Biomedical Signal Processing and Control 66 (2021) 102495

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

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

# A fast and yet efficient YOLOv3 for blood cell detection

Ashkan Shakarami [a,*], Mohammad Bagher Menhaj [b], Ali Mahdavi-Hormat [c], Hadis Tarrah [d]

[a] Department of Computer Engineering, Afarinesh Institute of Higher Education, Boroujerd, Iran
[b] Department of Electrical Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
[c] Department of Electrical, Computer and Biomedical Engineering, Islamic Azad University, Qazvin, Iran
[d] Department of Electrical, Computer and Biomedical Engineering, Islamic Azad University, Qazvin, Iran

ARTICLE INFO

*Keywords:*
Blood cell detection
YOLOv3
EfficientNet convolutional neural network
Dilated convolution
Depthwise separable convolution
Distance-Intersection over Union

ABSTRACT

These days, blood cell detection in microscopic images plays a vital role in cognition, the health of a patient. Since disease detection based on manual checking of blood cells is mostly time-consuming and full of errors, analysis of blood cells using object detectors can be considered as an effective tool. Hence, in this study, an object detector has been proposed which is used for detecting blood objects such as white blood cells, red blood cells, and platelets. This detector is called FED (Fast and Efficient YOLOv3) and it is a One-Stage detector, which is similar to YOLOv3, performs detection in three scales. For the purpose of increasing efficiency and flexibility, the proposed object detector utilizes the EfficientNet Convolutional Neural Network as the backbone effectiveness. Furthermore, the Dilated Convolution is indeed applied in order to increase receptive view of the backbone. In addition, the Depthwise Separable Convolution method is utilized to minimize the detector's parametersand the Distance Intersection over Union is further used for bounding box regression. Besides, for increasing the performance, the Swish activation function is employed. The experiments are run on the BCCD dataset that the average precision of platelets, red blood cells, and white blood cells become 90.25%, 80.41%, and 98.92%, respectively. The results of experiments and comparisons demonstrate that the proposed FED detector is more efficient than other existing studies for blood cell detection.

## 1. Introduction

The analysis of microscopic images is a broad field of study that helps in diagnosing disease using identifying the different blood cells. There are three important constituents in blood: White Blood Cells (WBC), Red Blood Cells (RBC), and Platelets [1,2]. Finding a new automated algorithm in blood cell detection reduces the time and produces more accurate results whereas, manual detection and counting the blood cells are difficult and sometimes full of error [3,4].

Object detection is a computer technology that is widely used in computer vision tasks, such as disease diagnoses, autonomous driving, video surveillance, and so on, with the purpose of locating instances or tracking movement [5,6].

Two main types of object detectors have been proposed up to now. One-Stage detectors and Two-Stage detectors. Two-Stage detectors, such as Faster R-CNN or Mask R-CNN utilize a Region Proposal Network to generate regions of interest and then send the region proposals down the pipeline for object classification and bounding-box regression [7,8]. Although such models are typically slower, reach the highest accuracy rates. One-Stage detectors, such as SSD (Single Shot MultiBox Detector) and YOLO (You Only Look Once) by taking an input image and learning the class probabilities and bounding box coordinates treat object detection as a simple regression problem [9–12]. Although these models are much faster than Two-Stage object detectors, reach lower accuracy rates [13].

In this research, a one-stage detector is proposed to solve the problem of the blood cell detection and low accuracy of One-Stage object detector, which is one of the most challenging problems in blood disease diagnosis. As a result, the proposed FED detector can provide high accuracy besides high efficiency. It provides benefits which are divided into the following items:

- Minimizing the model's parameters by Depthwise Separable Convolution method.
- Increasing receptive view (global view) of object detector using Dilated Convolution.

* Corresponding author.
*E-mail addresses:* ashkan.shakarami.ai@gmail.com (A. Shakarami), menhaj@aut.ac.ir (M.B. Menhaj), ali.mahdavi.hormat@gmail.com (A. Mahdavi-Hormat), hs.tarrah.88@gmail.com (H. Tarrah).

https://doi.org/10.1016/j.bspc.2021.102495
Received 23 September 2020; Received in revised form 18 January 2021; Accepted 4 February 2021
Available online 16 February 2021
1746-8094/

- Increasing efficiency and flexibility of detector using EfficientNet CNN and its compound Scaling method.
- Increasing the performance by Swish activation function and its smoothing feature, and improving loss function using Distance Intersection over Union.
- Ability to train and run the model with a midcore GPU and CPU, and usability of the model in embedded systems and portable equipment.

## 2. Background and literature review

In traditional computer vision approaches such as R-CNN, Fast R-CNN, and SSD, some techniques such as the sliding window and selective search have been used for seeking objects [9,14,15]. In addition, for identifying bounding boxes, Region Proposal Network has been utilized needed to be tested in Faster R-CNN [7]. These methods were too expensive because of the image's scan mechanism. Hence, researchers have proposed another way using the YOLO method concept. The three versions of the YOLO (YOLOv1, YOLOv2, and YOLOv3) have been published so far, that they forward the whole image only once through the network and use prior anchor boxes to look for objects [10–12]. These object detectors are much faster than others while achieving very comparable accuracy. In the following, some blood cell detection approaches have been described which use these methods.

In a blood microscopy image, the distribution of WBC is relatively sparse and easy to count, while the distribution of RBC is relatively dense and is prone to overlap adhesion. In [16] for counting RBC and WBC in microscopic images has been proposed an algorithm which utilizes YOLOv3 to detect discrete RBC and WBC and aggregated RBC. Then aggregated RBC has been counted by image density estimation method. In [16], the maximum mAP is reported as 88.26% by YOLOv3, while this method generates approximately 62 million parameters. These too large numbers of parameters show this method needs powerful hardware systems for training and running. In the method proposed in this paper, a solution is presented for overcoming the afore-mentioned problems by a depthwise separable convolutional network along with its remarkable ability of parameter reduction [17–19].

Increasing the depth of the network is the most common way which has been used in convolutional networks to increase accuracy, but training the deeper networks is more difficult due to the problem of gradient instability [20–23]. In addition, the wide networks tend to find low-level features and do not have the capability to extract high-level features [24–26]. The input image with higher resolution helps the network for better extraction of partial patterns of images however, it increases the number of network parameters. Although increasing the dimensions of the convolutional neural network increases the accuracy, all of these dimensions cannot always increase the accuracy [27–29].

To overcome this problem, one of the most effective methods is compound scaling. In this method, the depth, width, and resolution are changed according to a certain ratio. Therefore, in this study, EfficientNet CNN is used to apply this technique [27].

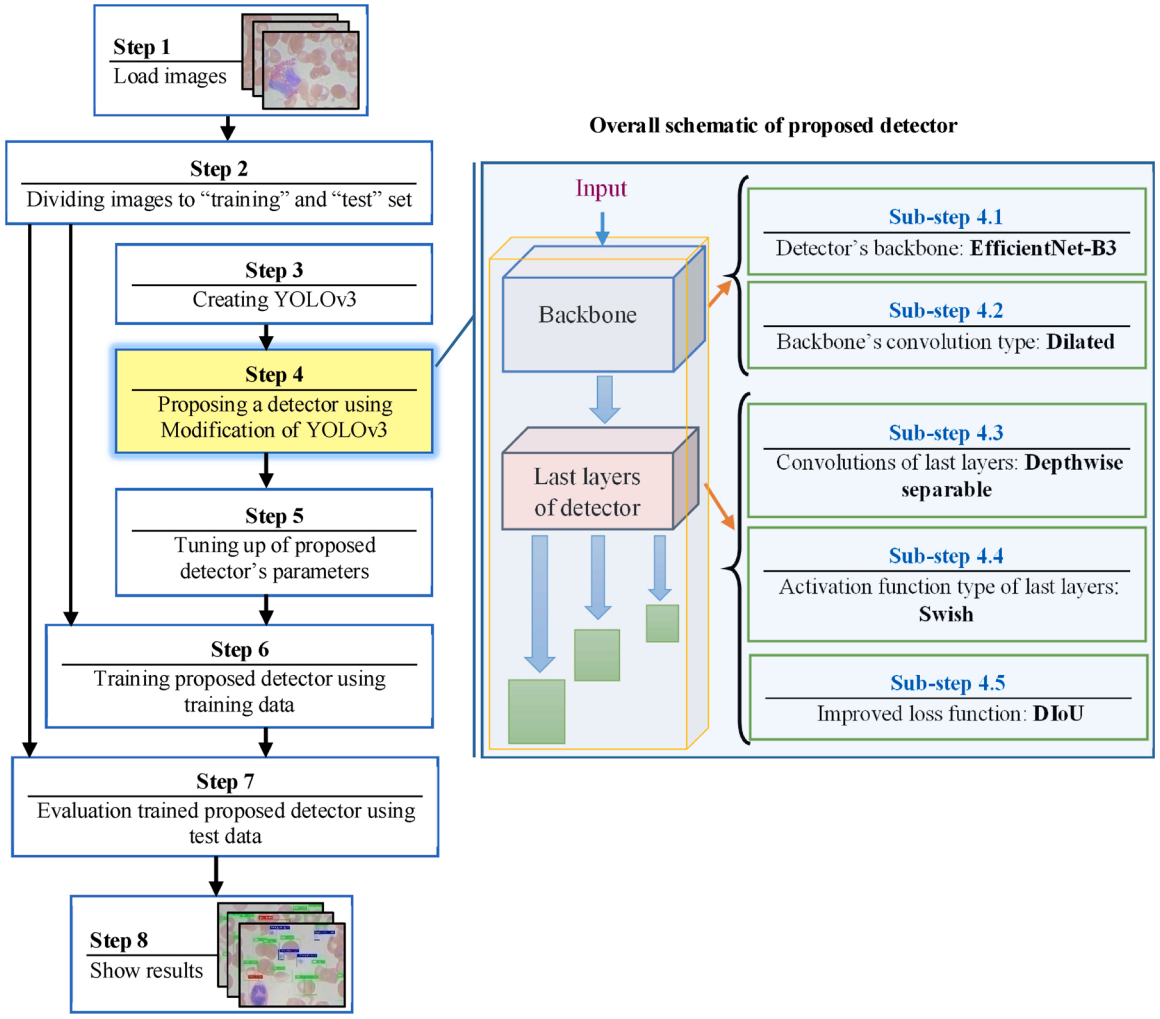



**Fig. 1.** Flow chart of the proposed method.

## 3. Material and methods

The goal of object detection is finding the location of objects in images and also recognition of its category. In this research, the proposed object detector by taking an input image and learning the class probabilities and bounding box coordinates, treat object detection as a regression problem. This proposed detector is a modified YOLOv3 which is described in the following [12].

In Fig. 1, the flow chart of the proposed method has been shown. According to this figure, the method of this research includes 8 steps. These steps are described in the following.

- **Step 1:** In this step, the images are loaded in a dataset.
- **Step 2:** In the second step, the loaded images have been divided into two sets such as training and test.
- **Step 3:** In this step, firstly YOLOv3 has been created for creating the proposed detector.
- **Step 4:** In the fourth step, the object detector of this research is proposed using a modification of YOLOv3. In Fig. 1, on the right side of "Step 4″, the overall schematic of the proposed object detector has been illustrated. This schematic includes "Backbone", "Last layers of detector" and three green rectangles that denote the detection of the three scales concept of YOLOv3 used in the proposed detector.
  - **Sub-step 4.1**: According to this schematic, it has been seen that the first major modification has been applied in the proposed detector's backbone of the original YOLOv3. This modification includes changing Darknet53 CNN to EfficientNet-B3 CNN.
  - **Sub-step 4.2:** Second major modification is changing the type of a middle normal convolution layer of EfficientNet-B3 to dilated convolution with a dilation rate of 2.
  - **Sub-step 4.3:** In this sub-step, for creating the proposed detector, the last layers of YOLOv3 are modified. This modification includes changing normal convolution layers to depthwise separable convolution.
  - **Sub-step 4.4:** Another modification of the last layers is using the Swish activation function instead of LeakyReLU.
  - **Sub-step 4.5:** Furthermore, the loss function of YOLOv3 has been improved using DIoU, and then it has been used as the loss function of the proposed detector. All modifications of YOLOv3 in order to attain the proposed detector have been mentioned in Table 1. In addition, in sections 3.1,3.2, 3.3, 3.4, and 3.5 each of these modifications is fully described. Moreover, in Fig. 2, a pictorial form of the proposed detector is depicted in detail.
- **Step 5:** In this step, the parameters of the detector are tuned up through training. Some of these parameters are explained in sub-section 4.1 and other parameters, which are not mentioned, are tuned up similar to parameters of YOLOv3 [12].
- **Step 6:** The proposed detector, in this step is trained using the training data. Moreover, additional information about the training phase is described in sub-section 4.2.
- **Step 7:** In this step, the proposed detector after the phase of training is evaluated using test data and in order to evaluate the proposed method the Average Precision (AP), mean Average Precision (mAP), and Precision-Recall curve have been utilized. In addition, the efficiency of the proposed detector is evaluated in terms of the number of the model's parameters. These evaluations have been explained in sub-section 4.3 (parts A, B, and C). For more detail, refer to that.
- **Step 8:** In the last step of the method's flow chart, the visual results of the proposed method have been shown. These steps are also described in sub-section 4.3 (part D).

**Table 1**
Comparing methods of the proposed detector (modified YOLOv3) with YOLOv3.

| Method | YOLOv3 | Proposed method (modified YOLOv3) |
|---|---|---|
| Backbone | Darknet53 | EfficientNet-B3 CNN |
| Backbone's convolution view | Normal | Dilated |
| Convolutional last layers type | Normal | Depthwise separable |
| Activation function | LeakyReLU | Swish |
| Location loss function | IoU | DIoU |

The differences between the proposed detector and YOLOv3 have been described in Table 1. It should be said that the loss function of YOLOv3 includes the sum of three loss functions: classification, confidence, and location losses. In this research, the loss function of YOLOv3 has been improved using DIoU. In other words, the used loss function for the proposed detector is similar to the YOLOv3′s loss function though the DIoU method has been used for bounding box regression instead of using Intersection over Union (IoU). This method can prevent gradient vanishing due to the bonding box covering over-lapping. The improved loss function is described in section 3.5 in detail.

Fig. 2 has been illustrated by inspiration of the architecture of YOLOv3 [12] and considering the descriptions of the proposed method for depicting the pictorial form of the proposed detector. According to this figure, the EfficientNet-B3 CNN (among all B0 to B7 versions of EfficientNet) has been utilized for the FED detector's backbone. By consideration of EfficientNet CNN and its flexibility, the use of its other versions in FED's backbone is feasible [27]. For more details, see section 3.1.

As can be seen, the input dimension of the FED detector is $416 \times 416 \times 3$, which can be adjusted with different dimensions. After making the detector's backbone, three output branches are defined (indexes: 117, 264, and 381) which have been used as inputs of other blocks. Furthermore, it is shown that the standard Conv2D of DepConv2D (index: 250) has been modified to Dilated Convolution in order to increase receptive and global view (for more details, see section 3.2). The abbreviation of layers such as Conv2D, DepConv2D is shown in the "Layers guide" section of Fig. 1.

In the proposed FED detector, the last layers ($Y_i$) are 3D tensors for the bounding box's encoding, objectness, and class predictions. The dimension of $Y_i$ is calculated according to Eq. (1):

$$Y_i = N_i \times N_i \times [s \times (b + objn + c)] \quad (1)$$

In Eq. (1), the length or width of the tensor is denoted by $N_i$, "s" is the number of boxes at each scale, "b" is used for 4 bounding box offsets, "objn" is objectness which is equal to 1 and "c" is the number of class prediction which is set with the number of dataset's categories.

So, in experiments of this research with the BCCD dataset, FED detector (similar to YOLOv3) predicts 3 boxes at each scale (s = 3) with b = 4, objn = 1 and c = 3 (WBC, RBC and Platelets) and as a result:

- $Y_1 = 13 \times 13 \times 24$;
- $Y_2 = 26 \times 26 \times 24$;
- $Y_3 = 52 \times 52 \times 24$.

According to Fig. 1, it can be seen that any $Y_i$ is created using $X_i$ and $Z_i$ tensors. The $X_i$ tensors ($X_1$, $X_2$, and $X_3$) and $Z_i$ tensors ($Z_1$, $Z_2$, and $Z_3$) have composed of some layers which are shown in the "Layers guide" of Fig. 1. One of $X_i$ or $Y_i$' s components in the proposed FED detector is Depthwise_Separable_Conv2D_BN_Swish while this component for YOLOv3 is DarknetConv2D_BN_LeakyReLU. Using Depthwise_Separable_Conv2D instead of DarknetConv2D has offered some benefits such as minimizing the model size and parameters (for additional information, see section 3.3).

In addition, because of using the Swish activation function and its smoothing feature, the mAP has been increased in the FED detector (for more details, see section 3.4).

In Fig. 1, the used Concatenate method (with symbol ⊕) gets a list of tensors as input and returns the concatenation of them as a single tensor.

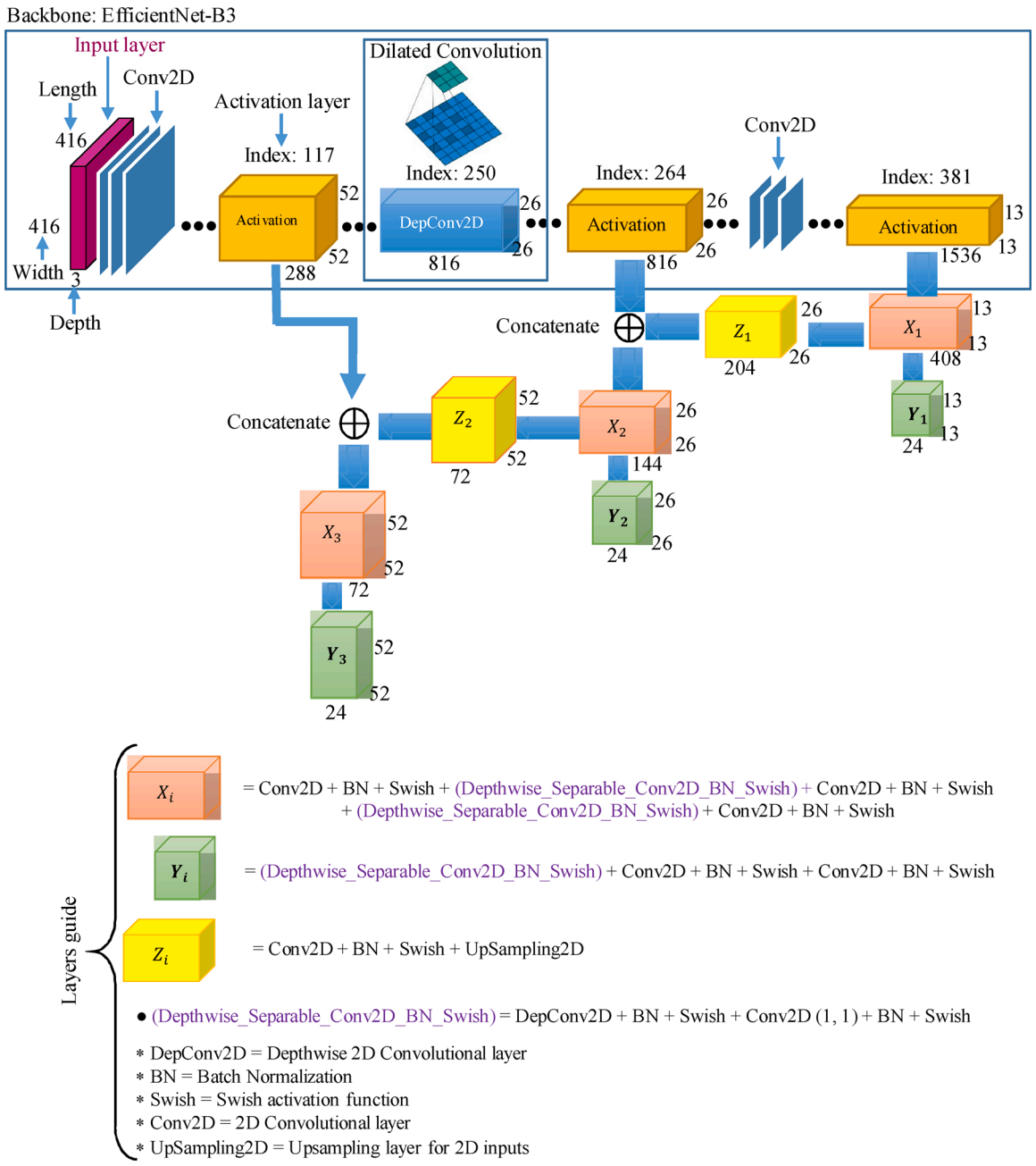



**Fig. 2.** Pictorial form of the proposed FED detector in detail.

Concatenation means sticking the data cubes back to back in the channel direction. There are residual blocks in the FED detector similar to YOLOv3 that element-wise addition has been used in them.

In this research to predict the size of candidate boxes, the prior anchors are utilized (similar to YOLOv3) which include the height and width of a set of candidate boxes.

The value of prior anchors in YOLOv3 was clustered by Pascal VOC and COCO datasets, but these anchors are not appropriate for detecting blood cells in microscopic images [12]. So, the analyzing and clustering of the used dataset's images are necessary to produce new anchors. In this research, similar to YOLOv3 the k-means clustering is utilized to carry out analysis of dimension clustering for new anchors on the training set. So, after doing this, 9 clusters and 3 scales are sorted and then the clusters evenly are divided up across scales. On the BCCD dataset the 9 clusters are: (29 × 28); (64 × 82); (66 × 64); (74 × 74); (76 × 52); (82 × 80); (84 × 68); (94 × 87); (162 × 148).

In most object detection algorithms, Non-Maximum Suppression is utilized to be sure that the algorithm detects each object once only. Using this method, the detected boxes which are redundant are removed when their overlap with the box which has the highest score exceeds a threshold. Hence, in this study, the Non-Maximum Suppression method has been utilized for non-maximum suppression similar to YOLOv3.

### 3.1. *EfficientNet CNN and its compound scaling method*

Compound scaling is a method that uses the changing depth, width, and resolution simultaneously. EfficientNet CNN uses compound scaling to achieve higher performance, accuracy, and flexibility. Although the increase of convolutional neural networks' depth, width, and resolution without special rule may increase the accuracy when the dimensions of the network become too large, these dimensions cannot increase the accuracy and in most cases have adverse effects. While the compound

scaling method is an effective idea that can increase accuracy [27].

The visual behavior and class activation map of Efficient Net's layers using the compound scaling method has been depicted in Fig. 3 so that each bounding box is the ground truth of one object. According to this figure, it is clear that the compound scaling method (Fig. 3e) enables the model to consider relevant zones that include more object details.

As a result, in this study by considering the benefits of the compound scaling method and the nature of blood cells images, the EfficientNet CNN has been used. Another merit is the flexibility of EfficientNet. This network is designed in B0 to B7 versions that each of them has a different compound scaling size so that it has different accuracy and needs different resources to run. For instance, the EfficientNet-B1, EfficientNet-B3, EfficientNet-B4 and EfficientNet-B7 produce 7.8, 12, 19 and 66 million parameters, respectively. On the other hand, the top-1 accuracies of them on the ImageNet dataset are 78.8%, 81.1%, 82.6%, and 84.4%, respectively. In this research, the EfficientNet-B3 has been selected in order to maintain the balance between accuracy and needing hardware and resourcing to run.

### 3.2. Dilated convolution

In the first layers of Convolutional Neural Networks, the low-level features are recognized and extracted while the last layers extract high-level features [30]. Although, the first layers of the Convolutional Neural Networks do not have an effective view of the image, by passing through these layers and reaching the middle layers, a relatively effective view of image features has been captured. However, this view is not as well as the last layers. By increasing the receptive view of the convolutional neural network in the middle layers, it is possible to converge an earlier model and achieve a better view and consequently, it has positive effects on the next layers. So, in this research in order to achieve this goal, a dilated convolution has been utilized in a middle depthwise block of the proposed FED detector.

Dilated Convolution is a convolution applied on input with defined gaps. This method is a technique to increase the receptive view (global view) of the Convolutional Neural Networks. Moreover, it can accrete linear parameters [31]. A 2D Dilated Convolution is definable according to Eq. (2).

$$y(m,n) = \sum_{i=1}^{M}\sum_{j=1}^{N} x(m + r \times i,\ n + r \times j) \times w(i,j) \tag{2}$$

In Eq. (2), $y(m,n)$ is the output of Dilated Convolution from input $x(m, n)$ and a filter $w(i,j)$ with the length and width of M and $N$, respectively. The parameter $r$denotes the dilation rate. Dilated Convolution turns into a normal Convolution when r is set with 1. In Fig. 4a normal Convolution is demonstrated while Fig. 4b is shown dilated convolution with $r = 2$.

### 3.3. Depthwise separable convolution

In this research, the last layers of the proposed detector are designed and inspired by Depthwise Separable Convolution. This technique can decrease the number of parameters greatly. Hence, the computation time and model size are reduced [17]. As a result, the produced model can be utilized in portable equipment.

The Depthwise technique factorizes a normal Convolution into a Depthwise Separable Convolution. The Depthwise Separable Convolution includes Depthwise and Pointwise layers according to Fig. 5. In this figure, the $1 \times 1$ Convolution is called Pointwise Convolution [25,18,19]. The standard Convolution and principle of Depthwise Separable Convolution are shown in sections 5.a and 5.b of Fig. 5 respectively.

### 3.4. Swish activation function

The Swish is an activation function for Neural Networks that is obtained by multiplying the input value in the sigmoid activation function. The Swish is defined as $f(x) = x.\sigma(\beta x)$. The $\sigma(z) = (1 + \exp(-z))^{-1}$ is the sigmoid function and $\beta$ is either a trainable parameter or a constant parameter. Fig. 6 shows the graph of Swish for $\beta = 1$ versus ReLU [32,33].

The benefits of Swish are divided into the following items [32]:

1 Swish is bounded below and very negative weights are zeroed. Also, it benefits from sparsity similar to ReLU.
2 It is unbounded above so that for very large values, the outputs do not saturate to the maximum value (i.e., to 1 for all the neurons).
3 Swish brings some benefits due to its smoothing output curve when converging the model towards the minimum loss.
4 Swish prevents small negative values from being zero, because they may still be relevant for capturing patterns underlying the data while the output of activation functions such as ReLU is zeroed out in negative parts.

As a result, in this research, the Swish activation function is utilized for the proposed FED detector.

### 3.5. Improving YOLOv3 loss function using distance intersection over union

The loss function of YOLOv3 includes three parts such as coordinate loss, classification loss, and confidence loss. This loss function can be denoted by the Eq. (3) that the $Coord_{loss}$ denotes the coordinate loss, the confidence loss is presented by $Conf_{loss}$, and $Class_{loss}$ is calculated for classifying loss [12,34].

$$Loss_{yolov3} = \sum(Coord_{loss} + Conf_{loss} + Class_{loss}) \tag{3}$$

The coordinate loss in Eq. (3) includes the summation of "(x, y)" loss and "(w, h)" loss in prediction boxes. In YOLOv1," x"," y"," w", and "h" are used to predict the actual value of the object. These values are really important so that small changes in them can expand to the whole of the image. Consequently, large coordinate fluctuations and also inaccurate predictions will happen. In YOLOv2 are proposed some equations such as (4) to (7) for addressing and improving these problems [11,34].

$$x = \sigma(t_x) + c_x \tag{4}$$

$$y = \sigma(t_y) + c_y \tag{5}$$

$$w = p_w e^{t_w} \tag{6}$$

$$h = p_h e^{t_h} \tag{7}$$

In Eq.s (4) and (5), $t_x$ and $t_y$ are values of the network's prediction. Plus, $c_x$ and $c_y$ are the coordinates of cells on the feature map. In Eq.s (6) and (7), $t_w$ and $t_h$ are the network prediction value too, and the width and height of the cell corresponding to the anchor box are expressed by $p_w$ and $p_h$ respectively.

According to Eq.s (4) and (5), in the prediction box, the center point coordinates "x" and "y" are activated using the sigmoid. In Fig. 7, the curves of the sigmoid function and its derivative have been depicted. According to these curves, if the output of the Neural Network becomes large, the sigmoid function's derivative will become exceedingly small. At this moment, the obtained squared error values are exceedingly small, but these values lead to the convergence speed of Neural Network become slow [34].

In the above problem, if the real value is only 0 or 1, the cross-entropy will be a common method that can be used for addressing this problem. It is formulated in Eq. (8) [34].

$$Loss_{cross\_entropy} = -1/n(\sum_{i=1}^{n}[a_i \times \log(\widehat{a}_i) + (1 - a_i) \times \log(1 - \widehat{a}_i)] \tag{8}$$

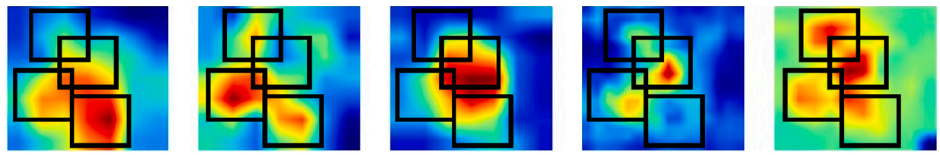



**Fig. 3.** A sample of class activation map for different scales in EfficientNet CNN (Tan and Li 2019).

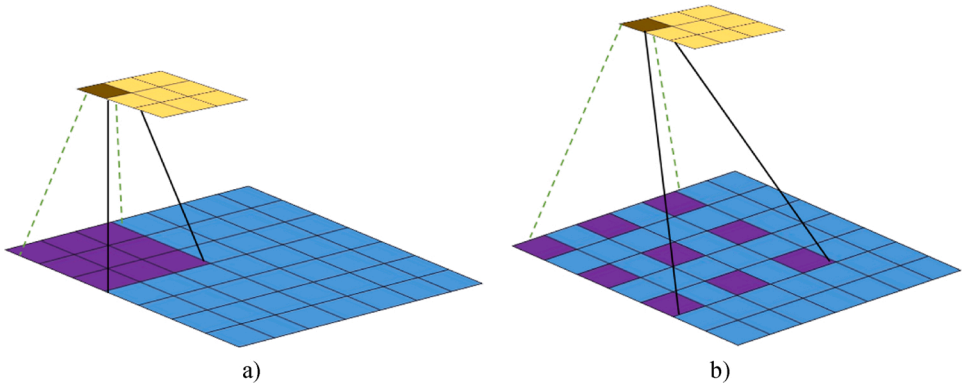



**Fig. 4.** a) Normal Convolution with 3 × 3 kernel, b) Dilated Convolution with dilation rate as 2.

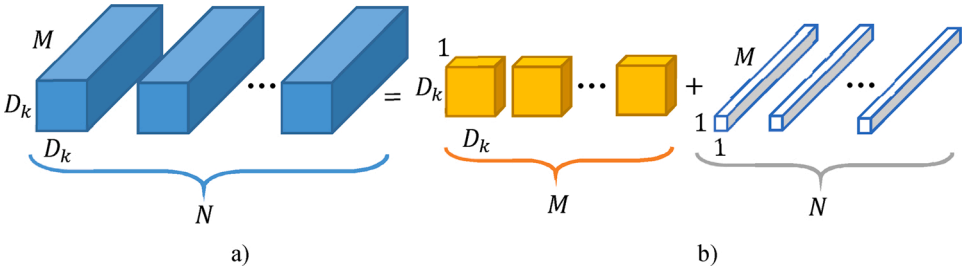



**Fig. 5.** a)Standard convolution kernels: N × DK × DK × M, b) depthwise convolution kernels: M × DK × DK and pointwise convolution kernels: 1 × 1 × M.

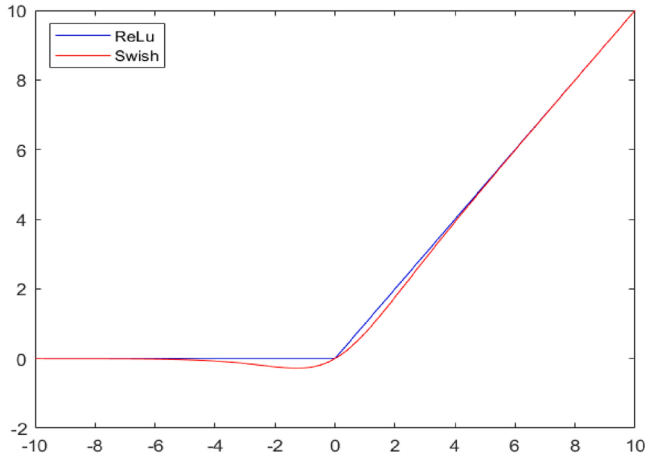



**Fig. 6.** The Swish activation function versus ReLU.

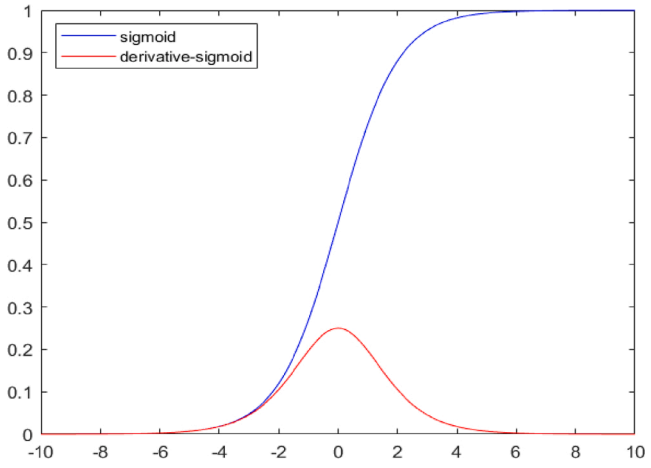



**Fig. 7.** Sigmoid function and its derivative.

In Eq. (8), $a_i$ denotes the true value and $\widehat{a}_i$ is the output value after the sigmoid function. According to this equation, the defined loss function of cross-entropy meets the requirements when the true value, $a_i$ can only take 0 or 1. In other words, when both of $a_i$ and $\widehat{a}_i$ are equal to 0 or 1 ($[a_i = \widehat{a}_i = 0]$ or ($[a_i = \widehat{a}_i = 1]$) the error are calculated using Eq. (8). For the prediction box center point coordinate," (x, y)" the true value is neither 0 or 1 rather, it is a value between 0 and 1. For instance, using $a_i = \widehat{a}_i = 0.7$, the value of cross-entropy loss is calculated according to $-0.7 \times \log(0.7) - 0.3 \times \log(0.3) = 0.27$ that it is not 0. So,

improving this loss function is feasible [34].

In our research, DIoU[1] is utilized to improve and modify the loss function of "(x, y)" (the center coordinate). Some solutions are described in the following in order to use DIoU. DIoU is considered as an improved version of IoU. It replaces the regression parameters in order to calculate the distance loss of the prediction box.

The bounding box regression is a vital step because the location of the target object is created by a rectangular box which is produced according to the bounding box regression [35]. The IoU-based loss is defined as Eq. (9). In this equation $\mathscr{R}(B,\ B^{gt})$ is the penalty term that is used for predicted target box $B^{gt}$ and box $B$.

$$\mathscr{L} = 1 - IoU + \mathscr{R}(B,\ B^{gt}) \tag{9}$$

For evaluating bounding box regression, IoU loss works for situations that bounding boxes have overlap, but it would not provide any moving gradient for other cases such as the non-overlapping one. For overcoming the IoU's problems, the Generalized IoU (GIoU) with $|C - B \cup B^{gt}|/|C|$ penalty term has been proposed where $C$ is the smallest box covering $B^{gt}$ and $B^{gt}$. When this term is used in GIoU, the predicted box shifts towards the target box in cases such as non-overlapping [36]. The GIoU has several limitations, though it can resolve the gradient vanishing problem for cases such as non-overlapping. To overcome these limitations, in our research, DIoU loss is used for bounding box regression.

In order to directly minimize the normalized distance between central points of two bounding boxes, a penalty term is added on IoU loss simply in DIoU. This term leads to a DIoU much faster convergence than GIoU [35]. This penalty term can be defined as Eq. (10).

$$\mathscr{R}_{DIoU} = \rho^2(b,\ b^{gt})/c^2 \tag{10}$$

In Eq. (10), $\rho$ is the Euclidean distance, $b$ and $b^{gt}$symbolizes the central points of $B$ and $B^{gt}$, and the smallest enclosing box's diagonal length covering the two boxes are mentioned by $c$. So, the loss function of DIoU is formulated as Eq. (11).

$$\mathscr{L}_{DIoU} = 1 - IoU + \rho^2(b,\ b^{gt})/c^2 \tag{11}$$

According to Fig. 8, GIoU aims to decrease the area of $C - B \cup B^{gt}$ while the distance between two central points is directly minimized by its penalty term. In this figure, the diagonal length of the smallest enclosing box covering two boxes is mentioned as $C$, and $d = \rho(b,\ b^{gt})$ denotes the central points' distance of two boxes.

DIoU has several advantages over IoU and GIoU that are mentioned below [35].

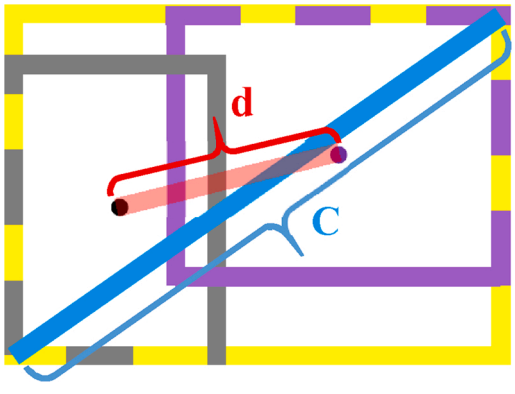



**Fig. 8.** Visual concept of DIoU loss for bounding box regression.

1 DIoU can converge much faster than GIoU because of directly minimizing the distance of two boxes.
2 In horizontal and vertical orientations and for the cases which include the inclusion of two boxes, the regression is made very fast by DIoU; however, GIoU loss is almost reduced to IoU loss.

## 4. Results

In this section, sub-sections 4.1, 4.2, and 4.3 are provided in order to describe implementing details, dataset, and also experiments and evaluations.

### 4.1. Implementing details

In this study, Python and Keras, 6 GB RAM, an Nvidia GeForce 920 M GPU, and an Intel® Core™ i5−7200U @ 2.50 GHz CPU have been utilized for implementing the proposed method. In Table 2, this information has been shown. In addition, mean average precision (mAP) criteria and Precision-Recall curve have been used to evaluate the proposed method [37,38]. Plus, the Adam training algorithm has been utilized to train the model [39]. Also, 38 epochs have been performed to train the model with a learning rate of 0.0001. In the following subsections, dataset, results, and comparisons are demonstrated and described in detail.

### 4.2. Dataset

In this study, for evaluating the proposed FED detector, the BCCD[2] dataset has been used. This dataset includes 364 images that each of them has dimensions of 640 × 480 × 3 and contains a small number of WBC and Platelets and a large number of RBC [16]. The division protocol of this dataset has divided images into training and test sets with a ratio of 8:2. In other words, 80% of data has been used for training, and reminded 20% has been utilized to evaluate the model. In addition, in this research, horizontal flip image, converting image to grayscale, vertical flip image, distorting the image in HSV color space and adjusting such as brightness, color level, contrast, and sharpness, are applied due to data augmentation [34,40].

In Table 3, the number of used objects for training and test has been shown in detail. According to this table, the 3943 objects have been used for training that RBC has the highest number. Also, the 945 objects have been utilized for the evolution of the proposed FED detector.

In this study, the cross-validation has not been used for reasons which are described in the following.

1 Very time-consuming training: Because of the nature of the object detection method and their high number of parameters.
2 Choosing the best part of data in the training phase: In this situation using Cross Validation the best model of object detection is made;

**Table 2**
Implementing tools and setting.

| Parameter | Type |
|---|---|
| Programming Language | Python |
| Library and wrapper | Keras with Tensorflow |
| GPU | Nvidia GeForce 920 M |
| CPU | Intel® Core™ i5-7200U @2.5 GHz |
| RAM | 6 Gigabyte |
| Training algorithm | Adam |
| Evaluation metrics | Mean average precision and Precision-Recall curve |

[1] Distance Intersection over Union

[2] https://github.com/experiencor/BCCD_Dataset

**Table 3**
The number of categories and objects per categories of the BCCD dataset.

| Data division | Type | Number of objects per category | Total |
|---|---|---|---|
| Training | RBC | 3350 | 3943 |
| | WBC | 301 | |
| | Platelets | 292 | |
| Test | RBC | 805 | 945 |
| | WBC | 71 | |
| | Platelets | 69 | |

however, this manipulation cannot be considered as an effective and yet accurate technique to evaluate the model [41].

3 The data division protocol of the used dataset: The used BCCD dataset has a data division protocol but cross-validation selects different training and test data for each fold. Hence, it is not reasonable to perform it on this dataset.

### 4.3. Experiments and evaluations

#### 4.3.1. Number of parameters

In Fig. 9, the number of parameters of both the YOLOv3 and the proposed FED detector using normal convolution and depthwise separable convolution are shown. According to this figure, the number of parameters of the FED is approximately 5 times as less as that of the YOLOv3 because of depthwise separable convolution and as a result, the size of the inference model and also consumed memory is reduced. Hence, it is useful for embedded systems and portable equipment. Moreover, it can be seen that the parameters' number of the FED detector with Depthwise Separable Convolution is approximately 2 times as less as that of the FED detector with normal Convolution.

#### 4.3.2. The effects of dilated convolution, Swish activation function, and DIoU

In Table 4 the effects of Dilated Convolution, Swish activation function, and DIoU for the FED detector have been mentioned. According to this table, when Dilated Convolution is used instead of normal Convolution, the mean average precision of the FED detector has been increased from 87.66%–88.33%. On the other hand, the Swish activation function has increased mAP as well. Moreover, the mean average precision increases when the FED detector uses DIoU instead of IoU. As a result, the mean average precision is equal to 87.66 when the FED detector uses the normal Convolution, LeakyReLU, and IoU while using Dilated Convolution, Swish, and DIoU, the mean average precision increase to 89.86%. Consequently, these methods increase efficiency.

The Average Precision (AP) of the proposed FED detector for any object is shown in Fig. 10. As can be seen, the FED detector has detected the WBC very well and the Platelets with 90.25%. The RBC has been detected with 80.41% because they are very dense and some of them are not separable adequately.

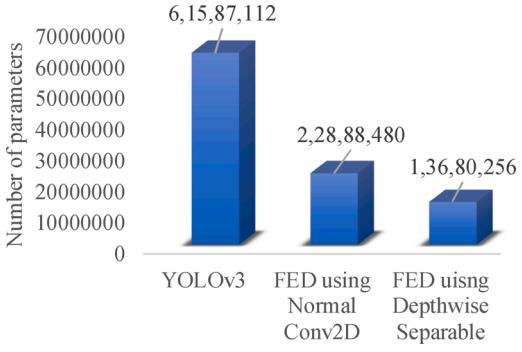



**Fig. 9.** The number of produced parameters by YOLOv3, proposed FED using normal convolution, and depthwise separable convolution.

**Table 4**
The effects of Dilated Convolution, Swish activation function, and DIoU in the proposed FED detector.

| Method | Mean average precision |
|---|---|
| Normal Convolution, LeakyReLU, IoU | 87.66 |
| Dilated Convolution, LeakyReLU, IoU | 88.33 |
| Dilated Convolution, Swish, IoU | 89.20 |
| Dilated Convolution, Swish, DIoU | **89.86** |

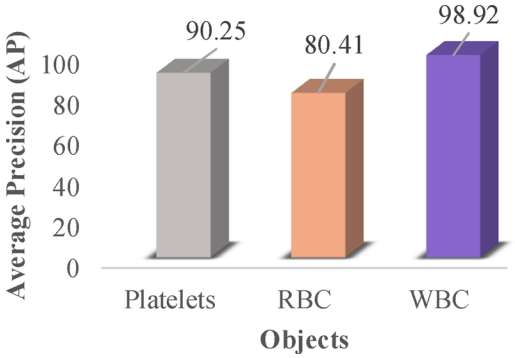



**Fig. 10.** The Average Precision (AP) of the proposed FED detector.

The Precision-Recall curves of Platelets, RBC, and WBC have been illustrated in Fig. 11 (sub-figures 11.a, 11.b, and 11.c, respectively). According to this figure, the detection precisions are high for most test images. In almost all test images the WBC has been found with the highest precision. In these experiments the IoU = 0.4.

#### 4.3.3. Detection time (speed) and the usability in portable equipment

In Table 5, the detection time of the proposed FED detector and the original YOLOv3 has been shown. According to this table, the average detection time of the FED detector is 0.28 s for each image on an Nvidia GeForce 920 m. In addition, because of the low parameters of the proposed detector, it is capable to run on CPU and its detection time is equal to 0.57 s averagely on Intel® Core™ i5−7200U @ 2.50 GHz and 6 gigabyte RAM. However, the detection time of the original YOLOv3 on Nvidia GeForce 920 m is 0.51 s and its time on this CPU is 0.79 s. So, it can be concluded that the proposed detector is approximately two times as fast as the original YOLOv3 is for the detection of objects on the used GPU. Moreover, the proposed detector is faster than the original YOLOv3 on the used CPU.

Therefore, due to the speed of the proposed detector and its low number of parameters (which was described in section 4.3.1), this method can be a proper choice for embedded systems and portable equipment.

#### 4.3.4. Visual results

The detection results of the FED detector have been visualized in Fig. 12 for two test images. According to this figure, it can be seen that the FED detector can detect the Platelets and WBC very well. Despite the density of RBC and difficulty in their detection, the proposed method has been able to detect most of them in test images correctly.

#### 4.3.5. Comparison results

The proposed FED is able to detect WBC, RBC, and Platelets with 98.92%, 80.41% and, 90.25%, respectively as given in Table 6. Moreover, for data division, the protocol of the BCCD dataset has been used without any change so that the rate of training and test data has been considered 80% and 20%, respectively.

Zhang et al. [16] have proposed a method for WBC and RBC detection and counting, using YOLOv3 and Image Density Estimation method and its average precision for WBC and RBC are respectively 86.49% and

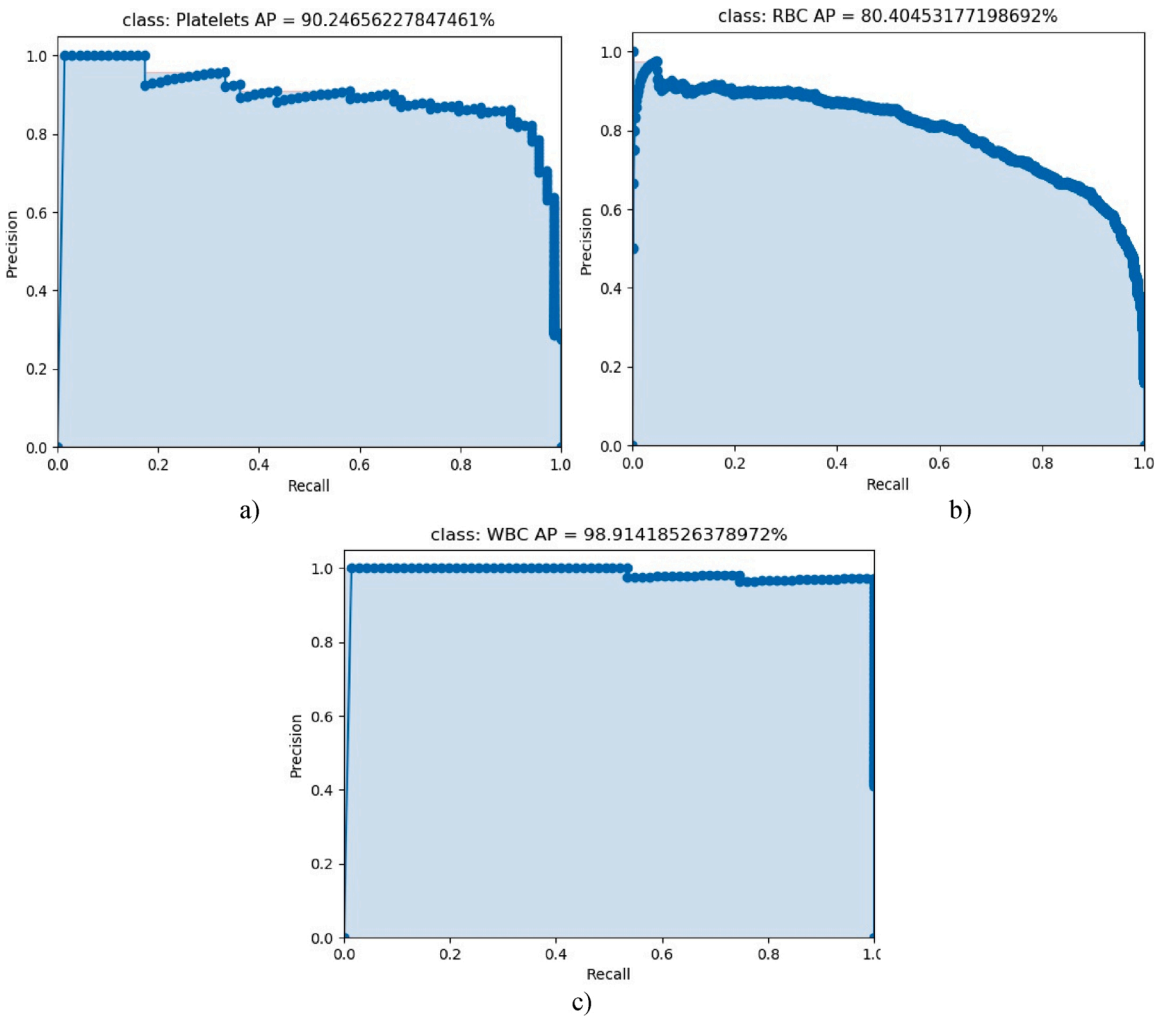



**Fig. 11.** The Precision-Recall curves for a) Platelets, b) RBC, and c) WBC.

**Table 5**
Comparing the detection time of the proposed FED detector to the original YOLOv3 based on the seconds for each image (or frame).

| Method | CPU: Intel® Core™ i5-7200U @2.5 GHz and RAM: 6 Gigabyte | GPU: Nvidia GeForce 920 m |
|---|---|---|
| Original YOLOv3 | 0.79 | 0.51 |
| The proposed FED detector | 0.57 | 0.28 |

83.28%. The detector of [16] which has led to the aforementioned results, produces 61,587,112 parameters while the number of our proposed detector's parameters is 13,680,256 that is approximately 5 times as less as [16]. All of these show the superiority of the proposed detector over the detector of [16] in terms of parameter numbers.

Other approaches such as [42] and [43] have utilized the concept of Faster R-CNN to detect WBC and the average precision of [42] and [43] is 98.40% and 74% for detecting WBC respectively.

## 5. Discussion

An automated count of major blood cells is usually performed via Flow Cytometry Instrumentation [44]. But, in the real clinical situation for medical diagnostics, an important challenge of detectors, which are developed using computer vision methods, is that small changes, which may occur for many reasons in image sets, may cause the need for re-training [18,19].

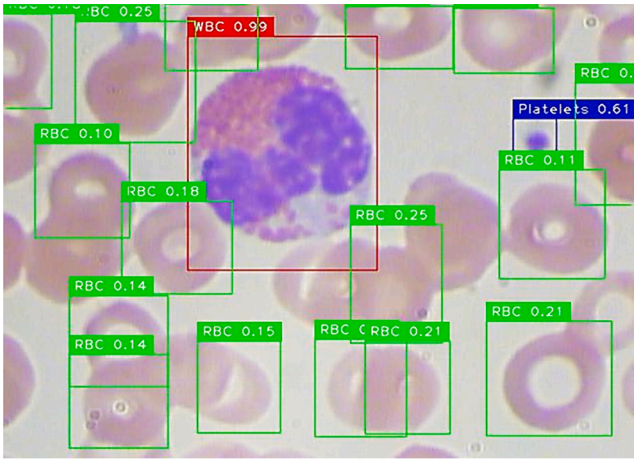
In this research, the proposed FED detector has utilized Convolutional Neural Networks as a feature extractor in the backbone, and such networks are not so sensitive to small changes instead mostly extract high-level features. Consequently, small changes are not able to reduce the accuracy of the system noticeably [27].

Besides, by transfer learning technique in computer vision, the proposed FED detector actually does not need a time-consuming, re-training if needed. If a large number of new images are accessible after the first training, transfer of weights and performing a just light re-training with small number of epochs can be considered as an effective way [45].

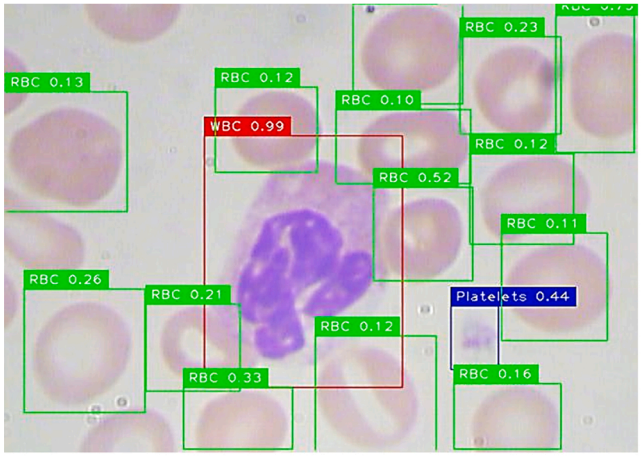
In addition, the neural networks have significant generalizability due to their high function- approximation property. Therefore, they can solve many new problems very well according to their last training. This is one of the reasons that the proposed FED detector does not need re-training in most cases; thanks to the EfficientNet neural network as the backbone. Conclusively, it is much more probable that the systems such as the proposed FED detector overcomes devices like the flow cytometry instrumentation in near future in terms of automatic counting blood cells; thanks to the remarkable progress of Artificial Intelligence [46].

## 6. Conclusion

In this research, a fast and yet efficient object detection method for blood cell detection has been proposed which utilizes compound scaling




a)




b)

**Fig. 12.** The visualization detection result of the proposed FED detector for two test images.

**Table 6**
Results of the proposed method.

| Number of parameters | Training rate %, test rate % | Average Precision (%) | Mean average precision (mAP %) |
|---|---|---|---|
| 13,680,256 | Based on the BCCD dataset's protocol (80%, 20%) | Platelets = 90.25<br>RBC = 80.41<br>WBC = 98.92 | 89.86 |

method, Dilated Convolution, Depthwise Separable Convolution, Swish activation function, and DIoU for bounding box regression. The results of experiments and comparisons demonstrate that the proposed FED detector is more efficient than other existing methods for blood cell detection. In this study, the Depthwise Separable Convolution method could dramatically reduce the number of parameters of the FED detector. Hence, it can be an appropriate choice for embedded systems and portable equipment. Moreover, the FED detector can be trained on a mediocre GPU and CPU. Besides, the proposed FED is a flexible detector, because its backbone can be changed to other Efficient Net's version (to achieve a higher accuracy or lighter model) without programming difficulties.

Due to the high importance of activation functions and their direct impact on the models, the proposed method employs the Swish activation function. The results of experiments show that it improves efficiency compared with other functions such as LeakyReLU. In addition, in this study, for bounding box regression and improving loss function, the DIoU method has been applied. This method minimizes the normalized distance between central points of two bounding boxes directly which leads to much faster convergence than other methods such as IoU and GIoU.

Due to the pandemic of COVID-19 coronavirus and late diagnosing dangerous, by considering its effect on the number of white blood cells, it can be vital to develop a Computer-Aided Diagnosis system using the proposed FED detector for counting white blood cells to trace symptoms of COVID-19 disease. This solution can be considered as future research. Besides, expanding the FED detector on chest CT-Scan or X-Ray imaging for detecting COVID-19′s infection and damaged areas of lungs can be also considered for future work.

## CRediT authorship contribution statement

**Ashkan Shakarami:** Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing - review & editing, Resources. **Mohammad Bagher Menhaj:** Supervision, Formal analysis, Validation, Writing - review & editing. **Ali Mahdavi-Hormat:** Conceptualization, Supervision, Formal analysis, Validation. **Hadis Tarrah:** Validation, Writing - review & editing.

## Declaration of Competing Interest

The authors report no declarations of interest.

## References

[1] C.G. Atkins, K. Buckley, M.W. Blades, R.F. Turner, Raman spectroscopy of blood and blood components, Appl. Spectrosc. 71 (5) (2017) 767–793.
[2] O. Garraud, J.D. Tissot, Blood and blood components: from similarities to differences, Front. Med. (Lausanne) 5 (2018) 84.
[3] S. Biswas, D. Ghoshal, Blood cell detection using thresholding estimation based watershed transformation with Sobel filter in frequency domain, Procedia Comput. Sci. 89 (2016) 651–657.
[4] P. Tiwari, J. Qian, Q. Li, B. Wang, D. Gupta, A. Khanna, et al., Detection of subtype blood cells using deep learning, Cogn. Syst. Res. 52 (2018) 1036–1044.
[5] P. Soviany, R.T. Ionescu, Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction, in: 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), September, 2018, pp. 209–214. IEEE.
[6] Z. Zhang, X. Wang, C. Jung, DCSR: dilated convolutions for single image super-resolution, Ieee Trans. Image Process. 28 (4) (2018) 1625–1635.
[7] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[8] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, Proceedings of the IEEE InternationalConference on ComputerVision (2017) 2961–2969.
[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, Ssd: single shot multibox detector, in: European Conference on ComputerVision, October, Springer, Cham, 2016, pp. 21–37.
[10] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, Proceedings of the IEEE Conference on ComputerVision and PatternRecognition (2016) 779–788.
[11] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, Proceedings of the IEEE Conference on ComputerVision and PatternRecognition (2017) 7263–7271.
[12] J. Redmon, A. Farhadi, Yolov3: An Incremental Improvement, arXiv preprint arXiv: 1804.02767, 2018.
[13] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, R. Qu, A survey of deep learning-based object detection, IEEE Access 7 (2019) 128837–128868.
[14] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, Proceedings of the IEEE Conference on ComputerVision and PatternRecognition (2014) 580–587.
[15] R. Girshick, Fast r-cnn, Proceedings of the IEEE InternationalConference on ComputerVision (2015) 1440–1448.
[16] D. Zhang, P. Zhang, L. Wang, Cell countingalgorithmbased on YOLOv3 and imagedensityestimation, in: 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP), July, 2019, pp. 920–924. IEEE.
[17] F. Chollet, Xception: deep learning with depthwise separable convolutions, Proceedings of the IEEE Conference on ComputerVision and PatternRecognition (2017) 1251–1258.
[18] Y. Li, Z. Han, H. Xu, L. Liu, X. Li, K. Zhang, YOLOv3-lite: a lightweight crack detection network for aircraft structure based on depthwise separable convolutions, Appl. Sci. 9 (18) (2019) 3781.

[19] Y. Li, A. Mahjoubfar, C.L. Chen, K.R. Niazi, L. Pei, B. Jalali, Deep cytometry: deep learning with real-time inference in cell sorting and flow cytometry, Sci. Rep. 9 (1) (2019) 1–12.
[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, Proceedings of the IEEE Conference on ComputerVision and PatternRecognition (2015) 1–9.
[21] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: European Conference on ComputerVision, October, Springer, Cham, 2016, pp. 630–645.
[22] A. Sepas-Moghaddam, A. Etemad, F. Pereira, P.L. Correia, Facial emotion recognition using light field images with deep attention-based bidirectional LSTM, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May, 2020, pp. 3367–3371. IEEE.
[23] A. Shakarami, H. Tarrah, A. Mahdavi-Hormat, A CAD system for diagnosing Alzheimer's disease using 2D slices and an improved AlexNet-SVM method, Optik (2020) 164237.
[24] S. Zagoruyko, N. Komodakis, Wide Residual Networks, arXiv preprint arXiv: 1605.07146, 2016.
[25] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications, arXiv preprint arXiv:1704.04861, 2017.
[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.C. Chen, Mobilenetv2: inverted residuals and linear bottlenecks, Proceedings of the IEEE Conference on ComputerVision and PatternRecognition (2018) 4510–4520.
[27] M. Tan, Q.V. Le, Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks, arXiv preprint arXiv:1905.11946, 2019.
[28] A. Howard, M. Sandler, G. Chu, L.C. Chen, B. Chen, M. Tan, et al., Searching for mobilenetv3, Proceedings of the IEEE International Conference on Computer Vision (2019) 1314–1324.
[29] A. Shakarami, H. Tarrah, An efficient image descriptor for image classification and CBIR, Optik (2020) 164833.
[30] V. Dumoulin, F. Visin, A Guide to Convolution Arithmetic for Deep Learning, arXiv preprint arXiv:1603.07285, 2016.
[31] Y. Li, X. Zhang, D. Chen, Csrnet: dilated convolutional neural networks for understanding the highly congested scenes, Proceedings of the IEEE Conference on ComputerVision and PatternRecognition (2018) 1091–1100.
[32] P. Ramachandran, B. Zoph, Q.V. Le, Searching for Activation Functions, arXiv preprint arXiv:1710.05941, 2017.
[33] N. Patwardhan, M. Ingalhalikar, R. Walambe, ARiA: Utilizing Richard's Curve for Controlling the Non-monotonicity of the Activation Function in Deep Neural Nets, arXiv preprint arXiv:1805.08878, 2018.
[34] H. Ma, Y. Liu, Y. Ren, J. Yu, Detection of collapsed buildings in post-earthquake remote sensing images based on the improved YOLOv3, Remote Sens. (Basel) 12 (1) (2020) 44.
[35] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression, arXiv preprint arXiv:1911.08287, 2019.
[36] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: a metric and a loss for bounding box regression, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019) 658–666.
[37] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in: Proceedings of the 23rd InternationalConference on Machine Learning, June, 2006, pp. 233–240.
[38] D.M. Powers, Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation, 2011.
[39] D.P. Kingma, J. Ba, Adam: a Method for Stochastic Optimization, *arXiv preprint arXiv:1412.6980*, 2014.
[40] L. Perez, J. Wang, The Effectiveness of Data Augmentation in Image Classification Using Deep Learning, arXiv preprint arXiv:1712.04621, 2017.
[41] R.B. Rao, G. Fung, R. Rosales, On the dangers of cross-validation. An experimental evaluation, in: Proceedings of the 2008 SIAM International Conference on Data Mining, April, 2008, pp. 588–596. Society for Industrial and Applied Mathematics.
[42] T. Xia, R. Jiang, Y.Q. Fu, N. Jin, Automated bloodcelldetection and counting via deeplearning for microfluidic Point-of-caremedicaldevices, in: IOP Conference Series: Materials Science and Engineering, October, 2019 (Vol. 646, No. 1, p. 012048). IOP Publishing.
[43] H. Kutlu, E. Avci, F. Özyurt, White blood cells detection and classification based on regional convolutional neural networks, Med. Hypotheses 135 (2020), 109472.
[44] M. Büscher, Flow cytometry instrumentation–An overview, Curr. Protoc. Cytom. 87 (1) (2019) e52.
[45] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2009) 1345–1359.
[46] C. Chen, W. Bai, R.H. Davies, A.N. Bhuva, C.H. Manisty, J.B. Augusto, et al., Improving the generalizability of convolutional neural network-based segmentation on CMR images, Front. Cardiovasc. Med. 7 (2020) 105.