Heliyon 9 (2023) e22203

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Sickle cell disease classification using deep learning

Sanjeda Sara Jennifer [a], Mahbub Hasan Shamim [a], Ahmed Wasif Reza [a,*], Nazmul Siddique [b]

[a] *Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh*
[b] *School of Computing, Engineering and Intelligent Systems, Ulster University, UK*

ARTICLE INFO

Keywords:
Sickle cell disease
Classification
Ablation experiment
Deep learning model
Machine learning classifier

ABSTRACT

This paper presents a transfer and deep learning based approach to the classification of Sickle Cell Disease (SCD). Five transfer learning models such as ResNet-50, AlexNet, MobileNet, VGG-16 and VGG-19, and a sequential convolutional neural network (CNN) have been implemented for SCD classification. ErythrocytesIDB dataset has been used for training and testing the models. In order to make up for the data insufficiency of the erythrocytesIDB dataset, advanced image augmentation techniques are employed to ensure the robustness of the dataset, enhance dataset diversity and improve the accuracy of the models. An ablation experiment using Random Forest and Support Vector Machine (SVM) classifiers along with various hyperparameter tweaking was carried out to determine the contribution of different model elements on their predicted accuracy. A rigorous statistical analysis was carried out for evaluation and to further evaluate the model's robustness, an adversarial attack test was conducted. The experimental results demonstrate compelling performance across all models. After performing the statistical tests, it was observed that MobileNet showed a significant improvement ($p = 0.0229$), while other models (ResNet-50, AlexNet, VGG-16, VGG-19) did not ($p > 0.05$). Notably, the ResNet-50 model achieves remarkable precision, recall, and F1-score values of 100 % for circular, elongated, and other cell shapes when experimented with a smaller dataset. The AlexNet model achieves a balanced precision (98 %) and recall (99 %) for circular and elongated shapes. Meanwhile, the other models showcase competitive performance.

1. Introduction

Sickle cell disease (SCD) is a genetic disorder, brought on by a beta-globin gene mutation [1], leading to abnormal haemoglobin formation [2]. Haemoglobin is the main component of Red Blood Cells [3]. Blood disorders such as Sickle Cell anaemia and Diamond-Blackfan anaemia are genetic disorders that may have an impact on the body's ability to produce red blood cells [4]. It affects the bone marrow's capability to create RBC [5]. A transgenic mouse model of SCD was found to have a higher steady-state level of adenosine in the blood when metabolomic profiling was used [6]. People with SCD [7] are at a higher risk of cognitive dysfunction compared to their healthy peers [8]. Sickle cells contain more excessive haemoglobin S than haemoglobin A [9]. SCD arises from a mutation in the β-globin gene [10]. Red blood cells are in charge of carrying oxygen throughout the body. This inherited blood disorder causes the RBC to become sickle-shaped and restricts the oxygen supply and blood flow on different body parts due to their abnormal

[*] Corresponding author.
E-mail addresses: jennifer.sanjedasara@gmail.com (S.S. Jennifer), shamim.mahbub.hasan@gmail.com (M.H. Shamim), wasif@ewubd.edu (A.W. Reza), nh.siddique@ulster.ac.uk (N. Siddique).

https://doi.org/10.1016/j.heliyon.2023.e22203
Received 15 May 2023; Received in revised form 24 October 2023; Accepted 6 November 2023
Available online 12 November 2023

shape as they become stiff and get stuck in the blood vessels easily. It was reported by Platt et al. [11] from an investigation on 3685 patients that the typical life expectancy for males and females with SCD was 42 and 48 years, respectively. Ali et al. [12] suspected that sickle cells get stuck in the vascular system of the pulp and restrict blood flow causing a toothache. Recently, people with severe and mild-to-moderate SCD genotypes died from COVID-19 [13]. Treatments such as pain management, blood transfusions, and antibiotics can help manage symptoms and prevent complications. Besides this, several studies were conducted, such as the pilot study done by Esrick et al. [14] to evaluate the viability and safety of gene therapy. As a possible option to cure SCD, Eapen et al. [15] took a deeper look into transplantation from sibling donors and alternative donors. However, to stop the situation from getting worse, DeBaun et al. [16] suggested 19 recommendations for different conditions that can provide support to the SCD community. Therefore, classification of SCD from histopathological images is important for diagnosis and treatment, which makes it a challenging research problem for the machine learning community.

The importance of deep learning [17] in medical image processing is increasing due to the availability of many tools and techniques of deep learning (DL) along with the advent of new digital medical technologies [18]. DL can be applied to medical problems more easily with enhanced accuracy and efficiency for diagnosis leading to better patient care [19]. For example, deep learning and machine learning models [20] have been applied to brain tumor classification [21] through image segmentation, automated detection, multi-modal analysis, interpretable explanations and hybrid convolutional neural network architecture [22]. Another study by Javed et al. [23] showed that deep transfer learning can be used to improve the performance of brain tumor classification algorithms. Similar examples are breast cancer detection using mammography images [24,25] and lung cancer classification [26]. Even for the identification of COVID-19, Ardakani et al. [27] studied ten CNN architectures to find the most efficient model. Recent studies have indicated that DL can be applied to improve the diagnosis of SCD. For example, a study by Ref. [28] showed that a DL model was able to classify sickle cell images with an accuracy of 95 %. The contributions of this research are as follows:

Clinical method for the classification of SCD from histopathological data involving human experts is impractical due to the considerable time and financial requirements. An intelligent classification system can mitigate these issues, improve classification accuracy, and provide an automated approach to SCD identification. Thus, the need for such a DL-based application has motivated the development of the SCD classification system.

A number of CNN-based models with tuned hyperparameters have been applied to SCD classification but little success is achieved. Several researchers have also applied transfer learning to SCD classification without considering hyperparameter tuning. In this research, transfer models have been used for tuning the hyperparameters on the specific dataset leading to the second contribution.

This research used erythrocytesIDB dataset. The dataset erythrocytesIDB is a very small dataset containing only 629 images of 3 classes (circular, elongated, and other). The dataset is well balanced (see Table 2), but a dataset with only 629 images is too small against training a DL or transfer learning (TL) model. Thriving towards higher accuracy will simply lead to an overfitting problem or performance degradation. DL or TL will require several thousands of images. Data augmentation is required. This research explicitly addresses the data augmentation issue leading to the third contribution.

In this research, 15 fine-tuned transfer learning models are generated (five baseline models such as ResNet50, AlexNet, MobileNet, VGG16 and VGG19 with three classifiers such as fully connected network (FCN), support vector machine (SVM) and random forest (RF). The best performing models are identified according to classification accuracy and statistically significant tests, e.g., *t*-test and p-value. These models are refined through ablation experiments. This leads to the fourth contribution.

The remaining part of the paper is organized into multiple sections. The literature review on sickle cell disease is presented in Section 2. Section 3 discusses the materials and proposed methods of preparation, leading to section 4, which portrays the experimentation and result analysis. Finally, section 5 concludes the paper.

2. Previous works

This section is presented in three parts. The first part provides the threats of sickle cell disease, the second part presents the search procedure of how the literature survey was done, and finally, the last part provides the literature review on sickle cell disease.

2.1. Sickle cell disease and challenges

SCD is a complicated genetic illness that necessitates ongoing efforts to better understand, diagnose, and treatment. It is crucial to examine the causes of SCD, its effects, and the continuous challenges that patients and the medical profession face in this setting.

According to Panepintoa et al. [13], SCD can shorten lifespan, impair many organ systems, affect the kidney, and result in permanent disabilities. Individuals with SCD may experience pain, fatigue, and jaundice episodes [29]. Additionally, it may result in severe complications, including damage to the eyes, heart, lungs, liver, kidneys, and joints, increased susceptibility to infections and acute chest syndrome [30]. Even for children, SCD causes acute chest syndrome [31], and gall bladder disease [32].

2.2. Search procedure

Detailed research was conducted using the Google Scholar platform to locate academic sources and references pertaining to sickle cell disease, red blood cell problems, and the use of deep learning and machine learning within these domains. The keywords used in this investigation included “sickle cell,” “sickle cell disease,” “red blood cell disease,” “blood cell disease,” and others. Both independent and combined keyword queries were made to search for relevant supplemental sources and identified scholarly articles. These research articles went under an extensive evaluation to be included in the literature review section. The search results provided a thorough comprehension of earlier studies in this area, particularly highlighting the as-yet-unexplored use of deep learning approaches and ablation trials.

The use of deep learning for SCD investigation is still in its early phases. It has the prospect of changing the methods of diagnosis and treatment of SCD. In addition to the analyses mentioned above, recent work has also analyzed the use of deep learning for additional tasks related to SCD, such as identifying patients at risk of complications [33] and predicting the severity of SCD [34].

2.3. Literature review

There have been many studies related to cell segmentation using deep learning. For example, Kutlu et al. [35] detected and classified White Blood Cells (WBC) using AlexNet [36], VGG16 [37], GoogLeNet [38] and ResNet-50 architecture. Das et al. [39] made a complete and comprehensive examination of automated techniques for SCD Detection. In the study conducted by Bunn [40], he elaborated pathophysiology and management of SCD, discussed the importance of genetic counselling and prenatal diagnosis in managing the disease, available treatments such as hydroxyurea therapy, blood transfusions, and bone marrow transplantation along with the potential of new treatments, such as gene therapy and fetal haemoglobin induction. The study conducted by Piel et al. [41] used geostatistical modelling to estimate the prevalence of SCD in newborns in different regions of the world. One of the key public health strategies mentioned in the paper is to implement newborn screening programs to identify infants with SCD so that they can receive early care and medical treatment. The article makes several other recommendations for treating sickle cell disease, such as expanding access to complete care, taking preventative measures, managing pain, receiving psychosocial support, and enhancing the availability of secure blood transfusions. Another clinical practice involves pulse oximetry to detect SCD in adult patients and it was concluded as a reliable screening tool [42].

Unlike other healthcare sectors, there is little prior research on SCD combining deep learning. Most of the established literature is biased toward the classification and segmentation of RBC and sickle cells with machine learning algorithms. Petrović et al. [43] worked to select the best classification features and methods for the diagnosis of SCD. They used the erythrocytesIDB dataset [44] to perform segmentation and feature extraction on the microscopic images after pre-processing and applied machine learning algorithms for classification. The finest parameters for the classifiers were found through Randomized and Grid searches. From the microscopic images, they extracted the individual cells, reduced the noise of the images, used morphology opening to remove objects, and detected edges. They split the dataset into 70/30 train-test ratios and performed 10-fold cross-validation to avoid overfitting. Their research was centered on 3 main experiments. For the first experiment, Random Forest and Gradient boosting had the best results. However, the computational time for gradient boosting was much higher, because of which, decision tree was chosen as the next best classifier instead of gradient boosting after the random forest algorithm. Their second experiment concluded that using PCA and LDA did more harm than good to the output of the first experiment. Finally, the third experiment for validation in comparison with Ardakani et al. [27] confirmed that their two best classifiers with SDS score (GB and RF) and F-measure outperformed the previous state-of-the-art existing work. The limitation of their work is that they used only machine learning classifier centric implementation.

Khalaf et al. [45] used machine learning to classify medical data to guide disease-altering therapies for SCD patients. The focus is on using machine learning techniques to achieve high accuracy in medical data classification by improving the preprocessing of medical time-series data signals. The case study presented involves classifying the medication dosage for SCD patients. Different machine learning models were used and their accuracy and performance were weighted against each other. The outcome indicated that Random Forest Classifier produced the highest performance compared to other models such as recurrent networks and feedforward neural networks. They focused on a small number of architectures that couldn't fully depict the potential of their desired task using RNN, which constrained the size and scope of their research.

Alzubaidi et al. [46] put forth three deep-learning models in order to classify RBC into circular (normal), elongated (sickle cells), and other blood content. The authors collected 3 sets of data for their experimentation: Dataset 1 erythrocytesIDB, Dataset 2 [47,48] and Dataset 3 [49]. The authors have used data augmentation to address the limited number of training datasets. They utilized traditional and parallel CNN to implement their proposed approach and found the best one among the three models. They implemented three models in total. Model 1 is a 40-layer deep neural network with nine convolutional layers, batch normalization, and ReLU activations. It employs traditional convolutional layers followed by two blocks of parallel convolutional layers with concatenation. The architecture ends with fully connected layers, dropout, and softmax for classification. Model 2 is a 35-layer neural network with eight convolutional layers that is structurally similar to Model 1 but differs in some ways. It has two conventional convolutional layers rather than three, two fully connected layers with one dropout layer, and a different number of filters (depth). In addition, while Model 1 utilizes a 4 × 4 filter for average pooling, Model 2 uses a bigger 7 × 7 filter. Model 3 is a 29-layer neural network with six convolutional layers. With a few minor variations, its construction is very similar to that of Model 2. In contrast to Model 2, which uses three parallel convolutions in each of its blocks, Model 3 has a different number of filters (depth) and uses two parallel convolutions in each of its blocks. The results reveal that the proposed models outperform the latest erythrocyte classification methods with 99.54 % accuracy on the erythrocytesIDB dataset, 99.98 % when their model is combined with the SVM classifier, and 98.87 % on the collected dataset. They experimented with four scenarios: Scenario 1, Scenario 2, Scenario 3 and Scenario 4. Models are only trained in Scenario 1 using the original photos. The addition of augmented photos to the training data in Scenario 2 expands on Scenario 1. Scenario 3 involves transfer learning, where models are initially trained on Dataset 2 plus augmented images before fine-tuning on the original images of the datasets. Finally, Scenario 4 applies the transfer learning strategy from Scenario 3 to augmented photos, producing models that make use of the knowledge from Dataset 2 as well as a more varied set of training instances. The highest accuracy was achieved on Scenario 4.

Darrin et al. [50] worked with video input data for cell motion classification. They introduced a two-stage ML pipeline for the automatic classification of cell motions in videos for sickle cell disease patients. The dataset was collected from Ref. [51]. CNN and recurrent convolutional neural networks (RCNN) are combined and compared in the pipeline. It can eliminate 97 % of inconsistent cell sequences in the first stage, and in the second stage, it can classify red cell sequences that are highly and poorly deformable with 97 % accuracy and an F1-score of 0.94. They tested out 9-layer CNN, ResNet-18, ResNet-34, ResNet-50, and VGG-16 and selected the best performing ResNet-18 for the CNN-based approach and achieved improved results in the classification of blood cells. The second approach combined CNN and RNN forming a convolutional recurrent network (*C*-RNN). The results showed that the best architecture differs for the two tasks. The limitations of their work are the case of cell characterization, uniform down-sampling performed poorly, and the percentage of unreliable data was also high.

Patgiri and Ganguly [52] introduced a novel hybrid segmentation method for the segmentation of normal and sickle red blood cells (RBCs) in microscopic blood smears for classifying sickle cell anaemia with higher accuracy and efficiency and making the detection process automatic. The hybrid method combines fuzzy *C*-means and adaptive thresholding with four different thresholding techniques, feature extraction and image classification. The classifiers used for classification were Naive Bayes and K-nearest neighbour, which were evaluated on a dataset that contained 10 image samples. The outcome depicted that the integration of fuzzy *C*-means and NICK's thresholding with the K-nearest neighbour gave the optimal performance with a classification accuracy of 98.87 %. The proposed method proved to be efficient for classifying sickle cell anaemia. The limitation of their work is that they eliminated sickle cell blood images that overlapped with red blood cells while classifying.

Table 1 gives a summary of the literature survey, highlighting the classifiers that gave the best accuracy, the key contributions and the limitations thereof.

3. Materials and methods

This section provides a detail description of the dataset and the methodology. The dataset is collected from the erythrocytesIDB database which contains images of blood samples that were obtained from patients with sickle cell disease. The methodology comprises a series of investigations including the evaluation of a set of baseline transfer learning models, tuning hyperparameters, and ablation experiments with a set of classifiers.

3.1. Dataset

This image dataset used in this research was developed by the Special Hematology Department of the General Hospital, Cuba. Blood samples were collected from volunteer patients. The samples were then dispersed, dried, and then fixed with 100 % alcohol. Giemsa stain was then applied using a 2 % reagent to 1 ml of distilled water ratio. It was dried again for 15–20 min before being cleaned with distilled water. Afterwards, the image samples were captured with a Leika microscope and a Kodak EasyShare V803 camera equipped with a Kodak Retinar Aspheric All Glass Lens of 36–108 mm AF 3*X* Optical lens. The photos were examined by the special hematology department's first-grade clinical lab specialist. The samples were classified into three categories: circular, elongated, and cells with other types of deformations. Clusters where different cell types overlapped were also identified by specialists. The dataset is not publicly available.

3.2. Data sampling

The erythrocytesIDB dataset is organized into three parts: erythrocytesIDB1, erythrocytesIDB2, and erythrocytesIDB3. Firstly, erythrocytesIDB1 contains 196 full-field image samples as shown in Fig. 1. There are 629 individual cell images in the dataset. The

Table 1
Summary of literature survey on sickle cell disease.

References	Dataset	Best Classifiers	Highest Accuracy	Key Contributions	Limitations
[43]	erythrocytesIDB	DT RF	94.44 % 95.06 %	Chose the best ML classification method to detect SCD.	Limited Dataset and ML-centric implementation.
[45]	SCD dataset	LEVNN	99.1 %	The preprocessing of the medical time-varying data stream was improved.	Addressed a few architectures which limited their scale and scope.
[46]	Collected	Model 2 + SVM in Scenario 4	99.98 %	Classified 3 types of RBC using 3 deep-learning models.	Data imbalance. Data Quality Trade-off. Does not provide details about the sources or potential issues related to data quality, reliability and variability about Dataset 3 which is from different websites and internet searches.
[50]	Collected	NB KNN	98.212 % 98.87 %	Automatic classification of cell motions in videos.	Uniform downsampling was poor and unreliable.
[52]	Collected	CNN *C*-RNN	93.5 % 97 %	Hybrid segmentation made detection automatic.	Excluded sickle cell images overlapping with RBC.

Table 2
Dataset description for sickle cell disease.

Individual Cell	Original Dataset	Augmented Dataset
Circular	203	3334
Elongated	212	3334
Other	214	3334

individual cell images are divided into categories: circular images (203 images of circle-shaped cells like red blood cells), elongated images (212 sickle-shaped sample images) and others (214 images of cells with shapes other than circular or sickle), as illustrated in Fig. 2a, b and 2c.

Secondly, erythrocytesIDB2 and erythrocytesIDB3 contain 50 and 30 full-field source images respectively. Each of the source images has its own labelled image, mask image, and mask for the individual cells namely mask-circular, mask-elongated, and mask-other. The mask images are the binary images that keep only the targeted cell white and everything else black. The image set in erythrocytesIDB2 and erythrocytesIDB3 is shown in Fig. 3, where Fig. 3a represents the source images, 3 b presents the images after labelling, and 3c shows the images after masking. Similarly, the individual cells also had their masked images, that is, mask-circular, mask-elongated and mask-other, as illustrated in Fig. 3d, e and 3f correspondingly.

3.3. Data augmentation

Since only the shape of the cells is required for the experiment, the raw RGB images were also converted to grayscale keeping the ratio of the individual cells the same. Due to insufficient raw data, data augmentation is deemed necessary. Simple data augmentation technique is employed for creating more samples such as rotation (left and right), and zoom-in on the “circular”, “elongated” and “other” individual cells on both the RGB and Grayscale data. There is evidence of similar data augmentation, for example, Hosny et al. [53] used data augmentation extensively to overcome the challenge of data scarcity for classifying skin lesions. After the completion of the augmentation process, 3334 circular images, 3334 elongated images and 3334 other images were obtained, as shown in Table 2. That is the total of 10,002 RGB images and 10,002 Grayscale images used in this research for the multiclass classification.

3.4. Methodology

There have been many methods reported for diagnosing SCD in the literature. One of the clinical methods involves counting the sickle-like cells from a patient’s peripheral blood samples under a microscope for diagnosing SCD. Such a manual clinical method is time consuming and tedious. Morphological analysis of peripheral blood smears [54] is an important tool for diagnosis. In this research, a framework for the classification of cell morphology is proposed. A high-level view of the proposed methodology is illustrated in Fig. 4. Deep transfer learning models are proposed for the classification of the SCD images created in section 3.3.

In this research, five baseline transfer learning models $\{\text{ResNet50}, \text{AlexNet}, \text{VGG16}, \text{VGG19}, \text{MobileNet}\}$ and two machine learning classifiers $\{\text{SVM}, \text{RF}\}$ are chosen for experimentation. The baseline models have their own default hyperparameter set $H_p = \{batch\_size, learning\_rate, epoch, dropout\_rate, optimiser\}$. The default hyperparameters are tuned on the dataset $D = erythrocytesIDB$. The tuning of the default hyperparameters H_p of the corresponding baseline models is described by the function $\Gamma[.]$ defined by

$$TL\left(H_p^t\right) = \Gamma\left[TL\left(H_p\right), D\right] \tag{1}$$

where $TL \in \{\text{ResNet50}, \text{AlexNet}, \text{VGG16}, \text{VGG19}, \text{MobileNet}\}$, and H_p is the default hyperparameter set. The function $\Gamma[.]$ generates the five models $TL(H_p^t)$ with tuned hyperparameters on dataset D.

The five tuned baseline models $TL(H_p^t)$ will then undergo ablation analysis with the two classifiers $\{\text{SVM}, \text{RF}\}$. The ablation function $\Psi[.]$ is defined by

$$TL_{ab}\left(H_p^t, CF\right) = \Psi\left[TL_{ab}\left(H_p^t\right), CF, D\right] \tag{2}$$

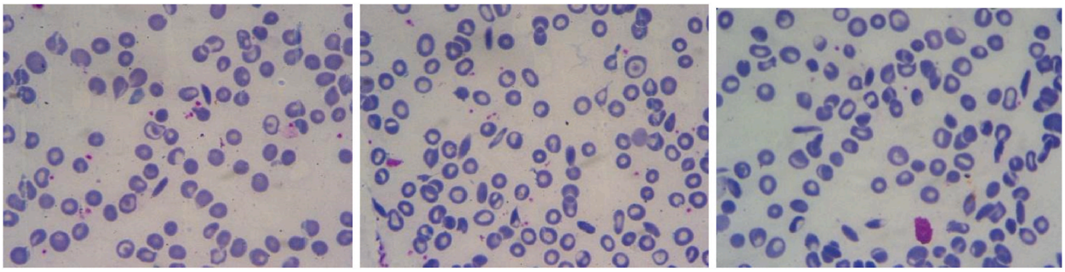

Fig. 1. Full-field source images.

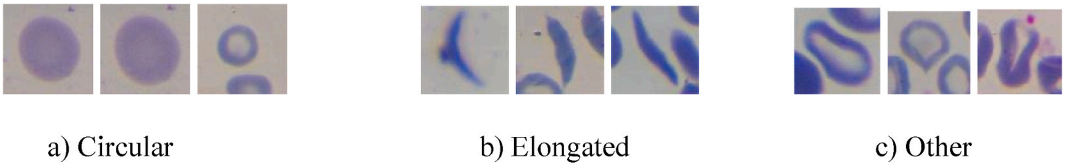

Fig. 2. Three categories of individual cell images.

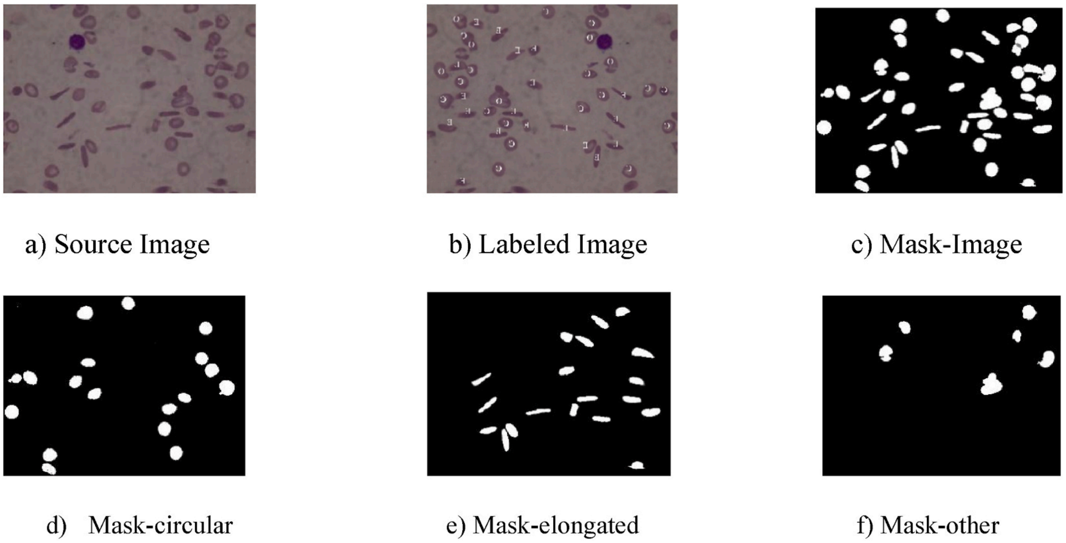

Fig. 3. Sample images in the dataset.

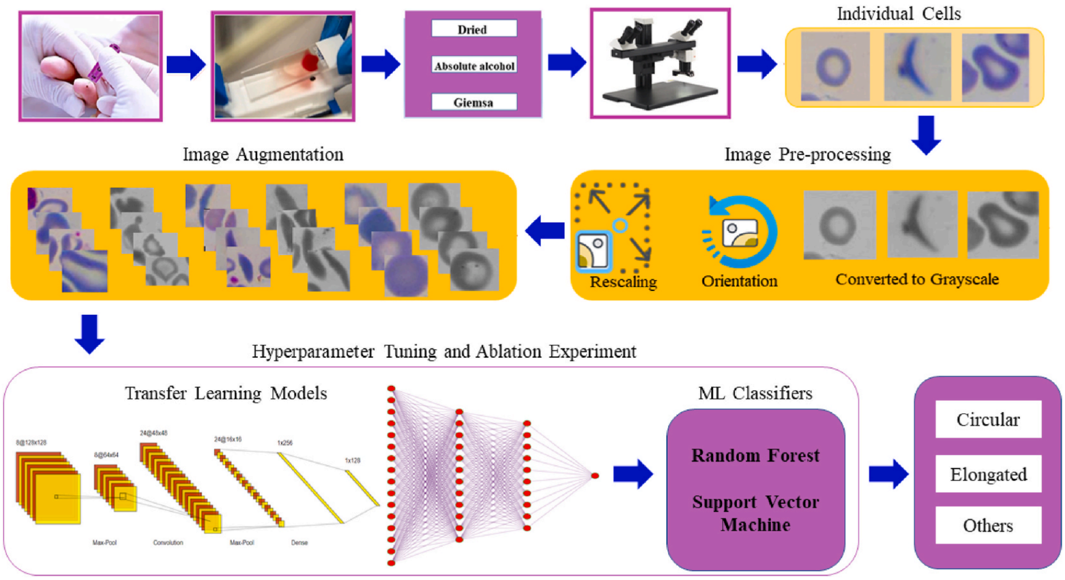

Fig. 4. A high-level overview of the proposed methodology.

where $CF \in \{\mathrm{SVM}, \mathrm{RF}\}$. The ablation function $\Psi[.]$ generates 10 models.

There will be 15 models (5 models with tuned hyperparameters, and 10 models generated by ablation). Statistical evaluations are carried out on these 15 models and the best performing model will be the outcome of the ablation experiment.

4. Experimentation and result analysis

In this research, experiments were conducted utilizing the Google Colab platform as our primary computational environment. Google Colab provided us with a cloud-based platform that facilitated the execution of complex computations, all without the necessity of high-powered local hardware resources. The selection of this environment was guided by several key considerations, including its accessibility, user-friendliness, and the presence of essential machine learning libraries. The predefined constraints in Colab included a 12.56 GB memory limit. Additionally, Google Colab imposed a 12-h runtime limit, a restriction carefully managed, especially when executing code segments with lengthier processing times. Despite these limitations, they were successfully accommodated within the experimental setup, ensuring the effective pursuit of research objectives.

To execute the experiment, at first, tuning of the deep transfer learning models is performed according to Eq. (1) and then the ablation experiment is carried out by integrating two classifiers according to Eq. (2).

4.1. Tuning of the deep transfer learning models

In deep transfer learning, pre-trained deep CNN is employed to leverage the knowledge already learned from a previous dataset by fine-tuning the model on new datasets. The fine-tuning is carried out by retraining some or all the layers. In general, the classification happens at the fully connected layer. It takes the outcome of the prior layers and passes it through a series of dense layers, which perform a linear transformation on the input data. The fully connected layer's final layer is a softmax layer, which gives the probability distribution over the different classes. The final layer of the network is the output layer, as it predicts the class of the input image based on the probabilities generated by the fully connected layer.

Five deep transfer learning models have been chosen for this study. These are ResNet50, AlexNet, VGG16, VGG19 and MobileNet. A comprehensive summary of the implemented models chosen for the classification is presented in the following sections.

ResNet-50: This deep learning model was developed by He et al. [55] for multiclass classification. It fine-tunes the pre-trained weights on the ImageNet dataset. It consists of a total of 50 layers, which include 1 fully connected layer and 49 convolutional layers, organized into residual blocks. The convolutional layers use kernel sizes of 3 × 3 and 1 × 1. The hyperparameters of ResNet50 include learning rate (0.001) and batch size (32). The architecture of ResNet50, with its residual blocks and skip connections, allows for effective learning of the features that distinguish between the different class labels in the image dataset. The final layers of the model, global average pooling and the fully connected layer allow for the prediction of the class label with high accuracy. ResNet50 can be expressed as shown in Eq. (3):

$$y = f(x) + x \tag{3}$$

where x is the input to a particular layer, and f (.) is a function that operates on the input and produces an output y. The output y is then passed as the input to the next layer. The residual connection is used to add the original input x to the output y, allowing the network to understand the dissimilarity between the input and output, rather than the whole mapping from input to output. The visualization for

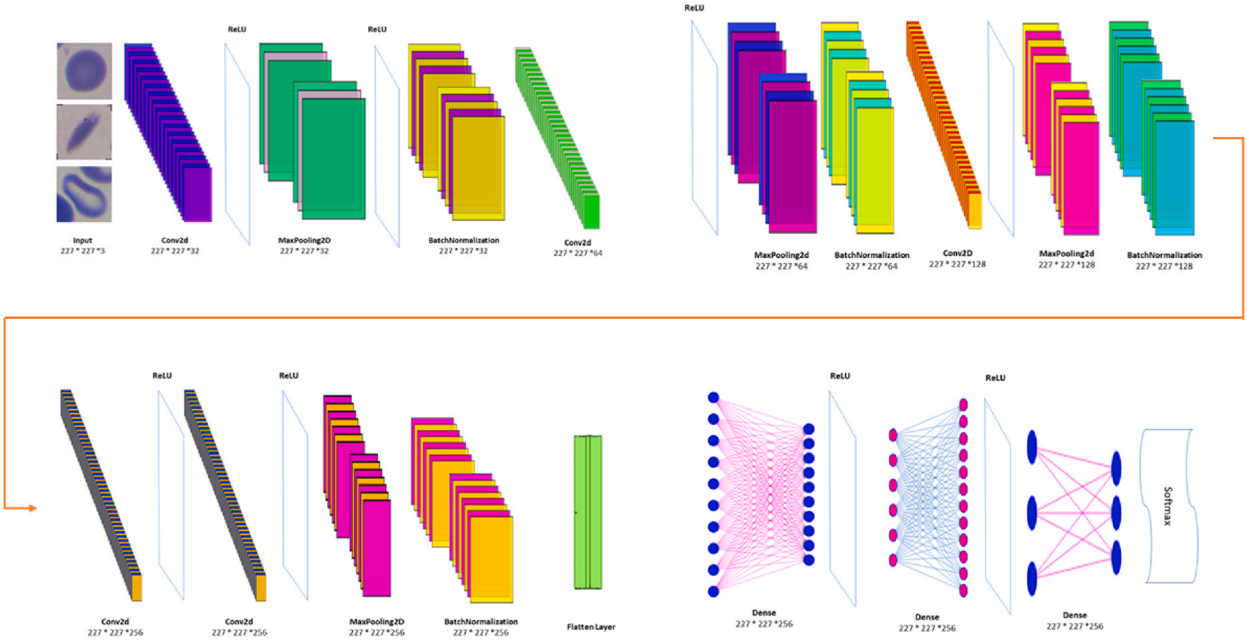

Fig. 5. Resnet-50 architecture.

the ResNet-50 architecture is illustrated in Fig. 5.

The accuracy was visualized using a confusion matrix, as shown in Fig. 6, created with the pd. crosstab function, which shows the number of accurate and inaccurate predictions for each class. The matrix was presented as a heatmap, using the Seaborn library's sn. heatmap function, where the colour intensity represents the frequency of the predictions. The accuracy was calculated as the ratio of correct predictions to the total number of predictions, and for the Resnet50 architecture, it was found to be 97.71 % as shown in the confusion matrix in Fig. 6. This high accuracy indicates that the model has efficiently mastered the patterns in the data resulting in a reliable model for making predictions. The classification report for the dataset with 10,002 images reveals that ResNet-50 exhibits high precision (99 %) for "circular" and "elongated" classes, indicating accurate positive predictions, while it struggles slightly with "other" (70 %). It achieves good recall (ability to correctly identify instances) for all classes, particularly excelling with "other" (98 %). The harmonic mean of precision and recall, the F1-score, is 0.86 for "circular", 0.91 for "elongated", and 0.82 for "other."

AlexNet: Krizhevsky et al. [36] developed AlexNet, a pre-trained model [56] which carries out multiclass classification tasks as shown in Fig. 7. It has eight layers: five convolutional, two fully connected, and one softmax layer. It uses kernel sizes of 11 × 11, 5 × 5, and 3 × 3 for the convolutional layers, with a stride of 4 pixels for the first layer, and 1 pixel for the rest. The hyperparameters used in AlexNet include learning rate (0.001) dropout rate (0.5), and batch size (3).

The network is trained on an image dataset that consists of multiple images, each labelled with one of the possible class labels. While training, the network adjusts its parameters to reduce the error between its predictions and the actual class labels in the data. After training, the network can classify new images into their respective classes. The image dataset must have a substantial number of images, each with a unique class label. AlexNet was even utilized for time-series classification [57]. Fig. 8 presents the AlexNet architecture.

The images are resized to (124 × 124). The AlexNet architecture is trained to classify input images into 3 classes. The model is then trained for 9 epochs with a batch size of 3. The validation accuracy was nearly 99.13 %. The early stopping callback was used to prevent overfitting, and it restored the weights of the best epoch based on validation loss. Fig. 9 depicts the accuracy of the model for training, testing and validation.

VGG16: The analysis utilized VGG16, a popular deep neural network architecture, to explore the parallels and contrasts in visual object representations between brains and deep networks [58]. This architecture [37] has a total of 16 layers:13 convolutional and 3 fully connected layers. It uses a fixed kernel size of 3 × 3 for all its convolutional layers. The hyperparameters used in VGG16 include learning rate (0.001) and batch size (32). VGG16 also uses dropout regularization with a rate of 0.5 in order to prevent overfitting during training. The convolutional layer's output can be written as Eq. (4):

$$y_{i,j,k} = \mathrm{ReLu}\Sigma_{r=1}^{R}\Sigma_{s=1}^{S}x_i + \left(r - \frac{R+1}{2}\right) * j + \left(s - \frac{S+1}{2}\right), l * w_{r,s,l,k} \tag{4}$$

The pooling layer's outcome can be defined as presented in Eq. (5):

$$y_{i,j,k} = max_{r=1}^{R} max_{s=1}^{S} x_{i*r,j*s,k} \tag{5}$$

In equations (7) and (8), x_i is the input feature map, $y_{i,j,k}$ is the output feature map, w is the convolution kernel, S and R are the spatial dimensions of the kernel, l is the input channel index, and k is the output channel index.

The output of a fully connected layer is given in Eq. (6):

$$y_k = \mathrm{ReLu}\left(\Sigma_{l=1}^{L} x_i * w_{l,k}\right) \tag{6}$$

where x_i and y_k are the input and output feature vectors respectively, $w_{l,k}$ is the weight matrix, and L is the number of inputs to the fully

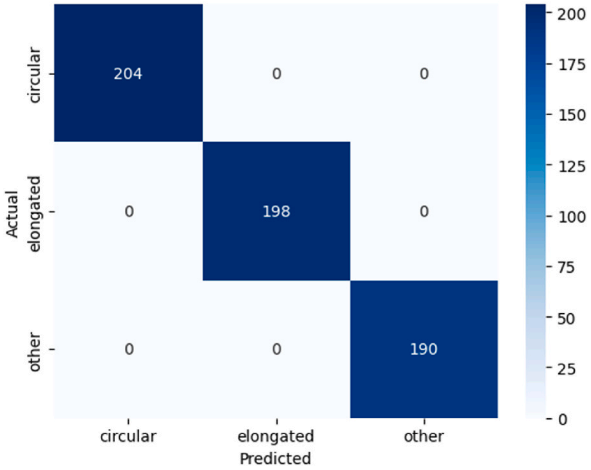

Fig. 6. Confusion Matrix for ResNet50 architecture.

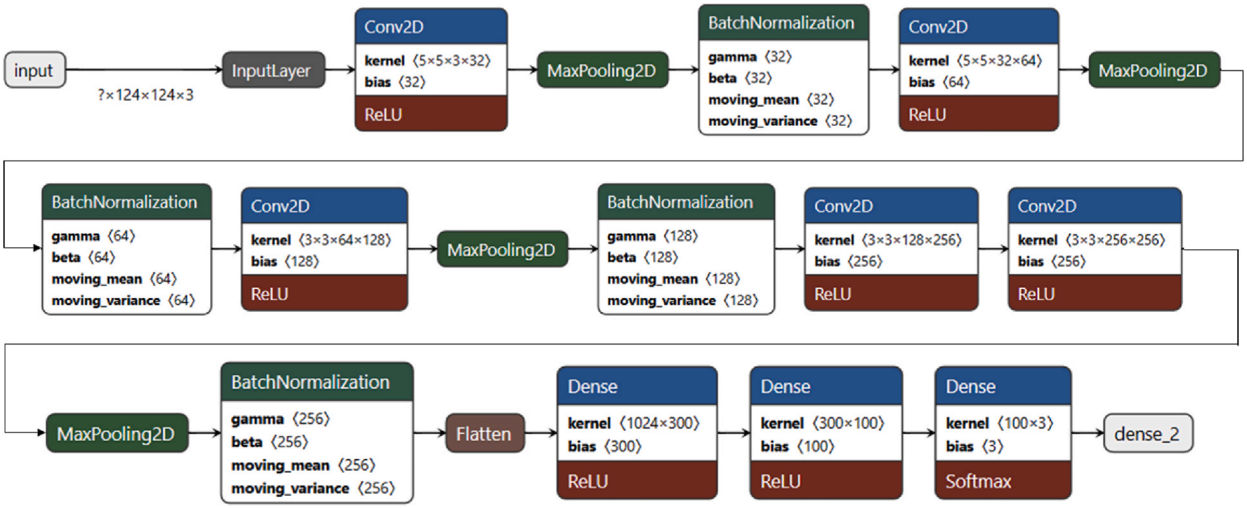

Fig. 7. AlexNet architecture.

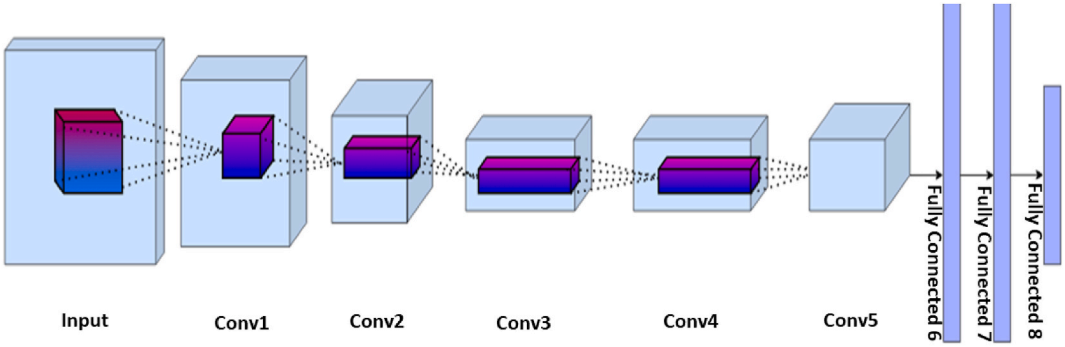

Fig. 8. Simple representation of the AlexNet Model.

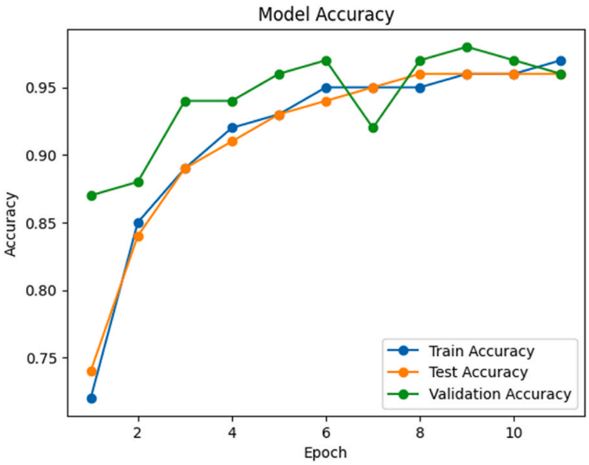

Fig. 9. Training, Testing, and Validation accuracy over time by AlexNet.

connected layer.

The VGG16 architecture has been fine-tuned for the task of classifying an image as either circular, elongated or others. The first step in building the model is to organize the data using the ImageDataGenerator class from the TensorFlow library. This class generates batches of image data for training and validation sets with batch size 32. The data is augmented using horizontal flipping, which can help to increase the robustness of the model. The images are then resized to a size of 224 × 224, which is the input size required by the VGG16 architecture. The VGG16 architecture is then loaded with the top layer removed, as it is being fine-tuned for a specific task. The outcome of the VGG16 architecture is then flattened and passed through a dense layer with a Softmax activation function. The model is then compiled using loss function (binary cross-entropy), optimizer (ADAM), and accuracy as the evaluation metric. The generic VGG architecture is demonstrated in Fig. 10.

The prediction part of the model is implemented as follows: a test image is loaded, preprocessed and passed to the model. The preprocessing involves resizing the image to the specified size (224 × 224). The results are then assessed and contrasted to determine the class of the image. If the first element of the result is greater than the second element, the image is classified as “circular”, if the first element is greater than the third element, the image is classified as “elongated”, and if neither condition is met, the image is classified as “others”. The final prediction is displayed as the output.

VGG19: The only difference between VGG16 and VGG19 is that the latter has a total of 19 layers: 16 convolutional and 3 fully connected layers. Fig. 11 portrays a general representation of the VGG model. VGG19 uses a kernel size of 3 × 3 for all of its convolutional layers, max-pooling layers with a 2 × 2 filter size and a stride of 2. The hyperparameters used in VGG19 include learning rate (0.001) and batch size (32). Ikechukwu et al. conducted experimentation with VGG19 instead of VGG16 with Resnet50 for segmentation and classification and to observe the difference in results when these pre-trained models are used and when they are trained from scratch [59]. VGG16 has 16 layers, whereas, VGG19 has 19 convolutional layers. The model can be trained on an image dataset, where an individual image is assigned a class label and adjusts its parameters during the training process to underrate the error between its predictions and the actual class labels in the data. Once trained, VGG19 can be used to classify new and previously unrecognized images into their respective classes with high accuracy.

The pre-trained model is taken as an input, and two dense (fully connected) layers are added on top of it. The first dense layer contains 128 units and implements the ReLu activation function, while the second layer has 3 units and employs the Softmax activation function to generate a probability distribution across the 3 class levels in the output. It is then compiled and trained for 20 epochs with early stopping applied, which stops training when there is no improvement for 5 consecutive epochs. The training and validation progress is reported for each epoch, showing the values of the loss and accuracy metrics. The model achieves an accuracy of 99 % on the train set and 98.58 % on the test set. A confusion matrix is generated, as shown in Fig. 11, for evaluating the effectiveness of the model. The confusion matrix and classification report indicated that the model performed well in recognizing all three classes. The classification report for the VGG-19 model displays remarkable performance in classifying objects into the three categories of “circular,” “elongated,” and “others.” With 99 % for “circular,” 99 % for “elongated,” and 95 % for “others,” the model gets astonishingly high precision scores across all classes, indicating a very low percentage of false positive predictions. In addition, the recall scores are also quite good, 99 % for “circular,” 96 % for “elongated,” and 98 % for “others”. The F1-scores are consistently high having values of 99 % for “circular,” 98 % for “elongated,” and 96 % for “others".

MobileNet: MobileNet [60] architecture is designed for efficient computation which reduces computational cost and memory usage. It consists of 18 layers: 13 convolutional layers and 5 depthwise separable convolutional layers. It uses 3 × 3 convolutional filters for most of its layers, which reduces parameters and makes it more efficient for deployment. The hyperparameters include a learning rate, weight decay, and batch size, which can be adjusted while training, to optimize the performance. Fig. 12 illustrates the MobileNet architecture.

The convolution can be expressed as a sequence of two operations: depthwise convolution and pointwise convolution. Among these

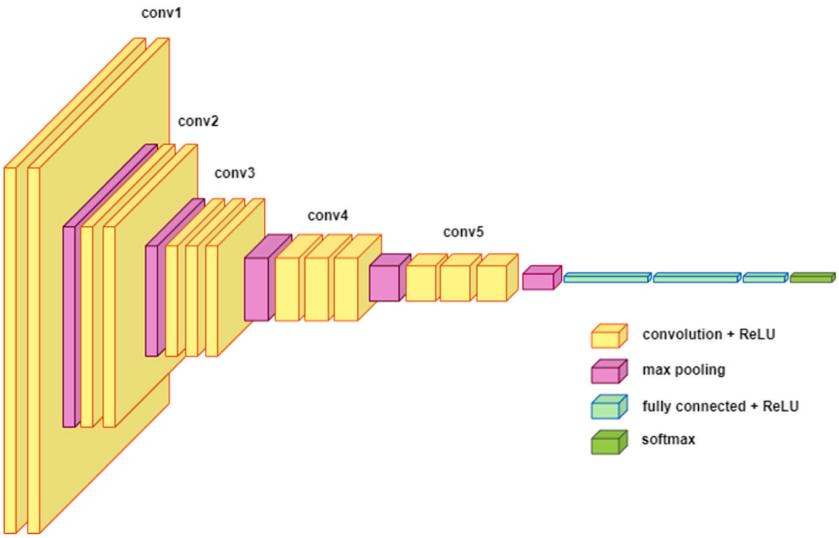

Fig. 10. VGG architecture.

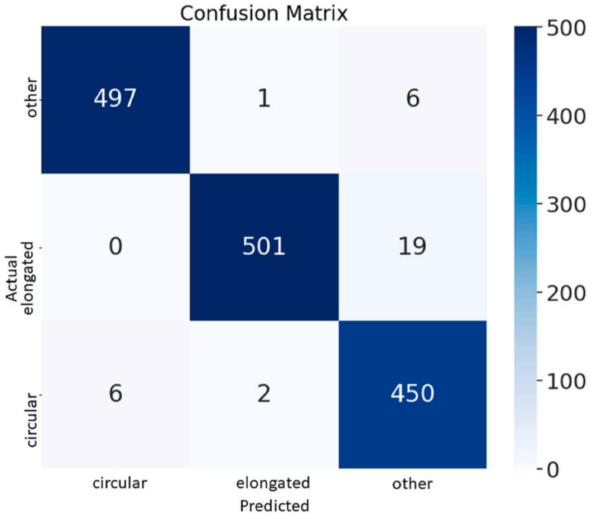

Fig. 11. Confusion Matrix for VGG-19 architecture.

two, the depthwise convolution can be written as Eq. (7):

$$y_{i,j,k} = \Sigma_{r=1}^{R}\Sigma_{s=1}^{S} x_i + \left(r - \frac{R+1}{2}\right) * s, j + \left(s - \frac{S+1}{2}\right) * t * w_{r,s,k} \tag{7}$$

where x_i is the input feature map, $y_{i,j,k}$ is the output feature map, $w_{r,s,k}$ is the kernel, and R and S are the spatial dimensions of the kernel.

The pointwise convolution is defined as Eq. (8):

$$z_{i,j,k} = \Sigma_{k=1}^{K} y_{i,j,k} * v_{k,l} \tag{8}$$

where z is the output feature map, v is the pointwise convolution kernel, K is the number of input channels, and l is the number of output channels.

The MobileNet model is trained for image classification through transfer learning with MobileNetV2 as the base. Final layers of this model are retrained for the specific image classification task and the data is preprocessed with the preprocess_input function. Data split

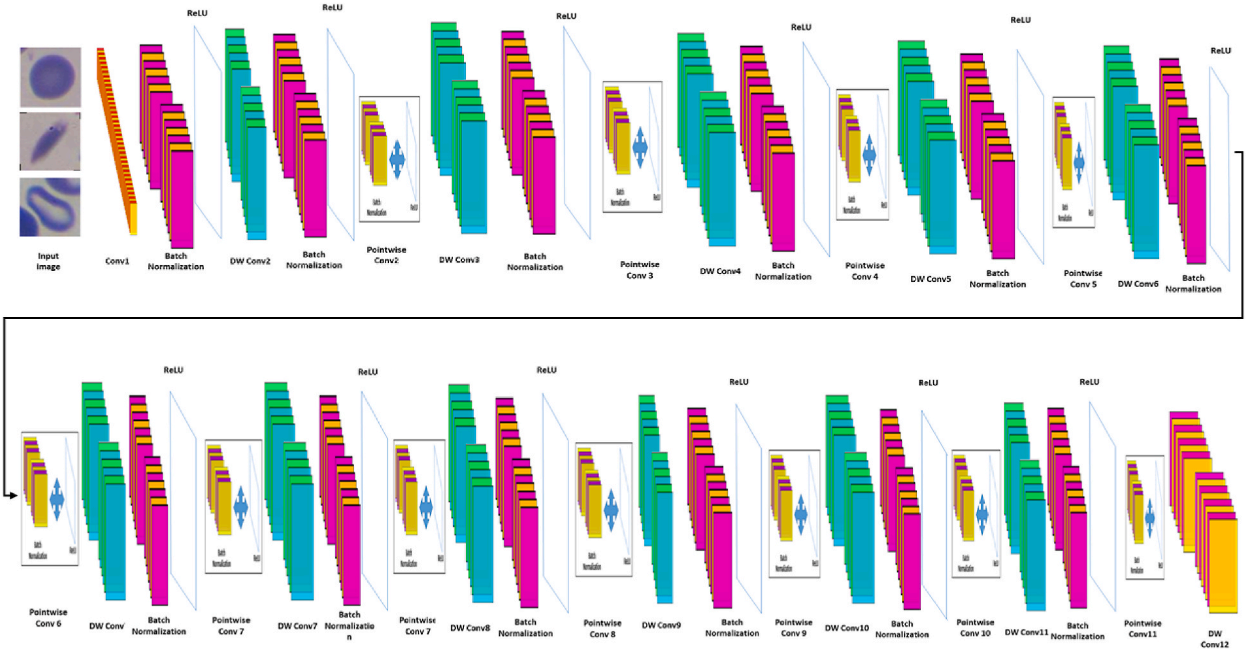

Fig. 12. MobileNet architecture.

ratio here is the same. The model uses an Adam optimizer, categorical cross-entropy loss function, and accuracy metric. It utilizes an early stopping function that stops executing the epochs once the validation accuracy stops improving. It displayed a test accuracy of 96.99 % and a validation accuracy of 94 %. Fig. 13 shows the training, testing and validation accuracy of the MobileNet architecture.

The accuracy of the architectures have already been discussed earlier, and it was shown that ResNet-50 had the best accuracy. A summary of the performance of the hyperparameter-tuned models is given in Fig. 14.

In the experimentation, five deep learning models were evaluated with varying batch sizes, learning rates, and epochs. The ResNet-50 model, utilizing a batch size of 32, trained for 10 epochs, coupled with the ADAM optimizer and a dropout regularization rate of 0.5, achieved a test accuracy of 97.71 % and a validation accuracy of 96.95 %. Moving on to VGG-16 with a batch size of 4, it attained a test accuracy of 98.96 % and a validation accuracy of 98.27 %. VGG-19, with a batch size of 5, reached a test accuracy of 99 % and a validation accuracy of 98.58 %. Lastly, the MobileNet model was trained with a batch size of 9, resulting in a test accuracy of 96.99 % and a validation accuracy of 94 %. For the AlexNet model, 19 epochs were used keeping all the other hyperparameters the same. That resulted in a very poor test and validation accuracy of 34.36 % and 34 % respectively. This experiment was taken further to figure out which combination gives the best output for AlexNet. When a batch size of 3, a learning rate of 0.001, and 17 epochs were employed, it yielded a test accuracy of 99.13 % and a validation accuracy of 98.34 %. Table 3 shows the performance of all the models that were investigated.

4.2. Ablation experiment

This section explores the ablation experiment that was designed to see the effectiveness of the combination of the tuned deep transfer learning models (carried out in section 4.1) with machine learning classifiers [61,62]. As discussed in section 2, Petrovich et al. [43] used this same erythrocytesIDB [44] dataset to choose the best classification method to detect SCD using machine learning. They executed three experiments involving seven machine learning models. Among those, random forest (RF) achieved the best accuracy (95.06 %), and the lowest (87.24 %) was acquired by the SVM algorithm. Therefore, the choice of machine learning classifier for this research is random forest (RF) [63] and SVM algorithm [64]. A total of ten ablation experiments are carried out combining the five tuned deep transfer learning models with the two classifiers. Fig. 15 presents a visual representation of the ablation experiment.

Hyperparameters are one of the most important factors of the experiment as they significantly affect the model's performance. As shown in Fig. 15, different hyperparameters for the deep transfer learning models and for the ML classifiers are employed. A pivotal contribution of this research is the execution of ablation experiments, a methodology previously unexplored within the domain of SCD classification. Departing from conventional fully connected networks, the final layers of all considered deep learning models were substituted with two distinct classifiers: (1) RF and (2) SVM. This approach culminated in the attainment of the highest classification accuracy observed in this study attained by the combination of MobileNet model and SVM as shown in Table 4, thereby establishing ablation experiments as a promising avenue for further exploration not just in the realm of SCD classification, but in other fields as well.

To ensure a rigorous and equitable comparison between the ablation experiments and the original deep learning models, meticulous attention was given to hyperparameter tuning. Specifically, all hyperparameters remained consistent across the ablation experiments, including those of the RF and SVM classifiers. Consequently, the comprehensive analysis encompassed not only the performance metrics of each ablation experiment but also the identification of the optimal hyperparameters, which were pivotal in achieving the highest accuracy across the various models. These findings collectively contribute to the robustness and validity of the

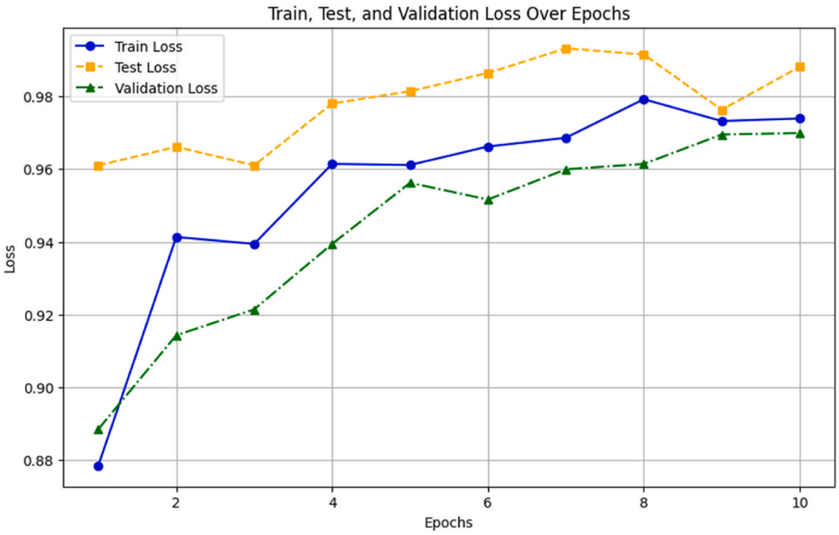

Fig. 13. Train, test and validation accuracy of MobileNet.

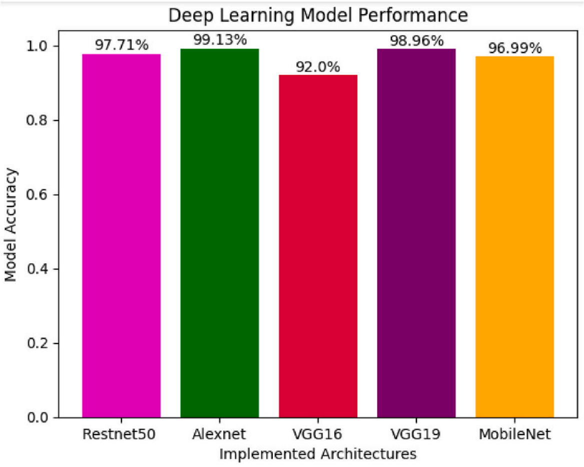

Fig. 14. Performance of the hyperparameter-tuned models.

Table 3
Performance comparison of the hyperparameter tuned models.

	Batch Size	Learning Rate	Epoch	Dropout Regularization	Optimizer	Test Accuracy	Val Accuracy
Resnet-50	32	0.001	10	0.5	ADAM	97.71 %	96.95 %
VGG-16			4			98.96	98.27 %
VGG-19			5			99 %	98.58 %
MobileNetV2			9			96.99 %	94 %
AlexNet			17			34.36 %	34 %
AlexNet	3	0.001	17			99.13 %	98.34 %

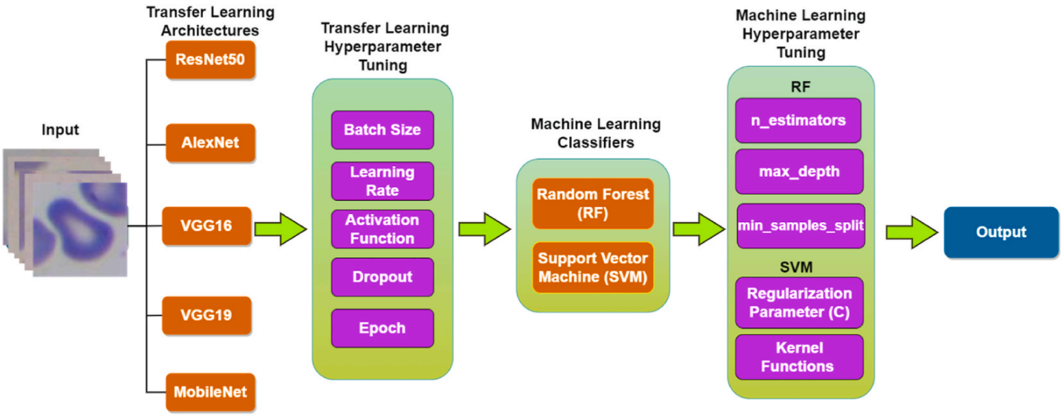
Transfer Learning Architectures
ResNet50
AlexNet
VGG16
VGG19
MobileNet
Input
Transfer Learning Hyperparameter Tuning
Batch Size
Learning Rate
Activation Function
Dropout
Epoch
Machine Learning Classifiers
Random Forest (RF)
Support Vector Machine (SVM)
Machine Learning Hyperparameter Tuning
RF
n_estimators
max_depth
min_samples_split
SVM
Regularization Parameter (C)
Kernel Functions
Output

Fig. 15. Ablation experiment design.

ablation experiments as a promising methodology in the context of SCD classification, underscoring the potential for further advancements in this area of research.

Table 4 provides a summary of the models and their performance on a classification task. Batch Size, Learning Rate, Dropout Regularization, and Optimizer are the hyperparameters for the transfer learning models. As for RF, the hyperparameters are the number of trees in the forest (n_estimators), maximum depth of the trees (max_depth) and minimum number of samples required to split an internal node (min_samples_split). The 'n_estimators' hyperparameter determines the number of decision trees in the forest. Increasing the number of estimators generally leads to a more robust and accurate model, but it also increases computational cost. In the table, values of 50, 100, and 200 are considered. On the other hand, 'max_depth' defines the maximum depth of each decision tree in the forest. "None" indicates that there is no maximum depth limit, allowing trees to grow until they contain very few samples. In the table, values of None, 10, and 20 are considered. The minimum number of samples required to split an internal node is specified by 'min_samples_split'. A smaller value can result in more splits and finer-grained trees, while a larger value can lead to simpler trees with fewer splits. In the table, values of 2, 5, and 10 are considered.

Table 4
Performance comparison of the hyperparameter tuned ablation models.

	Batch Size	Learning Rate	Dropout Regulari-zation	Optimizer	Random Forest						Val Accuracy	Test Accuracy
					Hyperparameters			Best Hyperparameter				
					n_ estimators	max_ depth	min_ samples_ split	n_ estimators	max_ depth	min_ samples_ split		
VGG16 + RF	32	0.001	0.5	ADAM	[50, 100, 200]	[None, 10, 20]	[2,5,10]	200	20	2	97.5 %	96.02 %
VGG19+ RF								200	None	2	96.89 %	95.82 %
Mobilenet + RF								200	None	5	97.29 %	96.63 %
Resnet50 + RF								200	20	2	97.36 %	97.03 %
Alexnet + RF								50	9	5	63.06 %	61.54 %
					Support Vector Machine						Val Accuracy	Test Accuracy
					Hyperparameters			Best Hyperparameter				
					C	Kernel		C	Kernel			
VGG16 + SVM					0.1, 1, 10	Linear, RBF		0.1	Linear		98.71 %	97.84 %
VGG19+ SVM								10	RBF		98.44 %	97.23 %
Mobilenet + SVM								**10**	**RBF**		**99.53 %**	**98.92 %**
Resnet50 + SVM								10	RBF		99.32 %	98.79 %
Alexnet + SVM								10	RBF		47.09 %	47.44 %

SVM was tested with two main hyperparameters, namely the Regularization Parameter (also known as the C parameter) and Kernels. The regularization parameter C trades off correct classification of training examples against maximizing the margin of the decision boundary. In the table, values of 0.1, 1, and 10 are considered. Conversely, the kernel function defines the type of decision boundary used by the SVM. Linear Kernel and RBF (Radial Basis Function) Kernel are used here. Linear Kernel creates a linear decision boundary, suitable for linearly separable data. RBF Kernel is a more flexible kernel that can capture non-linear relationships in the data by mapping it into a higher-dimensional space. The choice of the kernel function and the value of C significantly impact the SVM's ability to classify data. The SVM seeks to find the hyperplane (or decision boundary) that maximizes the margin between classes while minimizing misclassifications. Hyperparameter tuning for SVM involves finding the best combination of C and kernel type to achieve the highest classification accuracy on the validation or test dataset. GridSearchCV is used to search for these optimal hyperparameters systematically.

The purpose of tuning these hyperparameters is to find the combination that yields the best balance between model complexity and generalization performance. GridSearchCV systematically explores different combinations of these hyperparameters to find the best set that optimizes the model's performance. However, in order to keep the comparison fair, all the models were tested with the same set of hyperparameters, which are, batch size = 32, learning rate = 0.001, dropout and regularization = 0.5, and optimizer = ADAM. The RF classifier utilized three hyperparameters. The values for 'n_estimators' were [50, 100, 200], 'max_depth' were [None, 10, 20], and finally, 'min_samples_split' was [2,5,10]. When the SVM was incorporated, the C values were chosen to be 0.1, 1 and 10, and the kernels were Linear and RBF. Each model performed best under different combinations.

MobileNet combined with SVM achieved the highest performance, with a validation accuracy of 99.53 % and a test accuracy of 98.92 %. The best SVM hyperparameters used a regularization parameter (C) of 10 and an RBF kernel. MobileNet combined with RF achieved a validation accuracy of 97.29 % and a test accuracy of 96.63 %. The best RF hyperparameters were 200 estimators, no maximum depth limit, and a minimum of 5 samples required to split a node.

ResNet50 features coupled with SVM had the second-best performance with a validation accuracy of 99.32 % and a test accuracy of 98.79 %. The best SVM hyperparameters used a regularization parameter (C) of 10 and an RBF kernel. ResNet50 combined with RF achieved a validation accuracy of 97.36 % and a test accuracy of 97.03 %. The best RF hyperparameters were 200 estimators, a maximum depth limit of 20, and a minimum of 2 samples required to split a node.

VGG16 and SVM resulted in a validation accuracy of 98.71 % and a test accuracy of 97.84 %. The best SVM hyperparameters were a regularization parameter (C) of 0.1 and a linear kernel. VGG16 and RF achieved a validation accuracy of 97.5 % and a test accuracy of 96.02 %. The best RF hyperparameters were 200 estimators, a maximum depth limit of 20, and a minimum of 2 samples required to split a node.

VGG19 and SVM achieved a validation accuracy of 98.44 % and a test accuracy of 97.23 %. The best SVM hyperparameters used a regularization parameter (C) of 10 and an RBF kernel. VGG19 + RF achieved a validation accuracy of 96.89 % and a test accuracy of 95.82 %. The best RF hyperparameters were 200 estimators, no maximum depth limit, and a minimum of 2 samples required to split a node.

AlexNet features with SVM resulted in lower accuracy, with a validation accuracy of 47.09 % and a test accuracy of 47.44 %. The best SVM hyperparameters used a regularization parameter (C) of 10 and an RBF kernel. Again, with RF, it achieved a validation accuracy of 63.06 % and a test accuracy of 61.54 %. The best RF hyperparameters were 50 estimators, a maximum depth of 9, and a minimum of 5 samples required to split a node.

In this research, statistical tests were conducted to assess the significance of the differences among the compared models. The results of these tests are shown in Table 5.

Based on these results on the fine-tuned transfer learning models, it is observed that the Alexnet model exhibits the most significant difference compared to the other models, as indicated by its lowest *P*-Value of 0.0374. This suggests that the performance of the Alexnet model differs significantly from the rest of the models, supporting the validity of the proposed method when compared to the other models.

5. Comparison and discussion

As shown in Table 6, Petrovic et al. whose research was done using the same dataset as this study, achieved 95.06 % accuracy when they implemented the erythrocytesIDB dataset.

Alzubaidi et al. [46] reached their best accuracy with their Model1 (CNN) + SVM combination. However, it is important to note that they didn't use one single dataset, rather it was a combination of multiple sets. Their Dataset 1 is the erythrocytesIDB dataset

Table 5
Statistical test analysis of the models.

Model	Test	
	t-Statistic	*P*-Value
Resnet-50	**−2.2565**	**0.0870**
Alexnet	−1.3999	0.1950
Mobilenet	−2.6418	0.0229
VGG-16	−0.5762	0.6048
VGG-19	−0.0024	0.9981

Table 6
Comparison with state-of-the-art research.

References	Dataset	Best Classifiers	Accuracy	Comments
[43]	erythrocytesIDB (3 classes)	RF	95.06 %	This approach extracts the features and then classifies them using random forest classifier.
[46]	Dataset1 (erythrocytesIDB) Dataset2 (a composite dataset) Dataset3 (custom red blood smear dataset)	Model 2 + SVM (Scenario 4)	99.98 %	This accuracy was achieved for Scenario 4. For this, the authors had to incorporate their Dataset 2, which originally contained 367 pictures of white blood cells and other blood cells, 150 blood samples and 260 images of ALL-IDB2 dataset. They used both original and augmented images, before finally employing it with the Dataset 1 (erythrocytesIDB), Model and SVM classifier.
[52]	Collected	CNN C-RNN	93.5 % 97 %	A different dataset used in this research.
Proposed method	erythrocytesIDB (3 classes)	AlexNet ResNet50 + SVM MobileNet + SVM ResNet50	99.13 % 99.32 % 99.53 % 100.00 %	MobileNet + SVM provided the best performance using the same dataset [43]. When trained with only 5000 samples.

containing 196 full-field images, 202 circular, 211 elongated, and 213 other types of cell images. Their Dataset 2 comprised 3 more sub-datasets. Each image in the first sub-dataset, which has 367 images totaling 640 × 480 pixels, is used to categorize the four different types of white blood cells (neutrophils, eosinophils, lymphocytes, and monocytes). 150 blood pictures from the Wadsworth center (400,298 pixels) make up the second sub-dataset. The third sub-dataset, called ALL-IDB2, has 260 images of lymphocytes for the diagnosis of acute lymphoblastic leukemia, divided into mature and immature cells (257,257 pixels in size per image). Dataset 3 consists of 200 red blood smear samples collected from various websites and internet searches. Using Dataset 1 as their target dataset, Dataset 2 to employ transfer learning and Dataset 3 to demonstrate the robustness of proposed models, they achieved 99.98 % accuracy with Model 2 and Scenario 4. In Scenario 4, the authors combine the strategies of transfer learning from Dataset 2 and data augmentation. This approach begins with the transfer learning step where the models are initially trained on Dataset 2, which has been augmented with additional images to boost the representation of various blood cell types. By leveraging the knowledge gained from this initial phase, the models are already equipped with a foundation in recognizing different blood components. Afterwards, in addition to the knowledge transferred from Dataset 2, the authors introduce augmented images into the training process. These augmented images include variations of the original images, obtained through techniques like rotation, flipping, and brightness adjustments. By combining the transferred knowledge with augmented data, incorporating both Dataset 2, they achieved such result. In the course of model training, the ResNet-50 model achieved a remarkable 100 % accuracy when trained on a dataset consisting of 5000 images after augmentation. However, for the purpose of a comprehensive investigation, this research extended the dataset size to encompass 10,002 images. It is essential to emphasize that the dataset employed in this research differs from that of used by Alzubaidi et al. underscoring the significance of this distinction in dataset composition and size. Furthermore, it is worth noting that this research yielded highly competitive results, with the MobileNet + SVM combination achieving an accuracy of 99.53 %, and the ResNet-50 + SVM combination achieving an accuracy of 99.32 % utilizing the augmented erythrocytesIDB dataset of 10,002 images.

In order to find the optimal solution and best result, hyperparameters were tuned. The parameters were set after running multiple experiments and observing their behaviour. All the models had the same batch size, learning rate, optimizer, and dropout regularization. Only the epochs were different for the models. Keeping the hyperparameters same for the models gave insight into how well each model performs under a given condition. The batch size was set to 32, learning rate was 0.001, ADAM optimizer was used and

Table 7
Time complexity.

	No of Epoch	Total Samples	Batch Size	No. of Layers	Input Size	Operations per Batch (approximate)	Batches per Epoch	Operations per Epoch (approximate)	Total Operations (approximate)
Alexnet	17	6900	3	8	224 × 224	1984	2300	4,571,200	77,609,600
MobileNetV2	9	6900	32	53	224 × 224	216	11,872	2,564,352	23,079,168
VGG16	4	6900	32	16	224 × 224	3584	216	772,864	3,091,456
VGG19	5	6900	32	19	224 × 224	4256	216	919,296	4,596,480
ResNet50	10	6900	32	50	224 × 224	32,768	216	7,083,648	7,083,680

An in-depth analysis of the computational complexity of the models is provided in Table 7. Understanding these complexities is vital for optimizing training and inference processes and making informed model selection decisions in diverse computational environments. In order to calculate the complexity, the important factors to take into consideration are the following: Number of Epochs, Total Samples, Batch Size, Number of Layers, Input Size, Operations per Batch (approximate), Batches per Epoch, Operations per Epoch (approximate) and Total Operations (approximate). Eq. (9), (10), (11), and (12) represent Operations per Batch, Batches per Epoch, Operations per Epoch and Total Operations sequentially.

dropout regularization was 0.5. .

$$\text{Operations per Batch (approximate)}: Ob = Ol \times B \tag{9}$$

Here, *Ob* represents the operations per batch, *Ol* denotes the operations per layer, and *B* signifies the batch size.

$$\text{Batches per Epoch}: Be = \frac{S}{B} \tag{10}$$

Here, Be signifies the batches per epoch S is the total number of samples, and B represents the batch size.

$$\text{Operations per Epoch(approximate)} : Oe = Ob \times Be \tag{11}$$

Here, *Oe* represents the operations per epoch, *Ob* is the operations per batch, and *Be* stands for the batches per epoch.

$$\text{Total Operations(approximate)} : Ot = Oe \times N \tag{12}$$

Finally, *Ot* indicates the total operations, *Oe* is the operations per epoch, and N is the number of epochs.

AlexNet, recognized as a groundbreaking convolutional neural network architecture, encompasses 8 layers and operates with an input size of 224 × 224 pixels. When subjected to training for 17 epochs utilizing a dataset comprising 6900 samples and employing a batch size of 3, the computational complexity is as follows: approximately 1984 operations per batch, 2300 batches per epoch, resulting in an approximate 4,571,200 operations per epoch, totaling approximately 77, 609, 600 operations.

MobileNetV2 encompasses 53 layers and operates with an input size of 224 × 224 pixels. When trained over 9 epochs using a dataset consisting of 6900 samples with a batch size of 32, the computational complexity is as follows: approximately 11,872 operations per batch, 216 batches per epoch, resulting in an approximate 2,564,352 operations per epoch, totaling approximately 23, 079, 168 operations.

VGG16, a deep convolutional neural network comprising 16 layers and designed with an input size of 224 × 224 pixels, underwent training for 4 epochs using a dataset encompassing 6900 samples with a batch size of 32. The computational complexity is as follows: approximately 3584 operations per batch, 216 batches per epoch, resulting in approximately 772,864 operations per epoch, accumulating a total approximate computational load of 3,091,456 operations.

VGG19, an extension of VGG16 and featuring 19 layers, maintains the same input size of 224 × 224 pixels. During training for 5 epochs, utilizing a dataset comprising 6900 samples with a batch size of 32, the computational complexity is as follows: approximately 4256 operations per batch, 216 batches per epoch, resulting in an approximate 919,296 operations per epoch, accumulating to total approximate computational load of 4,596,480 operations.

ResNet50, a variant of the ResNet architecture encompassing 50 layers and designed with an input size of 224 × 224 pixels, underwent extensive training for 10 epochs using a dataset comprising 6900 samples and employing a batch size of 32. The computational complexity is as follows: approximately 32,768 operations per batch, 216 batches per epoch, resulting in approximately 7,083,648 operations per epoch, accumulating a total approximate computational load of 70, 836, 480 operations.

6. Conclusion and future works

This study presents an approach for classifying sickle cell disease, conducting multiclass classification to distinguish circular (red blood cell), elongated (sickle-shaped cells), and other (non-circular and non-elongated cell shapes). Model effectiveness is assessed using a confusion matrix, including accuracy, precision, recall, and F1-measure values. Validation loss and accuracy graphs indicate the model's performance. The experimental results demonstrate superior performance across all models, with ResNet-50 achieving exceptional precision, recall, and F1-score of 100 % for circular, elongated, and other cell shapes. Similarly, the AlexNet model showed a balanced performance with 98 % precision and 99 % recall for circular and elongated shapes, while MobileNet also exhibited competitive metrics. The experimental results in this study demonstrate significant performance enhancements across multiple models. Notably, the MobileNetV2 model achieved an impressive accuracy rate of 96.99 %. Subsequently, the integration of the ablation experiment, which incorporated RF with the MobileNetV2 model, yielded an even higher accuracy of 97.29 %, signifying a substantial increase of 0.3 %. Furthermore, when the MobileNetV2 model was combined with an SVM, the accuracy soared to an outstanding 99.53 %, showcasing a remarkable improvement of 2.54 % compared to the base model. The ResNet-50 architecture also demonstrated exceptional performance, achieving an accuracy rate of 97.71 %. The integration of SVM with the ResNet-50 model further elevated the accuracy to 99.32 %, reflecting a noteworthy improvement of 1.61 %.

Notably, the methods and models employed in this research, such as the deep transfer learning models, tuning of the hyperparameters, ablation experiment, augmenting the dataset to make it more comprehensive and statistical tests outperformed other novel approaches, which mainly focused on machine learning and related techniques for sickle cell disease.

Data availability statement

The data associated with this study has not been deposited into a publicly available repository. The original dataset was obtained from the UGiVIA research group at the University of the Balearic Islands, Palma, Spain, following a formal request as it was not publicly accessible at the time of this research. Subsequently, the dataset has been augmented and transformed, creating our own derivative dataset for this study. While the data is not currently publicly available, it will be made available upon request.

CRediT authorship contribution statement

Sanjeda Sara Jennifer: Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization, Data curation, Writing - original draft, Writing - review & editing. **Mahbub Hasan Shamim:** Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization, Writing - original draft, Writing - review & editing. **Ahmed Wasif Reza:** Supervision, Resources, Project administration, Funding acquisition, Conceptualization, Writing - original draft, Writing - review & editing. **Nazmul Siddique:** Supervision, Conceptualization, Resources, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The dataset is collected from the UGiVIA research group, University of the Balearic Islands, Palma, Spain.

References

[1] H. Frangoul, D. Altshuler, M.D. Cappellini, Y.-S. Chen, J. Domm, B.K. Eustace, J. Foell, J. de la Fuente, S. Grupp, R. Handgretinger, T.W. Ho, A. Kattamis, A. Kernytsky, J. Lekstrom-Himes, A.M. Li, F. Locatelli, M.Y. Mapara, M. de Montalembert, D. Rondelli, A. Sharma, S. Sheth, S. Soni, M.H. Steinberg, D. Wall, A. Yen, S. Corbacioglu, CRISPR-Cas9 gene editing for sickle cell disease and β-thalassemia, N. Engl. J. Med. 384 (2021) 252–260, https://doi.org/10.1056/nejmoa2031054.

[2] G.J. Kato, F.B. Piel, C.D. Reid, M.H. Gaston, K. Ohene-Frempong, L. Krishnamurti, W.R. Smith, J.A. Panepinto, D.J. Weatherall, F.F. Costa, E.P. Vichinsky, Sickle cell disease, Nat. Rev. Dis. Prim. 4 (2018) 1–22, https://doi.org/10.1038/nrdp.2018.10.

[3] D.S. Charles Bishop, Function and red blood cell, Nature (1965) 435.

[4] D. Iskander, G. Wang, E.F. Heuston, C. Christodoulidou, B. Psaila, K. Ponnusamy, H. Ren, Z. Mokhtari, M. Robinson, A. Chaidos, P. Trivedi, N. Trasanidis, A. Katsarou, R. Szydlo, C.G. Palii, M.H. Zaidi, Q. Al-Oqaily, V.S. Caputo, A. Roy, Y. Harrington, L. Karnik, K. Naresh, A.J. Mead, S. Thongjuea, M. Brand, J. de la Fuente, D.M. Bodine, I. Roberts, A. Karadimitris, Single-cell profiling of human bone marrow progenitors reveals mechanisms of failing erythropoiesis in Diamond-Blackfan anemia, Sci. Transl. Med. 13 (2021), https://doi.org/10.1126/SCITRANSLMED.ABF0113/SUPPL_FILE/SCITRANSLMED.ABF0113_DATA_FILES_S1_TO_S6.ZIP.

[5] A.M. Taylor, E.R. Macari, I.T. Chan, M.C. Blair, S. Doulatov, L.T. Vo, D.M. Raiser, K. Siva, A. Basak, M. Pirouz, A.N. Shah, K. McGrath, J.M. Humphries, E. Stillman, B.P. Alter, E. Calo, R.I. Gregory, V.G. Sankaran, J. Flygare, B.L. Ebert, Y. Zhou, G.Q. Daley, L.I. Zon, Calmodulin inhibitors improve erythropoiesis in Diamond-Blackfan anemia, Sci. Transl. Med. 12 (2020), https://doi.org/10.1126/SCITRANSLMED.ABB5831/SUPPL_FILE/ABB5831_SM.PDF.

[6] Y. Zhang, Y. Dai, J. Wen, W. Zhang, A. Grenz, H. Sun, L. Tao, G. Lu, D.C. Alexander, M.V. Milburn, L. Carter-Dawson, D.E. Lewis, W. Zhang, H.K. Eltzschig, R. E. Kellems, M.R. Blackburn, H.S. Juneja, Y. Xia, Detrimental effects of adenosine signaling in sickle cell disease, Nat. Med. 171 (17) (2010) 79–86, https://doi.org/10.1038/nm.2280, 2010.

[7] H. Brody, M. Grayson, Sickle-cell disease 13 (515) (2014) 7526.

[8] J.F. Menezes, M.O.S. Carvalho, L.C. Rocha, F.M. dos Santos, E.V. Adorno, C.C. de Souza, R.P. Santiago, C.C. da Guarda, R.M. de Oliveira, C.V.B. Figueiredo, S. P. Carvalho, S.C.M.A. Yahouédéhou, L.M. Fiuza, C.S.A. Adanho, T.N. Pitanga, I.M. Lyra, V.M.L. Nascimento, A.A. Noronha-Dutra, M.S. Goncalves, Role of paraoxonase 1 activity and PON1 gene polymorphisms in sickle cell disease, Sci. Rep. 13 (131) (2023) 1–10, https://doi.org/10.1038/s41598-023-34396-1, 2023.

[9] H. Frumkin, G.N. Bratman, S.J. Breslow, B. Cochran, P.H. Kahn, J.J. Lawler, P.S. Levin, P.S. Tandon, U. Varanasi, K.L. Wolf, S.A. Wood, Nature contact and human health: a research agenda, Environ. Health Perspect. 125 (2017), https://doi.org/10.1289/EHP1663.

[10] E. Ben-Akiva, R.A. Meyer, H. Yu, J.T. Smith, D.M. Pardoll, J.J. Green, Biomimetic anisotropic polymeric nanoparticles coated with red blood cell membranes for enhanced circulation and toxin removal, Sci. Adv. 6 (2020), https://doi.org/10.1126/SCIADV.AAY9035/SUPPL_FILE/AAY9035_SM.PDF.

[11] O.S. Platt, D.J. Brambilla, W.F. Rosse, P.F. Milner, O. Castro, M.H. Steinberg, P.P. Klug, Mortality in sickle cell disease – life expectancy and risk factors for early death, N. Engl. J. Med. (1994), https://doi.org/10.1056/nejm199406093302303.

[12] R. Ali, C. Oxlade, E. Borkowska, Sickle cell toothache, 2008 20510, Br. Dent. J. 205 (2008), https://doi.org/10.1038/sj.bdj.2008.990, 524–524.

[13] J.A. Panepintoa, A. Brandow, L. Mucalo, F. Yusuf, A. Singh, B. Taylor, K. Woods, A.B. Payne, G. Peacock, L.A. Schieve, Coronavirus disease among persons with sickle cell disease, United States, march 20-may 21, 2020, emerg, Inf. Disp. 26 (2020) 2473–2476, https://doi.org/10.3201/EID2610.202792.

[14] E.B. Esrick, L.E. Lehmann, A. Biffi, M. Achebe, C. Brendel, M.F. Ciuculescu, H. Daley, B. MacKinnon, E. Morris, A. Federico, D. Abriss, K. Boardman, R. Khelladi, K. Shaw, H. Negre, O. Negre, S. Nikiforow, J. Ritz, S.-Y. Pai, W.B. London, C. Dansereau, M.M. Heeney, M. Armant, J.P. Manis, D.A. Williams, Post-transcriptional genetic silencing of BCL11A to treat sickle cell disease, N. Engl. J. Med. 384 (2021) 205–215, https://doi.org/10.1056/nejmoa2029392.

[15] M. Eapen, R. Brazauskas, M.C. Walters, F. Bernaudin, K. Bo-Subait, C.D. Fitzhugh, J.S. Hankins, J. Kanter, J.J. Meerpohl, J. Bolaños-Meade, J.A. Panepinto, D. Rondelli, S. Shenoy, J. Williamson, T.L. Woolford, E. Gluckman, J.E. Wagner, J.F. Tisdale, Effect of donor type and conditioning regimen intensity on allogeneic transplantation outcomes in patients with sickle cell disease: a retrospective multicentre, cohort study, Lancet Haematol 6 (2019) e585–e596, https://doi.org/10.1016/S2352-3026(19)30154-1.

[16] M.R. DeBaun, L.C. Jordan, A.A. King, J. Schatz, E. Vichinsky, C.K. Fox, R.C. McKinstry, P. Telfer, M.A. Kraut, L. Daraz, F.J. Kirkham, M.H. Murad, American Society of Hematology 2020 guidelines for sickle cell disease: prevention, diagnosis, and treatment of cerebrovascular disease in children and adults, Blood Adv 4 (2020) 1554–1588, https://doi.org/10.1182/bloodadvances.2019001142.

[17] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444, https://doi.org/10.1038/nature14539.

[18] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26, https://doi.org/10.1016/j.neucom.2016.12.038.

[19] N. Praljak, S. Iram, U. Goreke, G. Singh, A. Hill, U.A. Gurkan, M. Hinczewski, Integrating deep learning with microfluidics for biophysical classification of sickle red blood cells adhered to laminin, PLoS Comput. Biol. 17 (2021) 1–24, https://doi.org/10.1371/journal.pcbi.1008946.

[20] M.Z. Alom, T.M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M.S. Nasrin, M. Hasan, B.C. Van Essen, A.A.S. Awwal, V.K. Asari, A state-of-the-art survey on deep learning theory and architectures, Electron 8 (2019), https://doi.org/10.3390/electronics8030292.

[21] S. Deepak, P.M. Ameer, Brain tumor classification using deep CNN features via transfer learning, Comput. Biol. Med. 111 (2019), 103345, https://doi.org/10.1016/j.compbiomed.2019.103345.

[22] A. Çinar, M. Yildirim, Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture, Med. Hypotheses 139 (2020), 109684, https://doi.org/10.1016/j.mehy.2020.109684.

[23] U. Javed, K. Shaukat, I.A. Hameed, F. Iqbal, T.M. Alam, S. Luo, A review of content-based and context-based recommendation systems, Int. J. Emerg. Technol. Learn. 16 (2021) 274–306, https://doi.org/10.3991/IJET.V16I03.18851.

[24] K. Gupta, N. Chawla, Analysis of histopathological images for prediction of breast cancer using traditional classifiers with pre-trained CNN, Procedia Comput. Sci. 167 (2020) 878–889, https://doi.org/10.1016/j.procs.2020.03.427.

[25] L. Abdelrahman, M. Al Ghamdi, F. Collado-Mesa, M. Abdel-Mottaleb, Convolutional neural networks for breast cancer detection in mammography: a survey, Comput, Biol. Med. 131 (2021), 104248, https://doi.org/10.1016/j.compbiomed.2021.104248.

[26] R.S. Vanguri, J. Luo, A.T. Aukerman, J.V. Egger, C.J. Fong, N. Horvat, A. Pagano, J. de A.B. Araujo-Filho, L. Geneslaw, H. Rizvi, R. Sosa, K.M. Boehm, S.R. Yang, F.M. Bodd, K. Ventura, T.J. Hollmann, M.S. Ginsberg, J. Gao, R. Vanguri, M.D. Hellmann, J.L. Sauter, S.P. Shah, Multimodal integration of radiology, pathology and genomics for prediction of response to PD-(L)1 blockade in patients with non-small cell lung cancer, Nat. Can. (Ott.) 3 (2022) 1151–1164, https://doi.org/10.1038/s43018-022-00416-8.

[27] A.A. Ardakani, A.R. Kanafi, U.R. Acharya, N. Khadem, A. Mohammadi, Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: results of 10 convolutional neural networks, Comput. Biol. Med. 121 (2020), 103795, https://doi.org/10.1016/j.compbiomed.2020.103795.

[28] 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, https://doi.org/10.1109/ACCESS.2021.3102399.

[29] R.E. Ware, M. de Montalembert, L. Tshilolo, M.R. Abboud, Sickle cell disease, Lancet (2017), https://doi.org/10.1016/S0140-6736(17)30193-9.

[30] E.P. Vichinsky, L.D. Neumayr, A.N. Earles, R. Williams, E.T. Lennette, D. Dean, B. Nickerson, E. Orringer, V. McKie, R. Bellevue, C. Daeschner, E.A. Manci, Causes and outcomes of the acute chest syndrome in sickle cell disease. National Acute Chest Syndrome Study Group, N. Engl. J. Med. 342 (2000) 1855–1865, https://doi.org/10.1056/NEJM200006223422502.

[31] J.T. Horan, B.K. Fox, D.N. Korones, Management of children with sickle cell anemia and acute chest SYNDROME922, 1996, Pediatr. Res. 394 (39) (1996), https://doi.org/10.1203/00006450-199604001-00944, 156–156.

[32] G. Karayalcin, A.K. Jou, A.J. Aballi, P. Lanzkowsky, Gall bladder disease (GBD) in children with sickle cell disease (SS), 1977, Pediatr. Res. 114 (11) (1977), https://doi.org/10.1203/00006450-197704000-00619, 473–473.

[33] T.M. Alam, K. Shaukat, I.A. Hameed, S. Luo, M.U. Sarwar, S. Shabbir, J. Li, M. Khushi, An investigation of credit card default prediction in the imbalanced datasets, IEEE Access 8 (2020) 201173–201198, https://doi.org/10.1109/ACCESS.2020.3033784.

[34] M.H. Steinberg, Predicting clinical severity in sickle cell anaemia, Br. J. Haematol. 129 (2005) 465–481, https://doi.org/10.1111/J.1365-2141.2005.05411.X.

[35] H. Kutlu, E. Avci, F. Özyurt, White blood cells detection and classification based on regional convolutional neural networks, Med. Hypotheses 135 (2020), 109472, https://doi.org/10.1016/j.mehy.2019.109472.

[36] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (2017) 84–90, https://doi.org/10.1145/3065386.

[37] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. (2015) 1–14.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn. (2015) 1–9, https://doi.org/10.1109/CVPR.2015.7298594, 07-12-June.

[39] P.K. Das, S. Meher, R. Panda, A. Abraham, A review of automated methods for the detection of sickle cell disease, IEEE Rev. Biomed. Eng. (2020), https://doi.org/10.1109/RBME.2019.2917780.

[40] H.F. Bunn, Pathogenesis and treatment of sickle cell disease, N. Engl. J. Med. (1997), https://doi.org/10.1056/nejm199709113371107.

[41] F.B. Piel, A.P. Patil, R.E. Howes, O.A. Nyangiri, P.W. Gething, M. Dewi, W.H. Temperley, T.N. Williams, D.J. Weatherall, S.I. Hay, Global epidemiology of Sickle haemoglobin in neonates: a contemporary geostatistical model-based map and population estimates, Lancet (2013), https://doi.org/10.1016/S0140-6736(12)61229-X.

[42] F.O. Ortiz, T.K. Aldrich, R.L. Nagel, L.J. Benjamin, Accuracy of pulse oximetry in sickle cell disease, Am. J. Respir. Crit. Care Med. (1999), https://doi.org/10.1164/ajrccm.159.2.9806108.

[43] N. Petrović, G. Moyà-Alcover, A. Jaume-i-Capó, M. González-Hidalgo, Sickle-cell disease diagnosis support selecting the most appropriate machine learning method: towards a general and interpretable approach for cell morphology analysis from microscopy images, Comput. Biol. Med. 126 (2020), https://doi.org/10.1016/j.compbiomed.2020.104027.

[44] erythrocytesIDB (Version 2, October 2017), (n.d.). http://erythrocytesidb.uib.es/(accessed December 15, 2022).

[45] M. Khalaf, A.J. Hussain, R. Keight, D. Al-Jumeily, P. Fergus, R. Keenan, P. Tso, Machine learning approaches to the application of disease modifying therapy for sickle cell using classification models, Neurocomputing 228 (2017) 154–164, https://doi.org/10.1016/j.neucom.2016.10.043.

[46] L. Alzubaidi, M.A. Fadhel, O. Al-shamma, J. Zhang, Y. Duan, Deep learning models for classification of red blood cells in microscopy images to aid in sickle cell anemia diagnosis, Electron 9 (2020), https://doi.org/10.3390/electronics9030427.

[47] D. Parthasarathy, wbc-classification/Original_Images at Master · Dhruvp/wbc-Classification · GitHub, 2019. https://github.com/dhruvp/wbc-classification/tree/master/Original_Images. (Accessed 15 November 2019).

[48] K. Al-Dulaimi, V. Chandran, J. Banks, I. Tomeo-Reyes, K. Nguyen, Classification of white blood cells using bispectral invariant features of nuclei shape, Int. Conf. Digit. Image Comput. Tech. Appl. DICTA 2019 (2018), https://doi.org/10.1109/DICTA.2018.8615762, 2018.

[49] Sickle Cell Anemia. https://sicklecellanaemia.org/teaching-resources/resources/scooter44-63/, 2019. (Accessed 1 September 2019).

[50] M. Darrin, A. Samudre, M. Sahun, S. Atwell, C. Badens, A. Charrier, E. Helfer, A. Viallat, V. Cohen-Addad, S. Giffard-Roisin, Classification of red cell dynamics with convolutional and recurrent neural networks: a sickle cell disease case study, Sci. Rep. 13 (2023) 1–12, https://doi.org/10.1038/s41598-023-27718-w.

[51] D. Maxime, S. Ashwin, S. Maxime, A. Scott, B. Catherine, C. Anne, H. Emmanuele, V. Annie, C.-A. Vincent, G.-R. Sophie, Classification of Blood Cells Dynamics with Convolutional and Recurrent Neural Networks: a Sickle Cell Disease Case Study, 2021, https://doi.org/10.5281/ZENODO.5723606.

[52] C. Patgiri, A. Ganguly, Adaptive thresholding technique based classification of red blood cell and sickle cell using Naïve Bayes Classifier and K-nearest neighbor classifier, Biomed. Signal Process Control 68 (2021), 102745, https://doi.org/10.1016/j.bspc.2021.102745.

[53] K.M. Hosny, M.A. Kassem, M.M. Fouad, Classification of skin lesions into seven classes using transfer learning with AlexNet, J. Digit. Imag. 33 (2020) 1325–1334, https://doi.org/10.1007/s10278-020-00371-9.

[54] A. Merino, L. Puigví, L. Boldú, S. Alférez, J. Rodellar, Optimizing morphology through blood cell image analysis, Int. J. Lab. Hematol. (2018), https://doi.org/10.1111/ijlh.12832.

[55] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn. (2015) 770–778, https://doi.org/10.48550/arxiv.1512.03385, 2016-December.

[56] H. Alaskar, N. Alzhrani, A. Hussain, F. Almarshed, The implementation of pretrained AlexNet on PCG classification, Lect. Notes Comput. Sci. 11645 (2019) 784–794, https://doi.org/10.1007/978-3-030-26766-7_71. LNAI.

[57] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D.F. Schmidt, J. Weber, G.I. Webb, L. Idoumghar, P.A. Muller, F. Petitjean, InceptionTime: finding AlexNet for time series classification, Data Min. Knowl. Discov. 34 (2020) 1936–1962, https://doi.org/10.1007/s10618-020-00710-y.

[58] G. Jacob, R.T. Pramod, H. Katti, S.P. Arun, Qualitative similarities and differences in visual object representations between brains and deep networks, Nat. Commun. 121 (12) (2021) 1–14, https://doi.org/10.1038/s41467-021-22078-3, 2021.

[59] A. Victor Ikechukwu, S. Murali, R. Deepu, R.C. Shivamurthy, ResNet-50 vs VGG-19 vs training from scratch: a comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images, Glob. Transitions Proc. 2 (2021) 375–381, https://doi.org/10.1016/j.gltp.2021.08.027.

[60] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, MobileNetV2 : Inverted Residuals and Linear Bottlenecks, 2018, pp. 4510–4520.

[61] N. Arya, S. Saha, A. Mathur, S. Saha, Improving the robustness and stability of a machine learning model for breast cancer prognosis through the use of multi-modal classifiers, Sci. Rep. 13 (2023) 1–10, https://doi.org/10.1038/s41598-023-30143-8, 2023 131.

[62] Y. Li, H. Zheng, X. Huang, J. Chang, D. Hou, H. Lu, Research on lung nodule recognition algorithm based on deep feature fusion and MKL-SVM-IPSO, Sci. Rep. 12 (2022) 1–18, https://doi.org/10.1038/s41598-022-22442-3, 2022 121.

[63] M. Reza, S. Miri, R. Javidan, A hybrid data mining approach for intrusion detection on imbalanced NSL-KDD dataset, Int. J. Adv. Comput. Sci. Appl. 7 (2016) 1–33, https://doi.org/10.14569/ijacsa.2016.070603.

[64] C. Cortes, V. Vapnik, Support-Vector Networks, Mach. Learn, 1995, https://doi.org/10.1023/A:1022627411411.