BLOOD CELL TYPE DETECTION USING DEEP LEARNING

Dr. D.V.Krishna Reddy[1], Associate Professor, Department of IT,
KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES,Vinjanampadu
Guntur Dt., Andhra Pradesh-522217.

Challa Jhansi Lakshmi Bhavani[2], **Dasari Hema Latha**[3], **Harshitha Telanakula**[4], **Alavalapati Pujitha**[5]
[2,3,4,5] UG Students, Department of IT,
KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES,Vinjanampadu
Guntur Dt., Andhra Pradesh-522217.
[1] **krishnareddydp@gmail.com**, [2] **jhansichalla45@gmail.com**,
[3] **hemadasari2003@gmail.com**,
[4] **hharshi347@gmail.com**, [5] **pujithaalavalapati27@gmail.com**

Abstract

Accurate and timely analysis of blood cells plays an important role in the diagnosis and treatment of many diseases. The manual blood testing method is labor-intensive, time-consuming, and prone to human error. In this context, the use of deep learning technology seems to be a great promise in terms of automating the process and increasing the efficiency of the treatment. This work presents a blood cell detection system based on deep learning algorithms. The YOLOv8 model is used to analyze microscopic images of blood samples and accurately identify identical cells. Periods must be trained to obtain higher predictions. The model is trained on many different data sets to ensure good performance across different blood types and conditions. Even with limited data, transfer learning can be used to optimize training models, thus improving the transfer process to different clinical settings. Rapid identification of blood cells allows doctors to make faster, more informed decisions and improve patient outcomes. Early detection of blood-related diseases leads to intervention and timely treatment, ultimately improving the quality of all medical services. Additionally, the automation of this application reduces the workload of laboratory workers, allowing them to focus on more complex tasks and be more efficient throughout the laboratory. The technology has the potential to be integrated into existing treatments, resulting in a variety of treatments.

Keywords: YOLO, pytorch, tensorflow, nvidia, flask, image processing.

Introduction

Cell division is usually done according to the morphological characteristics of the cells and requires a specialist physician. This process can be time consuming, tiring and expensive. For adults, there are approximately five liters of blood in the body, with blood cells making up approximately 45% of the tissue by volume. The three types of blood cells are red blood cells (RBCs), white blood cells (WBCs, including basophils, lymphocytes, neutrophils, mononuclear cells, and eosinophils), and platelets. Red blood cells are important transporters of oxygen, white blood cells are part of the immune system, and platelets have clotting functions and can heal bone injuries. Clinically, both physiological and pathological changes affect blood composition.

Therefore blood testing has become a direct function of self-diagnosis or disease diagnosis. A complete blood cell count (CBC) is a classic blood test that identifies and counts important blood cells to diagnose, monitor and manage blood transfusions.

Also, manual analysts can make mistakes when counting and separating different blood cells. This manual review can be replaced by machine learning-based automation technology, saving valuable time and reducing human effort and error.

Literature Review

You Only Look Once (YOLO) which acts as a good object detector to detect small objects in real-time and has seen significant advancements in recent years, with one important application is detection and analysis of blood cells to get accurate results. The Blood Cell Type Detection system requests an image from the user and generates a report accordingly. Traditional procedures are slow and may not be reliable but, recent developments have explored the use of various algorithms for more accurate blood cell detection and analysis.

Developments in machine learning have been significant in the creation of systems for detecting blood cells. In “**A Framework for White Blood Cell Segmentation in Microscopic Blood Images Using Digital Image Processing**”, Zainina Seman and Abdul Rahman Ramli's work, they used a machine learning approach for automatic identification and counting of three types of blood cells using ‘you only look once’ (YOLO) object detection model. Similarly, Bassem.S (2019) proposed “**A deep learning-based approach for detection of blood cells**” , which exceeded the traditional methods by giving more accurate results.In summary, the blood cell detection, counting and analysis using the YOLOv8 model holds great potential for creating more and responsive interfaces. The future research initiatives

might concentrate on enhancing the scalability, accuracy, and reducing development cost as well as resolving the privacy concerns for deploying.

Proposed Methodology

The proposed method for our blood cell type detection using deep learning consists of several key components:

1.YOLO based blood cell detection:

YOLO (You Only Look Once) is a popular object detection model known for its speed and accuracy. It was first introduced by Joseph Redmon et al. in 2016 and has since undergone several iterations, the latest being YOLO v12 Figure 1.1 shows Object detection algorithms are broadly classified into two categories based on how many times the same input image is passed through a network.

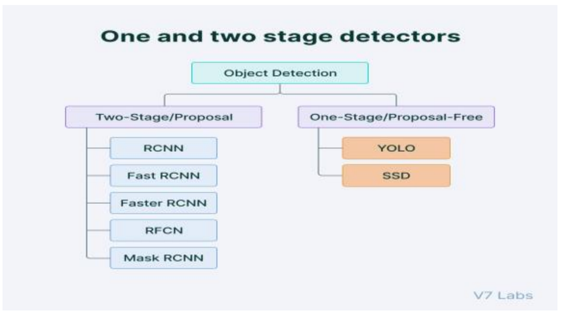


Fig.1.1 Object classifiers

While algorithms like Faster RCNN work by detecting possible regions of interest using the Region Proposal Network and then performing recognition on those regions separately, YOLO performs all of its predictions with the help of a single fully connected layer.

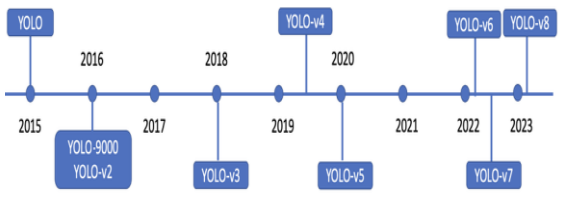
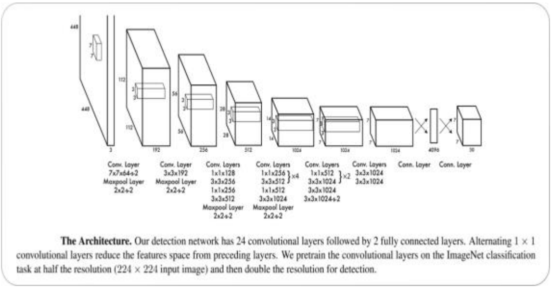


Fig.1.2. Timeline of YOLO

YOLO ARCHITECTURE

The YOLO algorithm takes an image as input and then uses a simple deep convolutional neural network to detect objects in the image. The architecture of the CNN model that forms the backbone of YOLO is shown below.



The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.

Fig.1.3. Process of YOLO model

2.YOLO Object Detection Work:

Imagine you built a YOLO application that detects blood cells from a given image

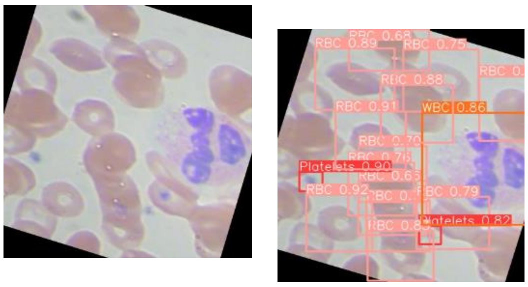


Fig.2.1.detection of blood cell before and after using YOLO

The algorithm works based on dividing the image into residual blocks and identifies the objects within the blocks with high speed and accuracy.

In terms of blood testing mode, YOLOv8 value optimizer can be used in medicine

to improve the accuracy, speed or efficiency of testing and classifying different blood cells such as egg red blood cells, white blood cells and platelets. photos or videos. This optimization will involve correcting or designing the YOLOv8 algorithm to better adapt to the characteristics of blood images and thus **3.Deep learning models**

The system was developed by a "You Only Look Once (YOLO)" model and along with PyTorch, TensorFlow frameworks. These are open-source frameworks well-suited

for blood cell analysis tasks like detection, count, and analysis and can be used to train models like YOLO for cell detection and classification in a single step. This eliminates the need for separate feature extraction and classification stages. This system offers a promising approach for efficient and consistent blood cell analysis and also provides faster analysis compared to traditional manual methods. This system can be scaled to handle large volumes of images.

1. Data Acquisition:

A dataset of public blood cell images is used to train the model. Identifying the blood cells and determining whether the cells are normal or abnormal. PyTorch integrates with libraries like TensorBoard for visualization of training data.

2. Data Preprocessing: The PyTorch and TensorFlow frameworks are used to Perform pre-processing and segmentation techniques on the blood smear image which is captured using the microscope camera and annotates the images with bounding boxes around each cell, specifying the cell type (RBC, WBC, Platelet). PyTorch offers functionalities for data loading, manipulation, and transformation steps and the TensorFlow functions are used to pre-process the images.

3. Post-Processing Unit: Receives the YOLO model's output image and applies a threshold on class probabilities to filter out low-confidence detections and further counts the number of detected cells of each type. By using TensorFlow we can monitor training progress using metrics like accuracy and precision/recall for each cell type.

4. User Interface: The interface generates a report containing the original image with detected cells highlighted by bounding boxes. Cell counts for each type.

4.Image processing techniques

The image processing techniques involve various steps like pre-processing (image reading, resizing, normalization,data augmentation), segmentation, feature extraction ,noise reduction,background removal and image sharpening.

1. Pre-processing: OpenCV offers functions for image reading, resizing to standardize image dimensions to match the YOLO model's input size, normalization of pixel values for better convergence during training and improved model performance , data augmentation.

2. Segmentation: OpenCV provides functionalities for various segmentation techniques:

Thresholding:Using Thresholding methods to achieve optimal segmentation of blood cells.

Watershed Transformation: Useful for separating touching or overlapping cells.

3. Feature Extraction: OpenCV offers functions to extract shape features,

intensity features to calculate area, perimeter, circularity, eccentricity to differentiate RBCs from WBCs.

4.Noise Reduction : Blood smear images might contain noise due to camera imperfections or staining processes. Techniques like median filtering or Gaussian filtering can help to remove noise without blurring important cell details.

5. Background Removal: In some cases, background regions in the image might not be relevant for analysis. Techniques like thresholding or background subtraction can isolate the foreground cells and improve model focus.

6. Image Sharpening: If cell boundaries appear blurry, sharpening techniques can enhance edges and improve YOLO's ability to localize cells accurately.

However, be cautious of over-sharpening, which can introduce artifacts.

Training Details:

In our training process, we opted for a total of 150 epochs. This choice was made after considering the complexity of the task, the size of the dataset, and computational resources. While training, the model learns from the dataset multiple times, allowing it to capture patterns and present them in the data.

We set the learning rate to 0.89 for our training process. This choice was based on initial experimentation and hyperparameter tuning. A higher learning rate can lead to faster convergence, but it must be carefully chosen to prevent the model from overshooting optimal parameters. Throughout training, the learning rate remains constant to ensure stable optimization.

For our object detection task, we employed a combination of loss functions to effectively train the model.

Bounding Box Regression Loss: This loss measures between predicted bounding box coordinates and the ground coordinates. Minimizing this loss encourages the model to accurately predict the locations of objects within the image.

Classification Loss:This loss encourages the model to correctly classify objects present in the image.

Implementation

The implementation of the project involves several components, including YOLO models, Deep learning techniques and some other required methods.Below is an overview of the implementation steps:

1. Environment Setup:

Install necessary libraries like TensorFlow, PyTorch (depending on chosen YOLO version), OpenCV for image processing, and scikit-learn for data manipulation.Download a pre-trained YOLO model (e.g., YOLOv5s) or choose a framework that allows easy model creation (e.g., YOLOv7 in PyTorch).

2. Data Preparation:

Obtain a blood cell image datasets using YOLOV8 data model with annotations for different cell types (bounding boxes and class labels as RBC,WBC,PLATELETS) from the roboflow and the epochs to get the more accuracy have to be trained to detect the images very effectively.

Epochs has to be trained to get the higher accuracy and predictions for the blood cell images

3. YOLO Model Customization:

If using a pre-trained model, modify the final layers to predict the desired number of blood cell types (e.g., RBC, WBC, Platelet).

Define the loss function by combining localization loss (e.g., IoU loss) and classification loss (e.g., cross-entropy) suitable for bounding box prediction and class probabilities.

4.Uploading image:

A python code is used to upload the image of blood cell smear microscopic images and click the analyze option.After, the detection and classification of blood cell types and return the count of total number cell regarding each type.

5.Deployment:

Depending on the application, the trained model can be deployed for real-time blood cell detection on new images.Here, we have trained the EPOCHs which is very useful for training the datasets very accurately and give the correct predictions.This might involve saving the model in a lightweight format and integrating it with a user interface for image processing and result visualization with high accuracy and speed.

Results

The results of the project will be as follows as:

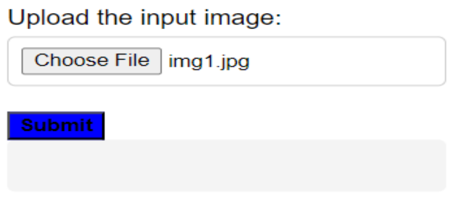


Fig.1. upload page

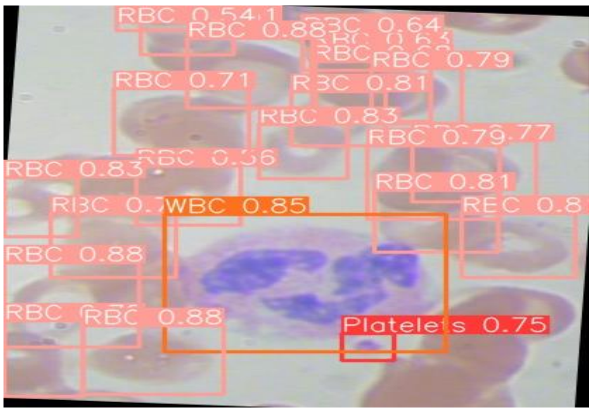
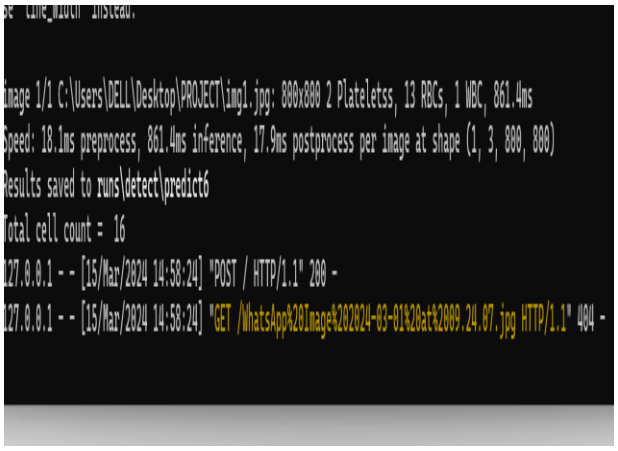


Fig.2. Blood cell types detection



```
se  line_width  instead.

image 1/1 C:\Users\DELL\Desktop\PROJECT\img1.jpg: 800x800 2 Plateletss, 13 RBCs, 1 WBC, 861.4ms
Speed: 18.1ms preprocess, 861.4ms inference, 17.9ms postprocess per image at shape (1, 3, 800, 800)
Results saved to runs\detect\predict6
Total cell count =  16
127.0.0.1 - - [15/Mar/2024 14:58:24] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [15/Mar/2024 14:58:24] "GET /WhatsApp%20Image%202024-03-01%20at%2009.24.07.jpg HTTP/1.1" 404 -
```

Fig .3. count of blood cells as class labels

Conclusion

"The Blood Cell Type Detection project successfully employs machine learning techniques to accurately classify blood cell types. Leveraging advanced algorithms, it enhances medical diagnostics with potential applications across various sectors.

Particularly promising is its utility in healthcare, aiding in early disease detection. This innovation marks a significant advancement in medical technology, demonstrating machine learning's capability to revolutionize blood cell analysis and improve patient care."

Limitations

The Limitations for the project may be as follows:

1.Data Imbalance: Limited availability of diverse and balanced datasets for different blood cell types can lead to biased models.

2.Model Performance: While YOLO models are fast and efficient, they may not always achieve the highest accuracy compared to other deep learning architectures, especially for small or subtle cell types.

3.Generalization: YOLO models may have difficulty generalizing to new or unseen cell types or variations in cell appearances.

4.Interpretability: Deep learning models, including YOLO, can be challenging to interpret, making it difficult to understand the reasoning behind their predictions.

5.Resource Intensive: Training and deploying YOLO models can be computationally intensive, requiring powerful hardware and significant time and effort.

Future Work

1.Dataset Improvement: Creating larger, more diverse, and balanced datasets for training YOLO models to improve their accuracy and generalization.

2.Model Optimization: Developing techniques to optimize YOLO models for better performance and efficiency in blood cell type detection.

3. Interpretability: Exploring methods to improve the interpretability of YOLO models, such as visualization techniques or attention mechanisms.

4. Transfer Learning: Investigating the use of transfer learning to leverage pre-trained YOLO models on related tasks or datasets to improve performance on blood cell type detection.

5.Real-timeImplementation: Developing strategies to deploy YOLO models for real-time blood cell type detection in clinical or diagnostic settings.

Acknowledgements

We would like to express our sincere gratitude to everyone who contributed to the successful completion of this project. First and foremost, we extend our heartfelt appreciation to the developers and contributors of the open-source tools and libraries that formed the backbone of our implementation, including Django, OpenAI, and the FER 2013 dataset creators.

We are deeply thankful to our project Guide Dr.D.V. Krishna Reddy sir and project Coordinator Dr.Guru Kesava Dasu

sir for their invaluable guidance, support, and insightful feedback throughout the project's development. Their expertise and encouragement have been instrumental in shaping our approach and overcoming challenges.

This project would not have been possible without the collective efforts and support of everyone mentioned above. We are truly grateful for their contributions and collaboration.

References

[1] Yao, X., Sun, K., Bu, X., Zhao, C. & Jin, Y. “Classification of white blood cells using weighted optimized deformable convolutional neural networks”. Artif. Cells Nanomed. Biotechnol. 49, 147–155 https://doi.org/10.1080/21691401.2021.1879823 (2021)

[2] Sharma, S. et al. Deep learning model for the automatic classification of white blood cells. Compute. Intell. Neurosci. 2022, 7384131 https://doi.org/10.1155/2022/7384131 (2022).

[3] Alhazmi, L. Detection of WBC, RBC, and platelets in blood samples using deep learning. Biomed. Res. Int. 2022, 1499546 https:// doi.org/10.1155/2022/1499546 (2022).

[4] Zhang, C. et al. Hybrid adversarial-discriminative network for leukocyte classification in leukemia. Med. Phys. 47, 3732–3744 https://doi.org/10.1002/mp.14144 (2020).

[5] Eman Alajrami1 , Bassem S. Abu-Nasser2 , Ahmed J. Khalil, Musleh M. Musleh2 , Alaa M. Barhoom2 , Samy's. Abu-Naser2 ,Blood Donation Prediction using Artificial Neural Network; ResearchGate; Volume 3; Issue 10, October – 2019.

[6] Ziquan zhu; Zeyu RenSiyuan lu,Shuihua wang,Yudong Zhang; A Deep Learning Network for Classifying Blood Cells; Big Data and Cognitive Computing; Volume 7; Issue 2 ;14 April, 2023;DOI:**10.3390/bdcc7020075.**

[7] Shin-Jyee Lee,Pei Yun Chen,Jeng-Wei Lin;Complete Blood Cell Detection and Counting Based on Deep Neural Networks; Applied Sciences; Volume 16; Issue 16; 14 August 2022; DOI:**10.3390/app12168140.**

[8] A. M. Patil, M. D. Patil and G. K. Birajdar, "White blood cells image classification using deep learning with canonical correlation analysis", IRBM; volume 42, no. 5; pp. **378-389, 2021.**

[9] S. J. Lee, P. Y. Chen and J. W. Lin;"Complete Blood Cell Detection and Counting Based on Deep Neural Networks";Applied Sciences; volume 12, no. 16, DOI:**10.3390/app12168140,2022**.

[10] Ihab Zaqout,”Diagnosis of Skin Lesions Based on Dermoscopic Images Using Image Processing Techniques” ,Research gate, DOI:10.5772/intechopen.88065,2019

[11] Mohammad Mahmudul Alam , Mohammad Tariqul Islam; Machine learning approach of automatic identification and counting of blood

cells ; Healthcare Technology Letters; Volume 10; December 2023; DOI:**10.1049/htl.2018.5098**

[12] Mesut Togacar , Burhan Ergen , Zafer Comert ; Classification of white blood cells using deep features obtained from Convolutional Neural Network models based on the combination of feature selection methods; Applied Soft Computing; Volume 97 part B ; 16 October,2020; DOI:**10.1016/j.asoc.2020.106810**

[13] K. A. K. Al-Dulaimi," *White blood cells classification using higher order spectra and l-moments invariant features "*, Queensland University of Technology School of Electrical Engineering and Robotics Science and Engineering Faculty,2021.

[14] Lorenzo Putzu, and Cecilia Di Ruberto,"White Blood Cells Identification and Counting from Microscopic Blood Image",International Journal of Medical, Health, Biomedical and Pharmaceutical Engineering; Volume 7; 2013.

[15] Mesut Togacar , Burhan Ergen , Zafer Comert ; Classification of white blood cells using deep features obtained from Convolutional Neural Network models based on the combination of feature selection methods; Applied Soft Computing; Volume 97 part B ; 16 October,2020; DOI:**10.1016/j.asoc.2020.106810**