Automated Blood Cell Count using YOLOv7

Md. Abu Hena Shadid
Department of Computer Science and Technology
Islamic University of Technology (IUT), OIC
Gazipur, Bangladesh
abuhenashadid@iut-dhaka.edu

Wasswa Lutufi Sebbanja
Department of Computer Science and Technology
Islamic University of Technology (IUT), OIC
Gazipur, Bangladesh
wasswalutufi@iut-dhaka.edu

Mahajabin Tabassum Soikey
Department of Computer Science and Technology
Islamic University of Technology (IUT), OIC
Gazipur, Bangladesh
mahajabin@iut-dhaka.edu

***Abstract*—In clinical medical diagnosis, automatic detection and classification of blood cells has a great demand. Traditional methods like hematology analyzer or manual count are laborious, takes a lot of time and depends on the accuracy of analysts' knowledge and expertise. An improved version of YOLOv7 is provided in this paper for detecting and classifying blood cells automatically. First, from among the large number of detection classes, in this version they are reduced down to three blood cells classes in the modified YOLOv7 model. In addition, some unnecessary layers have been removed from the model to develop a robust detection system capable of identifying various blood cells with high accuracy. To ensure accessibility and ease of use, we designed a user-friendly website interface that allows users to upload a single blood slide image and receive the results in under ten seconds. The detection results are generated in a comprehensive PDF report, facilitating immediate and effective analysis by healthcare professionals. This automation not only streamlines the diagnostic process but also significantly reduces the time and effort required, ultimately contributing to improved patient care and medical outcomes. The model working on the dataset, for 20 epochs, provides the output that detects three different types of blood cells, with an accuracy of 78%. Moreover, the fast detection and high accuracy have opened the way for computer-aid diagnostic systems in the future. The GitHub repository of this project can be found here.**

***Index Terms*—Blood cell detection, YOLOv7, Blood Profile Automation, Clinical Diagnostics, Hematology Analyzers, Convolutional Neural Networks.**

I. Introduction

In medical term, Complete Blood Count or CBC is a set of medical diagnostic test that gives different information of cells in the blood. There are two components in human blood - Plasma and Cellular Components. The cellular components are divided into three different categories - Erythrocytes or red blood cells (RBCs), Leukocytes or white blood cells (WBCs) and Thrombocytes or platelets. The reb blood cells deliver oxygen to the cells and take CO_2 back from the tissues through blood flow. For the immune system, WBC plays a very important role. They protect human body from different diseases and infections. The platelets have coagulation technique that helps in blood clotting and it then heals the wounds of the body. CBC provides a complete report of the total counts and types of blood components like RBCs, WBCs, Platelets and Hemoglobin. The CBC might be an indication of illness, as abnormal change in count is often an expression of a type of illness of the human body. [1]. That's why most of the times doctors check a person's CBC result and accordingly take decisions.

Blood Cell detection, count and classification technology could help the doctors to diagnose diseases, including anemia, leukemia, dengue, malaria, infections etc. [2]. For example, a low count in Red Blood Cell indicates anemia [3]. A way too low count of Platelets indicates Thrombocytonia, which is an indication of acute leukemia and aplastic anemia. In case of infections and inflamation, a high count of WBCs can be noticed. Monitoring Blood Cell Count is necessary for the patients going through chemotherapy or radiation therapy treatment as these types of therapy results in the decrease of blood cell production in the bone marrow [4].

Counting blood cells was extensively utilized in clinical blood tests, involving classification and detection procedures. Traditionally, BCs were detected using methods like the hemocytometer, hematology analyzer, and manual counting [5]. While automated CBCs are now common using laboratory equipment, manual counting remains crucial for validating validating any discrepancies in the results. Nevertheless, manual counting is labor-intensive, demands a high level of skill and experience from clinical laboratory analysts, and can be imprecise and tedious ([7]). Hence, there is a growing need for an automated, efficient system. Blood smear image analysis for CBC plays a vital role in diagnosing different diseases, health issues and improving human health. With progress in deep learning techniques, there is increasing reliance on accurate and robust object detection in computer vision. Researchers are actively exploring DL methods to enhance automatic detection based on blood smear images ([7]).

Our project utilizes the YOLOv7 model for blood cell detection. YOLO, which stands for "You Only Look Once," is a state-of-the-art object detection algorithm known for its speed and accuracy. YOLOv7 builds upon previous versions by incorporating several enhancements to improve performance. The architecture of YOLOv7 includes:

1) **Extended Efficient Layer Aggregation Networks (E-ELAN)**: This design strategy focuses on controlling the longest shortest gradient path, enabling a deeper network to learn and converge more effectively. E-ELAN uses expand, shuffle, and merge cardinality to enhance the learning ability of the network without disrupting the gradient path.
2) **Model Scaling**: YOLOv7 employs a compound scaling method for concatenation-based models. This method modifies both the depth of computational blocks and the width of transition layers at the same time, preserving the model's optimal structure and ensuring efficient use of hardware.

These architectural innovations allow YOLOv7 to achieve faster and more accurate inferences, making it an ideal choice for real-time applications like blood cell detection.

Numerous studies have explored the application of deep learning techniques for blood cell detection and classification. For instance, researchers have developed models using convolutional neural networks (CNNs) to classify RBCs, WBCs, and platelets from blood smear images. These models have demonstrated high accuracy and robustness, significantly outperforming traditional methods.

In one study ([6]), a deep learning model was used to classify different types of WBCs, achieving a high degree of accuracy and reducing the need for manual intervention. Another study employed a CNN-based approach to detect malaria-infected RBCs, providing a reliable and efficient alternative to manual examination. These works highlight the potential of deep learning in automating and improving blood cell analysis.

However, many of these models are either too complex for real-time applications or lack the necessary accuracy for clinical use. Our project addresses these gaps by utilizing the YOLOv7 model, which combines high speed and accuracy, making it suitable for real-time blood cell detection. By developing a user-friendly web interface, we aim to provide an accessible and efficient tool for healthcare professionals to perform blood profiling and enhance patient care.

By leveraging the advanced capabilities of the YOLOv7 model, our project aims to automate the detection and classification of blood cells, providing a faster, more accurate, and cost-effective solution for clinical diagnostics. This innovation holds the potential to significantly improve healthcare outcomes, particularly in resource-limited settings.

The report is organized as follows. Section II introduces the methodology and detailed steps. Section III holds the result analysis part, describing experimental setup and model evaluation. Section IV depicts the prototype description. Section V presents the conclusions and provides future research directions. Section V points to all the references.

II. Methodology

In this section, we provide a comprehensive overview of the steps and processes followed in our project, focusing on datasets, data preprocessing, model architecture, and training procedures. This detailed methodology ensures that others can replicate our work.

A. YOLOv7 detailed overview

YOLOv7 (You Only Look Once version 7) is a state-of-the-art object detection model known for its remarkable speed and accuracy. It sets a new benchmark in real-time object detection, outperforming both transformer-based and convolution-based object detectors. YOLOv7 achieves the highest average precision of 56.8% and can efficiently process video inputs ranging from 5 fps to 160 fps. Compared to YOLOv4, YOLOv7 has 75% fewer parameters and 36% less computational time, while delivering 1.5 times higher average precision. The model's efficiency in prediction is achieved through several key innovations:

1) **Bag of Freebies:** This approach enhances model accuracy without increasing training costs. YOLOv7 incorporates batch normalization, implicit knowledge integration, and EMA (Exponential Moving Average) models to improve performance.
2) **Model Scaling:** YOLOv7 uses a compound scaling method for concatenation-based models, adjusting the depth of computational blocks and the width of transition layers simultaneously. This ensures efficient hardware utilization and maintains the optimal structure of the model.
3) **Intersection over Union (IoU):** This metric measures the overlap between predicted and ground truth bounding boxes, ensuring precise localization of objects.
4) **Extended Efficient Layer Aggregation Networks (E-ELAN):** E-ELAN controls the longest shortest gradient path, enabling a deeper network to learn and converge more effectively. It uses expand, shuffle, and merge cardinality to enhance learning ability without disrupting the gradient path.

YOLOv7's architecture consists of three main components: the input section, backbone, neck, and head. Each component plays a crucial role in the model's performance.

B. Dataset

The dataset used for this project is publicly available and can be accessed here.

Collaborators: The dataset was prepared and provided by Balakrishna Kumar (Owner).

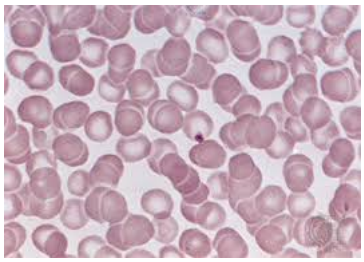

Fig. 1. Red Blood Cell (RBC)

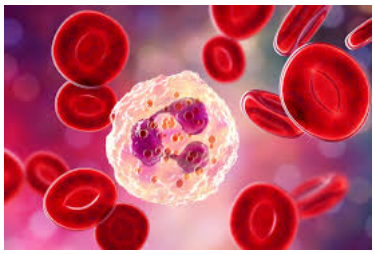

Fig. 2. White Blood Cell (WBC)

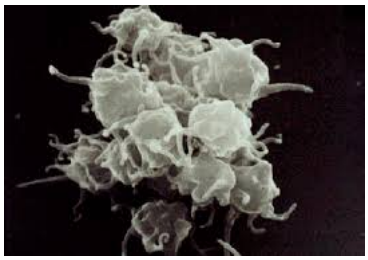

Fig. 3. Platelets

1) Description of the Dataset: The dataset contains blood smear images, annotated with bounding boxes and class labels for different types of blood cells: Platelets (Figure 3), RBCs (Red Blood Cells) (Figure 1), and WBCs (White Blood Cells) (Figure 2). The dataset includes:

- **Training Images and Labels:**The images are accompanied by .txt files containing annotations for the bounding boxes and class labels.
- **YAML File:** The dataset's YAML file, required for training the YOLOv7 model, is also provided. This file specifies the paths to the training and validation datasets, the number of classes, and the class names.

2) The structure of the dataset:

- **training:** /kaggle/working/blood-cell-count-and-types-detection/images/train
- **validation:** /kaggle/working/blood-cell-count-and-types-detection/images/valid
- **nc:** 3
- **names:** ['Platelets', 'RBC', 'WBC']

3) Dataset Overview: To efficiently handle the dataset, custom data loaders were implemented. These loaders perform various tasks, including Data Augmentation, Normalization, Batching, etc. These custom data loaders ensure that the dataset is preprocessed effectively, allowing the model to learn robust features and perform accurate detections.

4) Detailed Model Architecture: Our blood profile machine learning project leverages the power of YOLOv7, a state-of-the-art object detection framework. YOLOv7 is known for its exceptional speed and accuracy, making it ideal for real-time applications. In our model, we have customized the architecture to suit our specific needs, utilizing CSPDarknet53 as the backbone, and integrating several unique components to enhance performance. This detailed description walks through the entire model architecture, from input to output, highlighting each stage and its specific function. Figure 4

1) **Backbone:** The backbone of our model is CSPDarknet53, a robust feature extractor designed to capture essential characteristics from the input images. CSPDarknet53 is composed primarily of convolutional layers, which progressively downsample the input image and extract hierarchical features (Figure 5). The key components of our backbone include:
   a) **Initial Convolution Layers:** The initial layers consist of basic convolution operations designed to capture low-level features such as edges and textures. These layers are:
      - **Conv layer [3, 32, 3, 1]:** A 3x3 convolution with 32 filters.
      - **Conv layer [32, 64, 3, 2]:** A 3x3 convolution with 64 filters and a stride of 2 for downsampling.
      - **Conv layer [64, 64, 3, 1]:** A 3x3 convolution with 64 filters.
   b) **Downsampling and Feature Aggregation:** As we move deeper into the network, the convolutions increase in filter size and are often followed by downsampling operations to reduce the spatial dimensions of the feature maps:
      - **Conv layer [64, 128, 3, 2]:** A 3x3 convolution with 128 filters, followed by another set of convolutions that refine these features.
   c) **Cross Stage Partial (CSP) Blocks:** These blocks are designed to enhance the learning capability of the network by partitioning the feature map of the base layer into two parts and then merging them through a cross-stage hierarchy. This setup helps in gradient flow and reduces computation cost.

2) **Neck - Feature Pyramid Network (FPN) and Path Aggregation Network (PAN):** The neck of the model plays a critical role in collecting and combining feature maps from different stages of the backbone to provide a rich multi-scale feature representation. Our neck incorporates a combination of Feature Pyramid Network (FPN) (Figure 6) and Path Aggregation Network (PAN) (Figure 7) for efficient feature fusion:
   a) **Feature Pyramid Network (FPN):** FPN enhances the semantic strength of the low-resolution layers by adding top-down pathways and lateral connections:
      - Conv layers followed by upsampling operations to refine the features from different scales.
   b) **Path Aggregation Network (PAN):** PAN improves information flow and strengthens the learned features by introducing bottom-up paths:

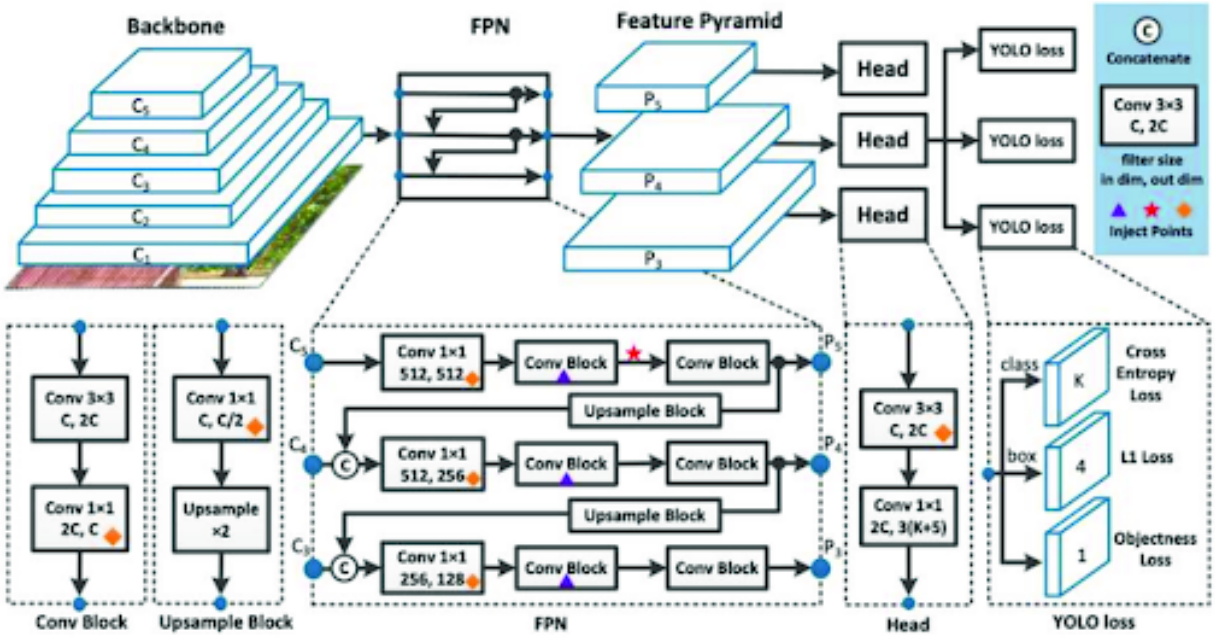

Fig. 4. YOLO Structure

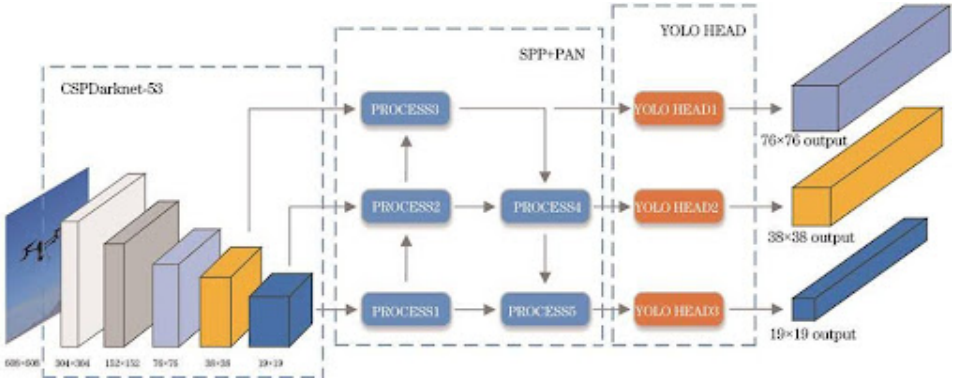

Fig. 5. CSPDarknet53 Structure

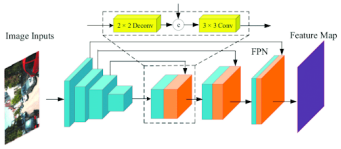

Fig. 6. FPN Structure

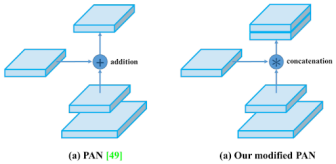

Fig. 7. PAN Structure

- Additional convolutional layers and concatenation operations to combine feature maps from various layers. Single RFB expanded. A number of these are used in the neck (Figure 8).

3) **Head:** The head of our model is where the actual detection happens. It consists of layers designed to predict bounding boxes, class probabilities, and objectness scores. It is made up an RFCN model. These predictions are made at multiple scales to accommodate objects of different sizes (Figure 9):

a) Convolutional Layers: These layers process the aggregated features from the neck and prepare them for the detection task:
   - A series of convolutions that further refine the features before final prediction.

b) **Prediction Layers:** The final layers of the head generate the output predictions. These layers include:
   - Bounding box predictions for localizing objects

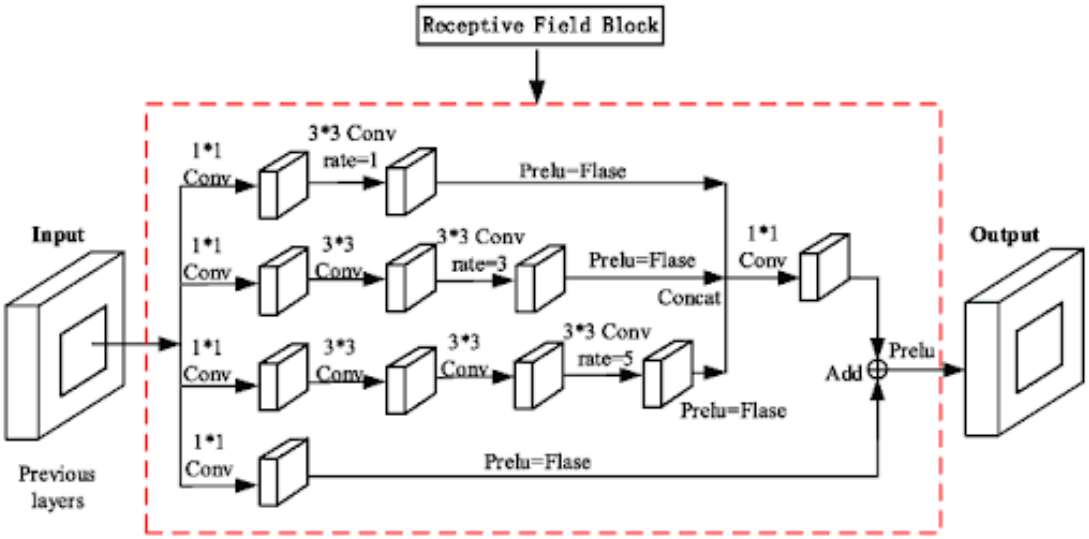

Fig. 8. Convolutional Layers

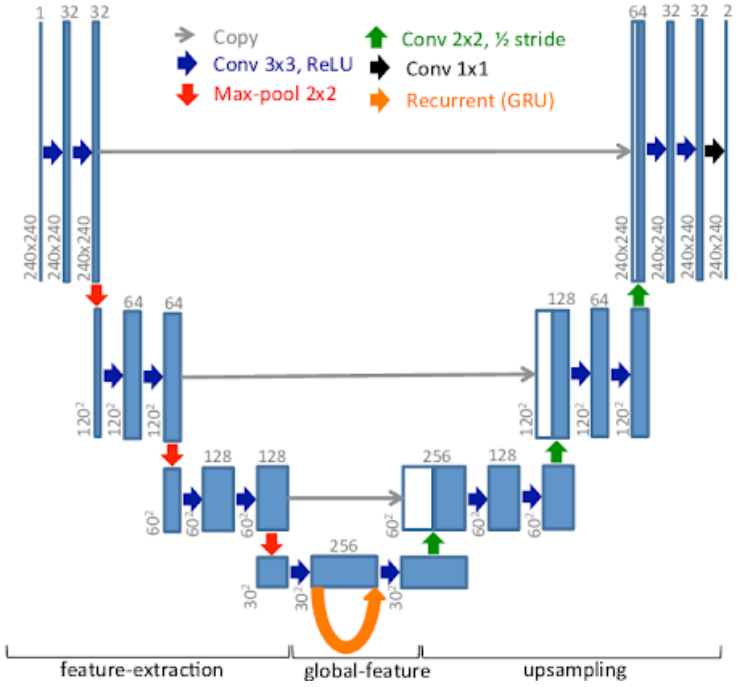

Fig. 9. YOLOv7 Head Structure

within the image.

- Class predictions to categorize the detected objects.
- Objectness score predictions to determine the presence of objects.

4) **Model Pipeline - From Input to Output:** The entire model pipeline can be summarized as follows (Figure 10):
   a) **Input:** The input is a two-dimensional array representing the image, typically with three color channels (RGB).
   b) **Backbone Processing:** The input image is processed through the CSPDarknet53 backbone, where it undergoes multiple convolutional operations and feature extraction stages. Each stage progressively captures more complex features, from edges and textures to more abstract patterns.
   c) **Neck Aggregation:** The feature maps from the backbone are fed into the neck, where FPN and PAN architectures work together to aggregate features from different scales, enhancing the model's ability to detect objects of varying sizes.
   d) **Detection Head:** The refined feature maps are passed to the head, where convolutional and prediction layers

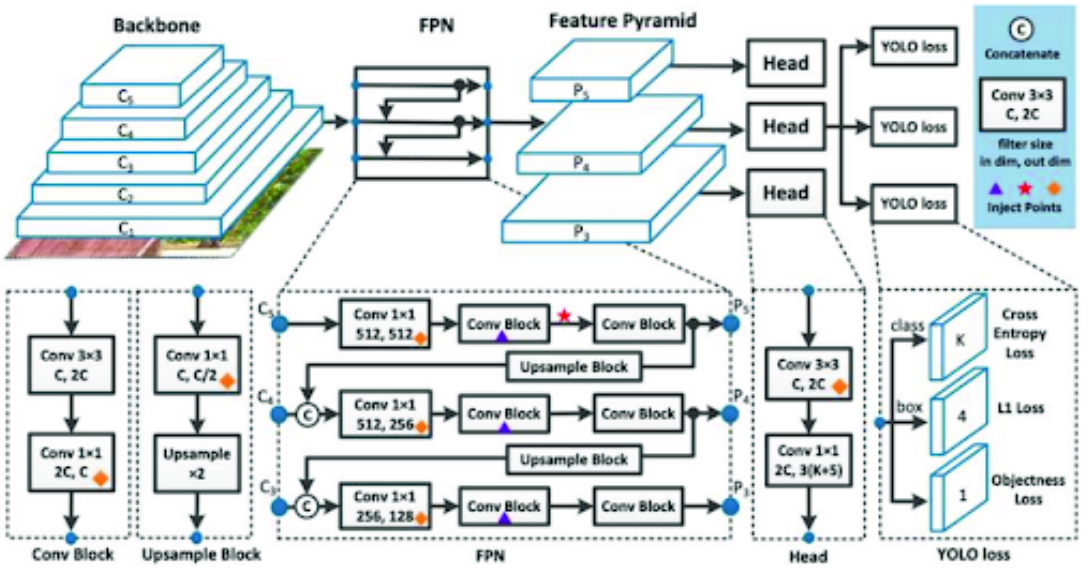

Fig. 10. YOLOv7 Full Model Structure

generate the final output. This includes bounding boxes, class probabilities, and objectness scores for each detected object.

By customizing YOLOv7 with CSPDarknet53 and integrating advanced feature aggregation techniques, our model achieves high accuracy and efficiency in detecting objects within blood profile images. This detailed architecture ensures that the model is well-suited for real-time applications, providing reliable and precise outputs.

III. Result Analysis

A. Experimental Outcomes

1) **Experimental Setup**
   - Initial Weights: Official YOLOv7 pre-trained weights
   - Batch Size: 4
   - Epochs: 20
   - Image Size: 640 x 640
   - Device: CUDA (NVIDIA GeForce MX250, 2047.875MB)
   - Torch Version: 2.3.1+cu118
   - Workers: 8
2) **Hyperparameters:**
   - Initial Learning Rate (lr0): 0.01
   - Scaled Weight Decay: 0.0005
3) **Data Dictionary:**
   - Train: ‘./yolov7/dataset/train‘
   - Validation: ‘./yolov7/dataset/test‘
   - Number of Classes (nc): 3
   - Class Names: [’Platelets’, ’RBC’, ’WBC’]
4) **Optimizer Groups:**
   - 95 .bias
   - 95 conv.weight
   - 98 other
5) **Model Summary:**
   - 415 layers
   - 37,207,344 parameters
   - 37,207,344 gradients
   - 105.1 GFLOPS
6) **Autoanchor Analysis:**
   - Anchors/Target: 5.91
   - Best Possible Recall (BPR): 0.9994

B. Evaluation Metrics

To evaluate the performance of our model, we utilized several metrics including precision, recall, F1-score, and confusion matrix analysis.

1) **Precision (P):** Measures the accuracy of the positive predictions. High precision indicates a low false positive rate. Our model has a fairly good precision recall (Figure 11).

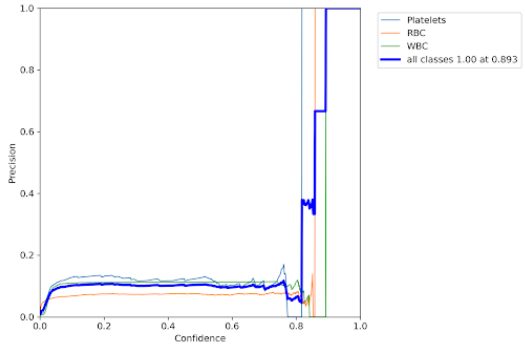

Fig. 11. Precision of the model

2) **Recall (R):** Measures the completeness of the positive predictions. High recall indicates a low false negative rate (Figure 12).

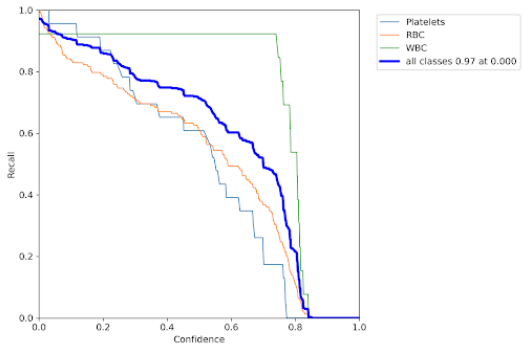

Fig. 12. Recall of the model

3) F1-Score: The harmonic mean of precision and recall, providing a single metric to evaluate the balance between precision and recall (Figure 13).

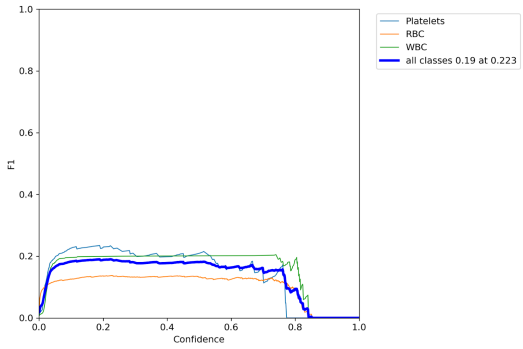

Fig. 13. F1 Score of the model

4) **Confusion matrix:** From the confusion matrix, we can identify the following areas of concern (Figure 14):

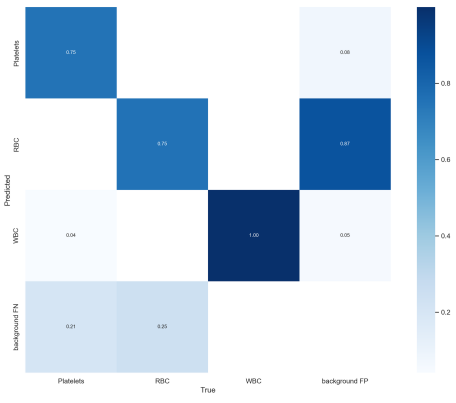

Fig. 14. Confusion Matrix

- False Positives: There is a noticeable rate of false positives in RBC (0.87), indicating the model sometimes misclassifies other objects as RBC.
- False Negatives: Platelets and RBCs have higher false negative rates (0.21 and 0.25, respectively), which means these classes are sometimes not detected when they should be.

IV. Prototype

To fulfill our mission for blood profile, we made a simple website where a user can submit a photo of a blood smear (preferably 640x640) and receive a pdf report of the blood profile results.

V. Conclusion

The blood profile machine learning project using YOLOv7 and CSPDarknet53 backbone has shown promising results. The model effectively detects and classifies Platelets, RBCs, and WBCs from blood samples. The confusion matrix indicates that the model achieves high accuracy, particularly for WBCs, with a true positive rate of 1.00. However, there are areas for improvement, particularly in reducing false positives and false negatives for Platelets and RBCs.

The limitations of our project include Class Imbalance, False Positives and Negatives, Limited Dataset and Computational Resources.

Our work can be further extended to Dataset Expansion, Addressing Class Imbalance, Hyperparameter Optimization, Advanced Architectures, Integration with Clinical Workflows and so on.

References

[1] R. Green and S. Wachsmann-Hogiu, “Development, history, and future of automated cell counters,” Clinics Lab. Med., vol. 35, no. 1, pp. 1–10, Mar. 2015, doi: 10.1016/j.cll.2014.11.003.

[2] D. Cruz, C. Jennifer, Valiente, L. C. Castor, C. M. T. Mendoza, B. A. Jay, L. S. C. Jane, and P. T. B. Brian, “Determination of blood components (WBCs, RBCs, and platelets) count in microscopic images using image processing and analysis,” in Proc. IEEE 9th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM), Dec. 2017, pp. 1–7, doi: 10.1109/HNICEM.2017.8269515.

[3] N. M. Deshpande, S. Gite, and R. Aluvalu, “A review of microscopic analysis of blood cells for disease detection with AI perspective,” PeerJ Comput. Sci., vol. 7, p. e460, Apr. 2021, doi: 10.7717/peerj-cs.460.

[4] Toyonobu Oshita, “Role of Bone Marrow in the Formation of Blood Cells and Its Functions”, vol. XI, J Osteopor Phys Act, Vol.11 Iss.2 No:1000334

[5] M. M. Alam and M. T. Islam, “Machine learning approach of automatic identification and counting of blood cells,” Healthcare Technol. Lett., vol. 6, no. 4, pp. 103–108, Aug. 2019, doi: 10.1049/htl.2018.5098.

[6] P. Tiwari, J. Qian, Q. Li, B. Wang, D. Gupta, A. Khanna, J. J. P. C. Rodrigues, and V. H. C. de Albuquerque, “Detection of subtype blood cells using deep learning,” Cogn. Syst. Res., vol. 52, pp. 1036–1044, Dec. 2018, doi: 10.1016/j.cogsys.2018.08.022.

[7] He Y. Automatic Blood Cell Detection Based on Advanced YOLOv5s Network. IEEE Access. 2024 Jan 30.

[8] Rodellar, J., Alférez, S., Acevedo, A., Molina, A. and Merino, A., 2018. Image processing and machine learning in the morphological analysis of blood cells. International journal of laboratory hematology, 40, pp.46-53.

[9] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. ”YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.” In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 7464-7475. 2023.