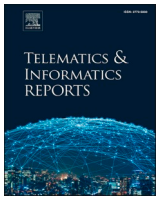
Telematics and Informatics Reports 16 (2024) 100163

Contents lists available at ScienceDirect

Telematics and Informatics Reports

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



A deep learning approach for white blood cells image generation and classification using SRGAN and VGG19

Jannatul Ferdousi, Soyabul Islam Lincoln, Md. Khorshed Alom *, Md. Foysal

Department of Electronics and Communication Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh

ARTICLE INFO

Keywords:
Deep learning
SRGAN
VGG19
WBC image generation
WBCs classification

ABSTRACT

The classification of White Blood Cells (WBCs) is crucial for diagnosing diseases, monitoring treatment effectiveness, and understanding how the immune system functions. In this paper, we propose a deep learning approach to classify WBCs using Super Resolution Generative Adversarial Network (SRGAN) and Visual Geometry Group 19 (VGG19). Firstly, microscopic images of WBCs are generated using the SRGAN to obtain more precise and high-resolution images, which are then classified with a pretrained VGG19 classifier. Low-resolution (LR) images are inputted into the generator of SRGAN, and its discriminator compares the High-resolution (HR) image with LR, generating super-resolution images to minimize misclassification risks. A large dataset of 12,447 images containing four classes of WBCs (Eosinophil, Lymphocyte, Monocyte, and Neutrophil) is utilized to train and validate our proposed model. Following extensive experimental analysis, our proposed model achieves a test accuracy of 94.87 %, surpassing traditional Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Hybrid CNN-RNN models, and other conventional approaches. The generated images of SRGAN overcome challenges associated with misclassification due to the poor resolution of microscopic images, while the use of a pretrained model as a classifier reduces classification complexity. The source code of the entire work is available at https://github.com/Jannatul-Ferdousi/SRGAN_VGG19_WBC.git.

1. Introduction

Human blood consists of three types of cells: Red Blood Cells (RBCs), WBCs and Platelets. Each of these three cell types has specific functions assigned to it. For example, RBCs are essential for removing carbon dioxide from the body's tissues and transferring oxygen from the lungs to those tissues; WBCs help fight against infections and diseases; and platelets are essential for clot formation and preventing excessive bleeding by promoting blood coagulation. The various types of WBCs in the human body include Neutrophils, Lymphocytes (T cells, B cells, and Natural Killer cells), Monocytes, Eosinophils, and Basophils. Different WBCs work together to detect and combat various illnesses found in the bloodstream, which is crucial in maintaining the body's health and preventing infections and diseases. The overall quantity and diversity of WBCs indicate a person's current state of health.

For proper classification of WBCs, a CNN might be considered as the classifier such as Residual Network 50 (ResNet50), VGG19, Inception Version 3 (InceptionV3), Mobile Network Version 2 (MobileNetV2), Densely Connected Convolutional Network (DenseNet) for their ability to learn high-level characteristics and specifications from image data. These models are tested with full parameter learning and transfer learning. Matrix transformation methods such as rescaling, data augmentation, Principal Component Analysis (PCA) [1], Learning Vector Quantization (LVQ) [2], and Non-Negative Matrix Decomposition (NNMD) are utilized to preprocess and prepare the data for analysis. Whereas, Generative Adversarial Networks (GANs) can be used for generative tasks like generating new images from a given distribution. Deep Convolutional GAN (DCGAN) is used for image generation tasks, while Wasserstein GAN (WGAN) is used to address stability issues [3,4]. CycleGAN is utilized for unsupervised image-to-image translation, while Progressive GAN (ProGAN) is employed for generating high-resolution images [5], and Information Maximizing GAN (InfoGAN) [6] is used to disentangle the factors of variation in the generated images. SRGAN is a type of GAN that produces high-resolution images with more details. These generated images can successfully overcome the complexities of data preprocessing and result in higher accuracy in image classification. The best type of GAN has to be chosen depending on the task requirements and data features.

* Corresponding author.
E-mail address: khorshed@ece.kuet.ac.bd (Md.K. Alom).

https://doi.org/10.1016/j.teler.2024.100163
Received 16 May 2024; Received in revised form 29 August 2024; Accepted 13 September 2024
Available online 16 September 2024

Classification of images can be achieved by training a GAN with a combination of real and synthetic images labeled with their corresponding classes and then utilizing the trained GAN to make predictions on new, unseen images. But, GANs are superior for image processing because they can generate images similar to given training images, rendering them useful for tasks like image synthesis, image-to-image translation and data augmentation. In a GAN, the generator initially generates new images using a random noise vector, while the discriminator architecture tries to distinguish between the generated or fake images and the real images present in the training set. There are other factors that make GAN a good choice for image processing, such as training GAN on large datasets to enable them to learn the underlying distributions of data [4].

However, classifying WBC images is challenging due to poor image quality, low resolution, and high noise. Staining techniques can cause cell deformation and alter the color of the nucleus, cytoplasm, and background. Light intensity can also affect the appearance of a picture. The nucleus may appear at various sizes and locations within cells and as WBCs progress from the immature blast stage to the mature form present in the blood, the cytoplasm, nucleus shape, location, and cell size change. Image noise and alterations in contrast can be attributed to blurry or poor image and the use of diverse imaging systems or cameras [7]. Again, data imbalance is a common issue in classifying WBCs, making it difficult for a classifier to identify them accurately. To enhance image quality for correct classification, improving visual quality is important [8].

Classification of WBCs are important for diagnosing different blood disease and disorders like leukemia, lymphoma. It also helps in detecting the viral, bacterial or parasitic infections of blood. Classifying WBCs is also crucial for patient health monitoring while undergoing chemotherapy, organ transplants etc. Miscount and classifications of WBCs can lead to severe damage of health, even causing death. Proper classification of WBCs is a crucial need for the proper diagnosis and treatment. However, poor resolution of WBC images can lead to misdiagnosis, potentially worsening health outcomes and even resulting in death. Hence, the major contributions of this research are:

- Developing a SRGAN network following the features of a pre-trained VGG19 classifier.
- Generating super-resolution images of WBCs by training the GAN with the low resolution of images.
- An extensive experimental analysis with other state-of-the-art CNN classification models such as Efficient Network B7 (EfficientNetB7), ResNet50, and Visual Geometry Group 16 (VGG16) to find out a robust network.
- Finally, presenting a comparative study with recent related works in WBC classification using deep learning approaches.

The remaining sections of this paper are structured as follows: Section 2 provides the theoretical background of this work, including a literature review of recent works. Section 3 explains the proposed methodology used in this work. Section 4 is the representation of experimental result analysis, while Section 5 discusses our outcomes, concluding in Section 6.

2. Related works

Feature extraction and classification are the two core functions of a machine learning system. The procedures of feature extraction and classification are handled independently and separately in traditional learning approaches. Deep learning, on the other hand, combines many machine learning methods using multi-layer neural networks. These networks use several layers to analyze the input, each of which captures ever more abstract representations of the material. Deep learning thereby unifies feature extraction and categorization into a single, cohesive process., enabling automatic feature learning from the data in a single step. The categorization of WBCs is crucial for diagnosing blood-related illnesses. Through the use of machine learning, several classification techniques have been developed, including K-Nearest Neighbors (KNNs), Support Vector Machine (SVM), random forest, Bayesian, CNN, and cyclic CNN algorithms. Supardi et al. [9] conducted studies on the k value and distance measurement of a well-known method, the KNN algorithm, and worked on acute leukemia by extracting 12 key features from its blood pictures. The outcomes demonstrate that when $k = 4$, the KNN algorithm has an accuracy of 80 % in differentiating between acute lymphocytic leukemia and acute myeloid leukemia. Image enhancement and arithmetic operations can be applied for WBC segmentation, as mentioned in [10], the authors proposed Otsu's threshold approach for automatically segment blood cell. In the 108-image public data set, this approach achieved an accuracy of 93 %. In [11], Leukemia cells were studied using SVM and KNN algorithm. The SVM map determines the largest marginal hyperplane in the cell images after mapping the data into kernel space. Leukemia cells are classified using the KNN classifier by identifying changes in their texture, geometry, color, and other properties. In [17], a novel and highly efficient computational Binary Random Forest Feature Selection (BRFFS) approach based on the decrease of Gini impurity was introduced. Through the effective removal of redundant characteristics, the classifier's performance was enhanced. But when processing extremely noisy WBC pictures, the BRFFS algorithm is prone to overfitting.

A method to detect blood changes automatically, was proposed by Sinha et al. [12], employing a histological image as input and a Bayesian algorithm to recognize and categorize WBCs. The authors of [13] used a watershed segmentation using markers with morphological operators and mathematical analysis to separate WBCs and nuclei in blood microscopy images. The classification and recognition of 150 cell pictures with the Bayesian approach achieved an accuracy of 83.2 %. Prinyakupt et al. [14] divided WBCs and segmented cells using threshold processing, morphological processes, and elliptic curve fitting. Accuracy data was collected at various intervals, revealing a 90 % accuracy rate in the end. A brand-new cell segmentation approach was introduced in [15], which seeks the WBC identification region in the Hue, Saturation, and Intensity (HIS) color space. The neural network utilizes geometric properties, color attributes, and Local Difference Pattern (LDP) based texture features to detect WBCs efficiently. Zhao et al. [16] developed a microscopic image WBC identification technique based on color and morphological operation. They categorized eosinophils and basophils based on SVM and particle size characteristics, then utilized CNN to extract high-level properties. With 215 photos evaluated, the classifier achieved an overall accuracy of 92.8 %, but only 74.8 % for lymphocytes. In [17], the authors suggested a probabilistic model with convolved cell templates for converting cell population pictures into atomic sums, while Wang et al. [18] introduced a novel technique called the Pattern Fusion Integrated Convolution Network (PECNN) to categorize WBCs. Twenty healthy blood donor samples and twelve medically abnormal blood donor samples were lysed to create a holographic image, and the findings were effectively used for the identification, counting, and categorization of WBCs. The error rate was <6.8 %.

Vision Transformer (ViT) models give higher accuracy in the field of medical images. Together YOLO and ViT provides 96.449 % accuracy in classifying WBCs [19] . Several experiments have done combining global ViT and CNNs for WBC classifications. In spite of providing higher accuracy, ViT is not reliable in the field of medical imaging as it can not focus on a known and exact area to predict, whereas the subclasses of WBCs are identified focusing the structure of the nucleus. Besides this method is not a good choice for smaller datasets at all.

Therefore, we may infer that typical machine learning approaches require pre-processing image segmentation before categorizing cells, which impacts the accuracy of the collected features and the final classification. In previous studies, we have observed various complex algorithms used to process images before classification. While this may slightly improve result accuracy, processing large amounts of data

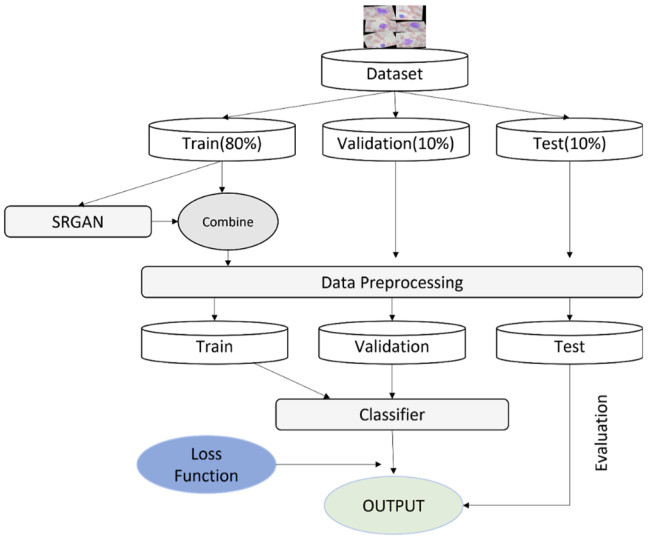


Fig. 1. Overview of procedural steps.

requires resources and may compromise automation. On the other hand, generating better images from microscopic images of WBCs using GAN models can mitigate these complexities and provide improved accuracy, even in a pre-trained model. DCGAN, along with ResNet50 achieved 91.7 % accuracy without relying on image preprocessing and segmentation [3].

After reviewing the research works above, it is concluded that there is still scope to improve the precise diagnosis system. In some studies, the preprocessing is quite complicated and time-consuming, which could pose a major problem for user-end implementation. Another significant issue is data imbalance, with some studies not taking it into account. Our research proposes a network structure that combines a GAN model (SRGAN) with the features of VGG19 and a pre-trained classifier (VGG19) to address problems such as overfitting, poor resolution, insufficient samples, annotations, and out-of-distribution inputs with the aim of achieving the highest classification accuracy. Although various neural network models provide much better accuracy, when the image are of great quality. But most of the architecture fails when the image quality is very poor or noisy. It is also often observed that the dataset is insufficient for the experiment, and the data samples are not evenly distributed, leading to frequent overfitting. To mitigate these issues, a new model architecture is introduced, combining SRGAN with a pre-trained classifier, VGG19. The techniques and concepts used in SRGAN help overcome such model uncertainties by providing more detail to the sampled images. Due to the adversarial nature of SRGAN, data distribution occurs extensively, addressing overfitting. Additionally, the transfer learning methods used for transferring model parameters increase convergence speed and enhance the model's robustness.

3. Methodology

This section outlines the key elements of WBCs classification methodology. It describes the idea of SRGAN for enhancing image quality, explores various loss functions essential for model training, and delves into the architecture of VGG19 for effective classification. The internal generator and discriminator mechanisms of the SRGAN structure are also briefly explained in this section. Together, these components form a robust framework for accurate and reliable WBCs classification. The full procedure of this experiment is shown in Fig. 1. Firstly, the dataset was divided into three parts: training, validation, and test sets, during the procedural phases. The WBC training photos are then combined with the original training images to create high-resolution images, which are subsequently used to train the classifier. During the classification and evaluation of data, a loss function is employed to minimize classification error.

3.1. Super-Resolution generative adversarial network (SRGAN)

GANs have been utilized by several researchers from a variety of domains. Typically, the Generator (G) and the Discriminator (D) constitute the two opposing parts of GAN and engage in a minimax game. The generator seeks to persuade the discriminator that the fake samples are real and aims to provide samples that are realistic and record data dispersion. Conversely, the discriminator's goal is to identify the origin of incoming samples. The whole procedure is described by the following formulation (Eq. (1)), where the discriminator and generator compete against each other by employing the value function $V(G, D)$. The success of the opposing component directly impacts the cost of each network.

$$\min_{G}\max_{D} V(G,D) = E_x[\log D(x)] + E_z[\log(1 - D(G(z)))] \quad (1)$$

When provided with a given noise z, the generator outputs $G(z)$. The discriminator then estimates $D(G(z))$, which shows the possibility that a fake occurrence is real. Additionally, the discriminator estimates $D(x)$, which shows the probability that a valid data object exists. E_x and E_z

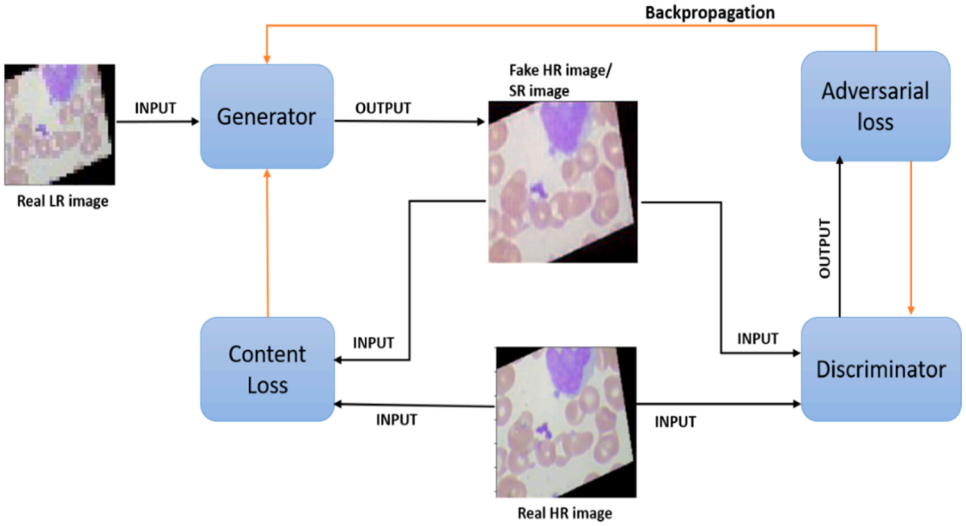


Fig. 2. General structure of the SRGAN training process.

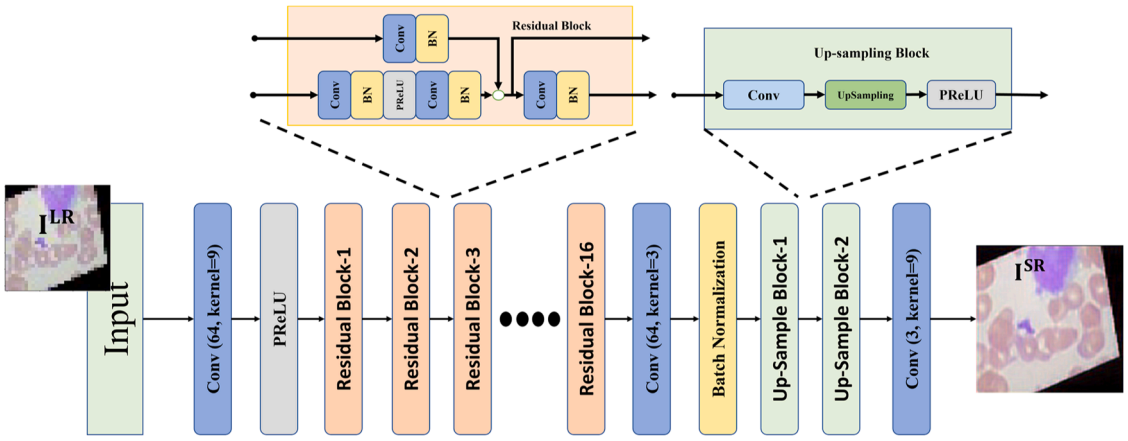


Fig. 3. Architecture of the Generator.

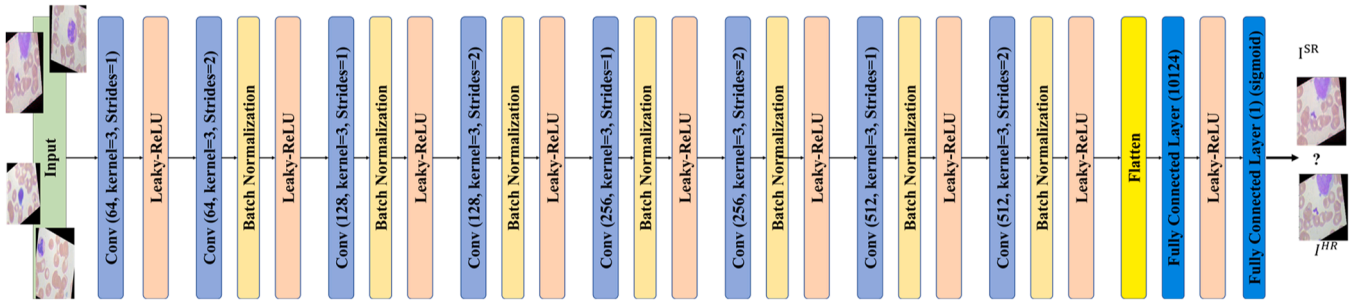


Fig. 4. Architecture of the Discriminator.

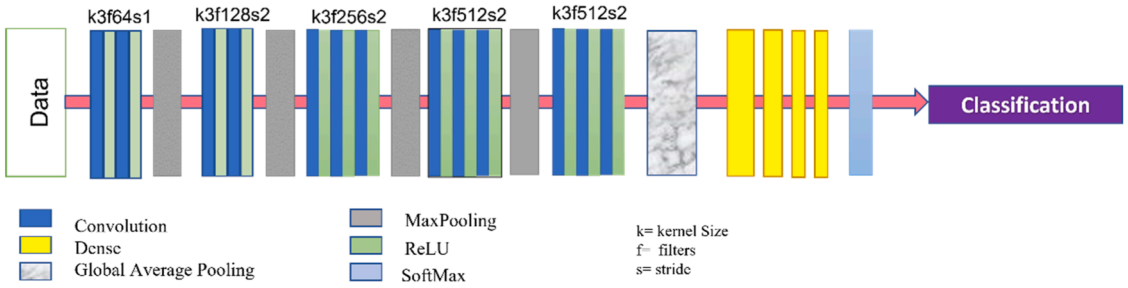


Fig. 5. Architecture of the VGG19 classifier.

represents the expected values for all real data samples and all random generator inputs used to produce these samples, respectively [10,20].

SRGAN is a super-resolution network that has the ability to create an adversarial network. Here, "super-resolution" refers to the process of generating HR images from LR images, which encompasses two sub-categories: multi-frame super-resolution and Single-Image Super-Resolution (SISR). In SRGAN, LR images are regarded as noise, which is fed into G. D compares the output of G with HR images, and the loss is calculated to adjust the weights and biases. The generator network is continuously aiming to make the LR picture closer to the true HR image. In contrast, the discriminator network generates a discrimination probability that may be used to assess the accuracy of the assertion by attempting to differentiate between the produced photos and actual HR images. In addition to the identification loss, the loss function is incorporated into the network to guide it towards producing artificially high-quality photos [21]. Fig. 2 provides a visual representation of this entire process.

The goal of the SISR in SRGAN is to transform a LR input image (I^{LR}) into a HR image, super-resolution image (I^{SR}). This instance involves the low-resolution counterpart of its high-resolution version, I^{HR}, is referred to as I^{LR}. High-resolution images are used solely for training purposes. During training, I^{HR} undergoes Gaussian filtering before being down-sampled with a down-sampling factor of r. We define I^{LR} as a real-valued tensor of size $W \times H \times C$ for an image with C color channels, and I^{HR} and I^{SR} as $W_r \times H_r \times C$, respectively.

The main objective of this study is to train a generating function G that can predict the associated high-resolution (HR) picture from a given low-resolution (LR) input. We develop a generator network as a feed-forward CNN G_{θ_G}, where the $\theta_G = \{W_{1:L}, b_{1:L}\}$ represents the generator network's weights and biases across l-layer . This parameter is fine-tuned by minimizing the loss function I_{SR}. For high resolution training images I_n^{HR} and their corresponding low-resolution images,

$$\widehat{\theta}_G = \underset{\theta_G}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} l_{SR}\left(G_{\theta_G}\left(I_n^{LR}\right), I_n^{HR}\right) \tag{2}$$

Here, l_{SR} represents the loss function for super resolution images and $\widehat{\theta}_G$ is the updated parameter of generator network. The discriminator

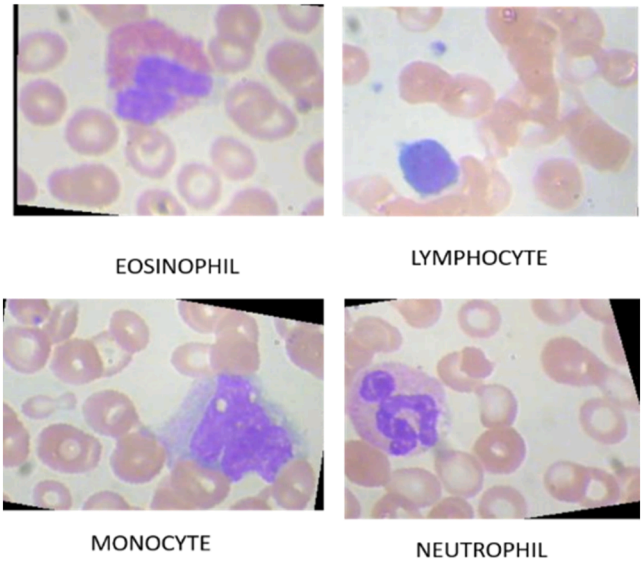


Fig. 6. Different types of White Blood Cell images.

network D aims to detect between fake and real image. Here, in this equation, $-\log\left(D_{\theta_D}\left(I_n^{HR}\right)\right)$ correspond to the loss while classifying the real high-resolution image and $-\log\left(1-\left(G_{\Theta_G}\left(I_n^{LR}\right)\right)\right)$ corresponds to the loss to classify the generated high-resolution image.

$$\widehat{\theta}_D = \arg\min_{\theta_D} \frac{1}{N}\sum_{n=1}^{N}\left[-\log\left(D_{\theta_D}\left(I_n^{HR}\right)\right) - \log\left(1 - D_{\theta_D}\left(G_{\Theta_G}\left(I_n^{LR}\right),\ I_n^{HR}\right)\right)\right] \tag{3}$$

3.1.1. Generator

The generator portion comprise of 16 residual blocks and 2 up-sampling blocks along with several convolutional blocks. The residual block consists of two convolutional blocks, each having 64 filters and a kernel size of 3 × 3, maintaining the spatial dimensions of the input through 'same' padding. In the residual block, the input tensor is first passed through a convolutional layer, followed by a batch normalization layer to stabilize and accelerate the training process by normalizing the output coming from the convolutional layer. This is followed by a Parametric Rectifies Linear Unit (PReLU) activation function, which provides nonlinearity while allowing for adaptively learnt activation parameters shared across spatial dimensions. The resulting feature map is then processed by a second convolutional layer, which is identical to the first, and then normalized again using batch normalization. Finally, the output of this sequence is added element-wise to the original input tensor (residual connection), allowing the network to learn identity mappings and handle difficulties such as vanishing gradients. This arrangement is repeated throughout 16 residual blocks, which aids in the acquisition of deeper representations. This residual block

Table 1
Parameters summary for Generator.

Layer	Number of Parameters
Convolutional Layer	15,616
PReLU	64
Residual Blockx16	
Convolutional Layer	36,928
Batch Normalization	256
PReLU	64
Convolutional layer	36,928
Batch Normalization	256
Convolutional layer	147,712
Upsampler	0
PReLU	256
Convolutional layer	590,080
Upsampler	0
PReLU	256
Convolutional Layer	62,211
Total	**2044,291**

Table 2
Parameters summary for discriminator.

Layer	Number of parameters
Convolutional Layer	1792
Convolutional Layer	36,928
Batch Normalization	256
Convolutional Layer	73,856
Batch Normalization	512
Convolutional Layer	147,584
Batch Normalization	512
Convolutional Layer	295,168
Batch Normalization	1024
Convolutional Layer	590,080
Batch Normalization	1024
Convolutional Layer	1,180,160
Batch Normalization	2048
Convolutional Layer	2,359,808
Batch Normalization	2048
Dense	33,555,456
Dense	1025
Total	**38,249,281**

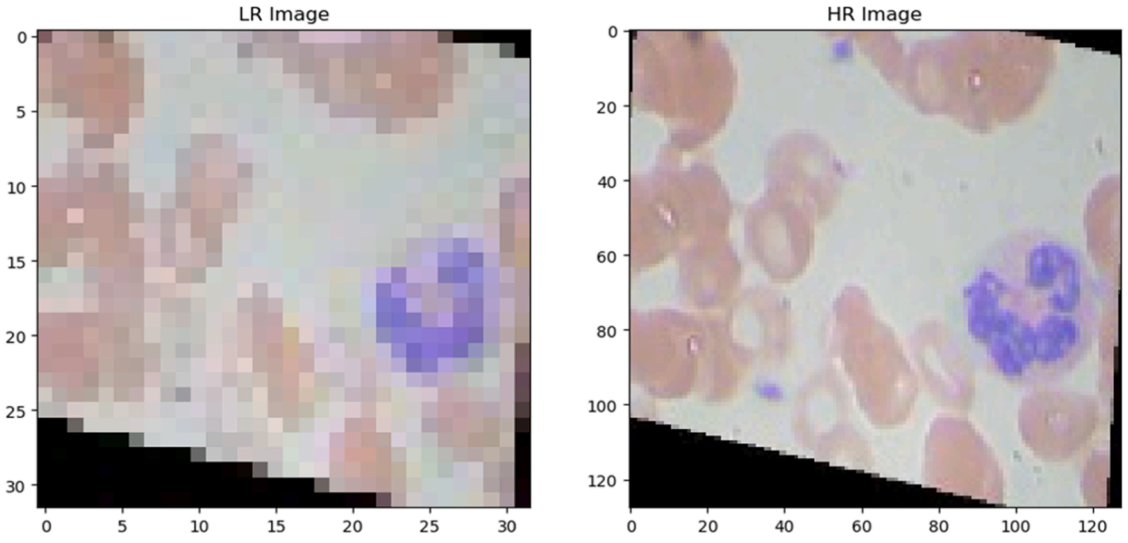


Fig. 7. LR images of 32×32 pixels (Left) and HR images of 128×128 pixels (Right).

Table 3
Total Parameters of VGG (extracted).

Layer	Number of parameters
VGG19 (pre-trained)	20,024,384
Dense_1	525,312
Dense 2	524,800
Dense_3	131,328
Dense_4	1028
Total	**21,206,852**

Table 4
Hyperparameters for Classification.

Hyperparameter	Value
Batch Size	4
Learning rate	0.001
Epoch	50
Activation	ReLU
Callbacks	EarlyStopping, ReduceOnPlateau
Loss	Sparse Categorical Crossentropy

arrangement used in here is called Gross and Wilber block layout, which was proposed by Shi et al. [22]. The up-sampling block consists of one convolutional layer with 64 filters and kernel size of 3 × 3. This convolutional layer increases the depth of feature map, enhancing the model's ability to learn complex features. After the convolutional layer, an upsampling layer is applied, which basically does a nearest-neighbor interpolation by doubling the spatial dimensions of the feature map (height and breadth) while keeping the depth constant. In order to enable the network to learn activation thresholds adaptively, this up-sampled output is then delivered via a PReLU activation function with spatially shared parameters. The whole architecture of generator is illustrated in Fig. 3.

Basically, the generator model receives a batch of low-resolution images as input in the form of tensors. Firstly, image tensor undergoes a convolutional layer with 64 filters and kernel size of 9 × 9. This is followed by a PReLU activation function. Next, the tensor passes through a series of residual blocks that allows the network to learn residual mappings that facilitate deeper representations. The skip connection in residual block helps preserving low-level details. Following the residual block sequence, a second convolutional layer with 64 filters and a 3 × 3 kernel processes the features before batch normalization. The network then applies two successive up-sampling block that increases the spatial resolution of the feature maps, preparing them for the final reconstruction phase. Lastly, the upscaled feature map is passed through a convolutional layer with 3 filters and a large kernel size of 9 × 9. This layer reduces the depth of tensor map to match the desired output channel dimension which is 3 for an red-Green-Blue (RGB) image.

3.1.2. Discriminator

The discriminator part follows a linear network structure, similar to the Visual Geometry Group Network (VGGNet) structure, which is based on the novel work of Xingchen et al. [23]. Primarily, the discriminator network is designed to distinguish between generated Super Resolution (SR) images and real HR images. In Fig. 4, we observe that the model's first layer comprises a Conv2D with 64 filters with a kernel size of 3 × 3 followed by a Leaky Rectified Linear Unit (Leaky-ReLU) activation function. The number of picture features increases with the augmented depth of network model layers, and the size of each extracted feature's convolution becomes smaller. The Leaky-ReLU activation function has a better impact than the ReLU function with negative value, which can reduce the likelihood of sparse gradient. Therefore, Leaky-ReLU is chosen as the activation function. The first convolutional layer is then followed by seven modules consisting of a convolutional layer, a Batch Normalization layer, and a Leaky-ReLU activation function. The filters of the convolutional layers increase progressively, while occasionally down-sampling the spatial dimension of feature maps. The number of filters doubles every two clocks, from 64 filters in the first block to 512 in the final block. This allows the network to learn increasingly complex features at different levels of abstraction. After the feature maps are flattened, the series of convolutional blocks ends with a fully connected (dense) layer. The network's last layer is a fully connected single-unit layer with a sigmoid activation function. This layer provides a probability value to determine if the incoming image is real or fake. numbers around 1 signify a greater probability of being real, whereas numbers near 0 suggest a fake picture.

3.2. Loss function

It is difficult to handle the uncertainty involved in recovering lost high-frequency information, such as texture, using pixel-wise loss functions like Mean Squared Error (MSE). To minimize the value of MSE tends to encourage the generation of pixel-wise averages for feasible solutions, even if they may appear overly smooth and lack perceptual quality. Low-level pixel-wise error metrics are suggested to be replaced

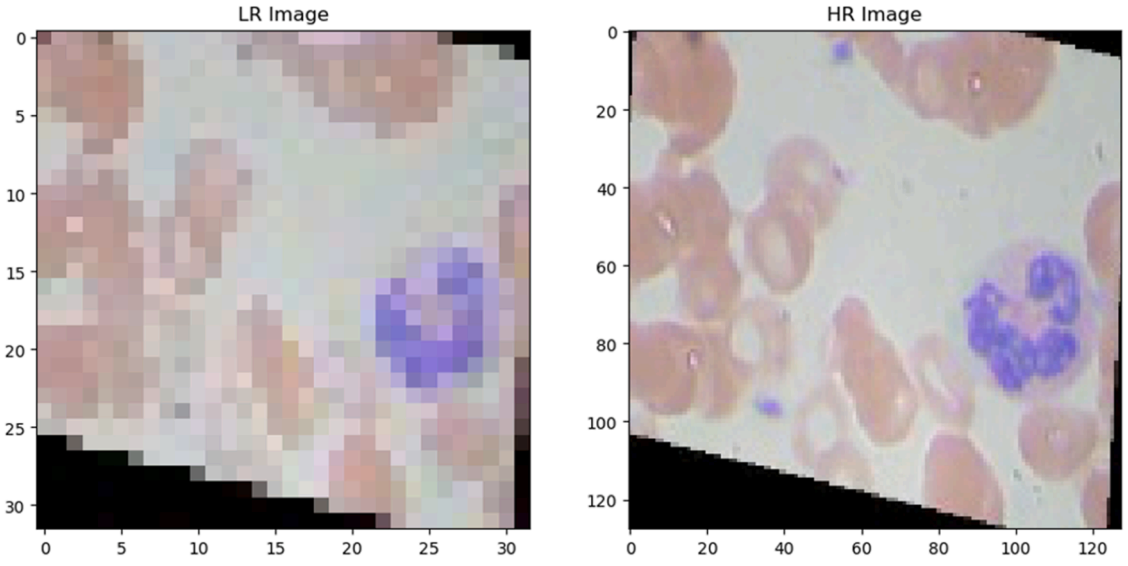


Fig. 8. Low-Resolution (LR) images and High-Resolution (HR) images.

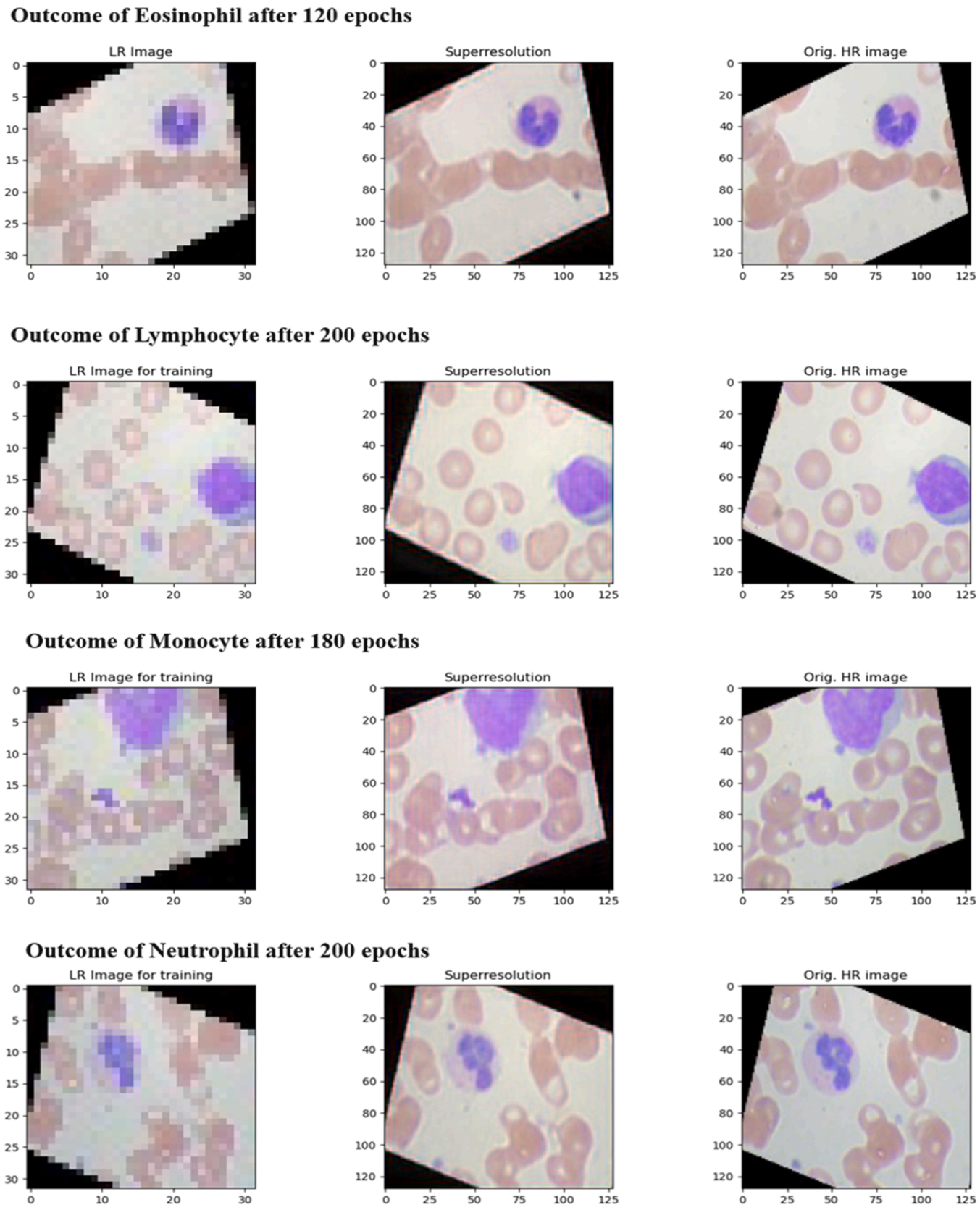


Fig. 9. The cell images at various epoch counts.

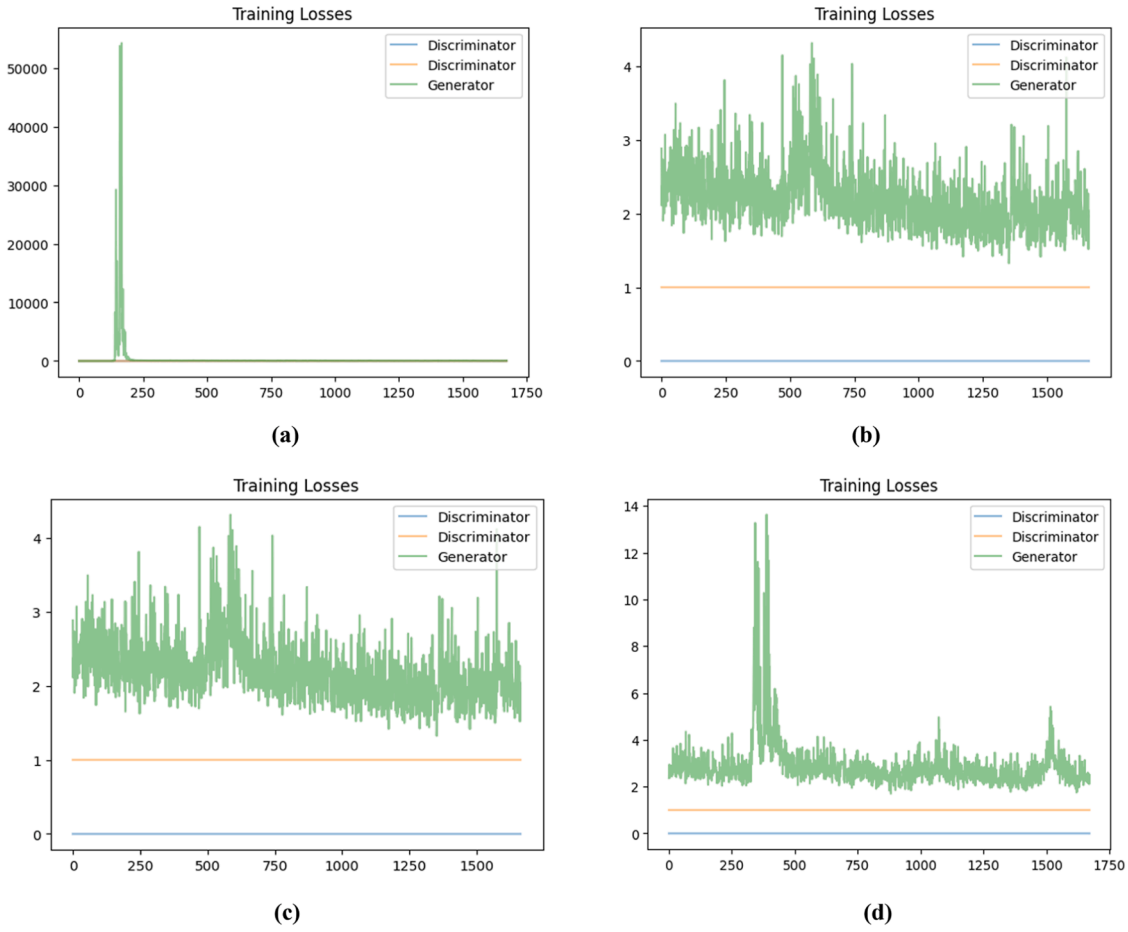


Fig. 10. Loss curves in generator and discriminator for (a) Eosinophil, (b) Lymphocyte, (c) Monocyte, (d) Neutrophil.

by features obtained from a pre-trained VGGNet [24]. The authors created a loss function by measuring the Euclidean distance between feature maps produced by the VGG19 [4] network. More intriguing outcomes in super-resolution and creative style transfer were obtained with this method [30]. The formula for the loss function is the weighted total of a content loss, (I_x^{SR}) and an adversarial loss component. Both the discriminator and generator utilize the adversarial loss during the training phase. Both the adversarial loss and content loss affect how quickly the generator converges and how effectively it produces more accurate data points [25]. The loss function can be written as:

$$I^{SR} = I_x^{SR} + 10^{-3} I_{gen}^{SR} \tag{4}$$

Here, I_x^{SR} refers to the content loss, while I_{gen}^{SR} denotes the adversarial loss.

3.2.1. Content loss

The general form of MSE loss for each pixel is given by:

$$I_{mse}^{SR} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} \left(I_{x,y}^{HR} - G_{\theta_G}\left(I^{LR}\right)_{x,y} \right)^2 \tag{5}$$

The majority of state-of-the-art approaches for SR rely on this equation as their primary optimization objective [21,25].

The loss function applied in this study is based on the loss established by Bruna et al. [29] and Gatys et al. [30], as well as the ReLU activation function of the pre-trained VGG19 network specified in [4]. In the VGG19 network, the feature map has been denoted as acquiring the j^{th} convolution (after activation) before the i^{th} max pooling layer indicated by the notation $\varnothing_{i,j}$. The Euclidean distance between an image reconstruction's feature representation $G_{\theta_G}\left(I^{LR}\right)$ and the reference image I^{HR} is then used to define the VGG loss:

$$I_{vgg/i,j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left(\varnothing_{i,j}\left(I^{HR}\right)_{x,y} - \varnothing_{i,j}\left(G_{\theta_G}\left(I^{LR}\right)\right)_{x,y} \right)^2 \tag{6}$$

Here, $W_{i,j}$ and $H_{i,j}$ are the dimensions of the feature maps of the VGG network [11].

3.2.2. Adversarial loss

The adversarial loss, serving as a loss function for the discriminator network in the GAN design, is crucial to the architecture. During HR image reconstruction, even a small amount of image noise can significantly impact the outcome. The adversarial component aids in enhancing the discriminator's ability to correctly identify the data source [25]. The generative loss, denoted as I_{gen}^{SR} is defined using the discriminator probabilities $D_{\theta_D} G_{\theta_G}$ of all training samples.

$$I_{gen}^{SR} = \sum_{n=1}^{N} -\log\left(D_{\theta_D} G_{\theta_G}\left(I^{LR}\right)\right) \tag{7}$$

The probability of the reconstructed image G_{θ_G} being a natural HR image is given here as $D_{\theta_D} G_{\theta_G}$. Instead of minimizing log [1-$D_{\theta_D} G_{\theta_G}$], -log [$D_{\theta_D} G_{\theta_G}$] is minimized to improve gradient behavior.

3.3. VGG19

Simonyan and Zisserman [26] presented the VGG19 architecture, which was trained on the ImageNet [27] database containing one million photos across 1000 categories. The feature extraction layers are

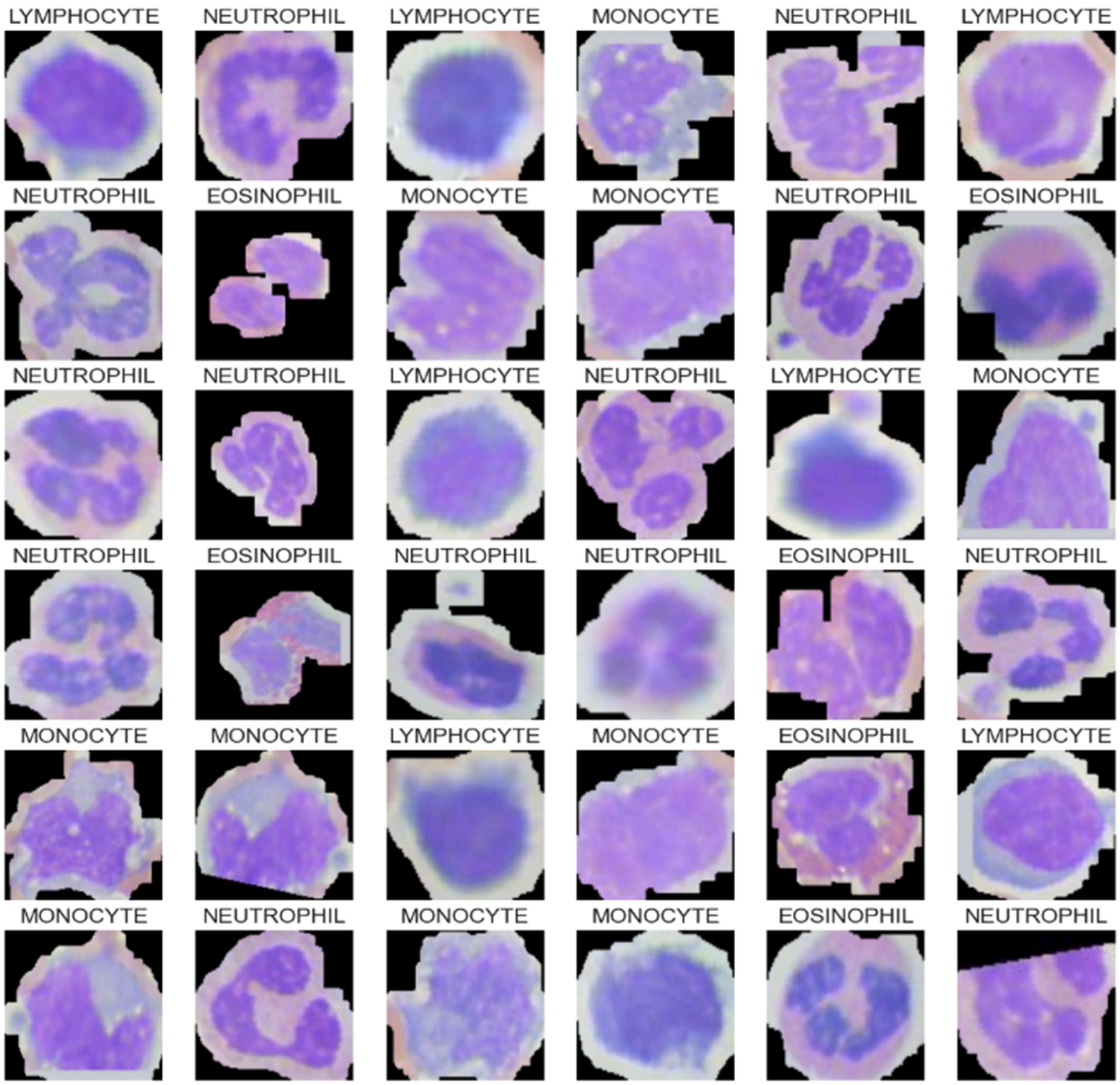


Fig. 11. Preprocessed images for VGG19 (only nucleus is extracted).

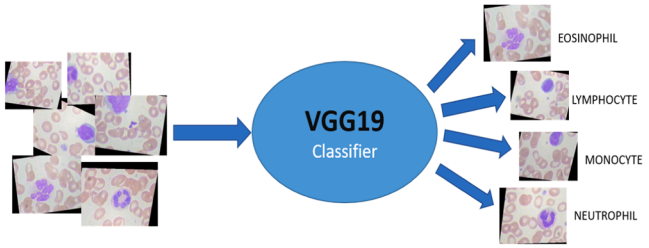


Fig. 12. Classification process of the images using VGG19.

organized into five groups, each followed by a maximum pooling layer. In Fig. 5, full architecture of VGG19 classifier is displayed. For classification problems, various machine learning algorithms are employed, including locality preserving projection. In our study, we utilize the pretrained VGG19 architecture to classify WBCs. Dimensionality reduction techniques [28] are applied within the structure to reduce the vector shape, thereby enhancing the robustness of the learning process.

4. Dataset description

The freely accessible Blood Cell Count and Detection (BCCD) dataset [29], which includes 12,447 Joint Photographic Experts Group (jpeg) formatted images of improved blood cells and cell type labels in CSV format, has been used in this work. The collection includes images of eosinophils, lymphocytes, monocytes, and neutrophils—four different kinds of cells. Fig. 6 displays the various cell image types. The training dataset contains 2497 eosinophils, 2483 lymphocytes, 2478 monocytes, and 2499 neutrophils images. There were 620 monocytes, 623 lymphocytes, 623 eosinophils, and 624 neutrophils for the validation and testing sets. They are collected from 410 blood images, each identified with an annotation box, including the WBCs. For the SRGAN model, images are divided into two types of images: HR images and LR images. LR images are taken into 32×32 pixels and HR images are resized into 128×128 shape in RGB format and converted to tensor. Samples of LR image and HR image are displayed in Fig. 7. In our collection, the quantity of pictures corresponding to various kinds of blood cells is fairly balanced. In this study, we have employed techniques for augmentation with the following parameters: a 15-degree rotation, 0.1-range width and height changes, 0.2 shearing factor, 0.2 zoom range, and horizontal flipping. This provides diversity in the model and make sure the model learns from a variety of photos. By doing this, the model's capacity to generalize across many picture changes is enhanced and biases are kept at bay.

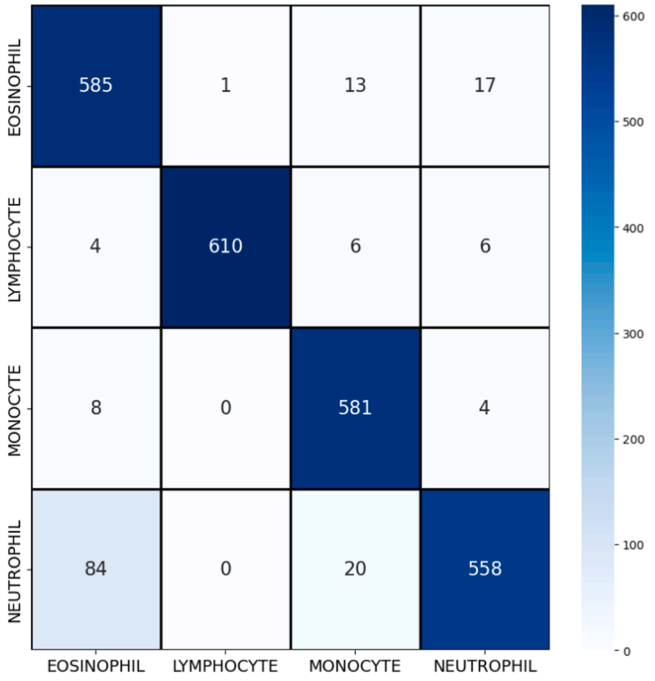


Fig. 13. Experimental results in the form of a confusion matrix.

Table 5
The result of classification precision, recall rate, F1 score and support.

Cell Type	Precision	Recall	F1-score	Support
EOSINOPHIL	0.89	0.93	0.91	616
LYMPHOCYTE	1.00	0.99	0.99	626
MONOCYTE	0.98	0.99	0.99	593
NEUTROPHIL	0.92	0.90	0.91	662
Macro Average	0.95	0.95	0.95	2497
Weighted Average	0.95	0.95	0.95	2497

5. Experiments and result analysis

5.1. Experimental setup

To conduct all the experiments for SRGAN and WBC classification, we utilized Python version 3.9 as our programming language. The experiments were executed on a device equipped with an Intel Core i7–6800k processor, 64GB of Random-Access Memory (RAM), and an NVIDIA Graphical Processing Unit (GPU) with 8GB of memory. We established a Python environment using Anaconda and employed Jupyter Notebook as our Integrated Development Environment (IDE).

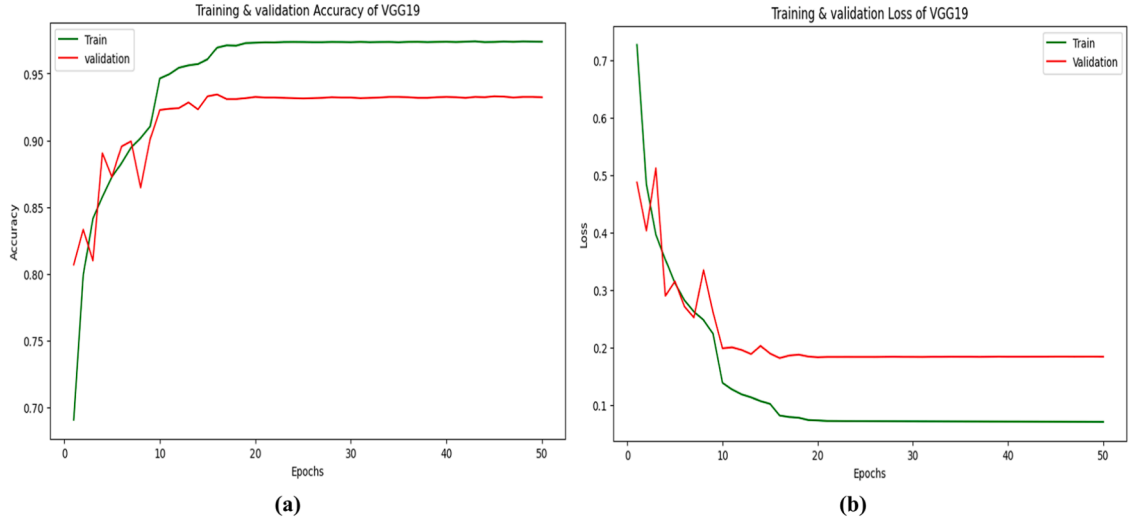
SRGAN model has been trained following the general structure as in Fig. 2. Images are preprocessed according to Section 3.2 before the training phase. The transfer learning approach is used in the process, in which ImageNet weights are utilized. These parameters are then applied to the network, and the real image dimensions are provided as input during the operation. The input size of the real images is used to initialize our model's weights of the convolutional layers. A random initializer is used for the input layer. The training data are resized into $128 \times 128 \times 3$, and divided into mini-batches for training. Generated images were fed into the classifier for the classification of cell images. For classification, we used the Adam optimizer with the loss function of Sparse Categorical Crossentropy. The callback functions were set to enrich the proposed model training and validation. The EarlyStopping() function with a minimum delta of 0.1 and the patience of 3 has been applied. The learning rate was reduced using the ReduceOnPlateau() function and the validation accuracy was also monitored. The reduction factor and patience were set to 0.2 and 2 respectively. Tables 1 and 2 provide the parameter summary of the SRGAN (generator and discriminator respectively) model. Table 3 presents the total parameters for the VGG19 model, whereas Table 4 shows the hyper-parameters.

5.2. SRGAN generated results

In this experiment, a few matrix transformations like clipping and flipping the images have been done before training the SRGAN. Primary weights are scaled with zero mean and zero standard deviation and then reloaded using the sum of the weights. All the models are trained with single images of high resolution of 128×128 pixels and low resolution of 32×32 pixel values which are displayed in Fig. 8.

SRGAN model generates 4000 images, consisting of 1000 images for each class. Based on the parameter tuning and several trial-and-error phases, we have used PReLU as the generator's activation function and Sigmoid as the discriminator. The whole model is trained for 200 epochs and after every 10 epochs, the trained model is saved to compare the outcomes for better result estimation. In Fig. 9, outcomes for different datasets in different epochs are displayed.

At the initial stage of training the discriminator loss was high, as soon as the generator improves learning, the generator loss starts to increase and the discriminator loss continues to decrease. The overall loss of



(a) (b)

Fig. 14. Training and validation accuracy versus epochs (left); and Training and validation loss versus epochs (right).

Table 6
Comparison among different pre-trained classifiers for SRGAN.

Model	Train Accuracy (%)	Val Accuracy (%)	Test Accuracy (%)	Train Loss	Val Loss	Test Loss
SRGAN-EfficientNetB7	47.65	46.47	53.103	1.161	1.18	1.0915
SRGAN-ResNet50	73.46	63.02	71.68	0.5807	0.90	0.659
SRGAN-VGG16	96.48	94.07	93.47	0.1011	0.1624	0.175
SRGAN-VGG19 (Proposed)	**97.41**	**93.22**	**94.87**	**0.0709**	**0.1842**	**0.1192**

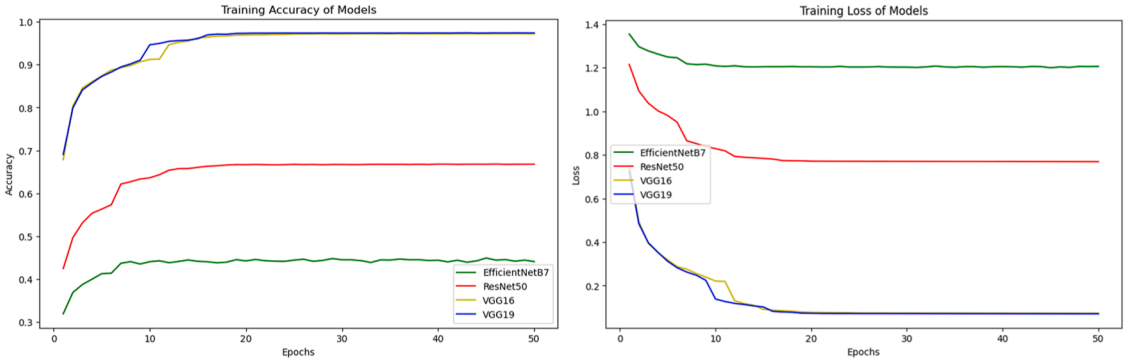


Fig. 15. (a) Training accuracy and (b) Training loss of the models.

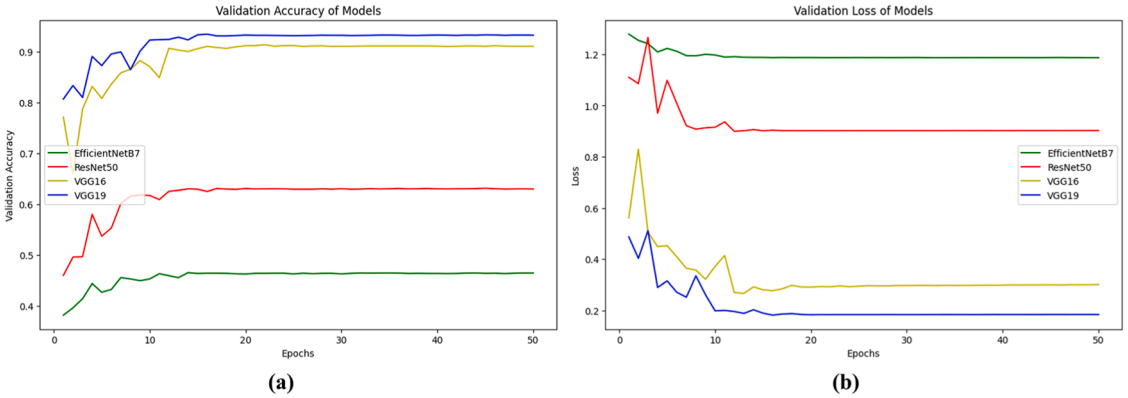


Fig. 16. (a) Validation accuracy and (b) Validation Loss of models.

Table 7
Comparative analysis with related works for WBC classifications.

Base Model	Author	Sub Models	Accuracy (%)
CNN	C. Szegedy et al. [32]	Inception V3	84.08
	K. Simonyan et al. [26]	VGG16	93.47
	T. Akiba et al. [30].	ResNet50	71.68
	T. Akiba et al. [30]	Inception-ResNet	87.02
	M. Tan et al. [31]	EfficientNetB7	53.103
CNN-RNN	G. Liang et al. [25]	Inception V3-LSTM	87.45
		ResNet50-LSTM	89.38
		Xception -ResNet50-LSTM	88.58
		Xception-LSTM	90.79
GAN—CNN	L. Ma et al. [3]	DCGAN-ResNet	91.68
	Proposed	**SRGAN-VGG19**	**94.87**

discriminator and generator of each of the subtypes are given below in Fig. 10. Here in this graph, the lower blue line of the discriminator loss indicates the classification loss of the real HR images, whereas the orange one indicates the loss of fake images which distinguish generated images from the real ones. Green line is the indication of generator loss which includes the content and adversarial losses.

5.3. *Image preprocessing before classification*

The nucleus of the generated image is preprocessed into a 128×128×3 shape and fed into a pretrained classifier to classify and evaluate the performance of the SRGAN model. Fig. 11 shows some batch examples of processed images. The images are processed individually maintaining a pipeline using image resizing, edge detection, and contour extraction. Then a bounding box is iteratively generated around the detected contours. The blood cell region from the original image is extracted using the coordinates of the bounding box and the mask.

5.4. *Classification results*

In this study, VGG19 architecture is selected as a classifier after simulating different state-of-the-art models. The main dataset is divided three parts. We have kept 80 %, 10 %, and 10 % for train, validation, and test respectively. The SRGAN model used the train dataset to generate

high-resolution images. The generated images and original images are fed into the VGG19 classifier for the training process. Then original images were utilized to assess the model performance. Fig. 12 represents the classification process of WBC images into 4 classes. The entire model showed good performance after 50 epochs, the test loss was at its lowest point, and the matching test accuracy was 0.9619. A common structure for evaluating accuracy is a confusion matrix, often called an error matrix. A 4×4 matrix (as for 4 classes) is used to represent the result of the white blood cell categorization model. Fig. 13 displays the experimental findings in a confusion matrix.

Four categories were created using the information gleaned from the confusion matrix: True Positive (TP), False Negative (FN), True Negative (TN), and False Positive (FP). It is evident from Fig. 13, that most of the cases model has the lowest false predictions. Various performance metrics were calculated using these values:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} \tag{11}$$

In Table 5, class-wise evaluation metric scores like precision, recall, and f1-score are shown. It is visible that lymphocyte has the highest precision score of 100 %. Among the 4 classes, eosinophils have the lowest precision score of 89 %. This result is based on the test dataset. The support column shows the sample numbers or images present in the test data on each class. The training and test accuracy are plotted against the number of training epochs on the left of Fig. 14, and the training and test loss are plotted against the same variables on the right of Fig. 14.

Initially, we have chosen some pre-trained classifiers to compare the performance of SRGAN. All the pretrained models were extensively trained on ImageNet [27] dataset. Our proposed model shows the best accuracy among them. We have done this comparison with the following models – VGG16 [26], ResNet50 [30], and EfficientNetB7 [31]. The accuracy and loss values for training, validation, and test datasets are shown in Table 6. It is fair to say that, our method outperforms other models in terms of classifying WBCs. SRGAN-VGG19 gets 94.87 % accuracy, whereas EfficientNetB7, ResNet50, and VGG16 have an accuracy score of 53.10 %, 71.68 %, and 93.47 % respectively. VGG16 model closely competes with our SRGAN-VGG19 structure. All four methods' loss and accuracy curves are shown in Fig. 15 and in Fig. 16. It is also clear from these figures that the proposed model can predict correctly better way than the others. Hyperparameters have been tuned several times and the outcome is compared with different experiments.

After getting the most suitable combination (SRGAN-VGG19), the performance of our proposed one is compared with some recent related models such as CNN, CNN-RNN, and GAN, which are shown in Table 7. From this table, it can be concluded that the proposed SRGAN-VGG19 model outperforms the state-of-the-art models in this research area. When the VGG19 is trained with SR images and HR images it gives a classification accuracy of 94.87 %. While using a single network to train the model may produce better results, it is challenging to further enhance the network's functionality. Our model's classification accuracy is constantly increasing, demonstrating the correctness of the entire framework and the robustness of the architecture. By employing the SRGAN, which produced additional SR images of blood cells to balance the quantity of images of various cell types, it is possible to make up for the shortage of data samples. The issue of low-resolution images is also effectively addressed by the proposed method.

While there are many other types of cells and five categories are typically used in clinical practice, there are only four categories for WBCs in this paper. The classification performs poorly when attempting to identify all cells and has some restrictions. As a result, more photos of various blood cells types are required to build a more complete database. The technological process of identifying WBC pictures under the microscope still presents difficulties, such as cell aberrations and cell maturity stages. This study initializes network architectural parameters using transfer learning, which necessitates an additional learning layer and further parameter tuning.

6. Conclusion

The technological process of identifying WBC pictures under the microscope still presents difficulties, such as cell aberrations and cell maturity stages. Therefore, classifiers perform poorly when attempting to identify all cells. This study proposed a new framework for classifying images of WBCs using an SRGAN and VGG19. The proposed strategy delivers a higher level of classification accuracy for WBC images than current pretrained model classifiers. Moving forward, we intend to upgrade the existing network architecture, develop a system for automatically generating high-resolution pictures, removing noise, and advancing the classification procedure. Explanation in medical applications is an important aspect. In our future work, we intend to implement explainable AI with our study. This will ensure that the deep learning models are learning from important features. For future research work, there can be two possible directions. The first will be to upgrade this generative model for more robust and state of the art performance with higher accuracy which will be able to deal with other medical image datasets like MRI images for Alzheimer's disease, X-ray images for osteoporosis etc. simultaneously. Then to explore the measurements to reduce the computational value and parameters to make this model able to boot in an electronic device for the faster diagnosis.

CRediT authorship contribution statement

Jannatul Ferdousi: Writing – original draft, Visualization, Software, Methodology, Investigation, Conceptualization. **Soyabul Islam Lincoln:** Writing – original draft, Validation, Software, Investigation. **Md. Khorshed Alom:** Writing – review & editing, Supervision, Project administration. **Md. Foysal:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Data availability

Data will be made available on request.

Acknowledgment

The authors are thankful to Aladdin Persson for giving the idea of SRGAN from scratch and making the concept easier for implementation.

References

[1] S. Nazlibilek, D. Karacor, T. Ercan, M.H. Sazli, O. Kalender, Y. Ege, Automatic segmentation, counting, size determination and classification of white blood cells, Measurement 55 (Sep. 2014) 58–65.

[2] P.R. Tabrizi, S.H. Rezatofighi, M.J. Yazdanpanah, Using PCA and LVQ neural network for automatic recognition of five types of white blood cells, in: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, IEEE, Aug. 2010, pp. 5593–5596.
[3] L. Ma, R. Shuai, X. Ran, W. Liu, C. Ye, Combining DC-GAN with ResNet for blood cell image classification, Med. Biol. Eng. Comput. 58 (6) (Jun. 2020) 1251–1264.
[4] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A.A. Bharath, Generative adversarial networks: an overview, IEEE Signal Process. Mag. 35 (1) (Jan. 2018) 53–65.
[5] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017.
[6] A. Spurr, E. Aksan, and O. Hilliges, "Guiding InfoGAN with Semi-supervision," in *Machine Learning and Knowledge Discovery in Databases*, vol. 10534, M. Ceci, J. Hollmén, L. Todorovski, C. Vens, and S. Džeroski, in Lecture Notes in Computer Science, vol. 10534., Cham: Springer International Publishing, 2017, pp. 119–134.
[7] K. AL-Dulaimi, J. Banks, V. Chandran, I. Tomeo-Reyes, and K. Nguyen, "Classification of white blood cell types from microscope images: techniques and challenges," pp. 17–25, 2018.
[8] M. Lippeveld, et al., Classification of human white blood cells using machine learning for stain-free imaging flow Cytometry, Cytometry A 97 (3) (Mar. 2020) 308–319.
[9] N.Z. Supardi, M.Y. Mashor, N.H. Harun, F.A. Bakri, R. Hassan, Classification of blasts in acute leukemia blood samples using k-nearest neighbour. 2012 IEEE 8th International Colloquium On Signal Processing and Its Applications, IEEE, Mar. 2012, pp. 461–465.
[10] J. M.D., K. A.H., and S. S., "White blood cells segmentation and classification to detect acute leukemia.," vol. 2, no. 3, pp. 147–151, 2013.
[11] C. N. and S. S., "Analysis of blood samples for counting leukemia cells using support vector machine and nearest neighbour," vol. 16, no. 5, pp. 79–87, 2014.
[12] N. Sinha, A.G. Ramakrishnan, Automation of differential blood count, in: TENCON2003. Conference on Convergent Technologies for Asia-Pacific Region, Bangalore, India, Allied Publishers Pvt. Ltd, 2003, pp. 547–551.
[13] Madhumala Ghosh, et al., Statistical pattern analysis of white blood cell nuclei morphometry, in: 2010 IEEE Students Technology Symposium (TechSym), Kharagpur, India, IEEE, Apr. 2010, pp. 59–66.
[14] J. Prinyakupt, C. Pluempitiwiriyawej, Segmentation of white blood cells and comparison of cell morphology by linear and naïve Bayes classifiers, Biomed. Eng. OnLine 14 (1) (Dec. 2015) 63.
[15] M.-C. Su, C.-Y. Cheng, P.-C. Wang, A neural-network-based approach to white blood cell classification, Sci. World J. 2014 (2014) 1–9.
[16] J. Zhao, M. Zhang, Z. Zhou, J. Chu, F. Cao, Automatic detection and classification of leukocytes using convolutional neural networks, Med. Biol. Eng. Comput. 55 (8) (Aug. 2017) 1287–1301.
[17] F. Yellin, S. Roth, R. Vidal, Multi-cell detection and classification using a generative convolutional model, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8953–8961.
[18] J.L. Wang, A.Y. Li, M. Huang, A.K. Ibrahim, H. Zhuang, A.M. Ali, Classification of white blood cells with PatternNet-fused ensemble of convolutional neural networks (PECNN), in: 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Louisville, KY, USA, IEEE, Dec. 2018, pp. 325–330.
[19] S.A. Tarimo, M.-A. Jang, E.E. Ngasa, H.B. Shin, H. Shin, J. Woo, WBC YOLO-ViT: 2 Way - 2 stage white blood cell detection and classification with a combination of YOLOv5 and vision transformer, Comput. Biol. Med. 169 (Feb. 2024) 107875.
[20] W. Li, Y.M. Tang, K.M. Yu, S. To, SLC-GAN: an automated myocardial infarction detection model based on generative adversarial networks and convolutional neural networks with single-lead electrocardiogram synthesis, Inf. Sci. 589 (Apr. 2022) 738–750.
[21] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (Feb. 2016) 295–307.
[22] W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network.," presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1874–1883.
[23] L. Xingchen, J. Juncheng, Z. Li, and H. Qinhan, "Feature concentration network for image super-resolution," 2020. [Online]. Available: http://kns.cnki.net/kcms/detail/11.2127.tp.20200820.1009.014.html.
[24] J. Bruna, P. Sprechmann, and Y. LeCun, "Super-resolution with deep convolutional sufficient statistics," 2015, doi: 10.48550/ARXIV.1511.05666.
[25] G. Liang, H. Hong, W. Xie, L. Zheng, Combining convolutional neural network with recursive neural network for blood cell image classification, IEEE Access 6 (2018) 36188–36197.
[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Apr. 10, 2015, *arXiv*: arXiv:1409.1556. Accessed: May 02, 2023. [Online]. Available: http://arxiv.org/abs/1409.1556.
[27] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, Li Fei-Fei, ImageNet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, IEEE, Jun. 2009, pp. 248–255.
[28] M. Bansal, M. Kumar, M. Sachdeva, A. Mittal, Transfer learning for image classification using VGG19: caltech-101 image data set, J. Ambient Intell. Humaniz. Comput. 14 (4) (Apr. 2023) 3609–3620.
[29] "WBC dataset." [Online]. Available: https://www.kaggle.com/datasets/paultimothymooney/blood-cells.
[30] T. Akiba, S. Suzuki, and K. Fukuda, "Extremely Large Minibatch SGD: training ResNet-50 on ImageNet in 15 Min," Nov. 12, 2017, *arXiv*: arXiv:1711.04325. Accessed: May 01, 2023. [Online]. Available: http://arxiv.org/abs/1711.04325.
[31] M. Tan, Q.V. Le, Efficientnet: rethinking model scaling for convolutional neural networks, in: PMLR 97, 2019, pp. 6105–6114.
[32] 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 (CVPR), 2016, pp. 2818–2826.