ITC 3/54

Information Technology and Control
Vol. 54 / No. 3/ 2025
pp. 918-936
DOI 10.5755/j01.itc.54.3.39342

Detection and Classification of Blood Cells Using Different Deep Learning Approaches

Received 2024/11/01 | Accepted after revision 2025/01/25

HOW TO CITE: Çakir, M. M., Çinarer, G. (2025). Detection and Classification of Blood Cells Using Different Deep Learning Approaches. *Information Technology and Control*, 54(3), 918-936. https://doi.org/10.5755/j01.itc.54.3.39342

Detection and Classification of Blood Cells Using Different Deep Learning Approaches

Mübarek Mazhar Çakir
Yozgat Bozok University, Department of Mechatronics Engineering, 66100, Yozgat, Turkey;
e-mail: mashar66@gmail.com

Gökalp Çinarer*
Yozgat Bozok University, Department of Computer Engineering, 66100, Yozgat, Turkey

Corresponding author: gokalp.cinarer@bozok.edu.tr

Blood cells have an important place in the human immune system. The amount of blood cells in the blood is used to determine whether human health is normal or unusual. For this reason, detecting and determining the amount of RBC, WBC and Platelets elements in the blood is very important for human health. In the management of all these processes, basic factors such as the complexity of cell structures, loss of time, and the necessity of expert opinion make the realization of these processes very complicated. In this study, cell detection was carried out using models of Detectron2 and Yolo algorithms to automatically detect and quantify blood cells quickly. The BCCD dataset was used to run the models. In the study, the performance results of the models of 21 different most recent artificial intelligence algorithms in detecting blood cells were analyzed. In this comprehensive research, the accuracy values of the train and test process of the models were examined comparatively, and the most suitable model was determined, which aims to provide a high success rate for small models. As a result of the study, the highest AP value with 93.7% in the train result among 5 models belonging to Detectron2 from 21 models belongs to the Faster R-CNN X_101_32x8d_FPN_3x and Faster R-CNN R_101_C4_3x models; Among the 16 models belonging to Yolo, the highest AP value as a result of the train belongs to the Yolov7-w6 model with 95.8%. When the test results are examined, among the Detectron2 models, the Faster R-CNN R_101_FPN_3x model achieved 90.3% AP value. In addition, among the Yolo models, the Yolov5-s model was the most successful algorithm with an AP value of 94.5%. When the train and test results of the models, training time, weight size values were examined, it was determined that the Yolov5-s model was the most successful model in classifying and detecting blood cells in the BCCD dataset.

KEYWORDS: Object detection, Yolo, R-CNN, Artificial Intelligence, Deep learning, Blood cells.

1. Introduction

Technological developments have been widely used in recent years to increase efficiency and reduce costs in the health sector as well as in every field. Especially in the field of medical imaging, the number of artificial intelligence applications is constantly increasing and the developed systems reduce the costs and reduce the workforce. In addition, since features such as blood cell structures and cell numbers are important effective parameters in the diagnosis and treatment of the disease in clinical studies, quantitative analysis of cell images and artificial intelligence-based detection are frequently used in these areas. For automatic detection of blood cells, it is necessary to separate the cells from the image background with conventional image processing methods. In this process, unnecessary workload arises due to the complexity of the image background and the similarity of cell structures. In addition to this workload loss of time, there is also a loss of quality and efficiency in the detection and classification of cells. To overcome all these problems, deep learning (DL) based object recognition algorithms were used.

In the field of health, the doctor or specialist should examine the patient and decide whether the disease is present by examining the results of the patient's analysis. It is vital for the patient that this process takes place quickly and accurately. The human-like decision-making structure of artificial intelligence algorithms, continuous development and continuous optimization studies have made the use of artificial intelligence in the field of medical imaging widespread [3]. In this study, it is aimed to determine the diagnosis of a person's health status by deep learning method, depending on the RBC, WBC and Platelets elements in the blood. By looking at the number of RBC in a person's blood, information can be obtained about the presence of heart failure and coronary artery disease [7]. WBC count was determined to be proportional to AIS disease in a person [35]. Again, WBC count is associated with anhedonia, fatigue, slowing down, low appetite, while platelet count is associated with insomnia, restlessness, suicidal ideation and suicide attempt [13].

The main contributions of this study are:

- Processing of image data of blood cells and classification with different deep learning models;
- Acceleration of blood analysis process with deep learning method considering that these elements in blood can be detected as a result of long-term training and experience in laboratory environment;
- Performing blood cell detection using weight files obtained from deep learning models to perform blood analysis;
- Comparative analysis of performance times, processing speeds and accuracy results of current object detection algorithms.

In the continuation of the study, firstly, the studies in the literature related to automatic object detection were examined. After the literature review, the algorithms and architectural structures used in object detection, the dataset used and their details were researched and shared. In line with the research, the models to be used in the study were determined and the details of the operation of the determined models were shared. The experimental results obtained from the models were evaluated comparatively and as a result of the evaluation, the most suitable model for the dataset was determined. The contributions of the study are mentioned by comparing with the studies in the literature that detect blood cells. Finally, details of future studies are given.

2. Related Works

Related works on object detection with deep learning in the last three years were examined. Details such as the results of the works and the algorithms used are shared. Leng et al. [12] used the Raabin dataset to detect leukocytes with blood cells. The dataset used was composed of three classes: eosinophil, monocyte and neutrophil. Instead of using normal CNN, Pyramid Vision Transformer was used in the backbone structure of the DETR algorithm. In addition to changing the backbone structure, the DAM structure, which interpolates position values and applies them to weight values, has been added to the DETR algorithm. Yolov3-TL, Faster R-CNN-TL, DETR-TL, Improved DETR-TL, Improved DETR-FS models were trained with the dataset. According to the results obtained, it was stated that the Improved DETR-TL model showed the best performance with a mAP value of 96.1%. Stating that platelet detection is important in evaluating blood status and liver diseases, Liu et al. [16] performed deep learning-based platelet detection. Researchers argued that datasets containing blood cells were insufficient and difficult to access and created a single-class dataset consisting of 412 images. The dataset is divided into three, 296 images are reserved for use in the training process, 33 images are reserved for use in the validation process, and the remaining 83 images are reserved for use in the testing process. Then, the number of iterations was set to 100, the batch size value was set to 4, the learning rate was set to 0.001 for the first 50 iterations and 0.0001 thereafter. Models were run with the adjusted hyperparameter values and it was determined that the highest AP value of 84.2% belonged to the Yolov3 model. Then, ablation experiments including multiscale fusion, anchor box clustering and match parameter structures were performed on the Yolov3 algorithm. As a result of the studies, it has been shown that the Yolov3 model, with simultaneous changes in the match parameter and anchor box clustering, reached an AP value of 87.3%. Hu et al. [4], who worked on feature extraction of small objects, used the BCCD dataset to detect blood cells. Based on the Faster R-CNN model, two different backbone structures were used: ResNet50 and FPN. Additionally, models were created using two different frameworks: Pytorch and MindSpore. In the created models, the optimization algorithm was SGD, the momentum value was 0.9, and the learning rate was 0.05. Pixel level balancing operations were used selectively in different parts of four different loss functions. As a result of the studies, a mAP value of 63.7% was obtained in the model in which two different pixel level balancing processes were used together, running on the Pytorch framework. Rahaman et al. [21] used the BCCD dataset in the study to enable doctors and physicians to automatically detect blood cells from real images. It has been stated that platelet count is common for COVID-19 and some diseases. In the study, Yolov5-s and Yolov5-m models were trained with the BCCD dataset and comparative results were shared. According to the results obtained, the precision value of Yolov5-s was recorded as 79.7%, while the precision value of the Yolov5-m model was recorded as 79.9%. As a result, it has been suggested that Yolov5-m is more successful in detecting and counting blood cells. To automatically identify and locate WBC types from blood cells, Kutlu et al. [11] conducted a study. BCCD and LISC datasets were combined to detect five WBC types called Eosinophils, Basophils, Neurophils, Monocytes and Lymphocytes. The size of 6250 images in the created dataset is set to 256x256. Four different models based on R-CNN were trained: VGG16, ResNet50, AlexNet and GoogLeNet. As a result of the studies, the ResNet50 model demonstrated higher performance than other models with a mAP value of 74%.

3. Material and Methods

3.1. Object Detection and Deep Learning

In recent years, it has been strengthened with artificial intelligence-based components. AI algorithm systems provide both faster and more accurate object detection. Object detection in the field of computer vision has emerged as one of the issues to be overcome in visual recognition. Object detection is the process of identifying the position of a detected or targeted object on the image by separating it from the image background and identifying which class the object belongs to [38]. Its success rate and high working speed compared to traditional methods have increased the use of DL algorithms in object detection. DL is a sub-branch of machine learning consisting of many layers and nonlinear processing units [36]. Thanks to CNN, deep learning learns the basic shapes in the first layer and the features on the images (such as color, shape, texture) in the next layers, that is, in the deeper layers [9].

3.2. Faster R-CNN

Faster R-CNN is one of the deep neural networks designed for automatic detection of target objects. When the development process was examined, the R-CNN system was first developed. R-CNN creates 2000 region suggestions on an image and feeds each region suggestion one by one to the CNN, acting as a feature extractor and creating a 4096-dimensional feature vector [18]. In convolutional meshing of R-CNN, applying forward pass for each object increases the computational load. In this situation, SPPnet and Fast R-CNN, in which the input image passes through the CNN in one go, are proposed for solving problem [6]. This process can be ordered as identifying region suggestions from the image feature map, resizing it in the ROI pooling layer, and then defining the objects in each region [18]. The innovations made in Fast R-CNN made it faster than R-CNN. However, the desired speed could not be reached. Developed Faster R-CNN, which uses the RPN structure to suggest quality region suggestions by suggesting the boundaries and accuracy scores of the objects in each location to increase temporal problems and accuracy [18].

3.3. Detectron2

Pytorch-based Detectron2 was launched by the Facebook AI team for use in various object detection tasks [34]. Considering the structure in which Base-RCNN and FPN structure are used together, it consists of three parts: backbone network FPN, RPN and ROI head (box head) [26]. The backbone network FPN extracts the characteristics of the input image; RPN can find objects' box locations and classification labels [30].

There are many state-of-the-art models of Detectron2. The difference between these models is due to the changes in the backbone, number of layers, backbone combination and learning rate (LR) scheduler structures. At the same time, these differences determine the names of the models. Backbone appears in two variants: ResNet (R) or ResNext (X); the number of layers is 50 or 101; backbone combination: Combination of ResNet and FPN, Conv5 head with ResNet Conv4 backbone (c4) or ResNet Conv5 with dilations (DC5); 1x or 3x is used for LR scheduler [17].

For example, when the Faster R-CNN R_50_FPN_1x model is examined: 50-layer ResNet is used as backbone, an FPN is used and 1x is used as LR scheduler.

3.4. Yolo

Yolo is an object detection algorithm created by Redmon J. and his team, using a neural network called Darknet, which leads to faster and more accurate results than its peers [27]. After the creation of Yolo, the improvement work on the algorithm continued. The aim of these improvement and optimization studies is to increase the performance of the Yolo algorithm in terms of speed and accuracy [22]. The third version of Yolo, which uses the Darknet-53 network structure by adding 53 more layers on top of the layers used in Yolov2, and can be classified into multi-labels [23]. Yolov3, CSPDarknet53, SPP and PAN are used in the backbone and neck parts, respectively, while the head part is the same as the head part of Yolov3 [2]. Yolov5 consists of four main parts: the network input section where images, namely data, are entered, the backbone where the features of the target objects are extracted, the neck and the head, where the extracted features are parsed [8]. While the focus structure undertakes the slicing of the image without entering the backbone part, the CSP structure in the backbone part increases the learning ability of CNNs and reduces memory usage [37]. In addition, it has strong semantic and positioning features with the FPN layer in the Yolov5 structure, improving the detection ability of objects of different scales [39]. Yolov7 is the use of ELAN structure in the backbone to gain the ability to continuously improve its learning ability; Again, in the backbone part, using the MP structure and ELAN together to extract the image features is one of the innovations [29]. The developers argue that RepConv will break the network structure as it will cause too much gradient diversity, and they designed RepConv's reparameterized convolution instead of the identity link [33]. When the Yolov7 architecture is examined, we see that the neck part of the model and the head part are combined and again called the head. This is one of the changes Yolov7 developers have made on the network. In addition, the new labeling method used is a feature that distinguishes the Yolov7 from other models. In this labeling structure, there are two heads called aux head and lead head that work in a hierarchy. The lead head has a stronger learning structure and the aux head is weaker in this regard. The hierarchy between two heads with different learning performances is that the learned information is transmitted by the lead head to the aux head and the lead head better focuses on the learning process of new information. The current released version of Yolov8 provides the ability to perform detection, segmentation and classification tasks [5]. In addition, there are five different models named n, s, m, l, x belonging to Yolov8. Yolo v8n is a model that requires low hardware requirements and has a fast processing time. In addition, while the accuracy increases in the Yolov8s model, the number of parameters also increases. The speed and resource requirements are higher in the Yolo v8m model. While Yolo v8l and Yolov8x reach maximum accuracy, they also need high hardware requirements. Although the number of parameters is low in small models such as Yolo v8n, the fps value is higher. In contrast, the FLOPs value is high and the FPS value is low in large models such as Yolov8x.

Considering the GPU requirements, large models need more performance. Latency represents the time a model takes to process the image and is usually shown in milliseconds. When Yolov8 models are examined, GPU and CPU latency is the highest in the Yolov8x model. In YOLO models, the width (α) and depth (β) factors allow the depth and width dimensions of the model to be interpreted. These values increase optimization and slow down the speed of the model in high models (l,x), while in low models (n,s) they give faster results.

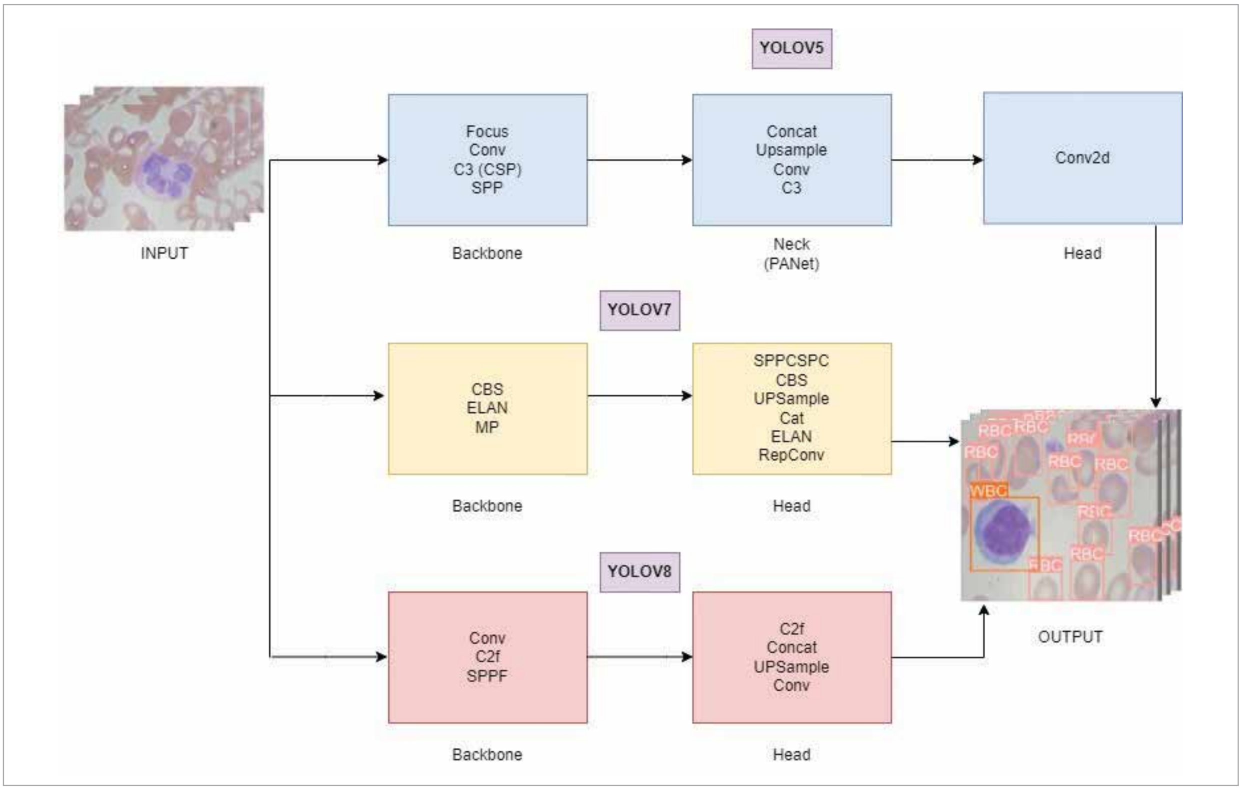
As seen in Table 1, the differences between the models are due to the changes in the depth multiple (d), width multiple (w) and ratio (r) values.

Since the anchor box is not used in the Yolov8 model, the center point of the object is estimated directly, indicating that the Anchor-Base was abandoned and the Anchor-Free idea was adopted. Another advantage of detection without anchor box is that it speeds up the Non-Maximum Suppression process due to the reduction in the number of predicted boxes [28]. In addition, in some deep learning models, data augmentation is performed during online training. One of these data augmentation processes is mosaic augmentation. Doing this throughout the entire train-

Table 1
Depth Multiple, Width Multiple and Ratio Values of Yolov8 Models

Models	d (depth multiple)	w (width multiple)	r (ratio)
Yolov8-n	0.33	0.25	2.0
Yolov8-s	0.33	0.50	2.0
Yolov8-m	0.67	0.75	1.5
Yolov8-l	1.00	1.00	1.0
Yolov8-x	1.00	1.25	1.0

Figure 1
Architecture of Yolo Models.



ing process greatly reduces performance. Mosaic augmentation process has been turned off in the last ten epochs to avoid the mentioned performance degradation. Due to the conflicts between classification and regression processes in object detection, Yolov8 uses the decoupled head for classification and localization. CIoU+DFL is used for Bbox. loss and BCE is used for class loss. CIoU loss includes distance, aspect ratio, and overlapping area geometric factors all. For this reason, it is called Complete IoU. DFL treats the continuous distribution of box positions as discrete probability. BCE creates a benchmark that measures the Binary Cross Entropy value between the target and the output.

C2f combined all output from Bottleneck, while C3 used only the last output. This reason in the first part of the stem, C2f, not C3, was used as the main structure, and 3x3 conv was chosen instead of 6x6 conv [14]. In the Bottleneck part, unlike Yolov5, the kernel size has been changed to 3x3. Finally, the same channel sizes are combined in the neck to reduce the overall size and number of parameters of the tensors.

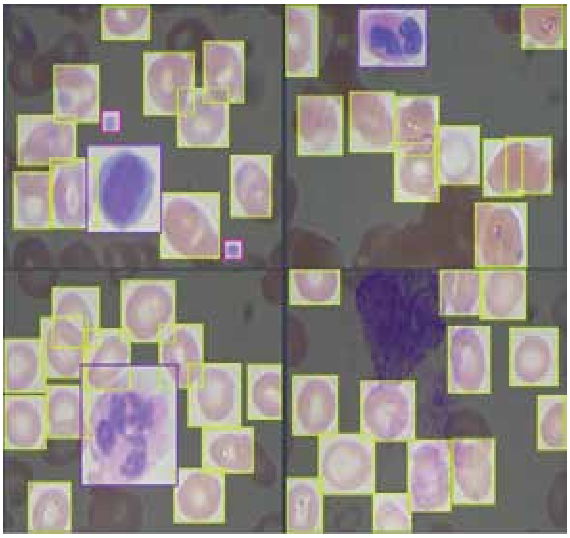
Details of the structures of the article Yolov5, Yolov7 and Yolov8 models are given in Figure 1.

3.5. Dataset

The most suitable dataset for this study was searched and it was deemed appropriate to use the images in the dataset named BCCD [25] in this study. There are three different classes in the dataset and a total of 364 images of the dataset were used. These classes are RBC, WBC and Platelets. There is a total of 4888 labels belonging to three classes. There are 36 images and 489 labels containing blood cells in the test set. Of the 489 tags in the test set, 36 belong to the Platelets class, 35 belong to the WBC class, and the remaining 418 tags belong to the RBC class. Labeled images of the dataset are shown in Figure 2.

File formats in which labeling information is hidden have been arranged for use in models of dataset Yolo and models of Detectron2. In Yolo models, txt files (364 txt files) for each image are stored, and a json file (3 json files) where label information is stored for each dataset group (train, test, validation) in Detectron2 models.

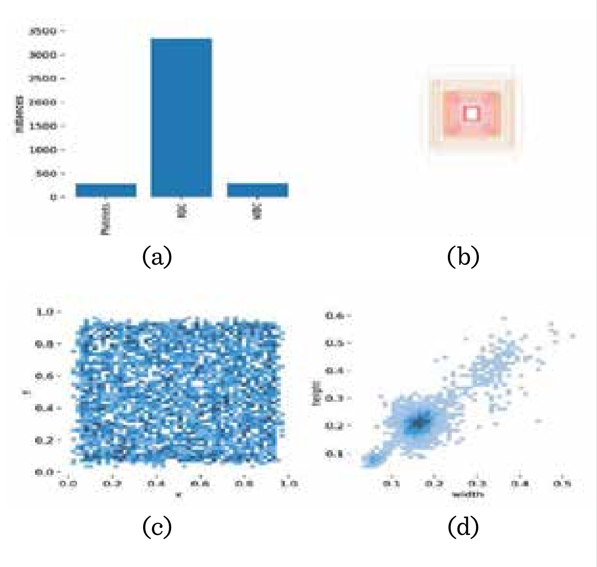
Figure 2
Image of Dataset.



The labels distribution of the classes of the image data in the dataset is given in Figure 3. When Figure 3 is examined, graphics showing the location, width and height of the label boundaries of the RBC, WBC, Platelets classes in the dataset are seen. The number of images belonging to each classification set is given in Figure 3(a). The distribution of RBC, WBC and platelet classes in the dataset is shown. It is understood from here that the class with the most data entry is the RBC class. Data entry of Platelets and WBC classes are close to each other. What we are referring to as data entry here is the number of tagged objects in the images in the dataset. Figure 3(b) chart represents the boundaries of the labels in the image data in the dataset. The abundance of the same label boundaries can be understood by the darkening of red color. It can be seen that the data density is concentrated in the center. This indicates that the data is clustered in a particular class or feature set. Graph in Figure 3(c) gives the coordinates of the center points of the labeled blood cells on the image. Here the image size is considered as 1.0 x 1.0 and the center point is determined on the x and y axis. It can be understood from the darkening of the blue color in which region of the image the center points of the labeled blood cells are located within the entire dataset. The distribution is seen to be quite homogeneous and random. Moreover, no specific pattern or relationship was detected. This may indicate that there is no significant correlation between the two variables. Figure 3(d) graph gives the width and height information on the image according to the center points of the labeled blood cells. Here, the image size is considered as 1.0 x 1.0 and represents the size of blood cells according to their height and width information in the same ratio of image. The graph shows a relationship between width and height, and a dense clustering in the center. This may indicate that the data is generally of similar size (e.g., object detection frame sizes), while the sparse points around it represent samples that deviate from the standard sizes. It helps us understand whether the dataset contains mostly small objects or large objects. If the objects throughout the dataset are small, the blue color on the graph becomes darker near the zero value; however, if objects are large, the blue color becomes darker near the value 1.

Figure 3

Distribution of Labels.



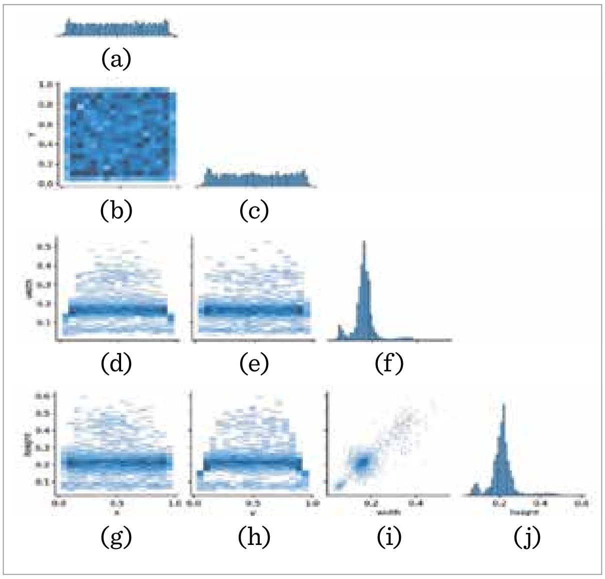
(a) (b)

(c) (d)

One of the best ways to analyze data is with a correlogram. Correlogram is the visualization of the relationships of the data in the dataset. The labels correlogram in Figure 4 is a 2d histogram that shows the axes of the image data relative to all axes in the xywh space. It shows the common features and differences between the labels. It specifies the common properties of the object classes in the dataset of the object detection model and the ability to capture special properties for each object class. Figure 4(a) graph shows the distribution of the center points of blood cells on the x-axis in the images in the entire dataset. Figure 4(b) graph shows the distribution of the center points of all labeled blood cells on the x and y axis. The more blood cell center points there are on the same axes of different images in the data set, the darker the color of the graph will be in that region. For example, if there are blood cells at the same point in 3 different images, this area will appear darker on the graph. Figure 4(c) graph shows the distribution of the center points of blood cells on the y-axis in the images in the entire dataset. The distribution seems balanced across the dataset. Figure 4(d) graph shows the width distribution of the center point of blood cells according to the x-axis. From this graph, information can be obtained on how many blood cells on the same x-axis are of the same width in different images in the dataset. If there are blood cells of the same width on the same axis of different images, that region is shown darker in the graph. Figure 4(e) graph shows the width distribution of the center point of blood cells according to the y-axis. From this graph, information can be obtained on how many blood cells on the same y-axis are of the same width in different images in the dataset. If there are blood cells of the same width on the same axis of different images, that region is shown darker in the graph. The Figure 4(f) graph shows the width distribution of blood cells over the images in the dataset. In order to see what width the blood cells are at most, it is necessary to look at the longest column on the chart. Here it looks like it's between 0.1 and 0.2. The Figure 4(g) graph shows the height distribution of the center point of blood cells according to the x-axis. From this graph, information can be obtained on how many blood cells on the same x-axis are of the same height in different images in the dataset. If there are blood cells of the same height on the same axis of different images, that region is shown darker in the graph. Figure 4(h) graph shows the height distribution of the center point of blood cells according to the y-axis. From this graph, information can be obtained on how many blood cells on the same y-axis are of the same height in different images in the dataset. If there are blood cells of the same height on the same axis of different images, that region is shown darker in the graph. Figure 4(I) graph and the Figure 3(d) graph are the same. This graph was interpreted while explaining Figure 3. Figure 4(j) graph shows the height distribution of blood cells over the images in the dataset. In order to see what height the blood cells are at most, it is necessary to look at the longest column on the chart. Here it looks like it is 0.2.

Figure 4

Labels Correlogram



Figures 3-4 provide information about the aspect ratios and position distribution of blood cells on the image. Since feature map extraction is determined according to anchor box sizes, the location distribution and aspect information of the objects in the dataset are of great importance. Thanks to the anchor boxes in the detection layers, the detection layer with the most appropriate anchor box according to the target object size is responsible for detecting the object. It is understood from this that the assignment of the object class is also based on the object size. During the object detection process, thousands of anchor boxes are created on the image and the intersection rate with the real object is calculated. Achieving the highest agreement with the real object boundaries ends with the detection of that object. For these reasons, appropriate anchor boxes should be created by examining the location and aspect ratio distribution of the objects in the dataset. The location information and aspect information of the blood cells in the BCCD dataset were examined and the default anchor box values of Yolov5 were found to be appropriate. An anchor box is not used in Yolov7 and Yolov8.

3.6. Running the Models

The Yolov5, Yolov7, Yolov8 and Detectron2 models were run on Google Colab, a virtual computer platform designed for artificial intelligence developers. The image data in the BCCD dataset were processed on Colab using Python software language and Pytorch, Keras, Tensorflow libraries. Of 364 image data, 292 (80%) were used for train, 36 (10%) for validation, and the remaining 36 (10%) for testing.

Transfer learning is to ensure that the model gives better performance by preventing the values in the weight file of a model to be trained as custom from being determined from scratch. Here, the train was performed based on the values in the weight files of the models trained with the COCO dataset.

The original BCCD dataset images are 640x418. In Yolo models, the width and length of the image must be equal in size adjustments. To avoid losing resolution the image data to be given as input for the train process of each model is set as 640x640. The epoch value, which means how many times the image data will be displayed to the neural networks, is determined as 50 and the iteration value is 2000. In each epoch period, images are displayed to the neural network as a group. The term describing the size of these image groups is called batch size. Batch size is set to 16 for all models. High GPU is required to increase the batch size value. For this reason, the batch size value is 16.

The learning rate, which works depending on the optimization algorithm and can be defined as the update rate of the parameters in the neural network, was determined as 0.01 in 16 models of Yolo and 0.001 in 5 models of Detectron2. SGD was used in all models as an optimization algorithm that increases the performance of the model in complex learning processes. The momentum value makes the SGD run faster. 0.937 for Yolo models and 0.9 for Detectron2 models. Finally, the activation function, which is a factor that increases the performance by converting the outputs to nonlinear values, was chosen as SiLu for Yolo models and ReLu for Detectron2 models. Additionally, the loss functions used in the models are as follows: rpn_losses function for Detectron2, GIoU for Yolov5, focal loss for Yolov7 and DFL Loss+CIOU for Yolov8.

The performance of the model was examined by measuring the correct and incorrect predictions of the model for classification, accuracy, precision, sensitivity, F1 score and ROC-AUC values.

4. Experimental Results

The train results of the Detectron2 models run with the BCCD dataset, the test results obtained by testing the models from the image data previously included in the dataset with data unknown to the models, the dimensions of the weight files obtained by running the models and the training time are given in Table 2.

When the training accuracy values of Detectron2 models are examined, the highest AP value is 93.7%, with Faster R-CNN R_101_C4_3x and Faster R-CNN X_101_32x8d_FPN_3x models. The model that follows these models is the Faster R-CNN R_101_FPN_3x model with an AP value of 88.7%. Looking at the test results, the model that gives the best results with an AP value of 90.3% is the Faster R-CNN R_101_FPN_3x model. The model that follows this model as a result of the test is the Faster R-CNN X_101_32x8d_FPN_3x model with an AP value of 88.9%. With the joint evaluation of Train and test results, we determine the model with the highest performance among the Detectron2 models. In this case, the Faster R-CNN X_101_32x8d_FPN_3x model with a value of 91.3% is the model with the highest performance among the Detectron2 models. When Detectron2 models are evaluated in terms of weight size and training time, the fastest model with 0.068 hours and the smallest weight size value with 126 MB is the Faster R-CNN R_50_C4_1x model. The models that follow this model with 0.086 hours, 0.109 hours training times and 158 MB, 199 MB weight size values are Faster R-CNN R_50_FPN_1x and Faster R-CNN R_101_C4_3x, respectively. In this case, the train AP and test AP values are low in the models where the training time and weight size values are low in Detectron2 models; It is seen that training time and weight size values are worse in models with high train AP and test AP values.

In Figure 5, there are classification loss charts of Detectron2 models. When Figure 5 is examined, the classification loss value decreases rapidly in the beginning in all models. This sudden and rapid decrease lasts up to approximately 400 iterations in all models. After this sudden decrease, the classification loss value in all models continues to decrease gradually horizontally. When the graph of the Faster R-CNN R_50_C4_1x model is examined, the classification loss value is approximately 0.25 when 2000 iterations are reached. When the graph of the Faster R-CNN R_50_FPN_1x model is examined, the classification loss value is approximately 0.3 when 2000 iterations are reached. When the remaining three graphs are examined, when 2000 iterations are reached in Faster R-CNN R_101_C4_3x and Faster R-CNN R_101_FPN_3x models, the classification loss value remains above 0.2, while Faster R -The value of 0.2 in CNN X_101_32x8d_FPN_3x model is closer than these two models.

In Figure 6, there are false negative results of Detectron2 models. False negative describes the situation

Figure 5

Classification Loss of Detectron2 Models.

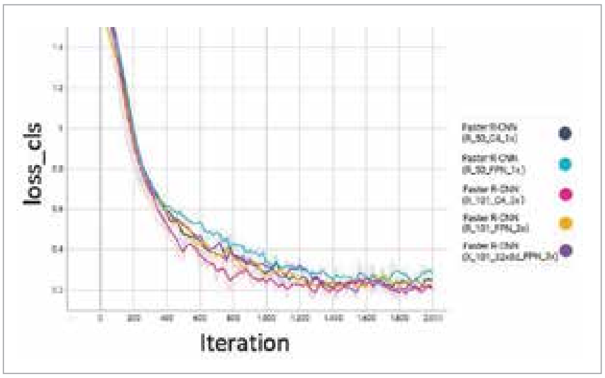


Table 2

Result For Detectron2 Models.

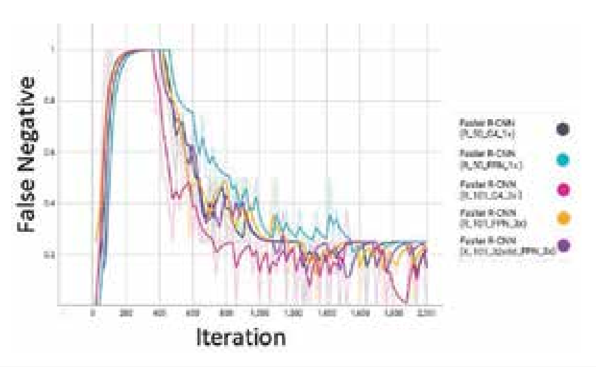
Models	Train Accuracy (AP_{50})	Test Accuracy (AP_{50})	Weight Size (Mb)	Train Time (h)
Faster R-CNN R_50_C4_1x	0.875	0.886	126	0.068
Faster R-CNN R_50_FPN_1x	0.875	0.868	158	0.086
Faster R-CNN R_101_C4_3x	0.937	0.836	199	0.109
Faster R-CNN R_101 FPN_3x	0.887	0.903	230	0.128
Faster R-CNN X_101_32x8d FPN_3x	0.937	0.889	399	0.256

where the model prediction is that the object does not exist, although there is a target object on the image. This means that the lower the false negative value, the better the performance of the model.

When Figure 6 is examined, the false negative value in all Faster R-CNN R_50_C4_1x, Faster R-CNN R_50_FPN_1x, Faster R-CNN R_101_C4_3x and Faster R-CNN R_101_FPN_3x models is between 20% and 30%. Faster R-CNN R_50_C4_1x and Faster R-CNN R_101_C4_3x models have a false negative value closer to 20%, thus outperforming Faster R-CNN R_50_FPN_1x and Faster R-CNN R_101_FPN_3x models. The false negative value of the Faster R-CNN X_101_32x8d_FPN_3x model is below 20%. This means that the Faster R-CNN X_101_32x8d_FPN_3x model outperforms the other four Detectron2 models in terms of false negatives.

Figure 6

False Negative of Detectron2 Models.



The train results of the Yolo models run with the BCCD dataset. Train results are examined, the model with the highest precision value is the Yolov5-s model with 91.6%. As a result of precision, this model is followed by Yolov7-w6 and Yolov8-x models with 91.1% and 90% values, respectively. The model with the highest value of 92.3% in terms of recall is the Yolov8-s model. Yolov8-s model is followed by Yolov5-m and Yolov8-n models, respectively, with recall values of 92% and 91.2%. The model with the highest F1 score value, which is used to evaluate the precision and recall values in a single unit, is the Yolov7-w6 model with 90.9%. Yolov7-w6 model is followed by Yolov8-s and Yolov8-n models with 89.9% and 89.7% values as F1 score.

The performance rate of the models is examined, the model with the highest success rate is the Yolov7-w6 model with an AP value of 95.8%. The Yolov7-w6 model is followed by the Yolov7-e6 and Yolov7-e6e models, with AP values of 95% and 94.9%, respectively.

Table 3 shows the training times of Yolo models and the weight file sizes obtained as a result of training. When the models are examined in terms of weight size, the smallest model with 3.72 MB is the Yolov5-n model. The Yolov8-n and Yolov5-s models follow the Yolov5-n model with a file size of 5.94 MB and 13.7 MB, respectively. When the training time results are examined, the fastest model is the Yolov5-n model with a training time value of 0.083 h. In terms of speed, Yolov5-n model is followed by Yolov5-s and Yolov5-m models with training time values of 0.096 h and 0.126 h, respectively.

Table 3

Weight File Sizes and Training Time for Yolo Models.

Models	Weight Size (Mb)	Training Time (h)
Yolov5-n	3.72	0.083
Yolov5-s	13. 7	0.096
Yolov5-m	40.2	0.126
Yolov5-l	88.5	0.19
Yolov5-x	165	0.344
Yolov7	71.3	0.288
Yolov7-x	135	0.424
Yolov7-w6	154	0.364
Yolov7-e6	211	0.505
Yolov7-d6	292	0.611
Yolov7-e6e	315	0.675
Yolov8-n	5.94	0.131
Yolov8-s	21.4	0.143
Yolov8-m	49.5	0.202
Yolov8-l	83.5	0.274
Yolov8-x	130	0.403

The results obtained by testing the image data in the BCCD dataset with data that the Yolo models have never seen are given in Table 4. As a result of the examination of the models in terms of the precision value in the test result, it is seen that the highest precision value among the Yolo models is the Yolov5-x model with 89.1%. This model is followed by Yolov5-l and Yolov7-x models, with precision values of 88.8% and 88.7%, respectively. When the recall value of the test results is examined, the highest recall value is the Yolov7-w6 and Yolov7-e6 models with 92.7%. These two models are followed by Yolov7-d6 and Yolov5-m models with recall values of 91.8% and 91.7%, respectively. The model with the best performance in F1 score results is the Yolov8-n model with 89.5%. The model that follows the Yolov8-n model with an F1 score of 88.6% is the Yolov5-s model.

The most important among all results is the success rate of the model in the testing process. In this respect, the highest AP value of 94.5% belongs to the Yolov5-s model. The models that follow this model as a test AP value are the Yolov5-m and Yolov5-l models. The AP value of the Yolov5-m model is 94% and the AP value of the Yolov5-l model is 93.3%.

Table 4

Testing Result for Yolo Models.

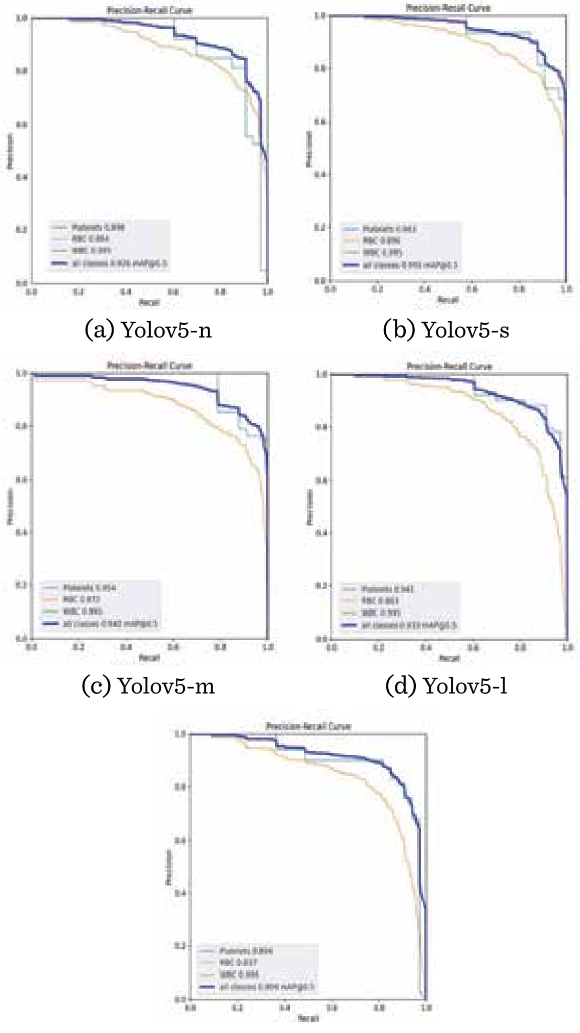
Models	Precision	Recall	F1 Score	AP_{50}
Yolov5-n	0.868	0.895	0.881	0.926
Yolov5-s	0.879	0.894	0.886	0.945
Yolov5-m	0.848	0.917	0.881	0.94
Yolov5-l	0.888	0.877	0.882	0.933
Yolov5-x	0.891	0.856	0.873	0.909
Yolov7	0.87	0.881	0.875	0.919
Yolov7-x	0.887	0.883	0.884	0.928
Yolov7-w6	0.837	0.927	0.879	0.923
Yolov7-e6	0.828	0.927	0.874	0.922
Yolov7-d6	0.84	0.918	0.877	0.904
Yolov7-e6e	0.828	0.916	0.869	0.931
Yolov8-n	0.884	0.908	0.895	0.93
Yolov8-s	0.866	0.905	0.885	0.919
Yolov8-m	0.867	0.905	0.885	0.924
Yolov8-l	0.85	0.901	0.874	0.926
Yolov8-x	0.846	0.904	0.874	0.908

The PR curves of the test results of the Yolov5 models are shown in Figure 7. When the PR curve is examined, the area under the curve gives us information about the success rate of the model. Here, a large area under the curve means that the model has a high success rate. Conversely, the small area under the curve indicates that the model's performance ratio is low. When Figure 7 is examined, the model in which the area under the curve is larger than the others is the Yolov5-s model. Therefore, we conclude that the Yolov5-s model performs better than other models.

When the performances of the models are evaluated separately for each class, the highest AP value of the Platelets class is the Yolov5-m model with 95.4%. This model is followed by Yolov5-s and Yolov5-l

Figure 7

Precision-Recall Curves of Yolov5 Models.



(a) Yolov5-n

(b) Yolov5-s

(c) Yolov5-m

(d) Yolov5-l

(e) Yolov5-x

models with 94.3% and 94.1% values, respectively. The highest AP value of the RBC class is the Yolov5-s model with 89.6%. The Yolov5-s model is followed by the Yolov5-n and Yolov5-m models, respectively, in terms of RBC class with an AP value of 88.4% and 87.2%. In the WBC class, all Yolov5 models showed the same performance with an AP value of 99.5%. The Yolov5 model with the highest AP value for all classes is the Yolov5-s model with 94.5%. With AP values of 94% and 93.3%, Yolov5-m and Yolov5-l models, respectively, follow the Yolov5-s model in performance.

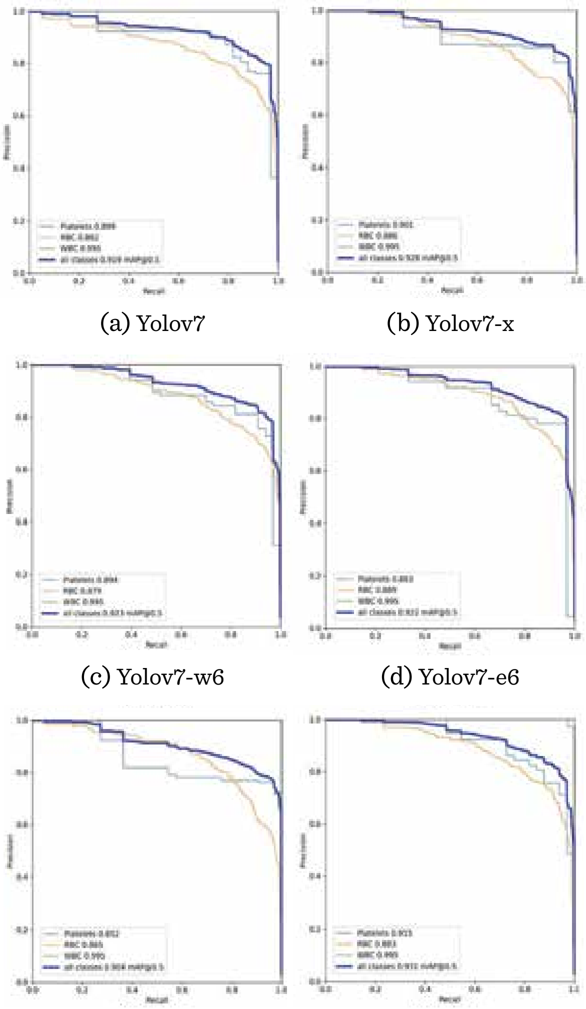
The PR curves of the test results of the Yolov7 models are shown in Figure 8. When Figure 8 is examined, the model in which the area under the curve is larger than the others is the Yolov7-e6e model. Therefore, we conclude that the Yolov7-e6e model performs better than other models. When the performances of the models are evaluated separately for each class, the highest AP value of the Platelets class is the Yolov7-e6e model with 91.5%. This model is followed by Yolov7-x and Yolov7 models with 90.1% and 89.9% values, respectively. The highest AP value of the RBC class is the Yolov7-e6 model with 88.9%. Yolov7-e6 model is followed by Yolov7-x and Yolov7-e6e models, respectively, in terms of RBC class with an AP value of 88.6% and 88.3%. In the WBC class, all Yolov7 models showed the same performance with an AP value of 99.5%. The Yolov7 model with the highest AP value for all classes is the Yolov7-e6e model with 93.1%.

The PR curves of the test results of the Yolov8 models are shown in Figure 9. When Figure 9 is examined, the model in which the area under the curve is larger than the others is the Yolov8-n model. Therefore, we conclude that the Yolov8-n model performs better than other models.

When the performances of the models are evaluated separately for each class, the highest AP value of the Platelets class is the Yolov8-l model with 93%. This model is followed by Yolov8-m and Yolov8-n models with 92.9% and 91.6% values, respectively. The highest AP value of the RBC class is the Yolov8-x model with 88.1%. The Yolov8-x model is followed by the Yolov8-n and Yolov8-s models, respectively, in terms of RBC class with an AP value of 87.7% and 86.5%. In the WBC class, all Yolov8 models showed the same performance with an AP value of 99.5%. The Yolov8 model with the highest AP value for all classes is the Yolov8-n model with 93%. With AP values of 92.6% and 92.4%, respectively, Yolov8-l and Yolov8-m models follow the Yolov8-n model in terms of performance. With AP values of 92.8% and 92.3%, Yolov7-x and Yolov7-w6 models, respectively, follow the Yolov7-e6e model in terms of performance.

Figure 8

Precision-Recall Curves of Yolov7 Models.



(a) Yolov7 (b) Yolov7-x

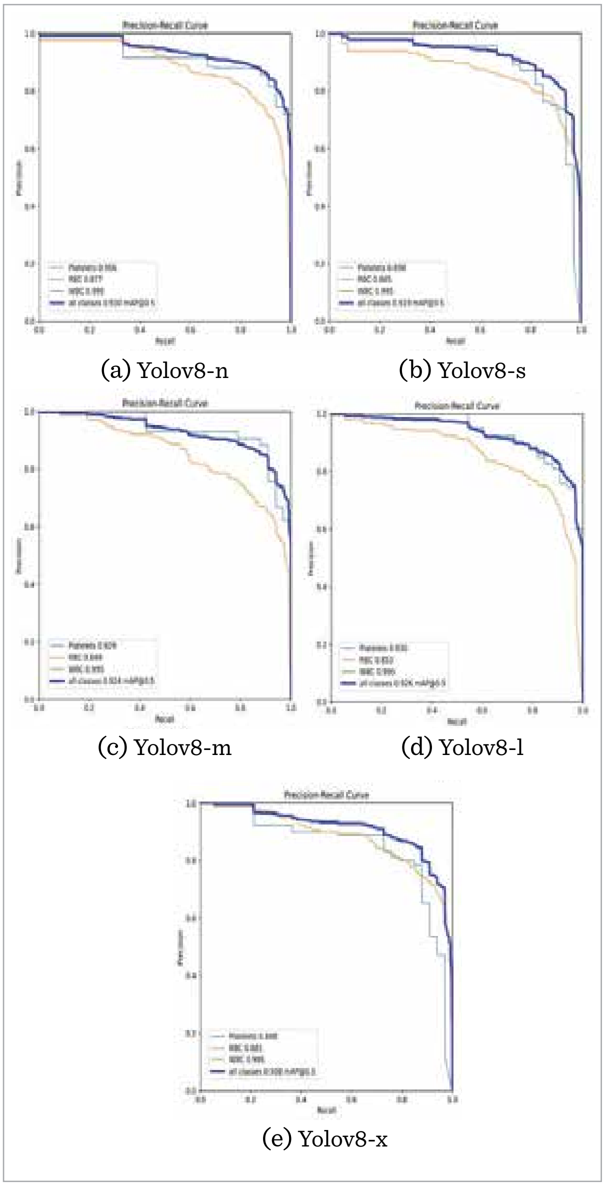
(c) Yolov7-w6 (d) Yolov7-e6

(e) Yolov7-d6 (f) Yolov7-e6e

The Yolov5-n model has achieved 100% accuracy in detecting the WBC class. When the other classes were examined, it was 91% for the Platelets class and 93% for the RBC class.

Figure 9

Precision-Recall Curves of Yolov8 Models.



(a) Yolov8-n

(b) Yolov8-s

(c) Yolov8-m

(d) Yolov8-l

(e) Yolov8-x

The detection of WBC and Platelets classes of the Yolov5-s model was 100% successful. It has achieved 90% success in determining the RBC class. The Yolov5-m model achieved 88%, 96% and 100% accuracy in the detection of Platelets, RBC and WBC classes, respectively. The Yolov5-l model provided 91%, 80% and 100% accuracy in the detection of Platelets, RBC and WBC classes, respectively. The Yolov5-x model provided 85%, 77% and 100% accuracy in the detection of Platelets, RBC and WBC classes, respectively. Among the Yolov5 models, the Yolov5-s model gave the best performance in determining the classes correctly.

The Yolov7 model achieved 97% accuracy in detecting the WBC and Platelets classes. It provided 96% success for the RBC class. A 97% success rate was achieved in the detection of Yolov7-x model WBC and Platelets classes. 91% success was achieved in the determination of the RBC class. The Yolov7-w6 model achieved 100% success in detecting the WBC class, respectively. When the other classes are examined, it has achieved 97% success in the Platelets class and 95% in the RBC class. The Yolov7-e6 model provided 97% success in the detection of Platelets and WBC classes, and 91% in the detection of RBC class. The Yolov7-d6 model provided 100% accuracy in the detection of Platelets and WBC classes, and 96% in the detection of RBC class. The Yolov7-e6e model provided 97% success in the detection of Platelets and WBC classes, and 94% in the detection of RBC class. Among the Yolov7 models, the Yolov7-d6 model gave the best performance in determining the classes correctly.

The Yolov8-n model has achieved 100% accuracy in detecting the WBC class. It provided 88% and 94% success for RBC and Platelets classes, respectively. A 100% success rate was achieved in detecting the WBC class of the Yolov8-s model. In the detection of RBC and Platelets classes, 92% and 91% success were achieved, respectively. The Yolov8-m model achieved 100% accuracy in detecting the WBC class. When the other classes are examined, it has achieved a success rate of 94% in the Platelets class and 91% in the RBC class. The Yolov8-l model provided 85%, 80% and 100% accuracy in the detection of Platelets, RBC and WBC classes, respectively. The Yolov8-x model provided 91%, 93% and 100% accuracy in the detection of Platelets, RBC and WBC classes, respectively. Among the Yolov8 models, the Yolov8-m model gave the best performance in determining the classes correctly.

Training accuracy alone is insufficient to evaluate the performance of models. In order for the performance evaluation to be carried out correctly, the results of the train should be examined together with the test results. The reason for this is that the models perform object detection on the data they do not see after they are trained with the appropriate dataset for the desired job-specific object detection. This shows that the results obtained in the test process are more important than the results obtained in the train process. When the testing accuracy (AP50) values are examined, the model with the highest test value with 94.5% is the Yolov5-s model. Following this model are Yolov5-m and Yolov5-l models with testing accuracy values of 94% and 93.3%, respectively. When the testing accuracy values of Detectron2 models are examined, the best performance with a value of 90.3% belongs to the Faster R-CNN R_101_FPN_3x model. Detectron2 models following this model are Faster R-CNN X_101_32x8d_FPN_3x and Faster R-CNN R_50_C4_1x with 88.9% and 88.6% values, respectively. However, Yolo models seem to be more successful than Detectron2 models. Although the training results among Yolo models were higher in Yolov7 models, these values decreased after testing. In Yolov5 models, on the other hand, the difference between the results obtained after the train process and the results obtained after the test process is less. Apart from the training accuracy and testing accuracy results, the speed of the model and the weight file size it creates are also important in terms of both the performance of the model and its use in mobile systems. When the weight size values are examined, the lowest file size is the Yolov5-n model with 3.72 MB. In terms of weight size, Yolov5-n is modeled as Yolov8-n and Yolov5-s with file sizes of 5.94 MB and 13.7 MB, respectively. When the training time (h) values are examined, the lowest training time is 0.068 and the Faster R-CNN R_50_C4_1x model. This model is followed by Yolov5-n and Faster R-CNN R_50_FPN_1x models with training time values of 0.083 and 0.086, respectively. The Yolov5-s model follows the Yolov5-n model with 0.096 hours of training time. When these 4 fastest running models were compared in terms of weight size, it was seen that the model with the lowest weight size value was the Yolov5-n model with 3.72 Mb. This model is followed by the Yolov5-s model with 13.7 Mb. When looking at Detectron2 models, the Faster R-CNN R_50_C4_1x model has a weight size of 126 Mb, and the Faster R-CNN R_50_FPN_1x model has a weight size of 158 Mb. Of these 4 models with similar training times, the weight sizes of the Detectron2 models are approximately 10 times the weight sizes of the Yolo models. At the same time, training and testing accuracy values are lower than Yolo models. For this reason and as a result of the examinations on the graphs given above, Yolo models are more suitable for the detection of blood cells.

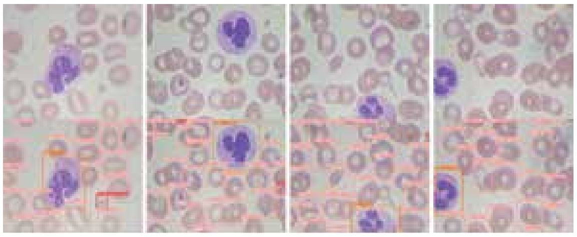
In general, according to the result values obtained for object detection, which is the main task of the models, the Yolov5-s model is more successful than other models with a value of 94.5%. At the same time, being the third model as weight size and the fourth model as training time among the models for detecting blood cells mobile by integrating with the mobile system makes the Yolov5-s model superior to other models.

4.1. Visualized Results

In this study, which was conducted in the field of medical imaging on blood cell detection and counting, the most suitable dataset was determined, the hyperparameter values of the models were determined and fine-tuned, and finally, training and testing operations were carried out in the Python environment. After all the procedures, blood cells were detected and counted with the Yolov5-s model, which has the best performance, to see the usability of the study in clinical studies and laboratory analysis processes. The original version of the 4 images used in the detection and counting process (Figure 10(a)-(d)) is given on the top line and the detected version is on the bottom line. Figure 10 was examined, it was seen that the blood cell detection process was carried out successfully.

Figure 10

Detection and Counting Process with Yolov5-s.



5. Discussion

Studies with current algorithms for the detection of blood cells are limited in the literature. Xu et al. [32] detected WBC, RBC and Platelets using BCCD dataset and Deformable DETR, Yolov3, YoloF,

YoloF+, TE-YoloF-B0, TE-YoloF-B1, TE-YoloF-B2, TE-YoloF-B3 models. Giving the comparative results of the results of the models, they suggested the TE-YoloF-B3 model, which has the highest AP50 value of 90.6%. EfficientNet was used in the backbone structure of the model used and Mish was used as the activation function to increase the precision value. At the same time, the original backbone of Yolov5 was used as the backbone structure, and SiLu was used in Yolo models and ReLu was used in Detectron2 models as the activation function. A better performance was achieved with Yolov5-s, with the train mAP value being 94.5% and the test mAP value being 94.5%. In addition, the test results of all Yolo models, except for the Yolov7-d6 model, whose test mAP value was 90.4%, showed better performance than the study by Xu et al. [32]. Among the train mAP values, all models except Yolov7-x and Yolov7-d6 models are better than the results obtained by Xu et al. [32]. In contrast to this study, higher accuracy was achieved with lower epoch numbers and optimized parameters with the developed models. At the same time, significant improvements were achieved in terms of training time and resource usage. Pfeil et al. [20] carried out the detection of blood cells with deep learning models. Researchers have created a dataset for the detection of three different classes in blood cells, namely RBC, WBC and Platelets. D2Det, MS R-CNN, Yoloact and Mask R-CNN models were run with the created dataset. According to the results obtained by running the models, the most successful model is the MS R-CNN model with an AP value of 57%. For the detection of blood cells, the researchers proposed the MS R-CNN model. In the test results of our study, none of the Yolo models fell below 90% in mAP value. As a result of Train, the lowest mAP value belongs to the Yolov7-d6 model with 88.3%. When Detectron2 models are examined, the lowest train AP value is 87.5% and the lowest test AP value is 83.6%. Detectron2 models performed worse than Yolo models. However, according to the study by Pfeil et al. [20], all models showed better performance. Mercaldo et al. [19] conducted a study using the BCCD dataset to detect blood cells. The researchers ran the dataset in the Yolov5s model and shared the results. They stated the success rate of the model as 91.8% in the detection of blood cells with the BCCD dataset and suggested the use of the Yolov5s model in the detection of blood cells. Before running the Yolov5-s model, the aothors set the size of the dataset images to 416x416. Batch size value is set to 16 and epoch number is set to 250. More epochs increase the time required to calculate each iteration, which can significantly extend the total training time. Therefore, using a high number of epochs in the study may cause a significant problem. After these settings were made, the train time of Yolov5-s lasted 33 minutes and 7 seconds. In our study, the image data in the dataset was set to 640x640 before being given as input to the network. Batch size value is set to 16 and epoch number is set to 50. When the Yolov5-s algorithm was run with these adjustments, the train time lasted only 5 minutes and 45 seconds. With the adjustments made, a 17.4% faster model and a more accurate model with a 2.7% higher mAP value were obtained than the study by Mercaldo et al. [19]. Alam and Islam [1] performed the detection of blood cells with the BCCD dataset. Yolo-tiny, VGG-16, ResNet50, InceptionV3, MobileNet models were used for automatic detection of blood cells. As a result of the study, the success rates of the models are: Yolo-tiny 62.3%, VGG-16 71.3%, ResNet50 74.3%, InceptionV3 68.2%, MobileNet 52%. On the other hand, in our study, as a result of the train and test processes, none of the mAP values of the Detectron2 and Yolo models are below 83%. In our study, not only accuracy (mAP) but also training time, data processing settings and model weight dimensions were taken into account in performance evaluations. Xia et al. [31] also worked with the same dataset, BCCD. They used Yolov3-tiny, Yoloc3-SPP and Yolov3 models in their studies. In line with their research and results, they suggested that the Yolov3-SPP model, which achieved an 88.6% success rate, should be used in the detection of blood cells because it outperformed other models. In their study, Xia et al. [31] set the epoch number to 100 and the train duration lasted 10 minutes. In our study, Yolov5-s, which was run over 50 epochs, is almost 2 times faster and is a more accurate model with a 5.9% higher mAP value. In our study, not only accuracy (mAP) but also training time, data processing settings and model weight dimensions were taken into account in performance evaluations. Kang et al [10], who conducted a study on blood cell detection and counting, evaluated models with three different datasets. BCCD dataset containing 364 images is divided into two: 327 for train and 37 for testing. CBC dataset containing 360 images is divided into two: 300 for train and 60 for testing. BCCD dataset containing 364 images is divided into three: 255 for training, 73 for validation and 36 for testing. Yolov5-x, Yolov7 and an improved model, CST-Yolov7, were used to perform the detection process. In the models, the learning rate value is determined as 0.001, the batch size value is 20 and the number of epochs is 150. After training the models with the BCCD dataset, mAP values of 92.3% for Yolov5-x, 89.6% for Yolov7 and 92.7% for CST-Yolov7 were obtained. In our study, the 364 image data in the BCDD dataset was divided into three: 292 train, 36 validation and 36 test. The epoch number is determined as 50 and the batch size value is 16. This means a faster running model and less processing power. In the Yolov5-s model, the learning rate value is determined as 0.01. With these adjustments, 94.5% mAP value was obtained in the train and test processes of the Yolov5-s model. This showed that a 1.8% higher mAP value was obtained than this study. Shakarami et al. [24] normal convolution and dilated convolution; Swish and LeakyReLU activation function; Loss functions IoU and DIoU and all combinations of these three parts were run by creating the Yolov3 model. Yolov3 models consisting of the specified combinations of the BCCD dataset were run in 38 epochs. The learning rate value was determined as 0.0001. Yolov3 combination with Dilated Convolution, Swish activation function, DIoU loss function reached 89.86% mAP value. In our study, the learning rate value of the Yolov5-s model was determined as 0.01 and the loss function GIoU was used. The activation function was set to SiLu and the models were run for 50 epochs. By making these adjustments, a mAP value of 94.5% was obtained, that is, a result that was 4.64% higher than the mAP value of the study.

It has been observed that a sufficiently comprehensive comparison is not made in deep learning studies for the detection of blood cells. In order to conduct a comprehensive comparative study in line with our observations, blood cells were detected and counted by running 21 different models with the image data in the BCCD dataset. The fact that 21 different models include models from Yolo and Detectron2 made the scope of comparison even wider. In addition, it was determined that Yolov8, the latest version of Yolo, was not used in the blood cell detection process in the literature, and in this study, five different models of Yolov8 were run with the BCCD dataset and compared with other models. Studies on automatic detection of blood cells are limited in the literature. Existing studies mostly have lower accuracy rates or longer processing times. For example, the accuracy rates of the models used in studies such as Pfeil et al. [20] (57% AP value) and Xu et al. [32] (90.6% AP value) are lower than the performance of more modern and optimized models. In many studies in the literature, training time and computational cost are high. For example, in Mercaldo et al.'s study [19], the YOLOv5-s model took 33 minutes, while the accuracy rate was only 91.8%. In this study, the model achieved higher performance and accuracy in 5 minutes 45 seconds. A comprehensive comparison between different AI models (such as Detectron2 and YOLO) is rare. This makes it difficult to determine which model is more suitable for a given dataset and purpose. In this study, one of the highest accuracy rates in the literature was achieved using the YOLOv5-s model with a test mAP value of 94.5%. The model's success in detecting small blood cells such as platelets is higher than other studies in the literature. The hyperparameter values used in the study were chosen appropriately so that the models, especially small models, can be used in the mobile system and give a high performance rate. As a result of what has been done, the 94.5% success rate of the Yolov5-s model is quite high compared to the studies in the literature. In addition, the weight size and training time values of the model are very small. This proves that the Yolov5-s model is very fast and is the most suitable model for mobile systems in blood cell detection and counting processes.

6. Conclusion

With the current study, blood cells were detected with deep learning technology and a suitable environment was created for the analysis of blood cells. Today, there are many deep learning models available for object detection. In order to determine the most suitable model for the need, it is necessary to compare the models, especially the current models.

Studies are insufficient for comprehensive comparison. Since there is no comprehensive research on the detection of blood cells with deep learning, our study was the first in its field by presenting the results of 21 different models. It is also seen that the success rate is higher than other studies.

As a result of research, it will provide the highest efficiency and benefit for the analysis of blood cells; A high-sensitivity blood cell detection model based on the Yolov5-s object detection algorithm was proposed, aiming to avoid the need for expert personnel, high-cost laboratory environment and complex equipment.

We suggested the most appropriate hyperparameters values to increase the accuracy of our model and to make it integrate into the mobile system by keeping the resulting file size small. Our results show that this system can be integrated into a mobile system. As a result, we implemented the highly accurate, minimum size Yolo model for analysis of blood cells. The following points should be emphasized here: first of all, the need for and workload of expert personnel has been reduced by the use of deep learning models. With software and image technology, the need for a costly laboratory has been eliminated. Third and lastly, a model that can be integrated into a mobile device that is available to everyone, such as a mobile phone, has been provided, eliminating the need for complex hardware.

In this study, hyperparameter values that can be referenced for the detection of blood cells are given. Developers of Yolov8, the most up-to-date model of Yolo, have achieved a higher mAP value with the COCO dataset compared to other versions of Yolo. Here, it is seen that the Yolov5-s model is superior to the Yolov8 models in terms of mAP value.

The study can be used as a reference source providing information about the performance of the BCCD dataset in object detection models. With the given hyperparameter values, it has been observed that Yolov5-s, a lightweight model, has higher performance than other models. Under normal circumstances, the speed of large-scale models should be slow but the performance should be higher than other models. The higher performance of Yolov5-s compared to large-scale models makes it possible to integrate the lightweight model into a mobile system that can be used in clinical studies.

In future studies, the integration of deep learning models into the mobile system is to provide an easy-to-use, low-cost and lightweight system for pathologists, biologists and laboratory workers. For this, it is a remarkable and worth researching method to diagnose diseases from blood cells by making minor adjustments to the general software flow of deep learning models. In addition, it is thought that the image data of the BCCD dataset will be run with different deep learning models with a large number of image data obtained by using data augmentation methods. We aim to further increase the success rate by continuing optimization studies on the high-performing hybrid deep learning model.

Acknowledgements

The authors are grateful to Editors for their valuable comments and contributions to the manuscript. This research paper was derived as a part of Mübarek Mazhar Çakır's MSc thesis, conducted under the supervision of Gökalp Çınarer at the Department of Mechatronics Engineering in School of Graduate Studies, Yozgat Bozok University, Yozgat, Turkey.

Author Contributions

Gökalp Çinarer and Mübarek Mazhar Çakir designed the study; Gökalp Çinarer created artificial intelligence models and performed the coding; Mübarek Mazhar Çakir implemented the experiments; Gökalp Çinarer and Mübarek Mazhar Çakir wrote and revised the manuscript. All authors reviewed the manuscript.

Funding

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

Declarations

Conflict of interest. All authors declare no conflict of interest related to this article.

Data availability. The dataset used in the study is free to the public. Therefore, no ethical approval is required.

References

1. Alam, M. M., Islam, M. T. Machine Learning Approach of Automatic Identification and Counting of Blood Cells. Healthcare Technology Letters, 2019, 6(4), 103-108. https://doi.org/10.1049/htl.2018.5098
2. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv Pre-print arXiv:2004.10934, 2020.
3. Chaddad, A., Peng, J., Xu, J., Bouridane A. Survey of Explainable AI Techniques in Healthcare. Sensors, 2023, 23(2), 634. https://doi.org/10.3390/s23020634
4. Hu, B., Liu, Y., Chu, P., Tong, M., Kong, Q. Small Object Detection via Pixel Level Balancing With Applications to Blood Cell Detection. Frontiers in Physiology, 2022, 13, 911297. https://doi.org/10.3389/fphys.2022.911297
5. Jocher, G. YOLOv8. https://github.com/ultralytics/ultralytics.
6. Jiang, H., Learned-Miller, E. Face Detection with the Faster R-CNN. International Conference on Automatic Face and Gesture Recognition, 2017, 650-657. https://doi.org/10.1109/FG.2017.82
7. Jin, F., Chag, X., Wang, X., Xiong, H., Wang, L., Zhang, B., Wang, P., Zhao, L. Relationship Between Red Blood Cell-Related Indices and Coronary Artery Calcification. Postgraduate Medical Journal, 2023, 99(1167), 4-10. https://doi.org/10.1093/postmj/qgac003
8. Jocher, G., Changyu, L., Hogan, A., Yu, L., Rai, P., Sullivan, T. ultralytics/yolov5: Initial Release. https://zenodo.org/record/ 3908560.
9. Jogin, M., Mohana, Madhulika, M. S., Divya, G. D., Meghana, R. K., Apoorva, S. Feature Extraction Using Convolution Neural Networks (CNN) and Deep Learning. IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, 2018, 2319-2323. https://doi.org/10.1109/RTEICT42901.2018.9012507
10. Kang, M., Ting, C.-M., Ting, F. F., Phan, R. CST-YOLO: A Novel Method for Blood Cell Detection Based on Improved YOLOv7 and CNN-Swin Transformer. arXiv Pre-print arXiv:2306.14590v1, 2023 https://doi.org/10.1109/ICIP51287.2024.10647618
11. Kutlu, H., Avci, E., Özyurt, F. White Blood Cells Detection and Classification Based on Regional Convolutional Neural Networks. Medical Hypotheses, 2020, 135, 109472. https://doi.org/10.1016/j.mehy.2019.109472
12. Leng, B., Wang, C., Leng, M., Ge, M., Dong, W. Deep Learning Detection Network for Peripheral Blood Leukocytes Based on Improved Detection Transformer. Biomedical Signal Processing and Control, 2023, 82, 104518. https://doi.org/10.1016/j.bspc.2022.104518
13. Lengvenyte, A., Strumila, R., Belzeaux, R., Aouizerate, B., Dubertret, C., Haffen, E., Llorca, P. M., Roux, P., Polosan, M., Schwan, R., Walter, M., D'Amoto, T., Kanuel, D., Leboyer, M., Bellivier, F., Etain, B., Navickas, A., Olie, E., Courtet, P. Associations of White Blood Cell and Platelet Counts with Specific Depressive Symptom Dimensions in Patients with Bipolar Disorder: Analysis of Data From the FACE-BD Cohort. Brain, Behavior, and Immunity, 2023, 108, 176-187. https://doi.org/10.1016/j.bbi.2022.12.002
14. Li, P., Zheng, J., Li, P., Long, H., Li, M., Gao, L. Tomato Maturity Detection and Counting Model Based on MHSA-YOLOv8. Sensors, 2023, 23(15), 6701. https://doi.org/10.3390/s23156701
15. Li, M., Zhang, Z., Lei, L., Wang, X., Guo, X. Agricultural Greenhouses Detection in High-Resolution Satellite Images Based on Convolutional Neural Networks: Comparison of Faster R-CNN, YOLO v3 and SSD. Sensors, 2020, 20(17), 4938. https://doi.org/10.3390/s20174938
16. Liu, R., Ren, C., Fu, M., Chu, Z., Guo, J. Platelet Detection Based on Improved YOLO_v3. Cyborg and Bionic Systems, 2022, 2022. https://doi.org/10.34133/2022/9780569
17. Lorente, Ò., Riera, I., Rana, A. Scene Understanding for Autonomous Driving. arXiv Pre-print arXiv:2105.04905, 2021.
18. Maity, M., Banerjee, S., Chaudhuri, S. S. Faster R-CNN and YOLO Based Vehicle Detection: A Survey. International Conference on Computing Methodologies and Communication, 2021, 1442-1447. https://doi.org/10.1109/ICCMC51019.2021.9418274
19. Mercaldo, F., Martinelli, F., Santone, A., Cesarelli, M. Blood Cells Counting and Localisation Through Deep Learning Object Detection. IEEE International Conference on Big Data, 2023, 4400-4409. https://doi.org/10.1109/BigData55660.2022.10020952
20. Pfeil, J., Nechyporenko, A., Frohme, M., Hufert, F. T., Schulze, K. Examination of Blood Samples Using Deep Learning and Mobile Microscopy. BMC Bioinformatics, 2022, 23(1), 1-14. https://doi.org/10.1186/s12859-022-04602-4
21. Rahaman, M. A., Ali, M. M., Ahmed, K., Bui, F. M., Mahmud, S. M. H. Performance Analysis Between YOLOv5s and YOLOv5m Model to Detect and Count Blood Cells: Deep Learning Approach. ACM International Conference Proceeding Series, 2022, 316-322. https://doi.org/10.1145/3542954.3543000

22. Redmon, J., Farhadi, A. YOLO9000: Better, Faster, Stronger. arXiv Pre-print arXiv:1612.08242, 2016. https://doi.org/10.1109/CVPR.2017.690

23. Redmon, J., Farhadi, A. YOLOv3: An Incremental Improvement. arXiv Pre-print arXiv:1804.02767, 2018.

24. Shakarami, A., Menhaj, M. B., Mahdavi-Hormat, A., Tarrah, H. A Fast and Yet Efficient YOLOv3 for Blood Cell Detection. Biomedical Signal Processing and Control, 2021, 66, 102495. https://doi.org/10.1016/j.bspc.2021.102495

25. Shenggan. BCCD (Blood Cell Count and Detection) Dataset. https://github.com/Shenggan/BCCD_Dataset.

26. Singh, R., Shetty, S., Patil, G., Bide, P. J. Helmet Detection Using Detectron2 and EfficientDet. International Conference on Computing Communication and Networking Technology, 2021. https://doi.org/10.1109/ICCCNT51525.2021.9579953

27. Stancel, M., Hulic, M. An Introduction to Image Classification and Object Detection using YOLO Detector. ICTERI PhD Symposium, 2019.

28. Terven, J. R., Cordova-Esparza, D. M. A Comprehensive Review of YOLO: From YOLOv1 and Beyond. arXiv Pre-print arXiv:2304.00501v5, 2023. https://doi.org/10.3390/make5040083

29. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y. M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-time Object Detectors. arXiv Pre-print arXiv:2207.02696, 2022. https://doi.org/10.1109/CVPR52729.2023.00721

30. Wen, H., Huang, C., Guo, S. The Application of Convolutional Neural Networks (CNNs) to Recognize Defects in 3D-Printed Parts. Materials, 2021, 14(10), 2575. https://doi.org/10.3390/ma14102575

31. Xia, T., Fu, Y. Q., Jin, N., Chazot, P., Angelov, P., Jiang, R. AI-enabled Microscopic Blood Analysis for Microfluidic COVID-19 Hematology. International Conference on Computational Intelligence and Applications, 2020, 98-102. https://doi.org/10.1109/ICCIA49625.2020.00026

32. Xu, F., Li, X., Yang, H., Wang, Y., Xiang, W. TE-YOLOF: Tiny and Efficient YOLOF for Blood Cell Detection. Biomedical Signal Processing and Control, 2022, 73. https://doi.org/10.1016/j.bspc.2021.103416

33. Yang, F., Zhang, X., Liu, B. Video Object Tracking Based on YOLOv7 and DeepSORT. arXiv Pre-print arXiv:2207.12202, 2022.

34. Yildirim, M. S., Dandil, E. Automated Multiple Sclerosis Lesion Segmentation on MR Images via Mask R-CNN. International Symposium on Multidisciplinary Studies and Innovative Technologies, 2021, 570-577. https://doi.org/10.1109/ISMSIT52890.2021.9604593

35. You, S., Ou, Z., Zhang, W., Zheng, D., Zhong, C., Dong, X., Qiu, C., Lu, T., Cao, Y., Liu, C. F. Combined Utility of White Blood Cell Count and Blood Glucose for Predicting in-hospital Outcomes in Acute Ischemic Stroke. Journal of Neuroinflammation, 2019, 16(1), 1-9. https://doi.org/10.1186/s12974-019-1422-7

36. Zhang, W. J., Yang, G., Lin, Y., Ji, C., Gupta, M. M.. On Definition of Deep Learning. Proceedings of the World Automation Congress, 2018, 232-236. https://doi.org/10.23919/WAC.2018.8430387

37. Zhang, Y., Guo, Z., Wu, J., Tian, Y., Tang, H., Guo, X. Real-Time Vehicle Detection Based on Improved YOLO v5. Sustainability, 2022, 14(19), 12274. https://doi.org/10.3390/su141912274

38. Zhao, Z. Q., Zheng, P., Xu, S. T., Wu, X. Object Detection with Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(11), 3212-3232. https://doi.org/10.1109/TNNLS.2018.2876865

39. Zhou, F., Zhao, H., Nie, Z. Safety Helmet Detection Based on YOLOv5. IEEE International Conference on Power Electronics, Computer Applications, 2021, 6-11. https://doi.org/10.1109/ICPECA51329.2021.9362711