CNN-based models for classifying WBC subtypes

3.1 Dataset description

In this study, we have used a WBC dataset that contains 9,975 augmented images of blood cells [19, 20]. There are approximately 2,500 images for each of 4 different cell types grouped into 4 different folders (according to cell type). The cell types are Eosinophil, Lymphocyte, Monocyte, and Neutrophil. Each image has a 320 x 240 pixel resolution and a depth of 24 bits. Table 5 shows the count of images for each while blood cell subtype.

Table 5. Details of dataset used [19]

WBC subtype	Count of images
EOSINOPHIL	2497
LYMPHOCYTE	2483
MONOCYTE	2478
NEUTROPHIL	2499

3.2 Hybrid CNN models

We have implemented our hybrid CNN models using Mobile-Net and VGG-16 as feature extraction methods, and SVM and QDA as classifiers. Figure 2 shows a simplified diagram of hybrid CNN-based models for classifying WBC subtypes.

Feature extraction. In order to train and develop our hybrid models, we have extracted 500 features using two different CNN techniques. The main reason for choosing CNN techniques for feature extraction is that these models are non-linear and therefore can learn non-linear features. The second reason is that feature vectors are often very large and dimensional.

To extract features to build our hybrid CNN models, we have used the Mobile-net and VGG-16 CNN techniques. The VGG-16 has 16 layers and a 3x3 convolutional kernel. The Mobile-net has 28 layers and it is considered a light-weight deep neural network with almost the same efficiency as VGG-16, but it is 32 times smaller. The VGG-16 is about 553 megabytes in size and it has 138 million parameters. The Mobile-Nets, on the other hand, is about 17 megabytes in size and it has 4.2 million parameters. In comparison to VGG-16, it has almost the same efficiency with 27 times less computation power.

Both techniques were designed to classify 1000 categories (Images classes) in their original versions. Their lower layers are used for general features (problem independent), whereas higher layers contain label-specific features (problem dependent). We have frozen the problem dependent layers before training our model to prevent their weights from being updated during the training. We extended these model by adding Dense, Dropout, and Batch normalization layers on top, where Dense layers are deeply connected neurons in that layer and the neurons in the previous layer, dropout and batch normalization layers are placed between each fully connected layer, and the whole thing is run on the input data images which were resized to 224 x 224.

Classification. Multi-class classification is a task of classifying categories into more than two classes. The categories are classified as belonging to one of several predefined classes. In our hybrid models, we have used the Quadratic Discriminant Analysis (QDA) and the support vector machines (SVM) to develop the models. We used Grid-SearchCV as a hyper-parameter tuning method.

The QDA classifier refers to the use of a non-linear combination of predictor variables to classify categorical responses. In addition, the classifier was chosen based on our experiments. Furthermore, it is a powerful classifier capable of capturing nonlinear features. The SVM method is a controlled machine learning method for classification. The Grid-SearchCV is the method used to find the best candidate parameters by looping through predefined hyper-parameters and fitting the model to the training data. Multiple parameters are tested by cross-validation and the best parameters can be extracted to apply for a predictive model. This method was chosen because setting the optimal hyper-parameter value can significantly improve the model's performance.

We trained our models many times to reach the best model. Training models start with 10 epochs, then we completed the training process until we get better results. We trained the models again with 20 epochs and 212 steps per epoch, and we got the best trained models.

Models optimization. The Adam optimizer was used as an optimization method in our proposed hybrid models. Based on the several experiments, Adam is the ideal optimizer for our classification task. Specifically, to more efficiently update network weights, which is based on adaptive estimation of first-order and second-order moments, with the advantage of less memory consumption.

3.3 Fine-tuned models

To develop our fine-tuned models, we have used the same CNN technique such as Mobile-Net and VGG-16 for feature extraction and classification. Figure 3 illustrates a simplified diagram of standalone CNN-based models for classifying WBC subtypes.

Feature extraction and Classification. For fine-tuned models, the transfer learning approach was used to adapt existing models to the new problem for feature extraction and classification tasks. We froze the weights of all the base layers in the pre-trained models; since their weights are frozen, all output from the base models is sent directly to the other model. We use the sequential model, which functions as layers. This benefits us by inserting the frozen model as a layer in our model and expanding the model by adding fully connected layers. Only the weights from the dense layers will be trained

using the 500 extracted features. We fine-tuned each model after training by unfreezing some layers in the base network and training both the unfrozen layers and the new parts we added jointly. Figures 4 and 5 show the frozen, unfrozen, and new layers when training models before and after fine tuning for both Mobile-Net and VGG-16 architectures, respectively

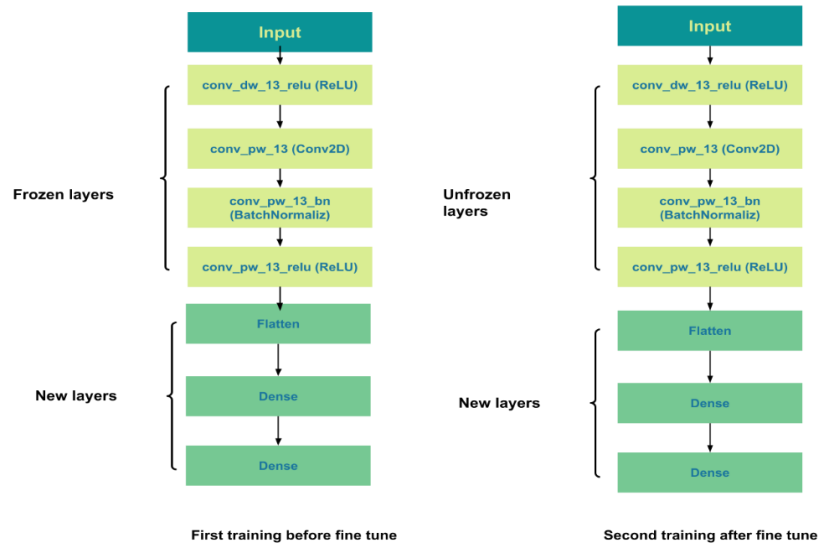


Fig. 4. Frozen, unfrozen, and new layers when training the MobileNet model before and after fine tuning

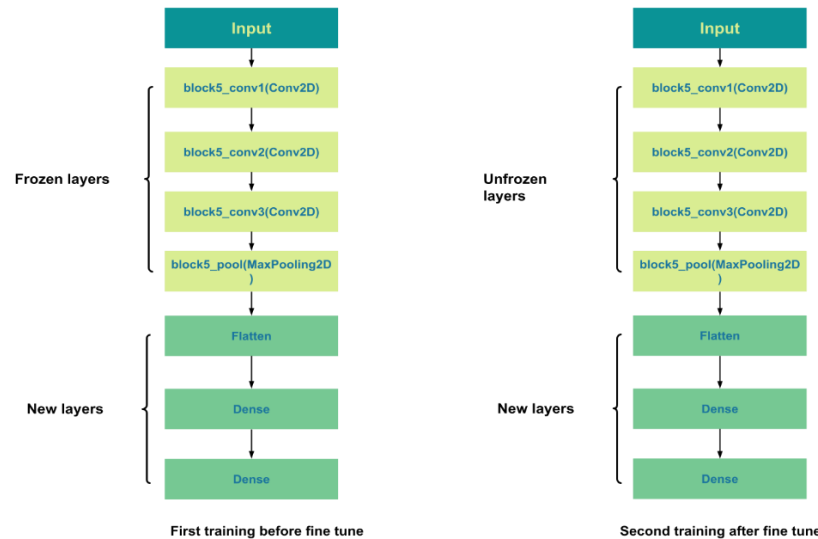


Fig. 5. Frozen, unfrozen, and new layers when training the VGG-16 model before and after fine tuning

In our experiments, we trained models many times until we got satisfied results. First, by freezing the convolutional base and expanding our model by adding additional layers. We trained the models with 20 epochs before fine-tuned, then we unfreeze some layers to fine tune the model by training it again with 30 epochs. We reduced the learning rate to RMSprop ($lr=1e-5$) as well. Extending models allow us to improve the model performance. Furthermore, to enhance classification accuracy, we have used the ImageDataGenerator class from Keras to apply data augmentation. It can generate augmented images dynamically during model training.

Models optimization. For the fine-tuning models, we first used RMSProp with a learning rate of $lr=2e-5$ to train the model with our modified layers. Once we had the result, we fine-tuned the model and trained it again with a learning rate of $lr=1e-5$, which is a very low learning rate, since large updates could damage the representations of the layers we were fine-tuning. Based on several experiments, we noticed that this optimization method gave us the best results rather than using other optimizers.

4 Results

4.1 Experimental environment

To conduct our experiments, we used keras (version 2.4.3), a Python-based Deep Learning library, on Google Colab (a Jupyter-based notebook cloud environment) for free usage of a NVIDIA Tesla P100 GPU.

4.2 Test results

The expected outcome of our models is to predict images of EOSINOPHIL, LYMPHOCYTE, MONOCYTE, and NEU-TROPHIL efficiently. After building our models, we evaluated their performances. Figure 6 and Table 6 show the prediction results of our four hybrid models. Furthermore, Tables 7 and 8 illustrate the prediction results of our standalone models before and after fine tuning, respectively. Among all our models, the model that achieved the best performance is the fine-tuned VGG-16, its accuracy is 99.81%. However, among our hybrid models, the VGG-16+SVM is the best, its classification performance is 98.44%.

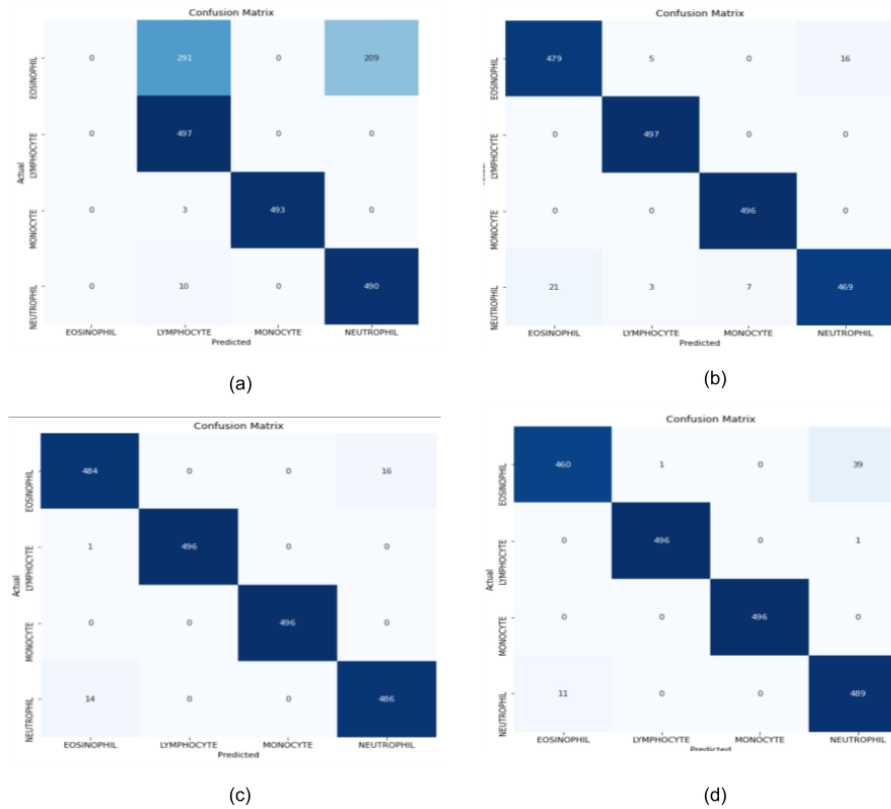


Fig. 6. The confusion matrices of (a) MobileNet+QDA model, (b) MobileNet+SVM, (c) VGG16+SVM, and (d) VGG16+QDA

Table 6. The performance of hybrid models

Feature Extraction	Classification	Learning Rate Reduction	Accuracy
MobileNet	SVM	Adam & Grid search	98.19%
	QDA	Adam	97.39%
VGG-16	SVM	Adam & Grid search	98.44%
	QDA	Adam	74%

As shown in Table 7, the results were not good before the fine-tuning was applied on the models and the best result is 87% for VGG-16, and 85.80% for Mobile-net. By training the models again and applying the fine-tuned we got very good results as shown in Table 8. The best result is 99.81% which is the classification accuracy of the VGG-16 model. While the accuracy of the fine-tuned Mobile-Net is 94.38%.

Table 7. The performance of CNN models before fine-tuning

Feature Extraction & Classification	Learning Rate Reduction	Accuracy
MobileNet	RMSprop(lr=2e-5)	85.80%
VGG-16	RMSprop(lr=2e-5)	87%

Table 8. The performance of CNN models after fine-tuning

Feature Extraction & Classification	Learning Rate Reduction	Accuracy
MobileNet	RMSprop(lr=1e-5)	94.38%
VGG-16	RMSprop(lr=1e-5)	99.81%

5 Discussion

5.1 Comparison of WBC classification models

This subsection compares the detection accuracy of our detection models with WBC detection models in the literature. Our fine-tuned VGG-16 outperforms the best WBC model proposed in related studies. The accuracy of our VGG-16 is 99.81% while the best accuracy of related models is 99.09%. Additionally, VGG-16+SVM and MobileNet+SVM models are considered the third and fourth best detection models among others. Table 9 shows the accuracy of the best five proposed WBC detection models in the related work and the accuracy of our proposed models.

Table 9. Comparison of our models with the best five WBC detection models

Ref.	Feature Extraction	Classification	Accuracy	
[15]	AlexNet	QDA	97.78%	
		SVM	97.33%	
		KNN	96.25%	
[2]	PatternNet-fused Ensemble of CNN	CNN(SoftMax)	99.09%	
[18]	MobileNet224	Logistic Regression	97.03%	
Proposed Models				
This Study	MobileNet	SVM	98.19%	
		QDA	97.39%	
	VGG-16	SVM	98.44%	
		QDA	74%	
	MobileNet			94.38%
	VGG-16			99.81%

5.2 Comparison of VGG-16 and mobilenet based models

This subsection compares the detection performances of our models with models that were developed using the VGG- 16 and MobileNet for feature extraction. The best result in the related studies was obtained by the MobileNet + Logistic regression model,

its classification accuracy is 97.03%. Four out of six proposed models in this study outperform that model. The performances of our VGG-16, VGG16+SVM, MobileNet+SVM, and MobileNet+QDA are 99.81%, 98.44%, 98.19%, and 97.39%, respectively. Table 10 illustrates the detection performance of VGG-16 based models. Table 11 shows the detection accuracy of MobileNet based models.

From the classifier's perspectives, using the suitable classifier to build hybrid models leads to enhanced detection performance. Our results show that the SVM classifier outperforms the QDA when used for classification in all hybrid models, see Tables 10 and 11.

In general, our findings illustrate that five of our proposed models are effective in classifying WBC subtypes. The lowest classification accuracy obtained was 74% which is the accuracy of the VGG16+QDA model.

Table 10. Comparison of VGG-16 based models developed using the same dataset to detect WBC

Ref.	Feature Extraction	Classification	Accuracy
[18]	VGG-16	FCN	69.26%
		Logistic Regression	95.80%
		Decision Tree	72.23%
		Random Forest	92.42%
		KNN	94.57%
		LDA	27.05%
		Naive Bayes	64.14%
[15]	VGG-16		27.34%
Proposed Models			
This Study	VGG-16	SVM	98.19%
		QDA	97.39%
	VGG-16		

Table 11. Comparison of MobileNet based models developed using the same dataset to detect WBC

Ref.	Feature Extraction	Classification	Accuracy
[18]	Mobile-Net	FCN	73.26%
		Logistic Regression	97.03%
		Decision Tree	61.68%
		Random Forest	86.07%
		KNN	94.67%
		LDA	92.42%
		Naive Bayes	53.28%
Proposed Models			
This Study	Mobile-Net	SVM	98.19%
		QDA	97.39%
	Mobile-Net		

6 Conclusion

In this study, we proposed six efficient CNN based models to classify white blood cells subtypes. Some of our models are fine-tuned CNN models, while the rest are hybrid CNN based models. In the fine-tuned models, we used the pre-trained VGG-16 and MobileNet architectures for both feature extraction and classification parts. To develop the hybrid models, we used the VGG16 and MobileNet architectures for feature extraction. For the classification part, we used the SVM and QDA classifiers. The best model among our models is the pretrained VGG-16, its accuracy is 99.81%. The best hybrid model is the VGG16+SVM, its accuracy is 98.44% which is better than the second fine-tuned model. The proposed models could help clinical laboratories to classify the white blood cell subtypes efficiently. To conclude, using transfer learning to solve new problems [20-21] instead of building models from scratch is a sufficient solution. In addition to the high performance, it reduces models' training time.

7 Future work

This study could be extended by using several machine learning classifiers and CNN architectures to develop models. Furthermore, additional pre-process steps (segmentation, object detection, etc.) could be applied in the future to enhance the classification performance. Finally, it is helpful to deploy the best WBC classification models.

8 References

- [1] Blood Disorders. (2021). Hematology.Org. <https://www.hematology.org/education/patients/blood-disorders>
- [2] Wang, J. L., Li, A. Y., Huang, M., Ibrahim, A. K., Zhuang, H., and Ali, A. M. Classification of white blood cells with pattern net-fused ensemble of convolutional neural networks (PECNN). International Symposium on Signal Processing and Information Technology (ISSPIT), December 2018, IEEE, pp. 325-330. <https://doi.org/10.1109/isspit.2018.8642630>
- [3] Tiwari, P., Qian, J., Li, Q., Wang, B., Gupta, D., Khanna, A., ... and de Albuquerque, V. H. C. (2018). Detection of subtype blood cells using deep learning. Cognitive Systems Research, 52: 1036-1044. <https://doi.org/10.1016/j.cogsys.2018.08.022>
- [4] Diouf, D., Seck, D., Diop, M., and Ba, A. (2019). Convolutional Neural Network and decision support in medical imaging: case study of the recognition of blood cell subtypes. arXiv preprint arXiv:1911.08010.
- [5] Loey, M., Naman, M., and Zayed, H. (2020). Deep transfer learning in diagnosing leukemia in blood cells. Computers, 9(2): 29. <https://doi.org/10.3390/computers9020029>
- [6] Cinar, A. C., and Yildirim, M. (2020). Classification of Malaria Cell Images with Deep Learning Architectures. Ingénierie des Systèmes d Inf., 25(1): 35-39. <https://doi.org/10.18280/isi.250105>
- [7] Baydilli, Y. Y., and Atila, Ü. (2020). Classification of white blood cells using capsule networks. Computerized Medical Imaging and Graphics, 80: 101699. <https://doi.org/10.1016/j.compmedimag.2020.101699>

- [8] Ahmed, N., Yigit, A., Isik, Z., and Alpkocak, A. (2019). Identification of leukemia subtypes from microscopic images using convolutional neural network. *Diagnostics*, 9(3): 104. <https://doi.org/10.3390/diagnostics9030104>
- [9] Hegde, R. B., Prasad, K., Hebbar, H., and Singh, B. M. K. (2019). Comparison of traditional image processing and deep learning approaches for classification of white blood cells in peripheral blood smear images. *Biocybernetics and Biomedical Engineering*, 39(2): 382-392. <https://doi.org/10.1016/j.bbe.2019.01.005>
- [10] Chatap, N., and Shibu, S. (2014). Analysis of blood samples for counting leukemia cells using Support vector machine and nearest neighbour. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 16(5): 79-87. <https://doi.org/10.9790/0661-16537987>
- [11] Ryabchykov, O., Ramoji, A., Bocklitz, T., Foerster, M., Hagel, S., Kroegel, C., and Popp, J.). Leukocyte subtypes classification by means of image processing. *Federated Conference on Computer Science and Information Systems (FedCSIS)*, September 2016, IEEE, pp. 309-316. <https://doi.org/10.15439/2016f80>
- [12] Abou El-Seoud, S., Siala, M., & McKee, G. (2020). Detection and Classification of White Blood Cells Through Deep Learning Techniques. *International Journal of Online & Biomedical Engineering*, 16(15). <https://doi.org/10.3991/ijoe.v16i15.15481>
- [13] Baydilli, Y. Y., Atila, U., and Elen, A. (2020). Learn from one data set to classify all—A multi-target domain adaptation approach for white blood cell classification. *Computer Methods and Programs in Biomedicine*, 196: 105645. <https://doi.org/10.1016/j.cmpb.2020.105645>
- [14] Simon, A., Vinayakumar, R., Sowmya, V., Soman, K. P., and Gopalakrishnan, E. A. A. (2019). A deep learning approach for patch-based disease diagnosis from microscopic images. In *Classification Techniques for Medical Image Analysis and Computer Aided Diagnosis*, Academic Press., pp. 109-127. <https://doi.org/10.1016/b978-0-12-818004-4.00005-4>
- [15] Togacar, M., Ergen, B., and Sertkaya, M. E. (2019). Subclass separation of white blood cell images using convolutional neural network models. *Elektronika ir Elektrotechnika*, 25(5): 63-68. <https://doi.org/10.5755/j01.eie.25.5.24358>
- [16] Adoghe, A., Noma-Osaghae, E., & Yabkwa, R. (2020). Photonic Crystal and its Application as a Biosensor for the Early Detection of Cancerous Cells. *International Journal of Online & Biomedical Engineering*, 16(3). <https://doi.org/10.3991/ijoe.v16i03.12523>
- [17] Patil, A. M., Patil, M. D., and Birajdar, G. K. (2020). White blood cells image classification using deep learning with canonical correlation analysis. *IRBM*. <https://doi.org/10.1016/j.irbm.2020.08.005>
- [18] H Mohamed, E., H El-Behaidy, W., Khoriba, G., and Li, J. (2020). Improved White Blood Cells Classification based on Pre-trained Deep Learning Models. *Journal of Communications Software and Systems*, 16(1): 37-45. <https://doi.org/10.24138/jcomss.v16i1.818>
- [19] Blood Cell Images. (2018, April 21). Kaggle. <https://www.kaggle.com/paultimothymooney/blood-cells>
- [20] Ahmed, S., Shaikh, A., Alshahrani, H., Alghamdi, A., Alrizq, M., Baber, J., & Bakhtyar, M. (2021). Transfer Learning Approach for Classification of Histopathology Whole Slide Images. *Sensors*, 21(16), 5361. <https://doi.org/10.3390/s21165361>
- [21] Jabir, B., Falih, N., & Rahmani, K. (2021). Accuracy and Efficiency Comparison of Object Detection Open-Source Models. *International Journal of Online & Biomedical Engineering*, 17(5). <https://doi.org/10.3991/ijoe.v17i05.21833>

9 Authors

Najla Alofi received the BSc degree in Information Technology from Department of Information Technology, College of Computer, Qassim University, Saudi Arabia in 2021.

Wafa Alonezi received the BSc degree in Information Technology from Department of Information Technology, College of Computer, Qassim University, Saudi Arabia in 2021.

Wedad Alawad received the BSc degree in Computer Science from Department of Information Technology, College of Computer, Qassim University, Saudi Arabia in 2008, and the MSc degree in Computer Science from Bowling Green State University, United States in 2014, and Ph.D. degree in Computer Science from Oakland University, United States in 2018. She is currently an assistant Professor of Computer Science at Qassim University, Saudi Arabia. Her research activity is related to artificial intelligence, cyber security, and cloud computing.

Article submitted 2021-09-06. Resubmitted 2021-10-13. Final acceptance 2021-10-13. Final version published as submitted by the authors.