Computer Methods and Programs in Biomedicine 224 (2022) 107019

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

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

Enhancing classification of cells procured from bone marrow aspirate smears using generative adversarial networks and sequential convolutional neural network

Debapriya Hazra [a], Yung-Cheol Byun [a,*], Woo Jin Kim [b]

[a] Department of Computer Engineering, Jeju National University, Jeju 63243, South Korea
[b] Department of Laboratory Medicine, EONE Laboratories, Incheon 22014, South Korea

ARTICLE INFO

Article history:
Received 1 February 2022
Revised 13 June 2022
Accepted 9 July 2022

Keywords:
MSC41A05
41A10
65D05
65D17
Generative adversarial networks
Bone marrow aspirate smears
Cell classification
Convolutional neural network

ABSTRACT

Background and Objective: Leukemia represents 30% of all pediatric cancers and is considered the most common malignancy affecting adults and children. Cell differential count obtained from bone marrow aspirate smears is crucial for diagnosing hematologic diseases. Classification of these cell types is an essential task towards analyzing the disease, but it is time-consuming and requires intensive manual intervention. While machine learning has shown excellent outcomes in automating medical diagnosis, it needs ample data to build an efficient model for real-world tasks. This paper aims to generate synthetic data to enhance the classification accuracy of cells obtained from bone marrow aspirate smears. Methods: A three-stage architecture has been proposed. We first collaborate with experts from the medical domain to prepare a dataset that consolidates microscopic cell images obtained from bone marrow aspirate smears from three different sources. The second stage involves a generative adversarial networks (GAN) model to generate synthetic microscopic cell images. We propose a GAN model consisting of three networks; generator discriminator and classifier. We train the GAN model with the loss function of Wasserstein GAN with gradient penalty (WGAN-GP). Since our GAN has an additional classifier and was trained using WGAN-GP, we named our model C-WGAN-GP. In the third stage, we propose a sequential convolutional neural network (CNN) to classify cells in the original and synthetic dataset to demonstrate how generating synthetic data and utilizing a simple sequential CNN model can enhance the accuracy of cell classification. Results: We validated the proposed C-WGAN-GP and sequential CNN model with various evaluation metrics and achieved a classification accuracy of 96.98% using the synthetic dataset. We have presented each cell type's accuracy, specificity, and sensitivity results. The sequential CNN model achieves the highest accuracy for neutrophils with an accuracy rate of 97.5%. The highest value for sensitivity and specificity are 97.1% and 97%. Our proposed GAN model achieved an inception score of 14.52 ± 0.10, significantly better than the existing GAN models. Conclusions: Using three network GAN architecture produced more realistic synthetic data than existing models. Sequential CNN model with the synthetic data achieved higher classification accuracy than the original data.

1. Introduction

Recently machine learning approaches have gained significant outcomes for feature-based analysis of medical images or data. Not only does it help for accurate decision making, but it also contributes to the prognosis of the problem and accurate diagnosis with proper solution. There has been rapid development in the field of medical image analysis and the enhancement of classification results using the generative adversarial networks (GAN) framework [1]. GAN is a machine learning framework that is widely used to generate synthetic data that replicates the probability distribution of the original data. For medical imaging, the main problem is the lack of data that led us to expand the image dataset. To overcome this lack of data problem researchers, use data augmentation. Simple changes to dataset images, such as translation, rotation, flipping, and scaling, are typical data augmentation techniques. In computer vision [2], using traditional data augmentation to assist the training process of networks is a routine technique. However, synthetic data augmentation is a high-quality approach for this purpose. Synthetic data samples learned via a generative

* Corresponding author.
E-mail address: ycb@jejunu.ac.kr (Y.-C. Byun).

https://doi.org/10.1016/j.cmpb.2022.107019
0169-2607/

model provide more diversity and enrich the dataset, allowing the system training process to be improved even more.

Generative Adversarial Networks are a potential way to train a model that synthesizes images inspired by game theory. The GAN model is made up of two networks that are trained in an adversarial process, in which one network makes (generator) fake images and the second network discriminates (discriminator) between actual and fake images simultaneously. This continuous win-loss approach to improve the performance have gained tremendous popularity in computer vision. The drawback of medical image classification is that it requires large number of images for training any classification model. Zhao et al. developed a transfer learning architecture VGG16 based on convolutional neural network and combined it with DCGAN to generate synthetic images for lung-nodules image classification. [3]. [4] proposed a model named PG-GAN (progressively grown generative adversarial networks) to generate synthesized medical images of premature retinopathy vascular pathology (ROP) and MRI images. [5] trained GAN to build lung field and heart segmentation images from chest X-ray images. Later in another work, GAN was trained on healthy retinal patches to learn healthy tissue data distribution. The GAN model was then evaluated for anomaly detection in retinal pictures on patches of both undetected healthy and anomalous data [6]. Crime prevention relies heavily on the footage captured by CCTV cameras. GAN architecture can also be used to capture the complexity of sizeable natural video datasets [7]. Various GAN models were introduced to answer the problem of raw video modeling. UCF-101 and Kinetics-600 datasets, both of which feature high-quality video and diversity, have been used to demonstrate the capability of GAN in [8]. The authors have used semi-supervised multi-discriminator GANs (MDGANs) to evaluate if averaging the outcomes of many discriminators may improve judgment ability of GAN algorithms for spatial-spectral data.

There have been lots of research going on for skin lesion classification based on pathological images. GANs were utilized to create highly realistic skin lesions images. The study has been focused on the generation of images and synthesis of natural images, but it has yet to be shown how much performance increase can be realized by supplementing the training data with synthetic images. The data provided by GAN was used to improve performance on a liver lesion classification challenge [9]. As a result, their study is one of the first to officially measure performance gains using GAN-based augmentations on a skin lesion classification task. The authors in [10] presented deeply discriminated generative adversarial networks (DDGAN) to boost the resolution of generated images. [11] used the ISIC 2018 challenge dataset to identify seven skin lesion categories. The authors used GAN models to augment data from skin lesion images. In a similar work, the discriminator of the GAN were proposed to serve as the final classifier [12]. Currently, many research has been going on for Covid-19 detection using machine learning techniques. Since the pandemic is recent, there are only limited data available for Covid-19 classification. Therefore, the authors developed CovidGAN for artificially generating training data that could be used for further processing using CNN [13]. GANs can create images from scratch in any category, and when combined with other methods, they can give satisfactory results. Nalla et al. proposed YOLOv3 object detection technique for white blood cell detection and classification [14]. They achieved 90% accuracy on classifying four cell types:eosinophil, lymphocyte, monocyte and neutrophil.The authors used blood cell count detection (BCCD) dataset to evaluate their proposed approach. Ahmed et al. proposed classification of white blood cell leukemia through a hybrid approach [15]. The author proposed using VGGNet for feature extraction and then using statistically enhanced Salp Swarm Algorithm (SESSA) for filtering features and enhancing classification. The proposed work was evaluated on two datasets. The first dataset was obtained from Department of Information Technology - Universit degli Studi di Milano for which the proposed work acheived 96.11% accuracy. The dataset contained two classes benign and malignant. The second dataset was C-NMC dataset which also had two classes benign and malignant. For C-NMC, the proposed work achieved 83.3% accuracy [16]. proposed a multi-level convolutional neural network (CNN) approach for white blood cell classification. The proposed model uses a faster-RCNN network and two parallel CNN with the MobileNet structure to classify four cell types: lymphocyte, monocyte, segmented neutrophil and eosinophil. They used five datasets: BCD dataset, CBC dataset, WBC dataset, KBC dataset and LISC dataset. They achieved an average accuracy of 98.36%.

This work incorporates the advantages of generative adversarial networks for generating synthetic data for cell types with fewer images. We form a balanced dataset that can be used for enhancing cell classification accuracy from microscopic images. The main contribution of this work is as follows:

- We collaborated with medical experts to form a dataset that integrates three different sources of microscopic cell images obtained from bone marrow aspirate smears.
- We propose a three network GAN architecture, C-WGAN-GP, consisting of a generator, discriminator, and a classifier, to generate synthetic microscopic cell images that can be categorized into a specific cell type.
- We propose a sequential convolutional neural network (CNN) to classify cell types from microscopic images. The proposed CNN model uses the balanced dataset formed through C-WGAN-GP and enhances classification accuracy.

2. Methods

2.1. Dataset

For this work, we have collected microscopic cell images from three different sources and combined them to form a single dataset. We collected data from EONE Laboratories of South Korea [17]. We received images of seventeen cell types obtained from bone marrow aspirate smears. The total number of microscopic cell images in the dataset, including patients with and without any blood diseases, is 12,756.

The next dataset was produced by Munich University Hospital and was made available at the cancer imaging archive [18]. The dataset contains 10,300 individual cell images annotated by experts and collected from 200 patients 100 patients with blood diseases that are not malignant and 100 patients with acute myeloid leukemia. Cell types present in the dataset are monocyte, basophil, myeloblast, eosinophil, erythroblast, lymphocyte atypical, lymphocyte typical, metamyelocyte, monoblast, neutrophil band, neutrophil segmented, smudge cells, promyelocyte bilobed, myelocyte, and promyelocyte.

Our third dataset is from Mendeley data [19] containing 10,121 microscopic peripheral individual blood cell images. The dataset was prepared in the hospital clinic of Core Laboratory in Barcelona through a CellaVision DM96 analyzer. The eight kinds of cell types present in the dataset are basophil, immature granulocytes, neutrophil, eosinophil, monocytes, erythroblasts, platelets, and lymphocytes. Size of the images are 360×360. The cell images were obtained from people free from drug consumption as well as from any infectious, hematologic, or oncologic disease. Fig. 1, shows samples from each cell type present in the combined dataset.

2.2. Methodology

This section describes the overall flow and architecture of our proposed work. Our main goal is to enhance the classification of

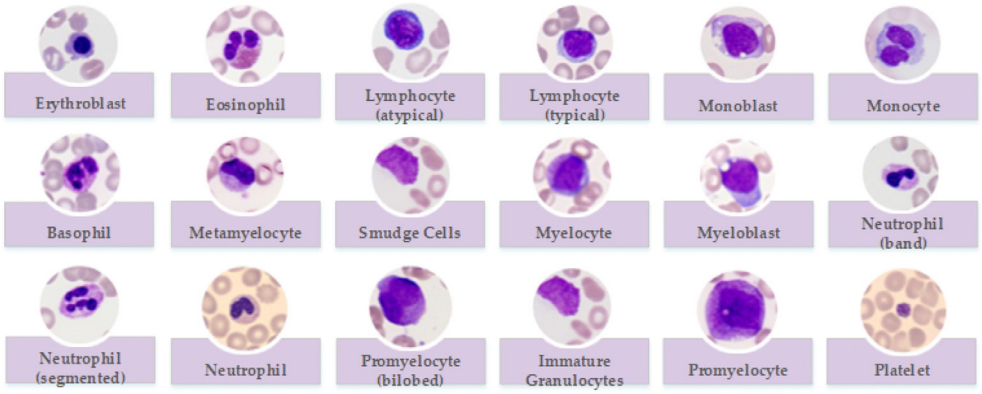

Fig. 1. Examples of types of cells obtained from bone marrow aspirate smears.

cells obtained from bone marrow aspirate smears by oversampling images for cell types containing fewer data and by developing a simple and efficient classification model. We divide our work into three stages as mentioned below:

- Stage 1- We collect data from three different sources and combine them to form a single dataset. In this stage, we take help from experts in this field to filter data, group images belonging to the same cell type, and eliminate redundancy in the dataset.
- Stage 2- In stage 2, we propose a GAN architecture consisting of three networks, namely generator, discriminator, and classifier, to eliminate the data imbalance problem. We generate synthetic images for cell types containing fewer data. With this GAN architecture, we create a whole new balanced dataset.
- Stage 3- We implement a sequential convolutional neural network to classify the cell types in the synthetic and the original dataset.

We describe the overall architecture of our proposed work in Fig. 2. The first stage is referred to as the data preprocessing stage. The second stage is the generative adversarial networks stage. We propose the C-WGAN-GP model that takes the classifier concept of auxiliary classifier GAN (AC-GAN) and inherits the architecture of WGAN-GP. Instead of implementing the classifier on top of the discriminator as in AC-GAN, we build a classifier separately and therefore name the model as C-WGAN-GP. The third stage is named on the model we use for classifying individual cell types from the images generated from C-WGAN-GP.

2.2.1. Data preprocessing

The dataset collected from three sources was first combined with the help of experts from EONE Laboratories. Before preprocessing, we had nineteen types of cells, namely myelocyte, basophil, erythroblast, immature granulocytes, eosinophil, lymphocyte, lymphocyte atypical, lymphocyte typical, monoblast, monocyte, metamyelocyte, myeloblast, neutrophil, neutrophil segmented, neutrophil band, platelet, promyelocyte, promyelocyte bilobed and smudge cells. Table 1 mentions the count of images per cell type.

The next task in the data preprocessing stage was to eliminate redundancy or remove duplicate data. By redundant data, we mean similar images or duplicate images. The images were filtered with the help of experts. After combining the dataset, redundancy removal, and filtering, we grouped cell types belonging to similar cell families to a single group or cell type. E.g., immature granulocyte, myelocyte, and metamyelocyte were combined to form a single cell type, i.e., immature granulocyte. Images from lymphocyte, lymphocyte atypical, and lymphocyte typical were combined to form a single lymphocyte group. Cell type promyelocyte was formed by combining promyelocyte and promyelocyte bilobed. Neutrophil was formed by merging images from neutrophil, neutrophil band, and neutrophil segmented. Subgrouping of the cell types were done by medical expert based on maturation sequence of human hematopoietic cells. After merging, the final count for twelve cell types is mentioned in Table 2.

Table 1
Distribution of total number of images per cell type before preprocessing.

Types of Cells	Count of Images
Eosinophil	3538
Erythroblast	1547
Basophil	1224
Lymphocyte	1213
Lymphocyte Atypical	7
Lymphocyte Typical	3818
Immature Granulocytes	2881
Monocyte	2583
Myeloblast	3104
Metamyelocyte	13
Monoblast	26
Myelocyte	39
Neutrophil	3316
Neutrophil Band	82
Neutrophil Segmented	7346
Promyelocyte	69
Platelet	2339
Promyelocyte Bilobed	17
Smudge Cells	15

Table 2
Distribution of total number of images per cell type after preprocessing.

Types of Cells	Count of Images
Eosinophil	3538
Erythroblast	1547
Basophil	1224
Lymphocyte	5038
Immature Granulocytes	2933
Monocyte	2583
Myeloblast	3104
Monoblast	26
Neutrophil	10,743
Promyelocyte	87
Platelet	2339
Smudge Cells	15

2.2.2. Generative adversarial networks (C-WGAN-GP)

Generative models have been widely used to produce samples by learning the probability distribution of a given latent variable. Ian Goodfellow et al. introduced generative adversarial networks (GAN) in the year 2014 [1]. GAN introduces a competitive and con-

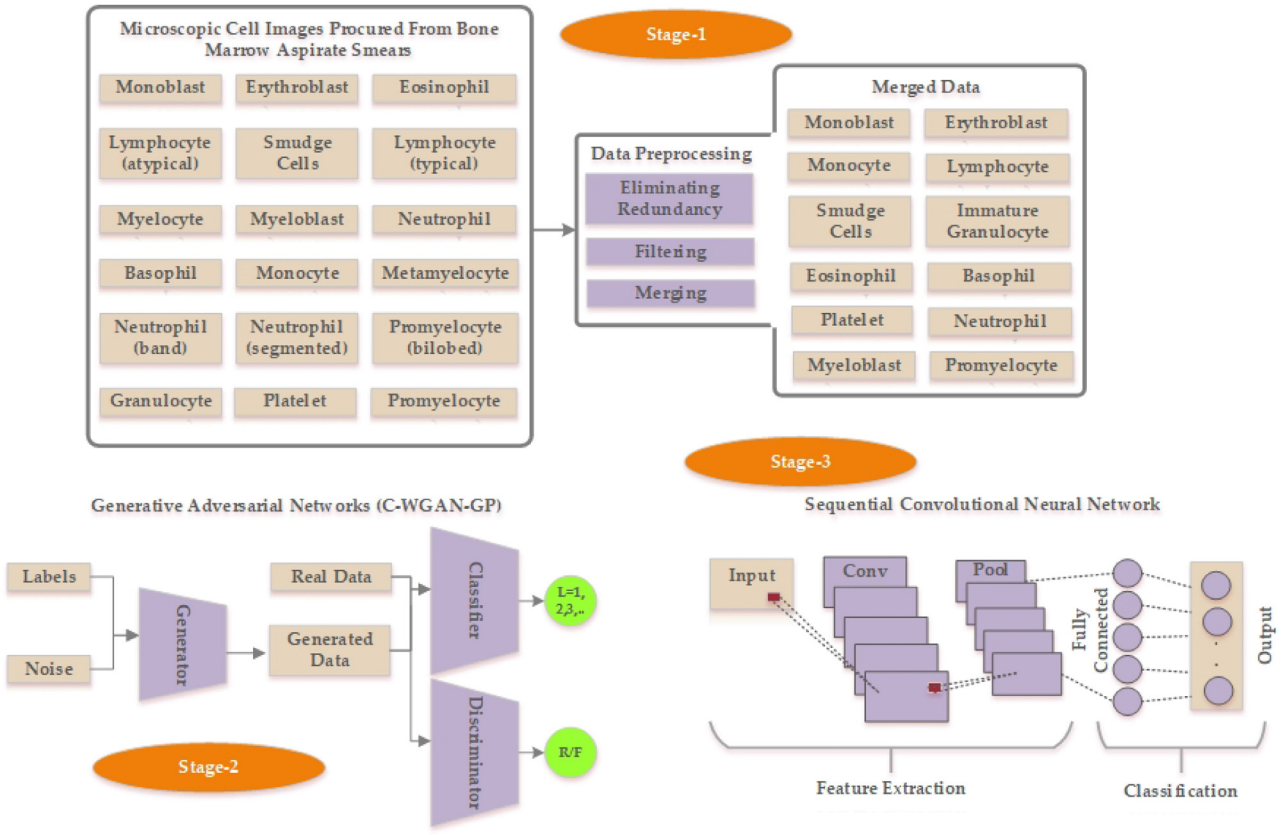

Fig. 2. Overview of the proposed methodology. Stage 1- data preprocessing and merging. Stage 2- classifier based generative adversarial networks (C-WGAN-GP) to oversample minority data. Stage 3- classification of cell types obtained from bone marrow aspirate smears through sequential convolutional neural network.

tinuous learning process through a two-network architecture, i.e., a generator network G and a discriminator network D. The task of the generator is to generate synthetic samples from randomly initialized variables, and the job of the discriminator is to distinguish between real and generated samples successfully. The generator tries to deceive the discriminator by generating realistic samples that cannot be distinguished from the real samples, and the discriminator tries to distinguish between original and synthetic samples correctly. Random noise variable (z) that preserves data distribution P_r over original distribution (x) is provided as input to the generator. The synthetic samples produced by the generator are then provided as input to the discriminator along with the original samples. The generator and discriminator take part in the zero-sum game where the generator tries to minimize $log(1 - D(G(z)))$, and the minmax objective is defined as:

$$\min_G \max_D f(D, G) = \mathbb{E}_{x\sim P_r}[logD(x)] + \mathbb{E}_{G(z)\sim P_s}[log(1 - D(G(z)))] \quad (1)$$

where, $P(s)$ is the distribution of generated data defined by $G(z).z \sim p(z)$. Initial GAN architecture or Vanilla GAN architecture used Jensen-Shannon divergence, which, when tried to minimize, led to vanishing gradient problem. The training process becomes more complex and the convergence rate slower. To resolve this issue, Wasserstein GAN [20] was introduced to improve the performance of GAN in terms of the loss function. Wasserstein, also known as the earth-mover distance, has the advantage of continuity, and its goal is to minimize the value function defined in 4 by minimizing the Wasserstein distance through parameter modification of the generator under an optimal discriminator.

$$\min_G \max_{D\in L} \mathbb{E}_{x\sim P_r}[D(x)] + \mathbb{E}_{G(z)\sim P_s}[1 - D(G(z))] \quad (2)$$

where L is 1-Lipschitz function. The critic function of WGAN helps in optimizing the generator. However, to add the Lipschitz constraint to the discriminator, a weight clipping method is used, where the weights of the discriminator are clipped in a compact space $[c, -c]$ and are limited to it. Although WGAN eliminated the problem of Vanilla GAN, the weight clipping method led to optimization difficulties. Therefore to solve this issue, WGAN with gradient penalty (WGAN-GP) was introduced by [21]. WGAN-GP replaces the weight clipping by complying with the condition of 1-Lipschitz and by enforcing a penalty on the gradient. The objective of WGAN-GP can be defined as the combination of original critic loss and gradient penalty as shown below:

$$OriginalCriticLoss = \mathbb{E}_{x\sim P_r}[D(x)] - \mathbb{E}_{G(z)\sim P_s}[D(G(z))] \quad (3)$$

$$GradientPenalty = \lambda \mathbb{E}_{x_r\sim P_{x_r}}[(\|\nabla_{x_r} D(x_r)\|_2 - 1)^2] \quad (4)$$

WGAN-GP improves the training of GAN significantly; therefore, we chose WGAN-GP. However, still, the GAN architecture lacked in controlling or conditioning the generator on producing specific samples. To address this problem, conditional GAN (CGAN) was proposed by [22]. CGAN takes labels (y) as input for the generator and the discriminator to train the network on the specified condition. The minmax objective in CGAN is defined as below:

$$\min_G \max_D \mathbb{E}_{x\sim P_r}[logD(x|y)] + \mathbb{E}_{G(z)\sim P_s}[log(1 - D(G(z|y)))] \quad (5)$$

Conditional GAN has the disadvantage, where the discriminator cannot output the category of the data directly. To get the data category, the data should be provided as input by mentioning each category one by one. Auxiliary classifier GAN (AC-GAN) [23] gives the discriminator the capability to classify the data as real or fake and to output the probability of the image belonging to a specific

Table 3
Network architecture details of the generator.

Blocks	Size of Kernel	Output
Input Concatenate(z,L)	-	500+12
Fully Connected	-	64 x 32 x 32
Residual Block (2)	3 x 3	64 x 32 x 32
Deconvolution (s=2)	5 x 5	64 x 64 x 64
Residual Block (4)	3 x 3	64 x 64 x 64
Deconvolution (s=2)	5 x 5	64 x 128 x 128
Residual Block (2)	3 x 3	64 x 128 x 128
Deconvolution (s=1)	5 x 5	64 x 128 x 128

Table 4
Network architecture details of the discriminator.

Blocks	Size of Kernel	Output
	Input (x, G(z,L))	
Convolution (s=2)	5 x 5	64 x 64 x 64
Residual Block (2)	3 x 3	64 x 64 x 64
Average Pooling	2 x 2	64 x 32 x 32
Residual Block (4)	3 x 3	64 x 32 x 32
Average Pooling	2 x 2	64 x 16 x 16
Residual Block (4)	3 x 3	64 x 16 x 16
Average Pooling	2 x 2	64 x 8 x 8
Fully Connected	-	128
Fully Connected	-	1

Table 5
Network architecture details of the classifier.

Blocks	Size of Kernel	Output
Input (x, G(z,L))		
Convolution (s=2)	5 x 5	64 x 64 x 64
Residual Block (2)	3 x 3	64 x 64 x 64
Average Pooling	2 x 2	64 x 32 x 32
Residual Block (4)	3 x 3	64 x 32 x 32
Average Pooling	2 x 2	64 x 16 x 16
Residual Block (4)	3 x 3	64 x 16 x 16
Average Pooling	2 x 2	64 x 8 x 8
Fully Connected	-	128
Fully Connected	-	12

class. AC-GAN defines a classifier on top of the discriminator, so the discriminator has two tasks. One is to distinguish generated image from a real image, and the second is to predict the class labels of the image. The discriminator in AC-GAN is not provided with the class label. Due to the sharing of weight parameters in the discriminator architecture AC-GAN has a complex and limited training process.

Therefore, in our proposed C-WGAN-GP model, we develop a three-player architecture that consists of three networks, i.e., generator, discriminator, and classifier. We have separated the classifier from the discriminator, and the generator is simultaneously trained from the feedback of the classifier and the discriminator. The generator tries to fool the discriminator and simultaneously targets to generate samples that can be classified correctly by the classifier. Table 3, Table 4 and Table 5 defines the network architecture we used to built the generator, discriminator and the classifier. For the generator, we use two hidden deconvolutional layers for each residual block, and every convolutional layer uses a pre-activation function. For the activation function, we have used LeakyReLU. In the discriminator architecture, we have used double hidden convolutional layers for each residual block. Same as the generator, the convolutional layer uses a pre-activation function, with LeakyReLU for the activation function. For the classifier, too, we have used two convolutional layers for each residual block.

2.2.3. Sequential convolutional neural network

In this section, we discuss the classification model we used to classify the cell types of each image. We used this model to evaluate the classification rate for original as well as the synthetic data generated through C-WGAN-GP. We have used a sequential convolutional neural network (CNN) to classify the type of cells obtained from the bone marrow aspirate smears. Each layer in the sequential CNN connects to the previous and the following layer. CNN is defined as the type of artificial neural network (ANN) that consists of convolution to extract crucial information or features from the input image. As we can see from Fig. 3, the convolutional layer in the proposed approach is followed by normalization, LeakyReLU, and 2D max pooling. This architecture is repeated four times. In the last layer, we have added a dropout layer to prevent overfitting problems. We also include a fully connected and a softmax layer. The classification layer is used to classify the types of blood cells from the features extracted from all the previous layers.

In the proposed architecture, we have used eight layers. In the input layer, we have received image input. In the convolutional layer, we have used a 2D convolutional layer. 2D convolutional layer automatically learns and extracts features while sliding on images. The batch normalization layer is used for normalizing the activations and the propagation gradients. Batch normalization optimizes and speeds up the training. ReLU layer provides thresholding operation to every input layer. The pooling layer is used for downsampling and reducing the computational overhead. The dropout layer is used for eliminating overfitting. A fully connected layer connects all the neurons to its previous layer so that the feature extracted in each layer is intact and can be used for classification. The output layer is built with softmax and classification layers. The softmax layer provides a probability for each cell type, and the classification layer classifies the cell type in a given image.

3. Results

We divide the experiments and results section into two parts. First, we evaluate the quality of the synthetic microscopic cell images generated through the proposed C-WGAN-GP model. We compare the performance of various GAN architectures based on error rates, structural similarity, inception score, and other evaluation metrics. In the second part, we measure the performance of the proposed classification model. We present the results for classification using original data as well as synthetic data to demonstrate how oversampling through GAN enhanced the classification rate. For generating new data, we excluded the cell type neutrophil since it already contained 10,000 images. We combined the original and the synthetic data generated by C-WGAN-GP. We dedicated same number of images for each cell types. The total number of images for each cell types were 10000. Since for neutrophil we had 10,000 images. We decided to keep the same number of images in each cell type. 70% data was used for training 20% for testing and 10% for validation. We show classification performance per cell type and compare different classification models to show the increased accuracy of our proposed model. The mentioned accuracy is an average of several trials. We also present screen grabs of classification and misclassification results as well as training and validation accuracy and loss.

3.1. Generative adversarial networks

We have computed and produced the results for six GAN models on generating synthetic microscopic cell images. We compared the outcomes of proposed C-WGAN-GP model with auxiliary classifier GAN (AC-GAN) [23], conditional Wasserstein GAN with gradient penalty (CWGAN-GP) [24], information maximizing GAN (InfoGAN) [25], deep convolutional GAN (DCGAN) [26] and conditional

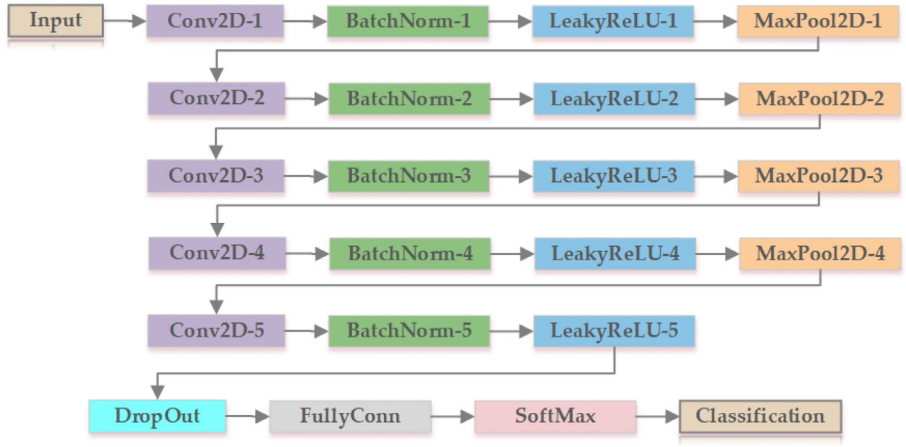

Fig. 3. Architecture of sequential convolutional neural network for classification.

Table 6
Quantitative analysis of different GAN models.

Models	FID	Precision	F1	Recall	LPIPS	IS
AC-GAN	64.36	94.33	93.42	93.67	0.25	11.32 ± 0.54
CWGAN-GP	77.57	88.15	88.10	88.67	0.38	8.23 ± 0.33
InfoGAN	75.44	90.23	88.36	89.91	0.35	9.22 ± 0.65
DCGAN	78.91	87.36	87.16	88.11	0.40	8.01 ± 0.36
CGAN	71.02	93.65	92.55	93.03	0.31	10.59 ± 0.77
C-WGAN-GP	60.33	97.10	96.16	96.99	0.23	14.52 ± 0.10

Table 7
Peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), l1 and l2 error comparison of different GAN models.

Models	PSNR	SSIM	l1 error	l2 error
AC-GAN	33.46	0.9012	13.7%	6.1%
CWGAN-GP	31.22	0.8956	13.9%	6.3%
InfoGAN	30.22	0.9288	11.6%	5.8%
DCGAN	31.42	0.9260	12.8%	6.3%
CGAN	30.22	0.9400	13.9%	6.5%
C-WGAN-GP	38.01	0.9714	8.7%	3.8%

GAN (CGAN) [22]. The evaluation metrics selected for quantitative comparison are Frchet inception distance (FID), precision, F1 score, recall, learned perceptual image patch similarity (LPIPS), and inception score. Also, we use peak signal-to-noise ratio (PSNR), structural similarity index metric (SSIM), and l1 and l2 error to evaluate the quality of the synthetic data generated through the proposed and existing GAN models.

Between the real and synthetic data, the distance of the feature vector can be measured through FID. FID helps us to find the distribution of the synthetic samples along the distribution of the original data while training the generator. The lower value of FID indicates better performance. If R_D is considered as original data and S_D as synthetic data, then FID is defined as follows:

$$F_{ID}=\|\mu_{R_D}-\mu_{S_D}\|^2+T_{R_D}(\Sigma_{R_D}+\Sigma_{S_D}-2(\Sigma_{R_D}\Sigma_{S_D})^{1/2} \quad (6)$$

Precision measures the quality of the synthetic microscopic cell images and also evaluates the GAN on the basis of how capable the classifier is in predicting the types of cells. In contrast, recall is for assessing the quantity of the synthetic images generated. The harmonic mean between precision and recall can be evaluated through the F1 score. It also contributes to the accurate measurement of a model on a specific dataset. The higher the value of F1, recall, and precision, the better the model's performance. LPIPS, on the other hand, measures the distance between the image patches, so a lower distance indicates more similarity and better results.

Synthetic data can be used for real-life problems only if it is as relevant as the original data. Inception score measures how realistic the synthetic data are as compared to the original data. The inception score can be defined as follows:

$$e(\mathbb{E}_a[K_{LD}(p(y|a)\parallel p(y))]) \quad (7)$$

where, K_{LD} refers to Kullback-Leibler divergence that computes difference between marginal $p(y)$ and probability distribution of image a, i.e. $p(y|a)$. Table 6, presents the quantitative analysis of various GAN models on our mentioned dataset.

Our experiments prove that C-WGAN-GP produces a higher value for PSNR and SSIM. Our proposed model generates better quality synthetic microscopic cell images than existing GAN models. The computation of the error rate is an essential step in judging a model's performance. We compute the loss functions l1 and l2, which is vital for evaluating the error rates. l1 measures the least absolute deviation and can be used for error evaluation that can be defined as the sum of all absolute differences between real and generated images as follows:

$$l1 = \Sigma_{i=1}^{n}|y_{R_D} - y_{S_D}| \quad (8)$$

l2 measures the squared differences between real and generated images. l2 is also an evaluation metrics to measure error rates. Table 7 shows PSNR, SSIM, l1 and l2 error values for different GAN models. Results in both the above tables show that the proposed GAN model (C-WGAN-GP) significantly improves the quality of the generated synthetic microscopic cell images compared to the other existing recent models.

3.2. Classification through sequential CNN

In this second part, we present how oversampling of minority classes or how generating synthetic data for cell types with less data and using sequential CNN has enhanced the performance of classification. We measured and compared the performance of sequential CNN and nine other classifiers through four evaluation metrics. Other classifiers are VGG19 [27], ResNet50 [28], InceptionV3 [29], Xception [30], EfficientNet [31], Logistic Regression [32], Naive Bayes [33], SVM [34] and Random Forest [35]. The evaluation metrics that we used to compare the performance are as defined below:

$$Recall = \frac{TruePositive}{(TruePositive + FalseNegative)} \times 100 \quad (9)$$

$$F1 = 2 \times \frac{Precision \times Recall}{(Precision + Recall)} \quad (10)$$

Table 8
Performance comparison of different classification models on original data.

Models	Recall	F1-Measure	Specificity	Precision
VGG19	65.88	63.11	77.55	64.15
ResNet50	79.18	81.26	81.14	78.37
InceptionV3	82.72	80.31	85.11	81.02
Xception	82.37	80.17	84.26	81.11
EfficientNet	86.33	84.22	88.46	85.03
Logistic Reression	78.25	74.53	82.40	77.29
Naive Bayes	74.11	74.74	77.65	74.93
SVM	86.72	85.44	91.30	85.47
Random Forest	79.15	81.33	83.21	79.46
Sequential CNN	90.48	89.11	89.97	90.72

Table 9
Performance comparison of different classification models on synthetic data.

Models	Recall	F1-Measure	Specificity	Precision
VGG19	70.23	65.86	81.36	69.80
ResNet50	82.38	83.67	84.16	83.45
InceptionV3	86.01	84.99	87.23	85.10
Xception	85.56	83.10	86.17	83.99
EfficientNet	88.91	87.19	91.10	88.56
Logistic Reression	82.13	82.66	84.31	82.18
Naive Bayes	77.01	77.33	79.77	77.60
SVM	90.71	88.33	92.30	89.91
Random Forest	83.56	84.73	85.07	83.55
Sequential CNN	96.55	96.01	95.10	96.93

$$Specificity = \frac{TrueNegative}{(TrueNegative + FalsePositive)} \times 100 \quad (11)$$

$$Precision = \frac{TruePositive}{(TruePositive + FalsePositive)} \times 100 \quad (12)$$

True positive refers to the correctly predicted positive values. True negative refers to the correctly predicted negative values. False positive refers to the incorrectly predicted positive value and false negative refers to the incorrectly predicted negative values. All the four metrics are important for evaluating any classification model. We have selected recall because identifying the true positives is important and we cannot afford false negatives for our classification results. To be more confident about the predicted positives for our classification result we have used precision metrics. F1-Score considers both recall and precision and conveys the balance between both recall and precision measure, which is important. Specificity has been selected because we want to know the ratio of true negatives to overall negatives for our classification result. In Table 8, we show the classification performance of all the ten models using original data. To demonstrate the improvement of classification results when using a balanced dataset obtained by oversampling images of minority classes through C-WGAN-GP, we present Table 9.

Using sequential CNN, for each twelve cell type, we present the accuracy, specificity, and sensitivity of the balanced dataset in Fig. 4 and for the original dataset in Fig. 5. The sequential CNN model classifies every cell with high accuracy, sensitivity, and specificity. The highest accuracy value is 97.5% for neutrophils, with the lowest being 95.6% for erythroblast. The highest value for sensitivity is 97.1% and for specificity is 97%.

In Fig. 6, we show the classification accuracy of all the ten models using our generated dataset with oversampled minority classes. As can be seen, sequential CNN performs the best with a testing accuracy of 96.98%.

Fig. 7, shows few samples from misclassification or incorrect classification. The model will show the image with the cell type. The percentage indicates how much percentage the code classified

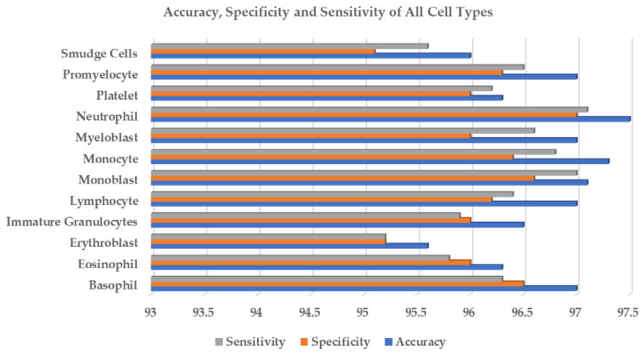

Fig. 4. Accuracy, Specificity and Sensitivity of All Cell Types for the Balanced Dataset.

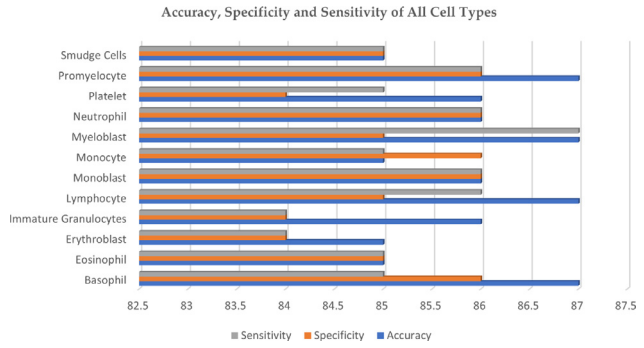

Fig. 5. Accuracy, Specificity and Sensitivity of All Cell Types for the Original Dataset.

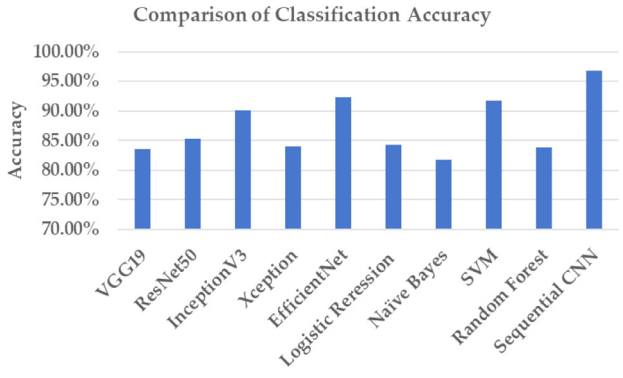

Fig. 6. Comparison of classification accuracy for different models.

it as some other cell type. E.g., in the bracket is the original cell type. So for the first image original is Basophil, but according to the code, the classification rate of 71% is for Neutrophil. Therefore, this is an incorrect classification. On the right, the graph shows the blue bar for original cell type and the red bar for incorrectly classified cell type with the highest percentage.

Fig. 8 shows the training and validation accuracy and loss from the Sequential CNN model. As we can see from the figure, sequential CNN performs well consistently. The model has been trained for 50 Epochs and 256 iterations for every epoch. Sequential CNN requires less computational overhead and performs significantly better than other classification models. It achieved training accuracy of 98.83% and validation accuracy of 98.42%. Training loss is 0.0599, and validation loss is 0.0891.

In Fig. 9 and 10, we show screen grabs of classification results for a few cell types. Our code generates how much percentage it is sure about the cell type. The cell type, written in the bracket, is the original cell type in our dataset, and on the left is the cell type classified by the sequential CNN model. For example, for Ba-

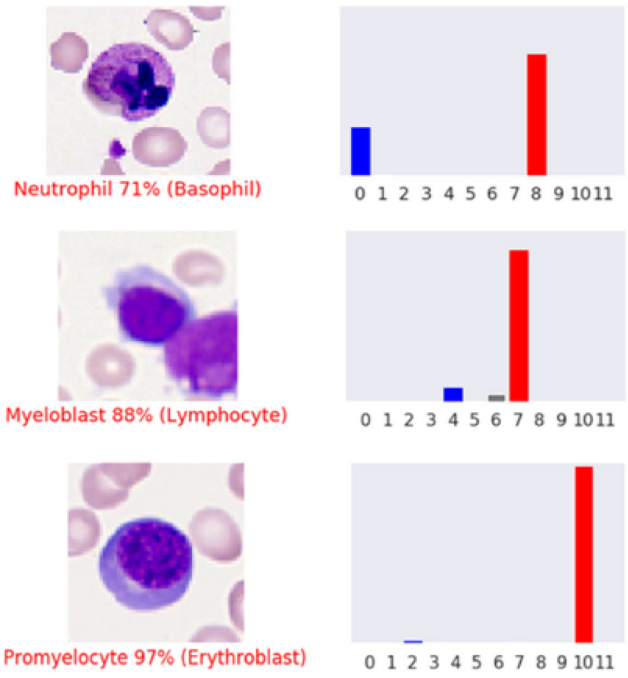

Fig. 7. Samples from misclassification.

sophil 100% (Basophil), the cell type within the bracket is original cell type and Basophil 100% represents the classification result of sequential CNN with confidence rate that is 100%. 0–11 are the classes of the cell types, which are also displayed as the result of the classification model. 0 represents basophil, 1 is eosinophil, 2 is erythroblast, 3 is immature granulocytes, 4 is for lymphocyte, 5 is for monoblast, 6 for monocyte, 7 represents myeloblast, 8 represents neutrophil, 9 is platelet, 10 is promyelocyte and 11 represents smudge cells.

4. Discussion

Label-Free White Blood Cell (WBC) and Red Blood Cell (RBC) segmentation and classification have been the goal for many deep learning researchers to improve the classification models' accuracy and build robust automated systems. ([36,37]). Also, deep learning models need powerful, compatible resources, which leads to an accessibility barrier. [38]. Data scarcity is a significant problem that makes the training of deep learning algorithms complex [39]. Our work tries to solve the data scarcity problem and implement a simple sequential convolutional network to enhance the classification accuracy for microscopic cell images. This work proposes a three-tier architecture to enhance the classification accuracy of microscopic cell images procured from bone marrow aspirate smears. We obtained cell images from three different sources. We collaborated with medical experts to preprocess the three datasets and combine them to form a single dataset. The data preprocessing stage included redundancy removal, filtering or eliminating images, and merging subtypes to form fewer classes. In total, we formed the final dataset with twelve cell types. In the second stage, we proposed C-WGAN-GP, a GAN architecture with three networks, namely generator, discriminator, and classifier. We built the model in a way that the generator trains itself from the feedback of the discriminator and the classifier simultaneously. Our proposed GAN model is different from the AC-GAN, which has the same goal but has a different design. AC-GAN deigns the classifier on top of the discriminator, which makes the training process complicated. We evaluated and compared our proposed GAN architecture with other GAN models to perform the same task with the same dataset. Results show that our proposed GAN model achieves an inception score of 14.52 ± 0.10, which is significantly better than the existing models. We also compare the results for LPIPS, recall, F1, precision, and FID. We also computed the error rates to illustrate how related our generated synthetic data is to the original data.

We develop a sequential CNN model to automate cell classification from microscopic images after generating synthetic microscopic cell images for the cell types containing fewer data. We compare the performance of various classification models using the original as well as the synthetic or balanced dataset. Recall, F1-measure, specificity, and precision were used as evaluation metrics. We present the accuracy, specificity, and sensitivity score for all the twelve cell types. Training and validation accuracy and loss show that our model is very consistent and requires less convergence time. We provide accuracy percentage for nine classification models and screengrabs of classification and misclassification outcomes. Our proposed architecture could achieve a classification accuracy of 96.98% with the balanced dataset, which is significantly better than existing models. The sequential CNN performs better than existing models because we tried to reduce classification error, through learning of the network parameters in the conv2D layers via backpropagation. We used batch normalization to enhance the convergence speed and improve generalization. We used optimum amount of Conv2D layers to avoid overfitting and increase accuracy. Also, ReLU activation function has been used to eliminate the saturation of the gradient in a deep network.

We have proposed a classifier-based GAN model which enhances the class conditioning power of the general GAN architec-

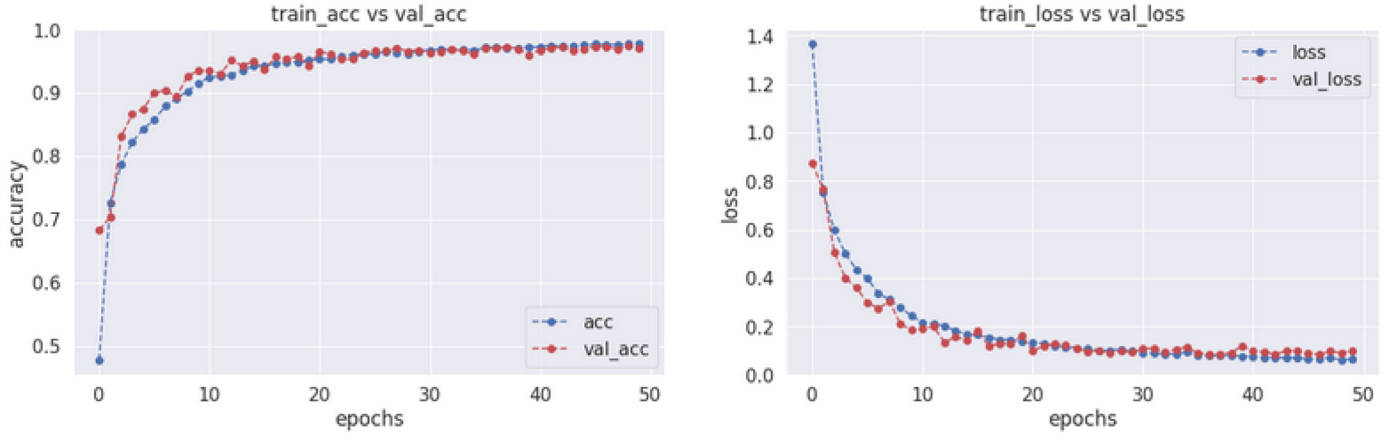

Fig. 8. Screen grabs of classification training and validation accuracy and loss.

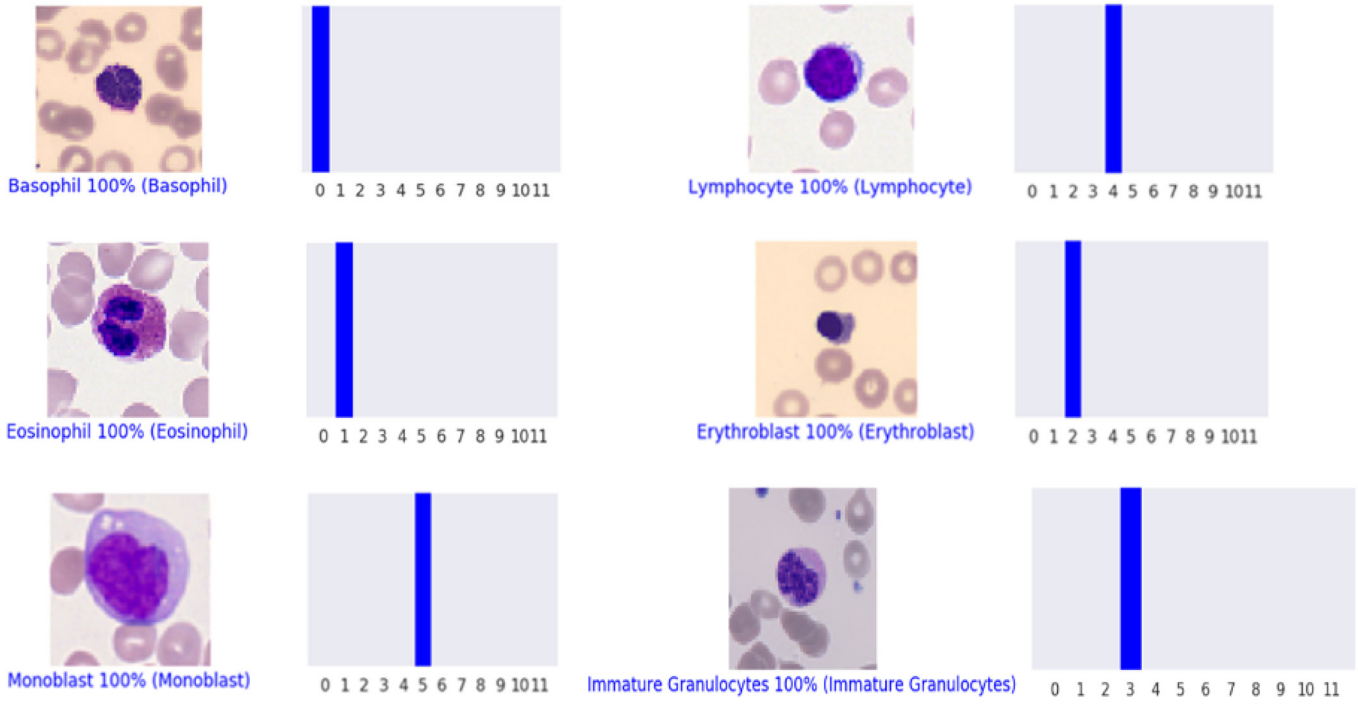

Fig. 9. Screen grabs of samples from classification result using proposed sequential CNN model.

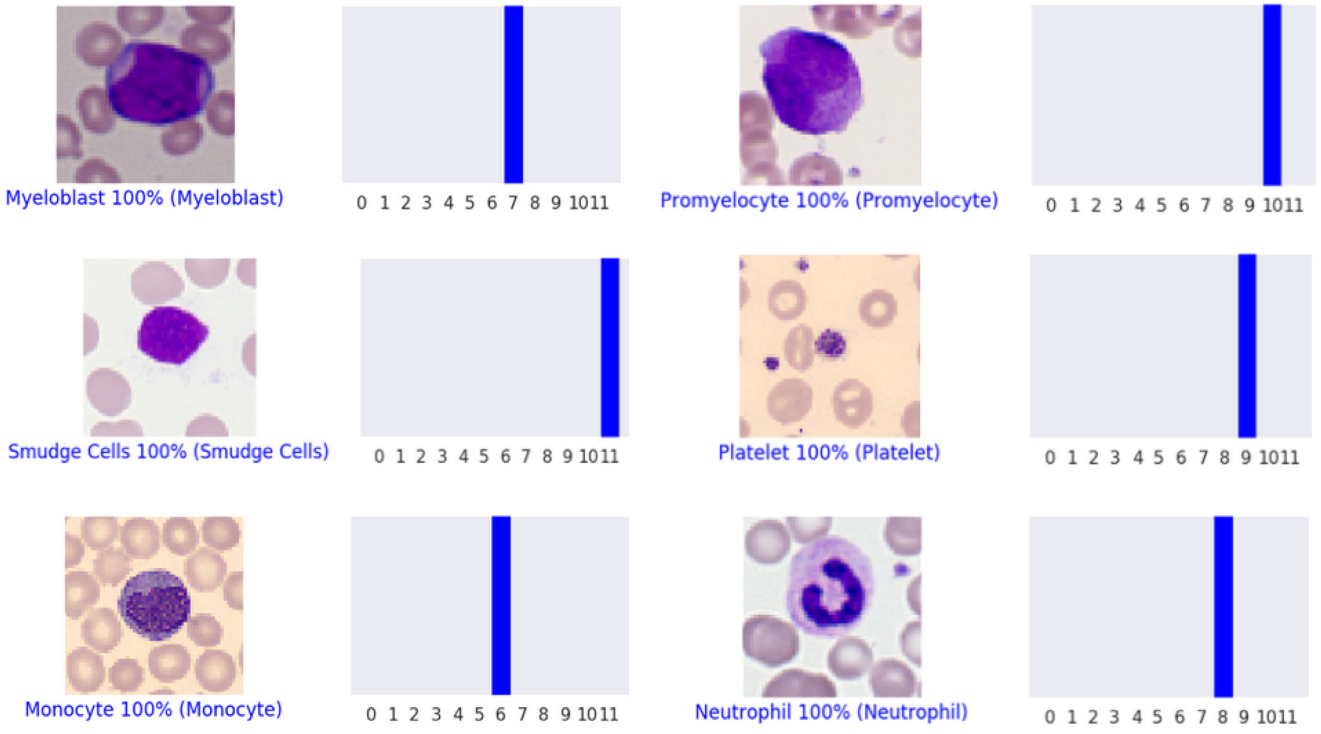

Fig. 10. Screen grabs of samples from classification result using proposed sequential CNN model.

ture. Also, the generator is trained on the outcome of not only the discriminator but also the classifier. Therefore, the proposed technique helped us to generate more realistic and accurate images of each cell types. We then use a simple sequential CNN model which requires less computational overhead and performs significantly better than other classification models. In most studies, four to six cell types are evaluated. We have evaluated the performance of sequential CNN on twelve cell types, and it performed better for each cell type as compared to existing models. We have presented the results of existing and proposed classification models on the original dataset as well as the balanced dataset, which shows that the proposed method is better than the existing solutions. In the future, we would try to improve the classification accuracy by learn-

ing from the misclassification results and developing a model that can learn from the errors.

Declaration of competing interest

The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgments

This research was financially supported by the Ministry of Small and Medium-sized Enterprises(SMEs) and Startups(MSS), Korea, under the "Regional Specialized Industry Development Plus Program(R&D, S3246057)" supervised by the Korea Institute for Advancement of Technology(KIAT).

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).
[2] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[3] D. Zhao, D. Zhu, J. Lu, Y. Luo, G. Zhang, Synthetic medical images using f&bgan for improved lung nodules classification by multi-scale vgg16, Symmetry (Basel) 10 (10) (2018) 519.
[4] A. Beers, J. Brown, K. Chang, J.P. Campbell, S. Ostmo, M.F. Chiang, J. Kalpathy-Cramer, High-resolution medical image synthesis using progressively grown generative adversarial networks, arXiv preprint arXiv:1805.03144 (2018).
[5] W. Dai, X. Liang, H. Zhang, E. Xing, J. Doyle, Structure correcting adversarial network for chest x-rays organ segmentation, 2020, (????). US Patent 10,699,412.
[6] T. Schlegl, P. Seeböck, S.M. Waldstein, U. Schmidt-Erfurth, G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in: International conference on information processing in medical imaging, Springer, 2017, pp. 146–157.
[7] D. Hazra, Y.-C. Byun, Upsampling real-time, low-resolution cctv videos using generative adversarial networks, Electronics (Basel) 9 (8) (2020) 1312.
[8] H. Gao, D. Yao, M. Wang, C. Li, H. Liu, Z. Hua, J. Wang, A hyperspectral image classification method based on multi-discriminator generative adversarial networks, Sensors 19 (15) (2019) 3269.
[9] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, Synthetic data augmentation using gan for improved liver lesion classification, in: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018), IEEE, 2018, pp. 289–293.
[10] C. Baur, S. Albarqouni, N. Navab, Melanogans: high resolution skin lesion synthesis with gans, arXiv preprint arXiv:1804.04338 (2018).
[11] N. Codella, V. Rotemberg, P. Tschandl, M.E. Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti, et al., Skin lesion analysis toward melanoma detection 2018: a challenge hosted by the international skin imaging collaboration (isic), arXiv preprint arXiv:1902.03368 (2019).
[12] H. Rashid, M.A. Tanveer, H.A. Khan, Skin lesion classification using gan based data augmentation, in: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2019, pp. 916–919.
[13] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, P.R. Pinheiro, Covidgan: data augmentation using auxiliary classifier gan for improved covid-19 detection, IEEE Access 8 (2020) 91916–91923.
[14] N. Praveen, N.S. Punn, S.K. Sonbhadra, S. Agarwal, M. Syafrullah, K. Adiyarta, White blood cell subtype detection and classification, in: 2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), IEEE, 2021, pp. 203–207.
[15] A.T. Sahlol, P. Kollmannsberger, A.A. Ewees, Efficient classification of white blood cell leukemia with improved swarm optimization of deep features, Sci. Rep. 10 (1) (2020) 1–11.
[16] C. Cheuque, M. Querales, R. León, R. Salas, R. Torres, An efficient multi-level convolutional neural network approach for white blood cells classification, Diagnostics 12 (2) (2022) 248.
[17] W. Kim, [23] eonelab.co.kr, molecular diagnostic research center', 1983, 1983, (https://www.eonelab.co.kr/global/en/main/main.asp). Accessed October 15, 2021.
[18] C. Matek, S. Schwarz, C. Marr, K. Spiekermann, A single-cell morphological dataset of leukocytes from aml patients and non-malignant controls (aml–cytomorphology_lmu), The Cancer Imaging Archive (TCIA)[Internet].[cited 29 Oct 2019]. Available: https://wiki. cancerimagingarchive. net/pages/viewpage. action (2019).
[19] A. Acevedo, A. Merino, S. Alférez, Á. Molina, L. Boldú, J. Rodellar, A dataset of microscopic peripheral blood cell images for development of automatic recognition systems, Data in Brief, ISSN: 23523409, Vol. 30,(2020) (2020).
[20] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: International conference on machine learning, PMLR, 2017, pp. 214–223.
[21] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of wasserstein gans, arXiv preprint arXiv:1704.00028 (2017).
[22] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784 (2014).
[23] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier gans, in: International conference on machine learning, PMLR, 2017, pp. 2642–2651.
[24] M. Zheng, T. Li, R. Zhu, Y. Tang, M. Tang, L. Lin, Z. Ma, Conditional wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification, Inf. Sci. (Ny) 512 (2020) 1009–1023.
[25] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, Infogan: Interpretable representation learning by information maximizing generative adversarial nets, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 2180–2188.
[26] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434 (2015).
[27] 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. (2021) 1–12.
[28] I.Z. Mukti, D. Biswas, Transfer learning based plant diseases detection using resnet50, in: 2019 4th International Conference on Electrical Information and Communication Technology (EICT), IEEE, 2019, pp. 1–6.
[29] X. Xia, C. Xu, B. Nan, Inception-v3 for flower classification, in: 2017 2nd International Conference on Image, Vision and Computing (ICIVC), IEEE, 2017, pp. 783–787.
[30] X. Wu, R. Liu, H. Yang, Z. Chen, An xception based convolutional neural network for scene image classification with transfer learning, in: 2020 2nd International Conference on Information Technology and Computer Application (ITCA), IEEE, 2020, pp. 262–267.
[31] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.
[32] S. Dreiseitl, L. Ohno-Machado, Logistic regression and artificial neural network classification models: a methodology review, J. Biomed. Inform. 35 (5–6) (2002) 352–359.
[33] M.M. Saritas, A. Yasar, Performance analysis of ann and naive bayes classification algorithm for data classification, Int. J. Intell. Syst. Appl. Eng. 7 (2) (2019) 88–91.
[34] A. Mathur, G.M. Foody, Multiclass and binary svm classification: implications for training and classification users, IEEE Geosci. Remote Sens. Lett. 5 (2) (2008) 241–245.
[35] M.S. Alam, S.T. Vuong, Random forest classification for detecting android malware, in: 2013 IEEE international conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing, IEEE, 2013, pp. 663–669.
[36] D. Ryu, J. Kim, D. Lim, H.-S. Min, I.Y. Yoo, D. Cho, Y. Park, Label-free white blood cell classification using refractive index tomography and deep learning, BME Front. 2021 (2021).
[37] X.-H. Lam, K.-W. Ng, Y.-J. Yoong, S.-B. Ng, Wbc-based segmentation and classification on microscopic images: a minor improvement, F1000Res 10 (1168) (2021) 1168.
[38] L. von Chamier, R.F. Laine, J. Jukkala, C. Spahn, D. Krentzel, E. Nehme, M. Lerche, S. Hernández-Pérez, P.K. Mattila, E. Karinou, et al., Democratising deep learning for microscopy with zerocostdl4mic, Nat. Commun. 12 (1) (2021) 1–18.
[39] M.A. Bansal, D.R. Sharma, D.M. Kathuria, A systematic review on data scarcity problem in deep learning: solution and applications, ACM Comput. Surv. (CSUR) (2022).