Computer Methods and Programs in Biomedicine 226 (2022) 107162

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

AIMIC: Deep Learning for Microscopic Image Classification

Rui Liu[a], Wei Dai[a], Tianyi Wu[a], Min Wang[a], Song Wan[b], Jun Liu[a,c,*]

[a] Department of Mechanical Engineering, City University of Hong Kong, Hong Kong, China
[b] Department of Surgery, Faculty of Medicine, The Chinese University of Hong Kong, Hong Kong, China
[c] Shenzhen Research Institute, City University of Hong Kong, Shenzhen 518057, China

ARTICLE INFO

Article history:
Received 26 June 2022
Revised 9 September 2022
Accepted 25 September 2022

Keywords:
Code-free deep learning
Artificial intelligence
AI platform
Microscopic image analysis

ABSTRACT

Background and Objective: Deep learning techniques are powerful tools for image analysis. However, the lack of programming experience makes it difficult for novice users to apply this technology. This project aims to lower the barrier for clinical users to implement deep learning methods in microscopic image classification.

Methods: In this study, an out-of-the-box software, AIMIC (artificial intelligence-based microscopy image classifier), was developed for users to apply deep learning technology in a code-free manner. The platform was equipped with state-of-the-art deep learning techniques and data preprocessing approaches. Furthermore, we evaluated the built-in networks on four benchmark microscopy image datasets to assist entry-level practitioners in selecting a suitable algorithm.

Results: The entire deep learning pipeline, from training a new network to inferring unseen samples using the trained model, could be implemented on the proposed platform without the need for programming. In the evaluation experiments, the ResNeXt-50-32×4d outperformed other competitor algorithms in terms of average accuracy (96.83%) and average F1-score (96.82%). In addition, the MobileNet-V2 achieved a good balance between the performance (accuracy of 95.72%) and computational cost (inference time of 0.109s for identifying one sample).

Conclusions: The proposed AI platform allows people without programming experience to use artificial intelligence methods in microscopy image analysis. Besides, the ResNeXt-50-32×4d is a preferable solution for microscopic image classification, and MobileNet-V2 is most likely to be an alternative selection for the scenario when computing resources are limited.

1. Introduction

Microscopic analysis of cells and medical samples is an essential procedure for health assessment and disease diagnosis in clinics. For example, motile sperms, oocytes, and embryos are routinely examined under the microscope for infertility assessment and treatment [1,2]. Blood cells are the main components of blood. There are three basic types of blood cells: red blood cell (RBC), white blood cell (WBC), and platelet [3]. WBC can be further classified into neutrophils, eosinophils, basophils, lymphocytes, and monocytes. Blood cell classification is to recognize different types of blood cells and the abnormal samples, which is critical for diagnosing blood cell-related diseases. For instance, In the diagnosis of Rhesus D (RhD) hemolytic disease, fetal RhD-positive cells need to be identified from maternal blood cells [4]. Traditionally, the examination of blood smears and recognition of blood cells are conducted under the microscope by experienced hematologists. Cell identification and classification can be challenging because manual microscopic analysis has low repeatability and is labor-intensive and error-prone.

Machine learning techniques have demonstrated potential superiority in a wide variety of areas [5–9]. This study focuses on the application of machine learning techniques in microscopic image recognition. To identify the recent progress in this topic, we queried the Web of Science, IEEE Xplore, ACM Digital Library, PubMed, ScienceDirect, and Wiley Online Library for publications with the terms 'machine learning and microscopic image classification', 'machine learning and blood cell classification', 'deep learning and microscopic image classification', and 'deep learning and blood cell classification' in their title, abstract and keywords. Besides, Google Scholar was used for forward and backward searches. Finally, 126 relevant projects were screened from the 2038 retrieved

[*] Corresponding author.
E-mail address: jun.liu@cityu.edu.hk (J. Liu).

https://doi.org/10.1016/j.cmpb.2022.107162
0169-2607/© 2022 Elsevier B.V. All rights reserved.

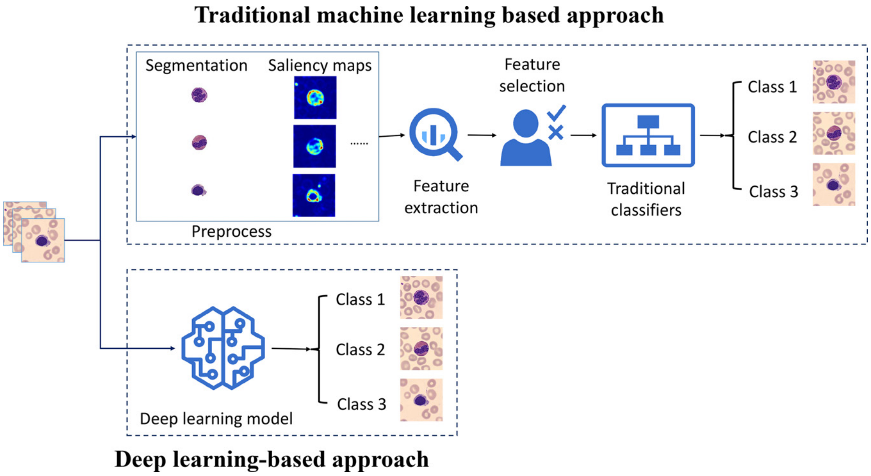


Fig. 1. Comparison of the general process of traditional machine learning approach and deep learning based method in blood cell classification.

items after removing the duplicated and irrelevant works. Next, we will introduce the representative contributions.

Computer-aided technology has shown its potential to help people diagnose various diseases [9–11]. Automated cell identification systems have been developed to overcome the challenges in manual cell classification. Automated detection systems can relieve experts from the heavy laboring of distinguishing different types of blood cells. Prior to the wide adoption of deep learning, the majority of studies employed conventional machine learning-based approaches, such as the k-nearest neighbors algorithm (k-NN) and support vector machine (SVM), to identify blood cells.

In conventional learning-based methods, segmentation and feature extraction are the prerequisite processes before the classification step. Despite the intricate tuning for feature extraction, these traditional methods were reported with unsatisfactory classification accuracy due to the lack of adaptive pre-processing methods and reliable segmentation algorithms [12,13]. In addition, the multi-stage workflow further introduces cumulative errors, lowering the classification accuracy. The limitations of existing methods motivate researchers to develop advanced algorithms based on deep learning to solve the problem in blood cell classification [14–16].

The past decade has witnessed a rapid evolution of machine learning, and deep learning networks have been proven critically helpful to serve as the right-hand man in dealing with real-world problems [17–19]. A deep learning network, inspired by the biological neural network, can be trained to accomplish complicated tasks, such as the detection of congenital heart disease [17], identification of diabetic retinopathy [20], diagnosis of esophageal adenocarcinoma [21], and screening of Covid-19 [22]. Different from conventional classification systems that involve crafting the features of the input data, deep learning-based classifiers can classify the raw images directly, making it feasible to build an end-to-end analyzer, as shown in Fig. 1.

Artificial intelligence (AI) technology has significant value to the medical community under applicable guidelines [23,24]. Although the deep learning-based methods have demonstrated success in the blood cell classification by computer scientists, the translational implementation for clinical use by medical practitioners remains a challenge, potentially due to the lack of sufficient coding experience. Moreover, selecting an optimal method from many deep learning algorithms is a nontrivial task. To promote clinical translations, we developed a platform, AIMIC, for medical practitioners to apply deep learning algorithms in microscopic image classification in a code-free manner. The software will simplify the implementation of AI-based methods for microscopic image analysis by offering a user-friendly solution. In summary, the major contributions of this study include:

- development of an out-of-the-box platform to lower the barrier for novice users to apply deep learning-based approaches for microscopic image classification. The users can carry out an end-to-end deep learning pipeline with the platform without any coding. Besides, the software provides an automatic k-fold cross-validation mode to facilitate the algorithm's evaluation.
- evaluation of ten state-of-the-art deep learning algorithms embedded in the proposed platform by measuring the standard performance metrics, i.e., accuracy, precision, recall, and F1 score. The evaluation outcomes provide a reference for selecting an appropriate model.
- assessment of the computational efficiency of these built-in networks. The assessment results provide a reference basis for the users to choose a solution that can balance the classification accuracy and computational cost when the computing resources are limited.

The rest of the article is organized as follows. Section 2 reviews the related works. The proposed deep learning-deployment platform and the experiments designed for deep learning model evaluation are introduced in Section 3, followed by the experimental results in Section 4. Section 5 discusses the evaluated models' performance and the advantages and limitations of the proposed software. Finally, Section 6 concludes this work.

2. Related Works

In the past few years, deep convolutional neural networks (CNNs) have been reported to achieve great success in blood cell classification. But before that, conventional machine learning techniques were often used to identify cell samples.

2.1. Conventional Machine Learning Approaches

To automate the blood cell counting task, Ongun et al. applied several machine learning algorithms to classify the blood cell images based on the segmentation output and feature extraction (shape, color, and texture) [25]. The experiment results demonstrated that SVM outperformed the other three methods (k-NN, linear vector quantization, and multi-layer perceptron) in terms of accuracy (91.03% vs. 80.76% vs. 83.33% vs. 89.74%). Kim et al. reported a framework to automatically recognize WBCs and RBCs based on the morphological information [26]. They binarized and segmented the input images and then extracted features from the preprocessed samples. Finally, the samples were classified by the neural network, linear vector quantization, and k-NN algorithm based on the extracted morphological features. To reduce the burden of differential blood counts on pathologists, Tai et al. presented a hierarchical SVM method to classify blood cells into seven types using the hierarchical features [27]. The proposed approach improved the recall rate by 56% to 95.27%, far exceeding its one-stage counterpart (66.84%). Noor et al. employed the speed up robust feature (SURF) method to increase the accuracy of the artificial neural network on the WBC classification task to 94.09% [28]. Abdeldaim et al. introduced a dedicated system to diagnose blood cells to be normal or affected [29]. The system started from segmenting the images, proceeded with extracting the features, and ended with normalizing the extracted features using three different techniques (z-score, min-max, grey-scaling). The system outputs were fed into four different classifiers, namely K-NN, Naive Bayes, SVM, and Decision Trees, to verify the efficiency of the proposed system. Finally, K-NN achieved the highest classification accuracy (96.42%).

2.2. Deep Learning Approaches

In 2017, Li et al. employed a multi-layer CNN to discriminate hyperspectral images into three classes: red cells, white cells, and background. The average accuracy of the CNN model exceeded that of the SVM algorithm by around 30% in two self-collected datasets consisting of 16,218 and 8,004 images, respectively [30]. Dastidar's group developed a customized platform, ShonitTM, to classify the subtypes of WBC, RBC, and platelet. The sensitivity of the shonitTM platform was reported over 91% for all cell types [31]. In a later study, Huang and his collaborators demonstrated that the attention-aware residual network-based manifold learning model outperformed the previous models with higher average accuracy (95.3%) when classifying 10,800 WBCs into six categories [32]. In the ISBI 2019 C-NMC Challenge, Prellberg et al. fine-tuned the ResNeXt50 to differentiate cancerous cells from normal ones with an 89.88% success rate [33]. Pan et al. achieved higher accuracy (91.73%) using a neighborhood-correction algorithm with modified ResNets in the same competition [34]. Acevedo et al. used a fine-tuning VGG-16 to classify eight types of blood cells in the PBC dataset with an overall accuracy of 96.2% [35]. Subsequently, Bagido et al. improved the classification accuracy of four categories of samples in the same dataset to 98.4% with the pre-trained Inception ResNetV2 [36]. Jung et al. boosted the recognition accuracy of W-Net from 91% to 96% on the LISC image set by pretraining the model on the collected samples [37]. Baydilli et al. solved the insufficient data problem in the blood cell identification task with a bespoke architecture called WBCaps [38]. In the WBCaps, a new convolution layer was added to the front of the original capsule network [39] to enhance the feature extraction. The experiment results showed that their method obtained a higher success rate (96.86%) in the LISC dataset without any data augmentation. Yao et al. attempted to classify the WBCs by using a two-module deformable CNNs approach [40]. In this method, the deformable convolutional layers and the transfer learning were used to improve the reliability and generalization of the framework. Their experiment results verified that the proposed scheme was superior to the other CNNs and traditional algorithms regarding precision, recall, F1 score, and area under the curve for three datasets with different characteristics. In a recent study, Das et al. utilized an efficient model that featured depthwise separable convolutions and inverted residual architecture to yield classification accuracies of 99.39% and 97.18% in two benchmark datasets, ALL-IDB1 and ALL-IDB2, respectively [41].

3. Methods

Our study bridged the gap for beginners to apply deep learning techniques to recognize microscope images in two steps. Firstly, we integrated the entire project pipeline, from training the network to inferring new samples using the trained model, into an easy-to-use graphical user interface (GUI). Subsequently, the state-of-the-art networks embedded in this software were evaluated to provide a reference for entry-level implementers to select the appropriate solution.

3.1. The Bespoke Microscopic Image Classification Software

We developed an out-of-the-box platform embedded with state-of-the-art deep learning networks to assist biologists and clinicians with limited coding experience in using deep learning techniques for microscopic image classification. This AI-based microscopic image classifier, AIMIC, was developed with PyQt5, PyTorch, and other supporting packages (Table S1). The platform has two different working modes: model training and sample prediction. The workflow of the AIMIC platform is summarized in Fig. 2, and a demonstration of the user guide is recorded in Video S1. The built-in models in the AIMIC can be trained from scratch or fine-tuned on top of the pre-trained weight parameters generated from transfer learning. In practice, a selected network is trained on the target domain data, followed by the recognition of the test samples with the trained model. In the first step, the dataset is uploaded to the AIMIC via a pop-up window to select the desired directory. During the uploading process, the samples will be automatically labeled and assigned into training and validation sets according to user settings. Then the inputs are adjusted (such as resizing images to the desired resolution, transforming them to tensors, etc.) for feature extraction. In addition to statistical information of the datasets, the platform provides visualization functions to examine the samples in each category. After the input images are prepared, the user can select a suitable model with desired hyper-parameters and settings to proceed with the training process. When transfer learning is applied, the pre-trained parameters transferred from a previous task can be loaded to the model during the initialization. If we want to assess a model in a k-fold cross-validation manner, the platform can automatically start the valuation process based on the given value of *k*. The indicators of each epoch, such as loss, accuracy, recall, and precision, are displayed real-timely in the information dialog when fitting the model. These indicators are also recorded in the training log for archiving and/or further processing. More importantly, the parameters of the epoch with the best performance will be saved for classifying new samples. In the inference procedure, the trained model is used to identify new unseen samples for a specific task.

3.2. Design of the Evaluation Experiment

In the evaluation experiment, ten commonly-used CNNs, including EfficientNet_B0 [42], MobileNet-v3 [43], MobileNet-v2 [44], ShuffleNet-v2 [45], ResNeXt-50 [46], ResNet-50 [47], ResNet-34 [47], Inception-v3 [48], GoogLeNet [49], and VGG-16 [50] were trained to classify samples from the four standard blood cell classification datasets (C-NMC [51], ALL-IDB2 [52], PBC [53] and LISC [54]), as shown in Fig. 3. The selected CNNs share a similar structure with the same basic units, including convolutional layers, pooling, fully connected layers, and softmax layers. According to Google Scholar Metrics, the models with the top three citation numbers (as of November 2021) among the general models and the efficiency models and/or their corresponding advanced versions were selected. Each model was trained and tested separately

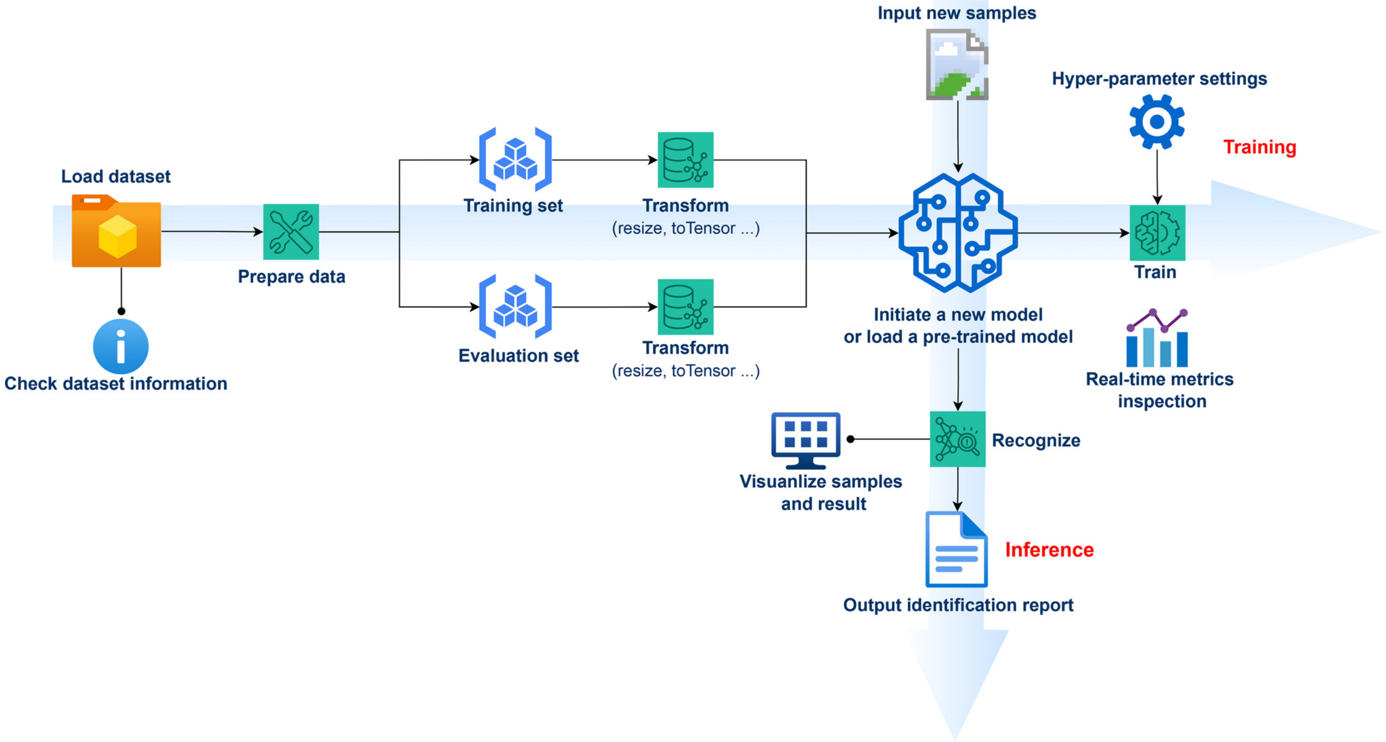


Fig. 2. The workflow of the proposed microscopic image classification platform (AIMIC).

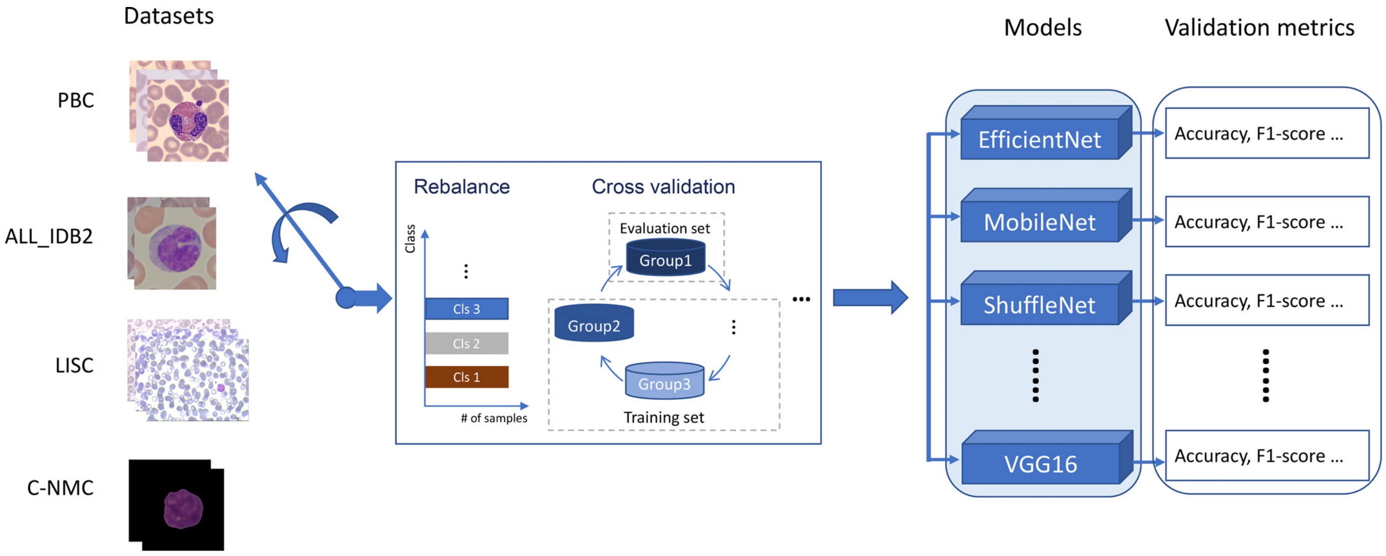


Fig. 3. The design of the evaluation experiment.

under the same settings on the four datasets. For every model-dataset pair, five-fold cross-validation was used to ensure an objective evaluation by avoiding biased results. In the five-fold cross-validation, the original dataset was partitioned into five equal-size subsets, of which one was reserved for testing the model, and the rest were used as training data. The process was then repeated five times until each subset had served as the validation set for once. The average of the five results was used as the evaluation reference for a single model-dataset pair. The evaluated CNN models are introduced below by emphasizing their characteristics.

The VGG network was proposed by the Visual Geometry Group at Oxford University and won the first place in the localization task and the runner-up in the classification task in the ImageNet competition in 2014. Unlike the pioneer AlexNet [55], VGG uses a smaller convolution kernel (3x3) and a deeper architecture. There are four different configurations of VGG networks, among which the VGG-16 is the most widely used.

The primary contribution of GoogLeNet is the introduction of the inception module, which can boost the performance of the model by fusing the feature information at different scales. Moreover, the GoogLeNet lowers the computational complexity by using the 1×1 convolution operation for dimensionality reduction.

Inception V3 is the advanced version of GoogLeNet with three significant improvements: the convolutions are modified with small filters to save computational costs; the filter banks are expanded to solve the representational bottleneck; and additional techniques with RMSProp optimizer and batch-normalization are used to improve the auxiliary classifier and label smoothing.

The deep residual network (ResNet), proposed by the Microsoft research team, is regarded as a breakthrough in deep learning. The residual learning principle (Fig. S1) is accomplished by shortcut connections in the feedforward process to address the degradation problem in model training. Since the residual principle enables building a network with increased depth, the performance is dramatically improved by adding more layers. ResNet-34 and ResNet-50 are two substructures in the ResNet family.

ResNeXt is modified from the ResNet with a strategy of splitting and merging the building block (Fig. S2). While preserving similar computational complexity, the ResNeXt divides the residual block into multiple substructures with the same topology and then aggregates them. ResNeXt-50 is the most commonly used configuration of ResNeXts.

ShuffleNet [56] is built by applying the shuffled channels and pointwise group convolutions to reduce the computational cost for implementation in portable devices. ShuffleNet-V2 is an improved version with four significant modifications, including equalizing the channel width, selecting a reasonable group number, avoiding network fragmentation, and reducing element-wise operations.

MobileNet is another lightweight CNN that uses the depthwise separable convolution (Fig. S3a) [57,58] for portable and mobile applications. The upgraded version of MobileNet-v2 applies two additional modifications, including the linearization of the bottlenecks and the use of inverted residuals (Fig. S3b) to improve performance. MobileNetV3 is an advanced architecture built in the light of network search. Furthermore, the resource-consuming layers are modified to decrease the latency by applying a neo-activation function and the squeeze-and-excite module [59] (Fig. S4).

EfficientNets is another computation-efficient CNN family with a balance of the input resolution and the architecture depth and width. EfficientNet-B0 was selected in this study because it requires the least computing resources in the EfficientNets family.

3.3. Deep Transfer Learning

Deep transfer learning (DTL) is the application of a deep neural network pre-trained from the source domain to solve problems in different target domains [60]. DTL is a helpful technique since deep learning models are usually data-hungry in many fields and training-expensive from scratch even if the data points are enough in a certain task[61–63]. Generally, transfer learning can be implemented by retaining all or part of the learnable parameters. In some cases, to save computing resources and improve efficiency to an extreme, users can freeze feature extraction parameters and only optimize the classifier's parameters. Here, to better reflect the capabilities of the individual model, we fine-tune the entire network on top of the pre-training parameters.

4. Results

In this section, the potential practicability of the AIMIC was illustrated with an application example. Furthermore, the results of the model evaluation experiments were examined in terms of classification performance and computational efficiency.

4.1. Details of the Datasets

In this study, four publicly available datasets of blood cells were employed to evaluate the deep learning models. The C-NMC and ALL-IDB2 are binary-class datasets with large (n = 15114) and small sample sizes (n = 260), respectively, while the PBC and LISC are multi-class datasets with large (n =17092) and small image sizes (n = 400), respectively.

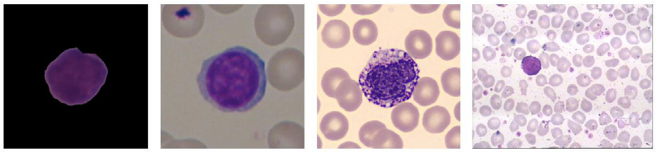


Fig. 4. Examples of the datasets. From left to right: C-NMC, ALL-IDB2, PBC, LISC.

The C-NMC dataset is a public hematological image set initially provided for the ISBI 2019 competition. In the competition, participants were required to identify the cancerous blood cells from the normal ones. The identification is challenging because these two cell groups are morphologically similar in the samples. In this dataset, 12,528 of the total 15,114 images are used to train the models. In the training data, 3389 healthy cells and 7,272 immature leukemic blasts were well labeled for the model to learn, while the remainder were for validation. In this study, the training dataset was reorganized by randomly sampling leukemic blasts with the same amount as the normal group to avoid the bias caused by imbalanced data [64].

ALL-IDB2 is a dataset collected by hematologists at M. Tettamanti Research Center in Italy. This collection contains 130 benign cell samples and the same number of the diseased counterparts (i.e., lymphoblasts).

PBC dataset, prepared by Hospital Clinic de Barcelona, is commonly regarded as a benchmark for evaluating machine learning models in differentiating different types of healthy peripheral blood cells. This dataset contains eight classes and 17092 high-definition images. Similar to the C-NMC dataset, we also resampled the categories with larger numbers of samples to have the same number as the group with the least amount to rebalance the data. Finally, the PBC dataset is reproduced in this study to have 9,712 images with 1,214 for each category.

LISC is an open-access database prepared by the Hematology-Oncology and BMT Research Center of Imam Khomeini Hospital. The dataset has 250 images and is divided into six classes unevenly. Since the eosinophil and mixture groups only have 39 and 8 images, respectively, which are not enough for five-fold cross-validation in deep learning models, these two categories were discarded. The remaining subset was processed in the same way as processing the PBC dataset, namely, removing redundant samples from classes with a larger number of pieces. As a result, 192 images in four categories were eventually selected for the experiment.

All images in the four datasets are of 24-bit color depth, and the representative samples are shown in Fig. 4. More detailed information about the databases is provided in Table 1.

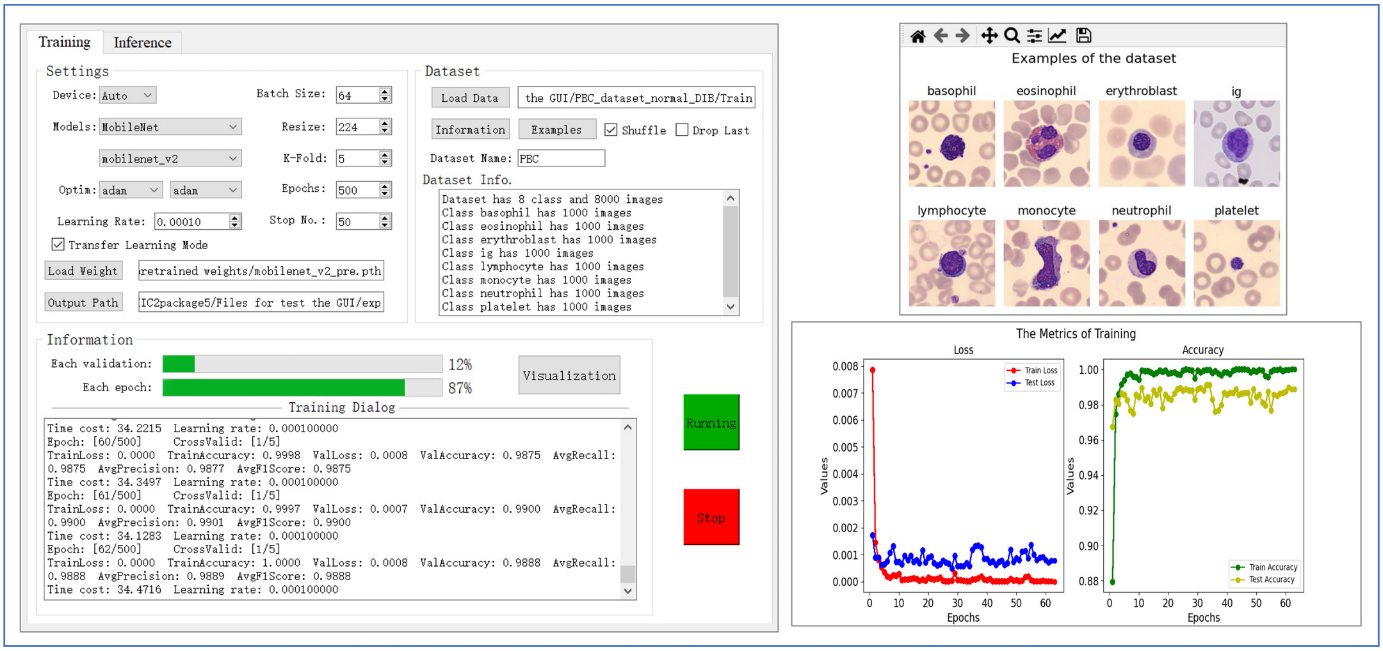
4.2. The Usability of the AIMIC

In this section, an example application is presented to demonstrate the practicality of the proposed deep learning software. The example was implemented on a laptop with a CPU of Intel(R) Core(TM) i5-9300H @2.40GHz and one NVIDIA GeForce GTX 1650 GPU.

In this application, a network (e.g., MobileNet-v2) first learned the samples' characteristics in the PBC dataset and then identified new blood cell samples. For simplicity, we picked 1000 samples from each category, a total of 8000 for the training phase and the remaining as the test data. The user interface of the training model is shown on the left of Fig. 5. The information and examples of the dataset can be visualized by clicking the corresponding function button (the upper right of Fig. 5). In the training phase, the maximum training epoch was set to be 500. At the same time, the process would stop early if the test accuracy was

Table 1
Dataset details

Datasets	Categories	Image Size (pixel)	Format
C-NMC	ormal cell, malignant cell	450×450	BMP
ALL-IDB2	healthy cell, blast cell	257×257	JPG
PBC	neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes, erythroblasts, platelets (thrombocytes)	360×363	JPG
LISC	basophil, eosinophil, lymphocyte, monocyte, neutrophil	720×576	BMP



Device: Auto
Batch Size: 64
Models: MobileNet
Resize: 224
mobilenet_v2
K-Fold: 5
Optim: adam adam
Epochs: 500
Learning Rate: 0.00010
Stop No.: 50
☑ Transfer Learning Mode
Load Weight: retrained weights/mobilenet_v2_pre.pth
Output Path: IC2package5/Files for test the GUI/exp

Load Data: the GUI/PBC_dataset_normal_DIB/Train
☑ Shuffle ☐ Drop Last
Dataset Name: PBC

```
Dataset has 8 class and 8000 images
Class basophil has 1000 images
Class eosinophil has 1000 images
Class erythroblast has 1000 images
Class ig has 1000 images
Class lymphocyte has 1000 images
Class monocyte has 1000 images
Class neutrophil has 1000 images
Class platelet has 1000 images
```

Each validation: 12%
Each epoch: 87%

```
Time cost: 34.2215  Learning rate: 0.000100000
Epoch: [60/500]     CrossValid: [1/5]
TrainLoss: 0.0000  TrainAccuracy: 0.9998  ValLoss: 0.0008  ValAccuracy: 0.9875  AvgRecall:
0.9875  AvgPrecision: 0.9877  AvgF1Score: 0.9875
Time cost: 34.3497  Learning rate: 0.000100000
Epoch: [61/500]     CrossValid: [1/5]
TrainLoss: 0.0000  TrainAccuracy: 0.9997  ValLoss: 0.0007  ValAccuracy: 0.9900  AvgRecall:
0.9900  AvgPrecision: 0.9901  AvgF1Score: 0.9900
Time cost: 34.1283  Learning rate: 0.000100000
Epoch: [62/500]     CrossValid: [1/5]
TrainLoss: 0.0000  TrainAccuracy: 1.0000  ValLoss: 0.0008  ValAccuracy: 0.9888  AvgRecall:
0.9888  AvgPrecision: 0.9889  AvgF1Score: 0.9888
Time cost: 34.4716  Learning rate: 0.000100000
```

Fig. 5. Demonstration of the training process of the proposed AIMIC.

not improving for 50 consecutive iterations. Adam was chosen as the optimizer for the training, and the learning rate was set to be 0.0001. Note that our program can automatically detect whether the device is configured with CUDA, and it will prioritize implementing the project on GPU(s) if the device has GPU(s). Moreover, transfer learning was used to accelerate training and improve accuracy here. During the training, two progress bars (one for a single epoch and the other for the whole fitting stage) indicated the completion percentage of the training progress. In addition to the real-time indicators printed in the dialog, their evolution with epoch would be plotted for visualization (the lower right of Fig. 5) through the corresponding command. In this case, it was seen that on top of the pre-trained parameters, the training process converged quickly, around 30 epochs, with a high validation accuracy (over 98%). The trained model would be saved automatically during training when the learnable parameters were optimized to the desired state, where the validation indicators reached their optimal value.

In this application example, it took approximately 50 minutes to train the network on the GPU mentioned above. It is worth noting that, with the same settings, training a model on a CPU takes several times as long as on a GPU. In addition, the number of floating-point operations (FLOPs) and the structure of the model also significantly influence the training speed.

After being trained over the training set, the model was used to recognize new samples in the testing set. Here, eight images from the testing set were used to verify the trained deep learning model (i.e., MobileNet-v2 on the left of Fig. 6). In the inference phase, the prediction results of each sample with the corresponding probability were updated in the information dialog in real time. The output file of the inference process includes the predicted category and the corresponding probability, as shown in the upper right of Fig. 6). In this example, the subtypes of eight blood cell samples were correctly inferred by the trained MobileNet-v2 with high probabilities ranging from 0.785 to 0.998, which could be visualized on the software for a user-friendly analysis, as shown in the lower right of Fig. 6.

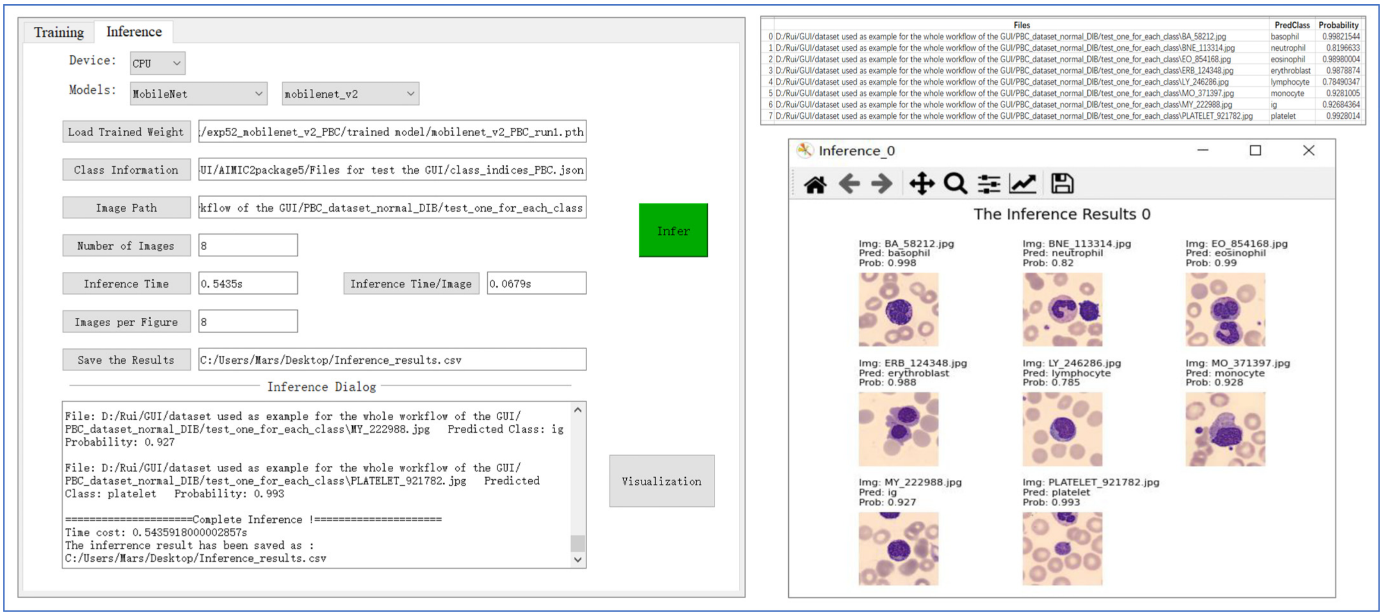
Table 2
Confusion matrix.

Data Class		Ground Truth	
		Positive	Negative
Prediction	Positive	TP	FP
	Negative	FN	TN

4.3. Classification Performance

In order to shorten the experimental period, the evaluation experiments were implemented on a DELL workstation with a CPU of Intel(R) Xeon(R) Gold 6226R @2.90 GHz and two NVIDIA Quadro RTX 4000 GPUs with 8GB memory. The proposed code-free platform AIMIC enables us to conduct the experiments expediently.

The performance of a classification model can be indicated by counting the number of correctly predicted positive samples (true positive or TP), the number of correctly recognized negative instances (true negative or TN), the number of negative samples incorrectly assigned as positive (false positive or FP), and the number of positive instances misclassified to be negative (false negative or FN). These four counts constitute a confusion matrix, as summarized in Table 2, from which various metrics can be derived [65]. We used accuracy, recall, precision, and F1 score for quantitative



Device: CPU

Models: MobileNet mobilenet_v2

Load Trained Weight :/exp52_mobilenet_v2_PBC/trained model/mobilenet_v2_PBC_run1.pth

Class Information UI/AIMIC2package5/Files for test the GUI/class_indices_PBC.json

Image Path kflow of the GUI/PBC_dataset_normal_DIB/test_one_for_each_class

Number of Images 8

Inference Time 0.5435s

Inference Time/Image 0.0679s

Images per Figure 8

Save the Results C:/Users/Mars/Desktop/Inference_results.csv

```
File: D:/Rui/GUI/dataset used as example for the whole workflow of the GUI/
PBC_dataset_normal_DIB/test_one_for_each_class\MY_222988.jpg   Predicted Class: ig
Probability: 0.927

File: D:/Rui/GUI/dataset used as example for the whole workflow of the GUI/
PBC_dataset_normal_DIB/test_one_for_each_class\PLATELET_921782.jpg   Predicted
Class: platelet   Probability: 0.993

====================Complete Inference !====================
Time cost: 0.5435918000002857s
The inferrence result has been saved as :
C:/Users/Mars/Desktop/Inference_results.csv
```

	Files	PredClass	Probability
0	D:/Rui/GUI/dataset used as example for the whole workflow of the GUI/PBC_dataset_normal_DIB/test_one_for_each_class\BA_58212.jpg	basophil	0.99821544
1	D:/Rui/GUI/dataset used as example for the whole workflow of the GUI/PBC_dataset_normal_DIB/test_one_for_each_class\BNE_113314.jpg	neutrophil	0.8196633
2	D:/Rui/GUI/dataset used as example for the whole workflow of the GUI/PBC_dataset_normal_DIB/test_one_for_each_class\EO_854168.jpg	eosinophil	0.98980004
3	D:/Rui/GUI/dataset used as example for the whole workflow of the GUI/PBC_dataset_normal_DIB/test_one_for_each_class\ERB_124348.jpg	erythroblast	0.9878874
4	D:/Rui/GUI/dataset used as example for the whole workflow of the GUI/PBC_dataset_normal_DIB/test_one_for_each_class\LY_246286.jpg	lymphocyte	0.78490347
5	D:/Rui/GUI/dataset used as example for the whole workflow of the GUI/PBC_dataset_normal_DIB/test_one_for_each_class\MO_371397.jpg	monocyte	0.9281005
6	D:/Rui/GUI/dataset used as example for the whole workflow of the GUI/PBC_dataset_normal_DIB/test_one_for_each_class\MY_222988.jpg	ig	0.92684364
7	D:/Rui/GUI/dataset used as example for the whole workflow of the GUI/PBC_dataset_normal_DIB/test_one_for_each_class\PLATELET_921782.jpg	platelet	0.9928014

Fig. 6. Demonstration of the inference process of the proposed AIMIC.

assessment as in previous research [66,67]. Accuracy measures the percentage of correct predictions overall, and it is expressed as follows:

$$Accuracy = \frac{1}{l}\sum_{i=1}^{l}\frac{TP_i + TN_i}{TP_i + FP_i + FN_i + TN_i} \tag{1}$$

where, TP_i,FP_i, FN_i and TN_i are the number of the true positive, false positive, false negative, and true negative of class C_i, respectively, i is the index of corresponding class number and l is the number of the classes. The precision is the fraction of true positive (TP_i) among all the retrieved instances (TP_i+FP_i), while the recall value is the fraction of the true positive (TP_i) among all the relevant cases (TP_i+FN_i). F1 score provides a harmonic mean of the precision and recall, and it is defined as:

$$F1score = \frac{1}{l}\sum_{i=1}^{l} 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{2}$$

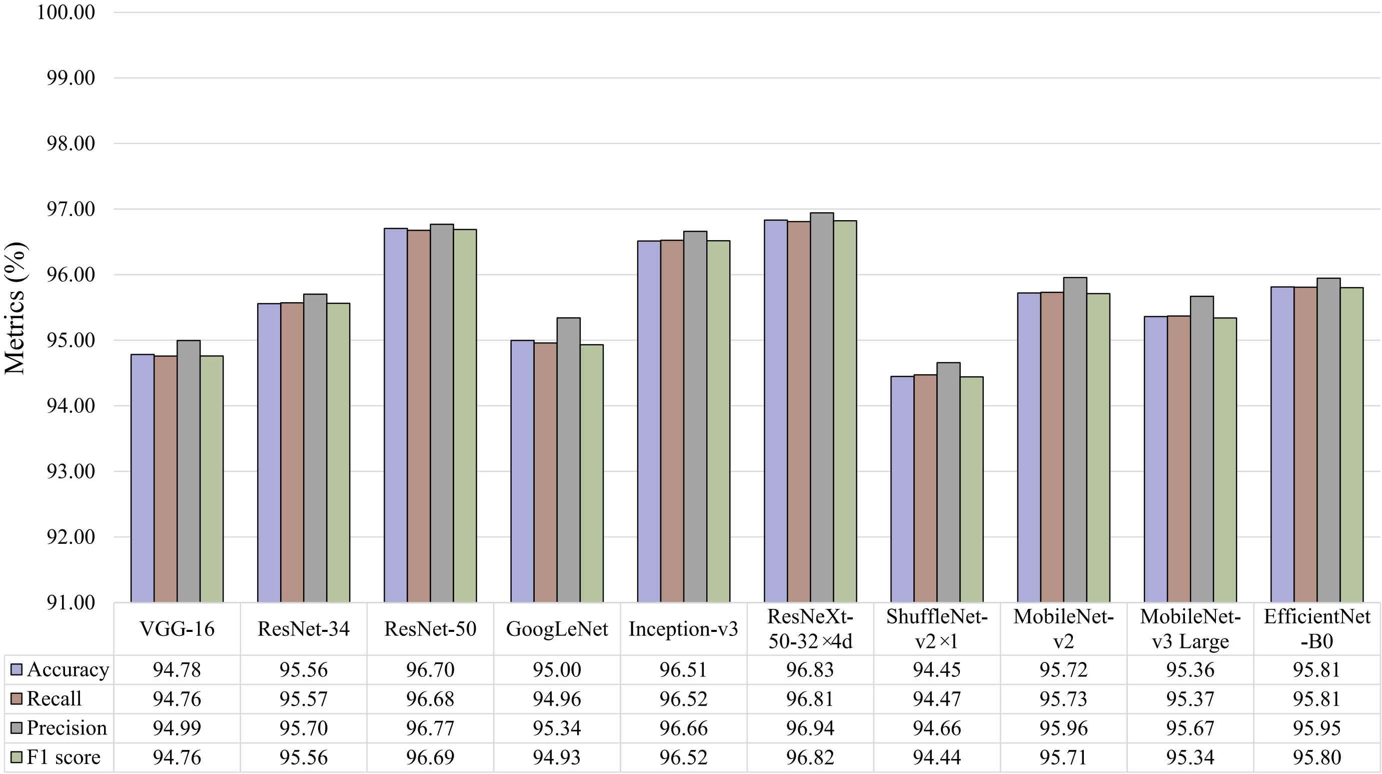
The average accuracy, average recall, average precision, and average F1 score for the evaluated deep learning models are summarized in Fig. 7. The average values refer to the average of the metrics of each model over the four datasets. As shown in Fig. 7, all the pre-trained CNNs have obtained values over 94% in terms of all four indicators. Among all the networks, ResNeXt-50-32×4d is the best selection with the average accuracy, average recall, average precision and, average F1 score being 96.83%, 96.81%, 96.94%, and 96.82%, respectively. Additionally, ResNet-50 also has relatively high scores on the given metrics (accuracy 96.70%, recall 96.68%, precision 96.77%, and F1 score 96.69%). In addition to the family ResNet, the inception architecture is also prevalent in computer vision research. Among the evaluated model, the performance of Inception-v3 is second only to ResNeXt-50-32×4d and ResNet-50 and higher than other networks. VGG16 attained scores only higher than that of ShuffleNet-v2×1 (accuracy 94.45%, recall 94.47%, precision 94.66%, F1 score 94.44%), which is a lightweight network developed for mobile devices.

The average F1 score is the harmonic measure of recall and precision and can be used as a comprehensive indicator to evaluate the overall performance of the models. As given in Fig. 8 and Table S2, the F1 score for different model-dataset pairs varies significantly, ranging from 89% to 99%. All the networks performed consistently well on the PBC with the highest value of 98.46% by ResNet-50 and the lowest value of 97.05% by ShuffleNet-v2×1). The performance of individual approaches varies significantly on the LICS dataset, with the maximum F1 score difference up to 4.30% (i.e., 97.72% by ResNet-50 vs. 93.42% by GoogLeNet). For the C-NMC dataset, the F1 score of the models also differs from 89.75% (by ShuffleNet-v2×1) to 92.77% (by ResNeXt-50-32×4d). The average F1 score of the ten evaluated models is significantly higher in ALL-IDB2 (98.00%) and PBC (97.85%) than that in C-NMC (91.72%) and LISC (95.06%). As for the performance of different algorithms, ResNeXt-50-32×4d attained the highest F1 score in C-NMC (92.77%) and ALL-IDB2 (98.85%), while ResNet-50 outperformed other methods in the PBC (98.46%) and LISC (97.72%).

4.4. Computational Efficiency

One of the drawbacks of deep neural networks preventing their application in some real-world applications, such as robots, cellphones, and quadcopters, where the resource is limited, is their expensive computation-consuming. Increasing the consumption complexity not only affects the experience but also does harm to the device. Therefore, in practical applications, the efficiency of the deep learning models must be aware carefully. Herein, we conducted extensional experiments to analyze each model's efficiency quantitatively. Instead of using indirect metrics, such as the volume of weight and the number of FLOPs, inference latency, a more direct indicator reflecting both software and hardware situation, was utilized to measure the efficiency of different networks. The models were run and assessed on the proposed software for a fair comparison.

The inference latency of the methods mentioned above for processing the samples from the PBC dataset is shown shown in Table 3. The experiments were conducted on an ordinary personal computer with a CPU of Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz, and a RAM of 8GB memory and the results were the average value over ten runs under the same settings (Table S3 and dummyTXdummy-(Table S4). More specifically, two fashions of batch processing with 100 images as input and single processing with one image as input were compared separately. When only



	VGG-16	ResNet-34	ResNet-50	GoogLeNet	Inception-v3	ResNeXt-50-32×4d	ShuffleNet-v2×1	MobileNet-v2	MobileNet-v3 Large	EfficientNet-B0
Accuracy	94.78	95.56	96.70	95.00	96.51	96.83	94.45	95.72	95.36	95.81
Recall	94.76	95.57	96.68	94.96	96.52	96.81	94.47	95.73	95.37	95.81
Precision	94.99	95.70	96.77	95.34	96.66	96.94	94.66	95.96	95.67	95.95
F1 score	94.76	95.56	96.69	94.93	96.52	96.82	94.44	95.71	95.34	95.80

Fig. 7. The average accuracy, recall, precision and F1 score of models over the four datasets.

Table 3
The inference latency of the models with the input of 100 images and one image, respectively. The scaling ratio refers to the latency of 100 images to that of one image.

Models	100 images (seconds)	1 image (seconds)	Ratio
VGG-16	17.179	1.677	10.24
ResNet-34	5.973	0.333	17.93
ResNet-50	9.195	0.405	22.71
GoogLeNet	3.444	0.209	16.46
Inception-v3	9.686	0.439	22.06
ResNeXt-50-32×4d	9.928	0.420	23.64
ShuffleNet-v2×1	1.857	0.080	23.18
MobileNet-v2	3.266	0.109	29.88
MobileNet-v3-Large	2.518	0.108	23.24
EfficientNet-B0	4.475	0.150	29.89

one image was fed into the platform, except for VGG-16 (1.677s), all other methods could identify it within 0.5 seconds. The mobile-size networks (less than 0.15s) run much faster than the conventional ones (more than 0.209s) as expected. The ShuffleNet-v2×1 is the most efficient taking only 0.08 s to recognize the input in single processing mode, following by MobileNet-v3-Large (0.108 s), MobileNet-v2 (0.109 s) and EfficientNet-B0 (0.150 s). The same trend was also observed when a batch of 100 samples was fed into the platform.

5. Discussion

The example application demonstrated in Section 4.2 shows the potential usability of the software that, by using AIMIC, deep learning models can be trained to recognize new samples with a high probability of correctness without the need for programming. Moreover, the proposed software can run on an ordinary laptop, making it possible for users to implement the method in a point-of-care manner. The training process converged in fewer epochs (around 30) with high accuracy, revealing that deep learning algorithms can effectively improve their versatility and robustness with knowledge learned from other domains.

As can be seen in Fig. 7, among all evaluated models, the ResNeXt-50-32×4d and ResNet-50 obtained the first and second best scores in terms of the standard metrics, respectively. Fundamentally, these two models share a similar backbone frame by stacking residual block layers repeatedly. The only improvement of the ResNeXt-50-32×4d is that it replaces the original building block of ResNet-50 by aggregating multiple transformations of the uniform topology. The experimental results certified that the residual-based backbone has robust feature extraction capabilities, which is of central importance in the current deep learning network dealing with computer vision tasks because the features extracted by the backbone are the basis for the further process. Interestingly, for the lightweight networks, MobileNet-v2 outperformed its next-generation counterpart, MobileNet-v3-Large, which is equipped with the squeeze-and-excite block [59], in all four criteria (i.e., accuracy 95.72% vs. 95.36%, recall 95.73% vs. 95.37%, precision 95.96% vs. 95.67%, and F1 score 95.71% vs. 95.34%). Although previous studies reported that MobileNet-v3-Large worked better than the V2 version in the ImageNet consisting of large images with various characteristics, MobileNet V3 is more prone to overfitting in microscopic datasets without adequate data and rich features. Moreover, while EfficientNet-B0 and MobileNet-V2 performed comparatively in classification performance metrics, MobileNet-V2 needs less time to identify a new sample (0.109s vs. 0.150s), as summarized in Table 2. Therefore, MobileNet-v2 is an appropriate choice for microscopic recognition of blood cells when computational resources are limited.

As shown in Fig. 8 and Table S2, the performance of different models on the PBC dataset is more steady than that on the LISC dataset. This is because, compared with LISC (Fig. S5), the target objects in the samples of the PBC are much more salient (Fig. S6). A similar situation (only 1.93% difference in F1 score of different

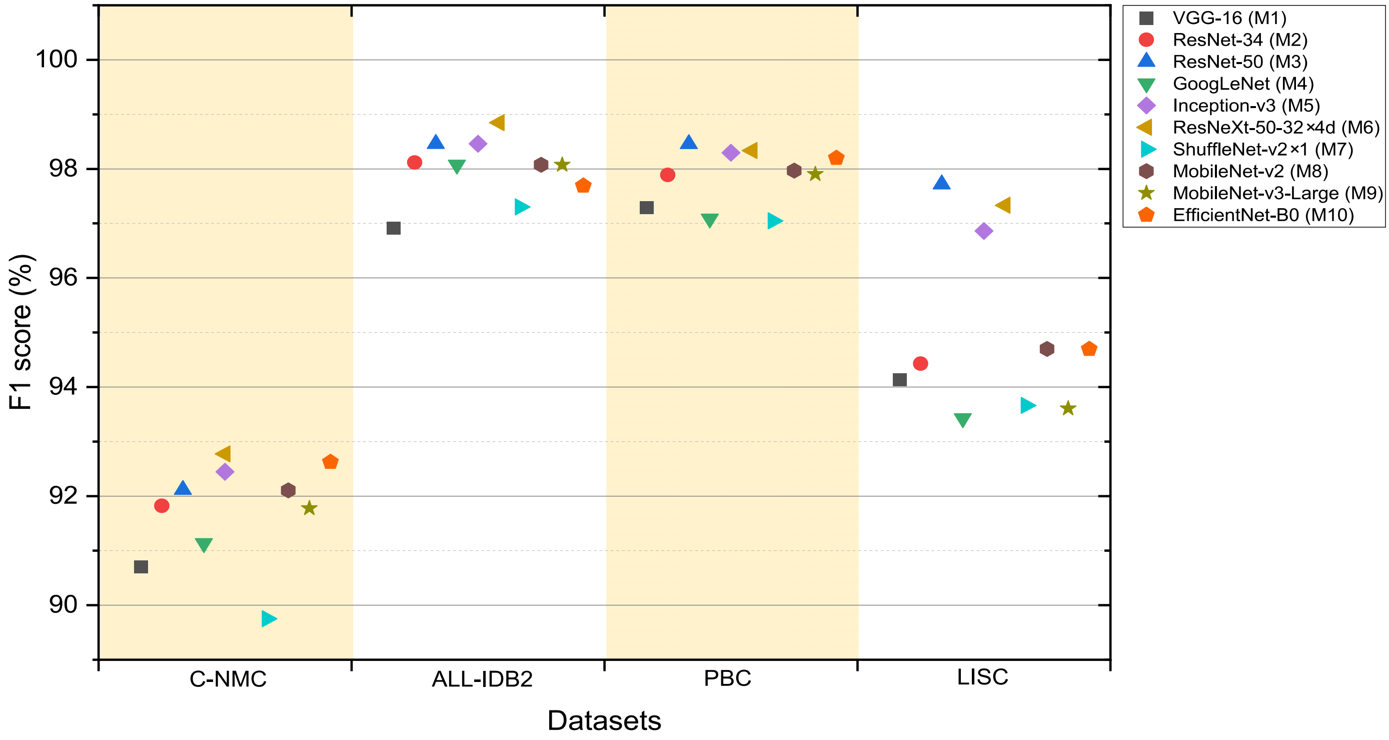


Fig. 8. The F1 score for each model-dataset pair. M1, M2, M3, M4, M5, M6, M7, M8, M9 and M10 denote VGG16, ResNet-34, ResNet-50, GoogLeNet, Inception-v3, ResNeXt-50-32×4d, ShuffleNet-v2×1, MobileNet-v2, MobileNet-v3-Large and EfficientNet-B0, respectively.

approaches) was also observed in ALL-IDB2 (Fig. S7), whose data has similar characteristics to that in PBC. The indicators of different models on the C-NMC dataset vary significantly since the features of the samples in the same class differ widely (Fig. S8), making it hard for the different algorithms to learn. Hence, the performance of CNNs is largely determined by the saliency of the target objects in the images and the consistency of their features. ResNeXt-50-32×4d and ResNet-50 achieved the top two highest scores on all four datasets, further verifying the feature capture ability of the backbone of the family ResNet.

In addition to the recognition performance, the time complexity is an essential indicator for assessing a machine learning model [70]. Traditionally, the FLOPs is used to measure the efficiency of AI models. Besides the FLOPs, the computational complexity of a deep learning network is also affected by several other factors, such as the parallelism degree of the architecture and the memory access cost [45]. In this study, inference latency is used to evaluate the time complexity as it is a composite metric taking into account all these factors. In the experiments evaluating computational efficiency, it was interesting that as the number of input images increased, the lightweight models' inference latency grew faster than that of most traditional models, see Table 3, Table S3, and Table S4. As the input batch grew from one to 100, the time of the efficient models spent to process the images increased by more than 23 folds, while the latency of all the conventional networks except ResNeXt-50-32×4d increased by less than 23 times. This can be attributed to the fact that the initial processing time of conventional CNNs for recognizing objects is longer than that of efficient networks. Hence, when the computing budget is affluent, the traditional models may be an ideal choice for large-scale classification in a batch. But for the limited resource scenarios, MobileNet-v2 might be a suitable choice for better trade-off accuracy and computational efficiency for microscopic image classification.

Compared to the ZeroCostDL4Mic [71], a cloud-based and browser-based deep learning-deploy platform for image processing, the AIMIC is a fully out-of-the-box solution independent of the Internet and browsers. Besides, the present platform aims at classification, while the ZeroCostDL4Mic focuses on image processing tasks, such as image segmentation, denoising, and restoration. The proposed AIMIC is more user-friendly to novice users than other AI software, such as AIDeveloper [72], by providing a concise yet functional user interface. Furthermore, a hands-off label technique tailored for beginner users is built into the platform. More importantly, the new software offers an automatic k-fold cross-validation mode to facilitate the evaluation of the selected algorithms.

We compare the classification performance of the proposed platform with the state-of-the-art methods. As summarized in Table 4, the classification accuracy obtained by AIMIC surpasses all competitors on the C-NMC, ALL-IDB2, and LISC datasets. It performed on par with the state-of-the-art on the PBC dataset. The best results among all built-in models are shown in Table 4. It is worth noting that a plain implementation of our method was conducted without fine-tuning, which further demonstrates the capabilities of the AIMIC.

The proposed platform has been proven to be a useful tool for microscopic image classification without the need for programming. However, training a deep learning model is a compute-intensive job. In the above example, it took nearly an hour to train a lightweight network (i.e., MobilNet-V2) on the high-end GPU. Without GPU acceleration, the training time could be several hours. In extreme scenarios, the training process cannot be performed effectively, such as training a large model on a device with insufficient memory, which may lead to unsatisfactory classification results. Furthermore, deep learning models are data-hungry. Generally, the larger the size of the training dataset, the better the learning effect of the deep learning network. Therefore, to train the model better, the user has to spend a lot of time and effort collecting and annotating the samples.

Table 4
Performance comparison with the state-of-the-art methods on C-NMC, ALL-IDB2, PBC, and LISC datasets. The best results are shown in bold, and the second-best results are indicated by underlining.

C-NMC		ALL-IDB2		PBC		LISC	
Methods	Accuracy	Methods	Accuracy	Methods	Accuracy	Methods	Accuracy
Zhao et al. [68]	86.29%	Abdeldaim et al. [29]	96.42%	Acevedo et al. [35]	96.2%	Noor et al. [28]	94.09%
Prellberg et al. [33]	89.88%	Sahlol et al. [69]	98.5%	Bagido et al. [36]	98.4%	Jung et al. [37]	96%
Pan et al. [34]	91.73%	Das et al. [41]	97.18%	Long et al. [67]	**99.3%**	Baydilli et al. [38]	96.86%
Proposed	**92.77%**	Proposed	**98.85%**	Proposed	98.46%	Proposed	**97.78%**

6. Conclusion

This paper presents a platform, AIMIC, for microscopic image classification to those with limited programming experience. The proposed software embedded with state-of-the-art algorithms and data preprocessing techniques can be used to identify images in a code-free manner. Moreover, the out-of-box platform has a user-friendly interface and can run on an ordinary personal computer. If GPUs are detected on the device, the AIMIC can automatically employ the GPUs for processing, which will be more convenient for novices. The extensive experiments have verified the platform's usability by obtaining robust classification results on multiple datasets. We believe the proposed software will lower the barrier for beginners to implement deep learning techniques in their work. In addition, to help the cytologist select a suitable algorithm for a specific application, we also investigated ten popular deep learning methods for classifying blood cells. The evaluation experiments tested the models on four publicly available datasets. The results show that two approaches in the ResNet family, ResNeXt-50-32×4d and ResNet-50, outperformed the other evaluated algorithms, suggesting that the backbone of ResNet has a robust learning capability. For mobile device-oriented models, MobileNet-v2 performs on par with EfficientNet-B0 but with less inference latency. Therefore, MobileNet-v2 is an appropriate solution for microscopic image classification when computational resources are limited. In addition to cell classification, other microscopic image processing (e.g., cell segmentation, cell counting, and cell detection) is also crucial for biomedical applications. In the future, the techniques for these applications will be investigated and added to the AIMIC platform.

In a nutshell, this study provides a critical reference for selecting a suitable deep learning model, and the proposed AIMIC system offers a simplified solution to take advantage of artificial intelligence development for microscopic image analysis.

Data Availability

All data that support the conclusion of this study are available in the main text and the Supplementary Materials. The code is available at https://github.com/RuiLiuvw/AIMIC.

Author Contributions

J. Liu supervised the project. R. Liu and S. Wan conceived the idea. R. Liu designed the experiments and prepared the draft of manuscript. R. Liu and W. Dai developed the software and conducted the experiments. R. Liu, T. Wu and M. Wang analyzed the data and discussed the results. J. Liu and S. Wan revised the manuscript

Declaration of Competing Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the Research Grant Council (RGC) of Hong Kong under Grant 11212321, 11217922, and Grant ECS-21212720, Basic and Applied Basic Research Foundation of Guangdong Province Fund Project 2019A1515110175, and Science, Technology and Innovation Committee of Shenzhen under Grant SGDX20210823104001011.

Supplementary material

Fig. S1. The Schematic diagram of residual learning. Fig. S2. (a) The original residual block in ResNet. (b) The corresponding transformation of (a) in ResNeXt with a cardinality of 32. Fig. S3. (a) is the depthwise separable convolutions, where a regular convolution is separated into a depthwise convolution operator followed by a pointwise one. (b) illustrates the inverted residual block, the size ratio between the input and the inner part of which is different from that of conventional residual block, where k <1. Fig. S4. Squeeze-and-Excite block. Fig. S5. Example of samples from the LISC database. From left to right: basophil, lymphocyte, monocyte, neutrophil. Fig. S6. Example of samples from the PBC dataset. Top (from left to right): basophil, eosinophil, erythroblast, immature granulocyte. Bottom (from left to right): lymphocyte, monocyte, neutrophil, platelet. Fig. S7. Example of samples from the ALL-IDB2 dataset. Left: probable lymphoblast from ALL patients. Right: healthy cell from non-ALL patients. ALL stands for Acute Lymphoblastic Leukemia. Fig. S8. Example of samples from the C-NMC image set. Top: acute lymphoblastic leukemia cells. Bottom: normal cells. Table S1. The packages used in the software. Table S2. The average F1 score of each model-dataset pair. Table S3. The inference latency of the models with the input of 1 sample. No.# denote the experiment number. Table S4. The inference latency of the models with the input of 100 samples. No.# denote the experiment number. Video S1. User tutorial of the software.

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cmpb.2022.107162.

References

[1] J. Liu, C. Leung, Z. Lu, Y. Sun, Quantitative analysis of locomotive behavior of human sperm head and tail, IEEE Transactions on Biomedical Engineering 60 (2) (2012) 390–396.

[2] A. Uyar, S. Torrealday, E. Seli, Cumulus and granulosa cell markers of oocyte and embryo quality, Fertility and sterility 99 (4) (2013) 979–997.

[3] R. Hoffman, E.J. Benz Jr, L.E. Silberstein, H. Heslop, J. Anastasi, J. Weitz, Hematology: basic principles and practice, Elsevier Health Sciences, 2013.

[4] J. Ge, Z. Gong, J. Chen, J. Liu, J. Nguyen, Z. Yang, C. Wang, Y. Sun, A system for counting fetal and maternal red blood cells, IEEE Transactions On Biomedical Engineering 61 (12) (2014) 2823–2829.

[5] K. Shaukat, S. Luo, V. Varadharajan, I.A. Hameed, S. Chen, D. Liu, J. Li, Performance comparison and current challenges of using machine learning techniques in cybersecurity, Energies 13 (10) (2020) 2509.

[6] K. Shaukat, S. Luo, V. Varadharajan, I.A. Hameed, M. Xu, A survey on machine learning techniques for cyber security in the last decade, IEEE Access 8 (2020) 222310–222354.

[7] M.I. Jordan, T.M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.

[8] W. Dai, R. Liu, T. Wu, M. Wang, J. Yin, J. Liu, Hierattn: Effectively learn representations from stage attention and branch attention for skin lesions diagnosis, arXiv preprint arXiv:2205.04326 (2022).

[9] C. Srinivas, N.P. KS, M. Zakariah, Y.A. Alothaibi, K. Shaukat, B. Partibane, H. Awal, Deep transfer learning approaches in performance analysis of brain tumor classification using mri images, Journal of Healthcare Engineering 2022 (2022).

[10] L. Devnath, P. Summons, S. Luo, D. Wang, K. Shaukat, I.A. Hameed, H. Aljuaid, Computer-aided diagnosis of coal workers' pneumoconiosis in chest x-ray radiographs using machine learning: A systematic literature review, International Journal of Environmental Research and Public Health 19 (11) (2022) 6439.

[11] T.M. Alam, K. Shaukat, I.A. Hameed, W.A. Khan, M.U. Sarwar, F. Iqbal, S. Luo, A novel framework for prognostic factors identification of malignant mesothelioma through association rule mining, Biomedical Signal Processing and Control 68 (2021) 102726.

[12] N. Ramesh, B. Dangott, M.E. Salama, T. Tasdizen, Isolation and two-step classification of normal white blood cells in peripheral blood smears, Journal of pathology informatics 3 (2012).

[13] A.I. Shahin, Y. Guo, K.M. Amin, A.A. Sharawi, White blood cells identification system based on convolutional deep neural learning networks, Computer methods and programs in biomedicine 168 (2019) 69–80.

[14] L. Bi, W. Gao, L. Meng, G. Gu, Z. Shi, Y. BAI, Deep learning for discovering and identifying morphological heterogeneity of neutrophils in primary hematological diseases based on bone marrow neutrophils analysis, Blood 136 (2020) 18.

[15] J.-W. Sidhom, I.J. Siddarthan, B.-S. Lai, A. Luo, B. Hambley, J. Bynum, A.S. Duffield, M.B. Streiff, A.R. Moliterno, P.H. Imus, et al., Deep learning for distinguishing morphological features of acute promyelocytic leukemia, Blood 136 (2020) 10–12.

[16] H. Shao, W. Gao, Q. Zhang, J. Li, D. Zhou, L. Bi, Y. BAI, Z. Shi, Transfer learning for identifying morphological heterogeneity of neutrophils nuclei in hematological diseases based on nuclei semantic segmentations of bone marrow smear, Blood 136 (2020) 1.

[17] R. Arnaout, L. Curran, Y. Zhao, J.C. Levine, E. Chinn, A.J. Moon-Grady, An ensemble of neural networks provides expert-level prenatal detection of complex congenital heart disease, Nature medicine 27 (5) (2021) 882–891.

[18] Y. Li, Y. Zhang, Z. Zhu, Error-tolerant deep learning for remote sensing image scene classification, IEEE transactions on cybernetics 51 (4) (2020) 1756–1768.

[19] S. Mozaffari, O.Y. Al-Jarrah, M. Dianati, P. Jennings, A. Mouzakitis, Deep learning-based vehicle behavior prediction for autonomous driving applications: A review, IEEE Transactions on Intelligent Transportation Systems 23 (1) (2020) 33–47.

[20] R. Gargeya, T. Leng, Automated identification of diabetic retinopathy using deep learning, Ophthalmology 124 (7) (2017) 962–969.

[21] M. Gehrung, M. Crispin-Ortuzar, A.G. Berman, M. O'Donovan, R.C. Fitzgerald, F. Markowetz, Triage-driven diagnosis of barrett's esophagus for early detection of esophageal adenocarcinoma using deep learning, Nature medicine 27 (5) (2021) 833–841.

[22] J. Zhang, Y. Xie, Y. Li, C. Shen, Y. Xia, Covid-19 screening on chest x-ray images using deep learning based anomaly detection, arXiv preprint arXiv:2003.12338 27 (2020).

[23] A.T. Holzinger, H. Muller, Toward human–ai interfaces to support explainability and causability in medical ai, Computer 54 (10) (2021) 78–86.

[24] H. Muller, M.T. Mayrhofer, E.-B. Van Veen, A. Holzinger, The ten commandments of ethical medical ai, Computer 54 (07) (2021) 119–123.

[25] G. Ongun, U. Halici, K. Leblebicioglu, V. Atalay, M. Beksaç, S. Beksaç, Feature extraction and classification of blood cells for an automated differential blood count system, in: IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222), volume 4, IEEE, 2001, pp. 2461–2466.

[26] K. Kim, J. Song, F. Golshani, S. Panchanathan, Automatic classification of cells using morphological shape in peripheral blood images, in: Internet Multimedia Management Systems, volume 4210, International Society for Optics and Photonics, 2000, pp. 290–298.

[27] W.-L. Tai, R.-M. Hu, H.C. Hsiao, R.-M. Chen, J.J. Tsai, Blood cell image classification based on hierarchical svm, in: 2011 IEEE International Symposium on Multimedia, IEEE, 2011, pp. 129–136.

[28] A.M. Noor, Z. Zakaria, A.M. Noor, A.N. Norali, Classification of white blood cells based on surf feature, Suranaree Journal of Science & Technology 28 (1) (2021).

[29] A.M. Abdeldaim, A.T. Sahlol, M. Elhoseny, A.E. Hassanien, Computer-aided acute lymphoblastic leukemia diagnosis system based on image analysis, in: Advances in Soft Computing and Machine Learning in Image Processing, Springer, 2018, pp. 131–147.

[30] X. Li, W. Li, X. Xu, W. Hu, Cell classification using convolutional neural networks in medical hyperspectral imagery, in: 2017 2nd international conference on image, vision and computing (ICIVC), IEEE, 2017, pp. 501–504.

[31] D. Mundhra, B. Cheluvaraju, J. Rampure, T. Rai Dastidar, Analyzing microscopic images of peripheral blood smear using deep learning, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Springer, 2017, pp. 178–185.

[32] P. Huang, J. Wang, J. Zhang, Y. Shen, C. Liu, W. Song, S. Wu, Y. Zuo, Z. Lu, D. Li, Attention-aware residual network based manifold learning for white blood cells classification, IEEE Journal of Biomedical and Health Informatics 25 (4) (2020) 1206–1214.

[33] J. Prellberg, O. Kramer, Acute lymphoblastic leukemia classification from microscopic images using convolutional neural networks, in: ISBI 2019 C-NMC Challenge: Classification in Cancer Cell Imaging, Springer, 2019, pp. 53–61.

[34] Y. Pan, M. Liu, Y. Xia, D. Shen, Neighborhood-correction algorithm for classification of normal and malignant cells, in: ISBI 2019 C-NMC Challenge: Classification in Cancer Cell Imaging, Springer, 2019, pp. 73–82.

[35] A. Acevedo, S. Alferez, A. Merino, L. Puigv, J. Rodellar, Recognition of peripheral blood cell images using convolutional neural networks, Computer methods and programs in biomedicine 180 (2019) 105020.

[36] R.A. Bagido, M. Alzahrani, M. Arif, White blood cell types classification using deep learning models, International Journal of Computer Science & Network Security 21 (9) (2021) 223–229.

[37] C. Jung, M. Abuhamad, J. Alikhanov, A. Mohaisen, K. Han, D. Nyang, W-net: a cnn-based architecture for white blood cells image classification, arXiv preprint arXiv:1910.01091 (2019).

[38] Y.Y. Baydilli, Ü. Atila, Classification of white blood cells using capsule networks, Computerized Medical Imaging and Graphics 80 (2020) 101699.

[39] S. Sabour, N. Frosst, G.E. Hinton, Dynamic routing between capsules, Advances in neural information processing systems 30 (2017).

[40] X. Yao, K. Sun, X. Bu, C. Zhao, Y. Jin, Classification of white blood cells using weighted optimized deformable convolutional neural networks, Artificial Cells, Nanomedicine, and Biotechnology 49 (1) (2021) 147–155.

[41] P.K. Das, S. Meher, An efficient deep convolutional neural network based detection and classification of acute lymphoblastic leukemia, Expert Systems with Applications 183 (2021) 115311.

[42] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105–6114.

[43] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1314–1324.

[44] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.

[45] N. Ma, X. Zhang, H.-T. Zheng, J. Sun, Shufflenet v2: Practical guidelines for efficient cnn architecture design, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 116–131.

[46] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1492–1500.

[47] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[49] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[50] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[51] A. Gupta, R. Gupta, Isbi 2019 c-nmc challenge: Classification in cancer cell imaging, Springer, Singapore. doi 10 (2019) 978–981.

[52] R.D. Labati, V. Piuri, F. Scotti, All-idb: The acute lymphoblastic leukemia image database for image processing, in: 2011 18th IEEE international conference on image processing, IEEE, 2011, pp. 2045–2048.

[53] A. Acevedo, A. Merino, S. Alférez, Á. Molina, L. Boldú, J. Rodellar, A dataset of microscopic peripheral blood cell images for development of automatic recognition systems, Data in brief 30 (2020).

[54] S.H. Rezatofighi, H. Soltanian-Zadeh, Automatic recognition of five types of white blood cells in peripheral blood, Computerized Medical Imaging and Graphics 35 (4) (2011) 333–343.

[55] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012).

[56] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6848–6856.

[57] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).

[58] L. Sifre, S. Mallat, Rigid-motion scattering for texture classification, arXiv preprint arXiv:1403.1687 (2014).

[59] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[60] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, in: International conference on artificial neural networks, Springer, 2018, pp. 270–279.

[61] S. Shao, S. McAleer, R. Yan, P. Baldi, Highly accurate machine fault diagnosis using deep transfer learning, IEEE Transactions on Industrial Informatics 15 (4) (2018) 2446–2455.

[62] Z. Huang, C.O. Dumitru, Z. Pan, B. Lei, M. Datcu, Classification of large-scale high-resolution sar images with deep transfer learning, IEEE Geoscience and Remote Sensing Letters 18 (1) (2020) 107–111.

[63] S. Khan, N. Islam, Z. Jan, I.U. Din, J.J.C. Rodrigues, A novel deep learning based framework for the detection and classification of breast cancer using transfer learning, Pattern Recognition Letters 125 (2019) 1–6.

[64] M. Khushi, K. Shaukat, T.M. Alam, I.A. Hameed, S. Uddin, S. Luo, X. Yang, M.C. Reyes, A comparative performance analysis of data resampling methods on imbalance medical data, IEEE Access 9 (2021) 109960–109975.

[65] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Information processing & management 45 (4) (2009) 427–437.

[66] J. Riordon, C. McCallum, D. Sinton, Deep learning for the classification of human sperm, Computers in biology and medicine 111 (2019) 103342.

[67] F. Long, J.-J. Peng, W. Song, X. Xia, J. Sang, Bloodcaps: A capsule network based model for the multiclassification of human peripheral blood cells, Computer methods and programs in biomedicine 202 (2021) 105972.

[68] C. Zhao, Y. Feng, R. Liu, W. Zheng, Application of lightweight convolution neural network in cancer diagnosis, in: Proceedings of the 2020 Conference on Artificial Intelligence and Healthcare, 2020, pp. 249–253.

[69] A.T. Sahlol, P. Kollmannsberger, A.A. Ewees, Efficient classification of white blood cell leukemia with improved swarm optimization of deep features, Scientific Reports 10 (1) (2020) 1–11.

[70] K. Shaukat, S. Luo, S. Chen, D. Liu, Cyber threat detection using machine learning techniques: A performance evaluation perspective, in: 2020 International Conference on Cyber Warfare and Security (ICCWS), IEEE, 2020, pp. 1–6.

[71] L. von Chamier, R.F. Laine, J. Jukkala, C. Spahn, D. Krentzel, E. Nehme, M. Lerche, S. Hernández-Pérez, P.K. Mattila, E. Karinou, et al., Democratising deep learning for microscopy with zerocostdl4mic, Nature communications 12 (1) (2021) 1–18.

[72] M. Kräter, S. Abuhattum, D. Soteriou, A. Jacobi, T. Krüger, J. Guck, M. Herbig, Aideveloper: deep learning image classification in life science and beyond, Advanced science 8 (11) (2021) 2003743.