Blood Cell Detection using SVM, Logistic Regression and YOLO V5

Burugupalli Sai Chaitanya, Kodati Jaideep & Himanshu

[1*]Computer Science Department, BML Munjal University, Gurgaon, Haryana, India.

Abstract

Background and Objectives: An integral part of contemporary healthcare diagnostics is blood cell analysis, which offers vital information about a patient's haematological health. Advanced technology and algorithms found in haematology analyzers provide quick and precise evaluation of red blood cells, white blood cells, and platelets. The efficiency of blood cell identification has increased with the introduction of sophisticated image processing and machine learning techniques, which are crucial to medical diagnostics. Using the BCCD dataset, the main goals of this study are to create and evaluate three distinct blood cell detection models: YOLO V5, Support Vector Machine (SVM), and Logistic Regression. The purpose of the study is to compare a deep learning model with conventional machine learning techniques and assess each model's accuracy and resilience. **Material & Methods**: In this project, the U-Net, Support Vector Machine (SVM), and Logistic Regression were employed for the segmentation and classification tasks. The YOLO V5 architecture was utilized for the Classification of red blood cells (RBCs) and white blood cells (WBCs), providing a detailed understanding of cell morphology. SVM was trained on features extracted from the segmented blood cell images. And Logistic Regression Logistic Regression was applied to the feature vectors derived from the segmented blood cell images. **Results**: The developed blood cell detection system demonstrates promising results in accurately identifying and classifying blood cell types. Logistic Regression, SVM and YOLO V5 models exhibit notable performance, achieving high accuracy and precision. The utilization of PCA aids in dimensionality reduction while preserving essential information. The system's effectiveness is validated on diverse datasets, showcasing its potential for application in medical diagnostics and research.

Keywords: Blood cell classification, Histogram of Oriented Gradients (HOG), Logistic Regression, Support Vector Machine (SVM), YOLO V5 Principal Component Analysis (PCA)**.**

1 Introduction

The automated detection and classification of blood cells hold significant importance in the realm of medical diagnostics, offering enhanced efficiency and accuracy in disease identification and treatment planning. The intricate task of distinguishing between red blood cells (RBC), white blood cells (WBC), and platelets has traditionally relied on manual examination, presenting challenges in terms of time consumption and subjectivity. To address these challenges, this study endeavors to develop a robust blood cell detection system that leverages advanced machine learning techniques.

The primary objective of this research is to contribute to the advancement of medical diagnostics by providing a reliable and automated method for blood cell classification. Through the utilization of the Blood Cell Count and Detection (BCCD) Dataset, a comprehensive collection of annotated blood cell images, we aim to create a model capable of accurately identifying and categorizing different blood cell types. This system is envisioned to have broad applications in medical research, facilitating studies related to hematological disorders, infection detection, and overall health assessment.

The methodology involves a multi-step process, including image preprocessing techniques for data enhancement, feature extraction using Histogram of Oriented Gradients (HOG) to capture relevant structural information, and the training of machine learning models such as Logistic Regression and Support Vector Machine (SVM) for accurate classification. Principal Component Analysis (PCA) is further employed to reduce dimensionality, optimizing computational efficiency without compromising classification accuracy.

In the context of blood cell detection, YOLOv5's ability to efficiently process high-resolution images with multiple objects aligns seamlessly with the requirements of medical image analysis. The model's precision and speed make it an ideal candidate for accurately identifying and classifying individual blood cells, including red blood cells (RBC), white blood cells (WBC), and platelets.

This section of the project leverages YOLOv5 to perform comprehensive blood cell detection on the Blood Cell Count and Detection (BCCD) dataset. By adopting YOLOv5, the study aims to assess the model's effectiveness in localizing and classifying blood cells within the images, contributing valuable insights to the broader field of medical image processing.

This study extends beyond the exploration of individual models and aims to compare their performance comprehensively. By evaluating the strengths and limitations of both traditional machine learning and pre-trained model approaches, the research seeks to contribute valuable insights to the evolving field of computer-aided diagnostics, especially in the context of blood cell analysis. The outcomes of this study may not only enhance the accuracy of blood cell detection but also inform future advancements in medical imaging and diagnostic methodologies.

2 Related Work

The field of blood cell detection and classification has witnessed significant advancements, as evidenced by several noteworthy research studies. In this section, we review key contributions from relevant literature, highlighting methodologies, findings, and the overall landscape of automated blood cell analysis.

The literature review encompasses a comprehensive examination of various studies conducted on blood cell image analysis, with a focus on methodologies employed, datasets used, and key findings. In the study by Grzegorz Drałus, Damian Mazur, and Anna Czmil, the authors utilized the Blood Cell Image Library (BCID) and Blood Cell Image Segmentation Benchmark (BCIS) datasets, employing Convolutional Neural Network (CNN) and Faster R-CNN methodologies. Their model achieved high accuracy, with 98.5% for blood cell detection and 97.5% for counting.

Shin-Jye Lee, Pei-Yun Chen, and Jeng-Wei Lin employed a modified VGGNet architecture and Faster R-CNN for blood cell detection, classification, and counting on the BCCD dataset. Their model demonstrated an impressive accuracy of 96.7% on the test set. In a study by Tiancheng Xia, Richard Jiang, YongQing Fu, and Nanlin Jin, Faster Region-based Convolutional Neural Networks (Faster RCNNs) were employed, yielding accuracy rates of 92% on the BCCD dataset and 85% on the BCID dataset.

Vilas B. Inchur, L. S. Praveen, and Preetham Shankpal utilized morphological operators, edge operators, texture region-based classification, and Circular Hough Transformation on BCID and BCIS datasets. Their findings indicated 91% accuracy with CHT and a morphological operator. G.P.M Priyankara, O.W Seneviratne, R.K.O.H Silva, W.V.D Soysa, and C.R. De Silva employed CNNs for blood cell detection and counting on the Blood Cell Image Library (BCCL) and Cell Tracking Challenge (CTC) datasets, achieving detection accuracy of 95.5%

and counting accuracy of 94.5% on CTC, and 97.5% detection and 96.5% counting accuracy on BCCL.

Hersh Abdulrahman Muhamad, Shahab Wahhab Kareem, and Amin Salih Mohammed compared Support Vector Machines (SVM) and CNN methodologies, with CNN achieving 98.4% correct classifications. Lamia Alhazmi utilized a CNN for blood cell identification, achieving 100% accuracy for white blood cells, 89% for red blood cells, and 96% for platelets. Mohammad Saied Rahnama, Farzad Khalili-Mahani, and Seyed-Javad Seyed-Hosseini combined image processing and machine learning with SVM classifiers, achieving 98% accuracy for red blood cell detection and 99% for white blood cells, along with 97% accuracy in cell counting.

Finally, Madhuri G. Bhamare and D.S.Patil employed SVM and morphological operations, achieving a detection accuracy of 90% on the test set. Collectively, these studies showcase the diverse approaches and notable successes in blood cell image analysis, offering valuable insights for future research in this domain..

3 Methodology

3.1 Data-Set

The Blood Cell Count and Detection (BCCD) Dataset is a comprehensive and publicly available collection of annotated blood cell images designed for research in automated blood cell analysis. This dataset serves as a valuable resource for developing and evaluating algorithms related to blood cell detection, classification, and counting. The dataset contains 410 images and annotations for red blood cells, white blood cells, and platelets. The images are 640x480 pixels and the annotations are in the VOC format. This dataset is accompanied by an additional dataset containing the original 410 images (pre-augmentation) bounding boxes for each cell in each of these 410 images (JPEG + XML metadata). BCCD dataset divided into training (70%) and testing (30%) sets. Each image in the dataset represents blood cells in varying conditions and configurations.

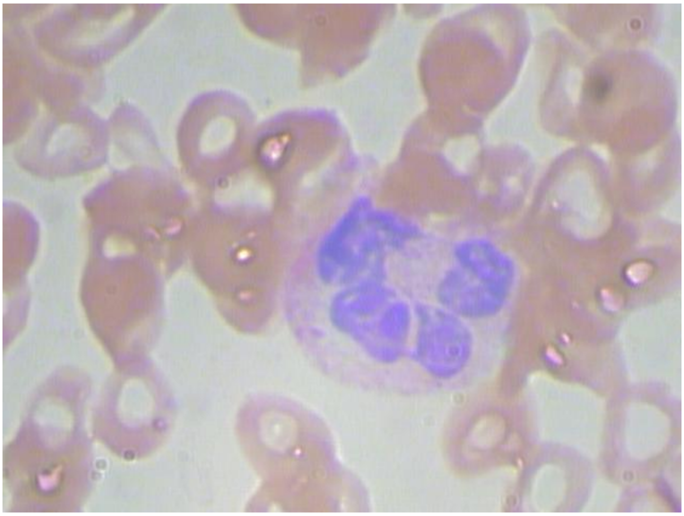


Fig. 3.1: BCCD Dataset Image

3.2 Pre-Processing

The pre-processing phase plays a crucial role in preparing the dataset for effective model training and ensuring uniformity in input data. The following steps outline the comprehensive pre-processing pipeline:

Resizing:

The process of resizing images to a standardized dimension of 64x64 pixels plays a crucial role in maintaining consistency across the dataset. Standardization is essential for ensuring that all images share the same dimensions, simplifying subsequent processing steps and facilitating the comparison and analysis of features across different samples.

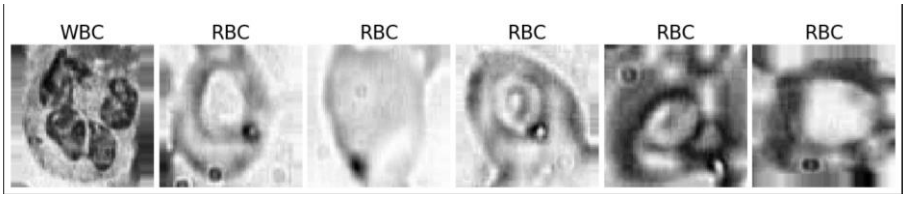


Fig. 3.2: Resized Images

Grayscale Conversion:

The conversion of RGB images to grayscale is a fundamental step aimed at simplifying the processing pipeline. Grayscale images represent a single channel of intensity values, as opposed to the three channels (red, green, and blue) present in RGB images. This simplification reduces computational complexity and focuses the analysis on essential features related to luminance and contrast.

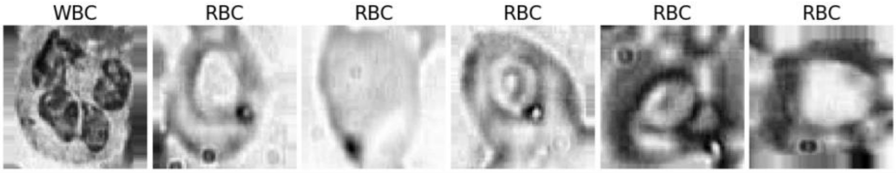


Fig. 3.3: Grayscale Images

Canny Edge Detection:

Canny edge detection is employed to enhance the visibility of edges within the images, a critical aspect in the analysis of blood cell morphology. Otsu thresholding is used to automatically determine the optimal threshold for binary edge detection, ensuring adaptability to variations in image intensity. Gaussian filtering is applied for image smoothing, reducing noise and creating a more refined edge map. The combination of these techniques enhances the precision of subsequent analyses by emphasizing significant features in the blood cell images.

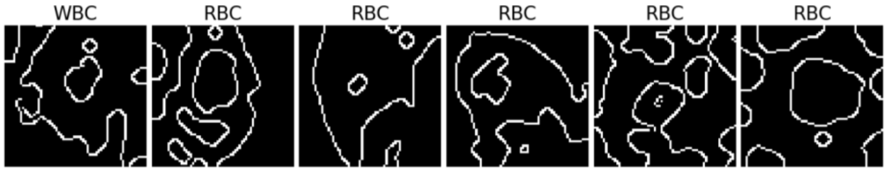
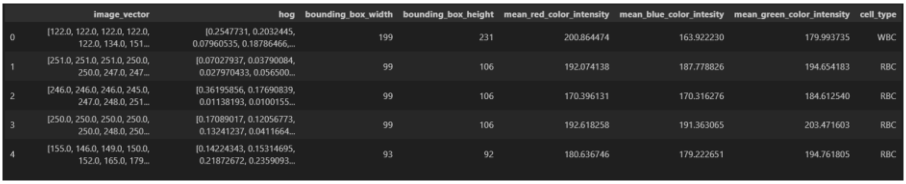


Fig. 3.4: Canny Edge Detection Images

Vectorization:

Following preprocessing steps, the grayscale images are flattened into one-dimensional vectors of size (4060, 1). This vectorization process transforms the pixel grid of each image into a linear format, facilitating compatibility with machine learning algorithms. The resulting vectors encapsulate the essential information regarding the intensity and spatial distribution of pixels, setting the stage for subsequent feature extraction and model training.



	image_vector	hog	bounding_box_width	bounding_box_height	mean_red_color_intensity	mean_blue_color_intesity	mean_green_color_intensity	cell_type
0	[122.0, 122.0, 122.0, 122.0, 122.0, 134.0, 151...	[0.2547731, 0.2032445, 0.07960535, 0.18786466,...	199	231	200.864474	163.922230	179.993735	WBC
1	[251.0, 251.0, 251.0, 250.0, 250.0, 247.0, 247...	[0.07027937, 0.03790084, 0.027970433, 0.056500...	99	106	192.074138	187.778826	194.654183	RBC
2	[246.0, 246.0, 246.0, 245.0, 247.0, 248.0, 251...	[0.36195856, 0.17690839, 0.01138193, 0.0100155...	99	106	170.396131	170.316276	184.612540	RBC
3	[250.0, 250.0, 250.0, 250.0, 250.0, 248.0, 250...	[0.17089017, 0.12056773, 0.13241237, 0.0411664...	99	106	192.618258	191.363065	203.471603	RBC
4	[155.0, 146.0, 149.0, 150.0, 152.0, 165.0, 179...	[0.14224343, 0.15314695, 0.21872672, 0.2359093...	93	92	180.636746	179.222651	194.761805	RBC

Fig. 3.5: Vectorization

HOG Feature Extraction: Histogram of Oriented Gradients (HOG) is employed to capture relevant features related to the distribution of gradients within the preprocessed images. By analyzing the orientation and magnitude of gradients in localized regions, HOG provides a descriptive representation of image structure. This feature extraction method is particularly effective in highlighting the distinctive patterns and contours of blood cells, serving as a foundation for subsequent model training. The HOG features offer a more nuanced understanding of the morphological characteristics critical for accurate blood cell detection and classification.

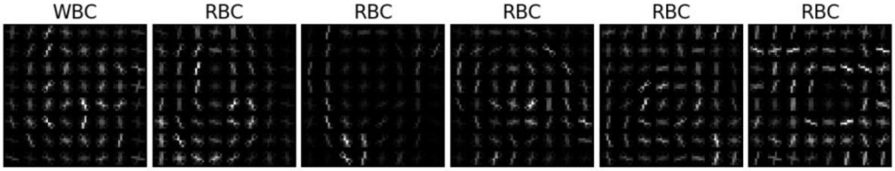


Fig. 3.6: HOG Images

Normalization:

Image pixel values are normalized to the range [0, 1]. Normalization ensures numerical stability during model training and helps achieve faster convergence.

Dataset Splitting:

The dataset is split into training and testing sets using the train_test_split function from scikit-learn. This division ensures an independent evaluation of the model on unseen data.

Class Balancing:

Balancing the class distribution is essential to prevent the model from being biased toward the majority class. Techniques such as oversampling or under sampling are applied, ensuring each cell type is adequately represented in the training set.

We can see that we have highly imbalanced classes. It is a problem because possibly any classification algorithm will predict RBC with the greatest probability.

Possible solutions:

- **Random Oversampling:** randomly duplicate samples in the minority class.
- **Random Undersampling:** randomly delete samples in the majority class.

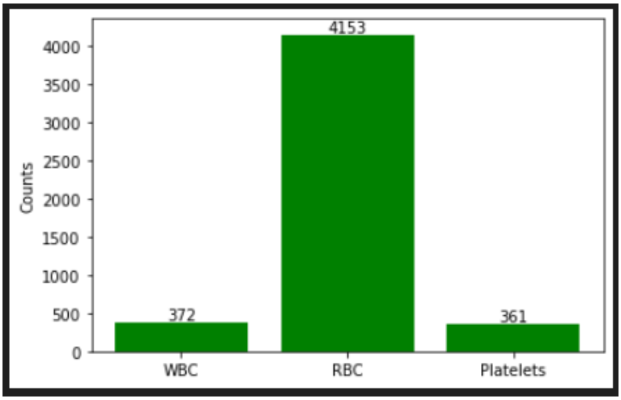


Fig. 3.7: Class Balancing

Of course, both have their pros and cons but random resampling provides a naive technique for rebalancing the class distribution for an imbalanced dataset. Actually in our case the data is representative as red cells are much more than white cells and platelets.

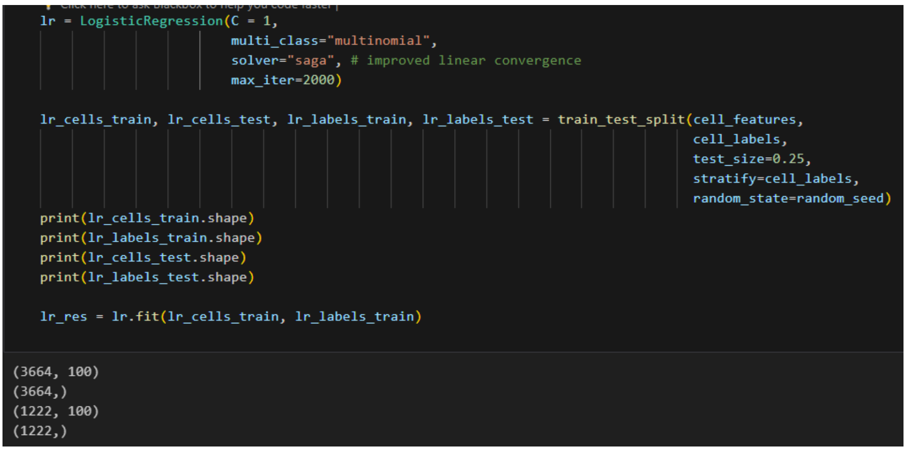
RBCs account for approximately 40 to 45 percent of the blood. This percentage of blood made up of RBCs is a frequently measured number and is called the hematocrit. The ratio of cells in the normal blood is 600 RBCs for each white blood cell and 40 platelets. So, at least initially we will not try to perform any resampling on the training data.

3.3 Model Architecture

1. Logistic Regression:

As Logistic Regression is one of the basic simple and most commonly used ML algorithms, we can use it as a baseline for our modelling. Actually, Logistic Regression is used for binary classification i.e. two-class classification. It describes and estimates the relationship between one dependent binary variable and an independent one.

However, in our case we have three classes and we will use the multinomial logistic regression i.e. the target variable has three or more nominal categories such as predicting types of cells. Multinomial logistic regression uses again a linear function.



```
lr = LogisticRegression(C = 1,
                        multi_class="multinomial",
                        solver="saga", # improved linear convergence
                        max_iter=2000)

lr_cells_train, lr_cells_test, lr_labels_train, lr_labels_test = train_test_split(cell_features,
                                                                                  cell_labels,
                                                                                  test_size=0.25,
                                                                                  stratify=cell_labels,
                                                                                  random_state=random_seed)
print(lr_cells_train.shape)
print(lr_labels_train.shape)
print(lr_cells_test.shape)
print(lr_labels_test.shape)

lr_res = lr.fit(lr_cells_train, lr_labels_train)
```

```
(3664, 100)
(3664,)
(1222, 100)
(1222,)
```

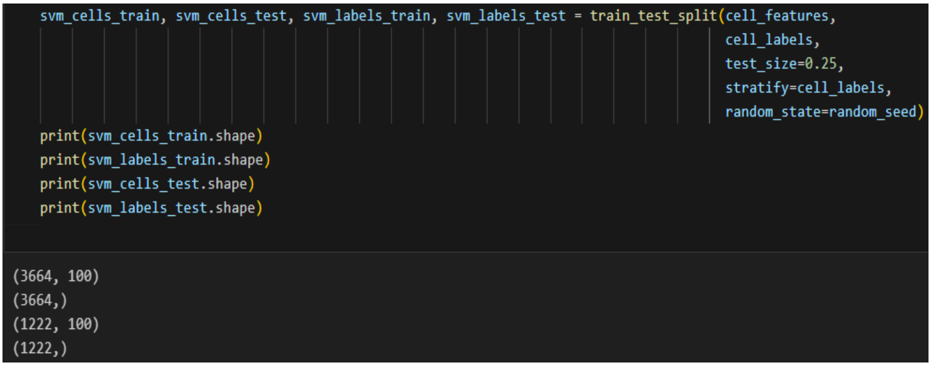
Fig. 3.8: Logistic Regression Model Architecture

The Linear Regression model employed in this study leverages a diverse set of features encompassing bounding box dimensions, color intensities (mean red, blue, green), and Histogram of Oriented Gradients (HOG) features. The architectural design revolves around a single-layer model, where this lone layer ingests the amalgamated input features. During operation, the linear regression model undertakes a linear combination of these diverse features to generate predictions directly aligned with the classification of blood cell types, namely Platelets, RBC, and WBC.

To equip the model with the ability to discern intricate patterns within the data, it has been trained rigorously on a comprehensive dataset that includes blood cell images along with their corresponding annotations. The efficacy of the model is subsequently gauged through a set of evaluation metrics, including mean accuracy, confusion matrix, and a detailed classification report, providing valuable insights into its performance characteristics.

2. Support Vector Machine (SVM):

The Support Vector Machine (SVM) model, in tandem with the Linear Regression counterpart, operates on a feature vector encompassing bounding box dimensions, color intensities (mean red, blue, green), and Histogram of Oriented Gradients (HOG) features. Distinguished by its discriminative nature, the SVM architecture is geared towards identifying the optimal hyperplane that effectively separates different classes within the data.



```
svm_cells_train, svm_cells_test, svm_labels_train, svm_labels_test = train_test_split(cell_features,
                                                                                      cell_labels,
                                                                                      test_size=0.25,
                                                                                      stratify=cell_labels,
                                                                                      random_state=random_seed)
print(svm_cells_train.shape)
print(svm_labels_train.shape)
print(svm_cells_test.shape)
print(svm_labels_test.shape)
```

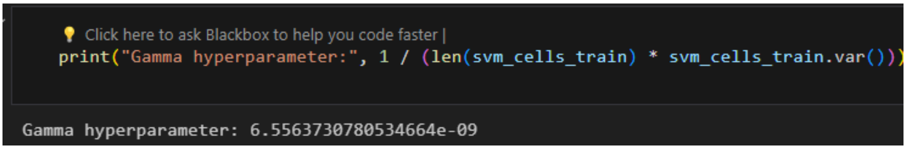
```
(3664, 100)
(3664,)
(1222, 100)
(1222,)
```

Fig. 3.9: Support Vector Machine Model Architecture

To accomplish this, the model harnesses the power of a kernel function, specifically the Radial Basis Function (RBF), for robust feature mapping. Predictions from the SVM model are dictated by the learned decision boundaries, offering a nuanced understanding of the blood cell types—Platelets, RBC, and WBC.

An essential facet of the SVM model lies in its use of the RBF kernel to map input features into a higher-dimensional space, enhancing its capability to discern intricate patterns. To address imbalances in the dataset, the model incorporates class weights during training, ensuring equitable representation of each class. Performance evaluation is conducted through various metrics, including accuracy, confusion matrix, and a

comprehensive classification report, offering a holistic perspective on the model's effectiveness and generalization.
We have quite a small value of gamma, so in this way we have a smoother decision boundary as more points, not just the ones closer to the support vectors, influence it.



```
print("Gamma hyperparameter:", 1 / (len(svm_cells_train) * svm_cells_train.var()))
```

```
Gamma hyperparameter: 6.5563730780534664e-09
```

Fig. 3.10: Gamma hyperparameter

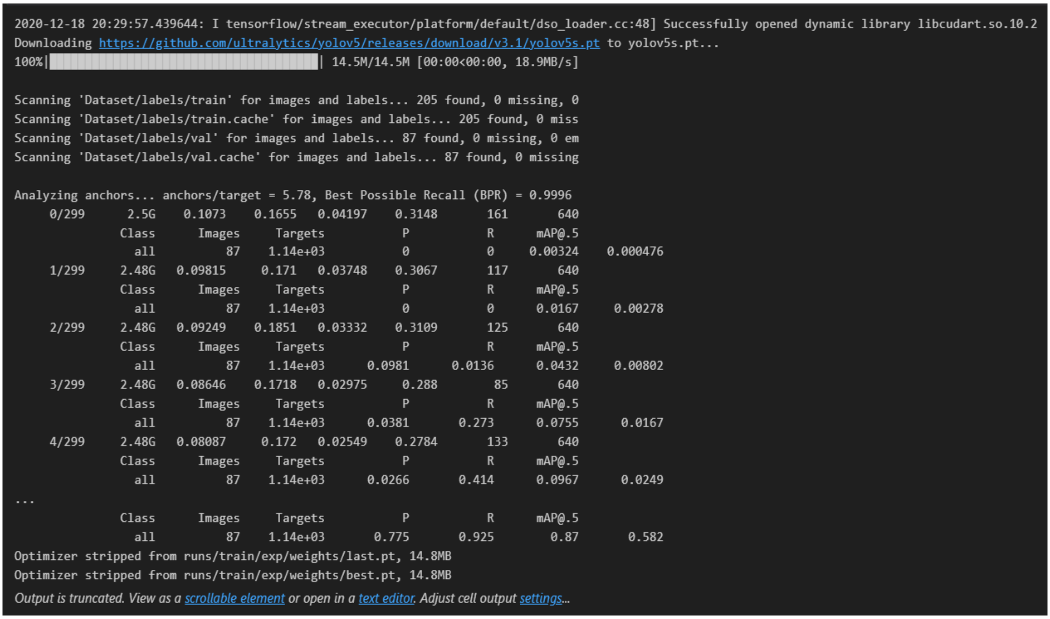
3. YOLO V5:

You Only Look Once version 5 (YOLOv5) stands at the forefront of modern object detection algorithms, representing a significant leap in performance and efficiency. As the latest iteration of the YOLO series, YOLOv5 excels in real-time object detection tasks, making it particularly suitable for applications requiring swift and accurate identification of multiple object classes within images.

The YOLO V5 architecture used in this project is designed to recognise blood cell types such as Platelets, RBC, and WBC within the BCCD dataset. Images from the BCCD dataset are used as input, along with XML files containing annotation metadata. Preprocessing entails converting XML annotations into Darknet format using the cord_converter function, which facilitates the translation of bounding box coordinates into YOLO format (x, y, w, h).

It renowned for its prowess in object detection, is harnessed for this task, leveraging the Darknet framework to construct a deep neural network tailored to object detection challenges. The training phase involves meticulous data organization, encompassing the creation of essential directories for YOLO V5 training, including images and labels. Training is executed with carefully configured parameters such as image size, batch size, epochs, and YAML configuration files. The progress of the training process is tracked and stored in the runs/train directory.

Upon completion of training, the trained YOLO V5 model is employed for inference on test images using the detect.py script. The resulting images, adorned with bounding box predictions, are visually inspected, facilitated by the Image class. To ensure the model's understanding of the dataset, a YAML configuration file (data/bccd.yaml) is generated, encapsulating



```
2020-12-18 20:29:57.439644: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully opened dynamic library libcudart.so.10.2
Downloading https://github.com/ultralytics/yolov5/releases/download/v3.1/yolov5s.pt to yolov5s.pt...
100%|████████████████████████████████████████| 14.5M/14.5M [00:00<00:00, 18.9MB/s]

Scanning 'Dataset/labels/train' for images and labels... 205 found, 0 missing, 0
Scanning 'Dataset/labels/train.cache' for images and labels... 205 found, 0 miss
Scanning 'Dataset/labels/val' for images and labels... 87 found, 0 missing, 0 em
Scanning 'Dataset/labels/val.cache' for images and labels... 87 found, 0 missing

Analyzing anchors... anchors/target = 5.78, Best Possible Recall (BPR) = 0.9996
     0/299      2.5G    0.1073    0.1655   0.04197    0.3148       161       640
               Class      Images     Targets           P           R      mAP@.5
                 all          87    1.14e+03           0           0     0.00324    0.000476
     1/299     2.48G   0.09815     0.171   0.03748    0.3067       117       640
               Class      Images     Targets           P           R      mAP@.5
                 all          87    1.14e+03           0           0      0.0167     0.00278
     2/299     2.48G   0.09249    0.1851   0.03332    0.3109       125       640
               Class      Images     Targets           P           R      mAP@.5
                 all          87    1.14e+03      0.0981      0.0136      0.0432     0.00802
     3/299     2.48G   0.08646    0.1718   0.02975     0.288        85       640
               Class      Images     Targets           P           R      mAP@.5
                 all          87    1.14e+03      0.0381       0.273      0.0755      0.0167
     4/299     2.48G   0.08087     0.172   0.02549    0.2784       133       640
               Class      Images     Targets           P           R      mAP@.5
                 all          87    1.14e+03      0.0266       0.414      0.0967      0.0249
...
               Class      Images     Targets           P           R      mAP@.5
                 all          87    1.14e+03       0.775       0.925        0.87       0.582
Optimizer stripped from runs/train/exp/weights/last.pt, 14.8MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.8MB
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Fig. 3.11: YOLO V5 Model Architecture

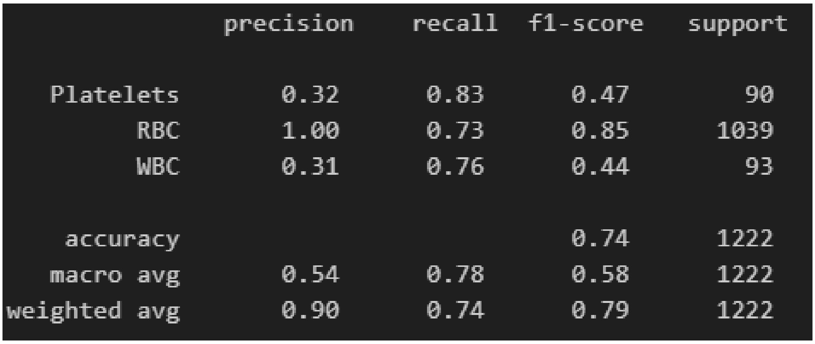
information about the dataset's structure, including directories for training and validation, the number of classes (3), and class names (Platelets, RBC, WBC).

4 Results

The performance of the models is tested through various evaluation metrics and techniques, Accuracy, Confusion Matrix, Precision, Recall, and F1-Score. The combination of these evaluation strategies offers a holistic view of the models' performance, enabling the identification of their strengths and areas for improvement. This thorough evaluation is crucial for selecting the most suitable model for blood cell detection in practical applications.

4.1 Logistic Regression

The logistic regression (LR) model exhibits a reasonable performance on the blood cell dataset. With an accuracy of approximately 74.2% on the test set, the LR model demonstrates its ability to make predictions based on features such as bounding box dimensions, color intensities, and Histogram of Oriented Gradients (HOG) features. The classification report further details the precision, recall, and F1-score for each class (Platelets, RBC, WBC). Notably, the LR model achieves a high recall for RBC, indicating proficiency in identifying red blood cells.



```
              precision    recall  f1-score   support

   Platelets       0.32      0.83      0.47        90
         RBC       1.00      0.73      0.85      1039
         WBC       0.31      0.76      0.44        93

    accuracy                           0.74      1222
   macro avg       0.54      0.78      0.58      1222
weighted avg       0.90      0.74      0.79      1222
```

Fig. 4.1: Classification report for logistic regression

As a whole, as a simple algorithm Linear Regression performs quite well as a percentage of the predicted cell types. It predicts a lot of WBCs and Platelets, not only RBCs but unfortunalely the precision is quite low. There are a lot of false possitives i.e. RBCs which are defined as WBCs and Platelets. Overall, it is not a very good result as actually the occurence of more WBCs is a sign of a disease.

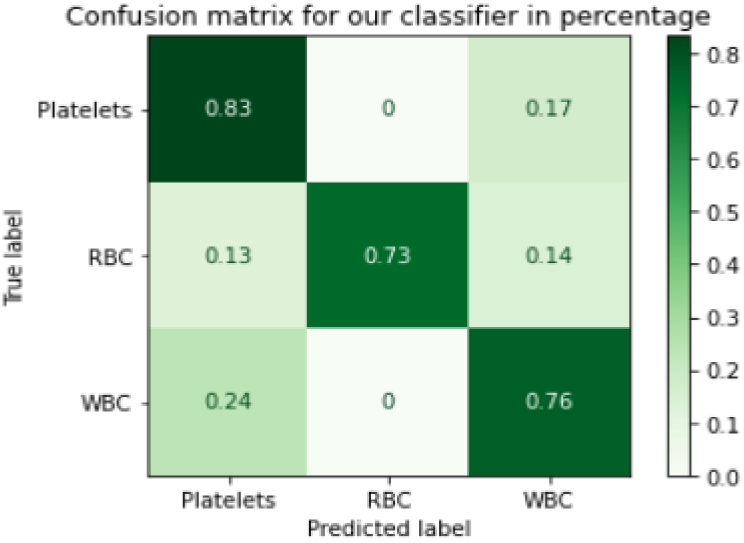
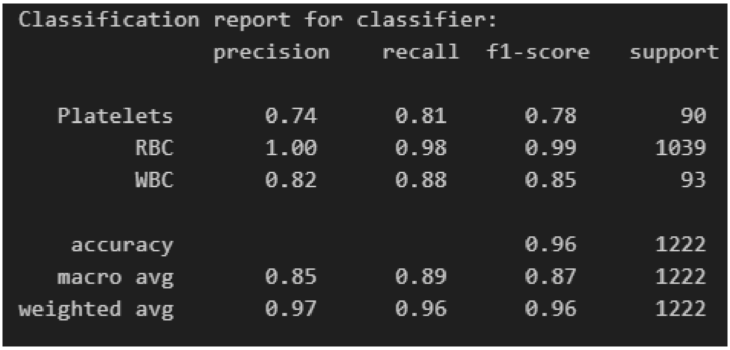


Fig. 4.2: Confusion matrix for logistic regression

4.2 Support Vector Machine(SVM)

The Support Vector Machine (SVM) model delivers superior results, achieving an impressive accuracy of 96% on the test set. The SVM model, leveraging the Radial Basis Function (RBF) kernel and incorporating class balancing, excels in accurately classifying Platelets, RBC, and WBC. The classification report underscores the SVM model's high precision and recall across all classes, emphasizing its robust performance on the blood cell dataset.



```
Classification report for classifier:
              precision    recall  f1-score   support

   Platelets       0.74      0.81      0.78        90
         RBC       1.00      0.98      0.99      1039
         WBC       0.82      0.88      0.85        93

    accuracy                           0.96      1222
   macro avg       0.85      0.89      0.87      1222
weighted avg       0.97      0.96      0.96      1222
```

.

Fig. 4.3: Classification report for SVM

It seems that the result is similar to the baseline logistic regression maybe a little bit better. Actually, if we look at the percentage of predicted WBCs and Platelets the classifier seems to perform quite well. Even the precision is definitely higher than the one of Logistic Regression.

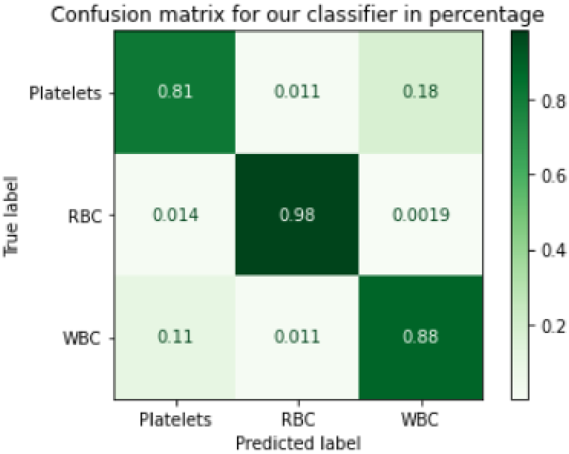


Fig. 4.4: Confusion matrix for SVM

Visualization of SVM Model Predictions:

For a more intuitive understanding of the SVM model's performance, a selected range of cells was visually inspected. The range, spanning from cell indices 100 to 105, was chosen for this illustrative purpose. The corresponding image vectors for these cells were extracted and reshaped into 64x64 images for visualization.

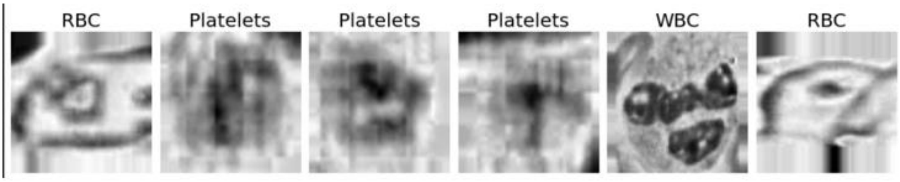


Fig. 4.5: Predicted Cells Images in SVM model

The SVM model was then applied to predict the cell types for the selected cells, providing insights into its classification capabilities. The visualized results showcase a side-by-side comparison of the actual cell types and the predicted cell types. This visual inspection serves as a qualitative assessment, offering a glimpse into the model's ability to generalize and accurately classify blood cell types.

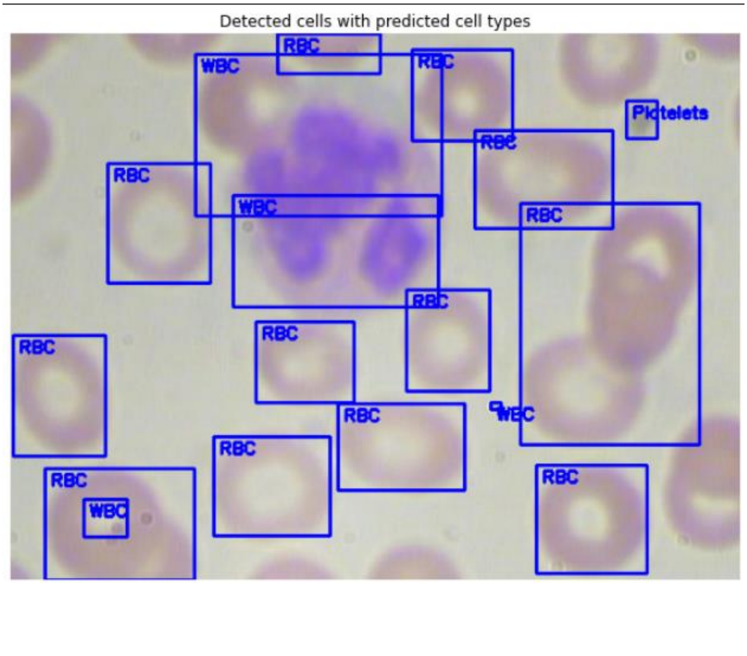


Fig. 4.6: Blood Cells Prediction using SVM model.

Extracting cell features, obtaining possible cell bounding boxes based on contours, and displaying images with predicted bounding boxes and cell types. Additionally, two specific images from the dataset, indexed at 398 and 388, are processed to showcase the detection and classification results. The SVM model is employed for predicting cell types.

4.3 YOLO V5

The YOLO V5 model, trained on the BCCD dataset with specific configurations (640x640 image size, batch size of 8, 300 epochs), exhibited robust performance in blood cell detection. The training process optimized the YOLO V5 architecture, showcasing consistent decreases in classification and regression losses. Evaluation on a validation set of 87 images revealed varying performance metrics (precision, recall, mAP) across Platelets, RBC, and WBC classes, with potential improvements observed over epochs.

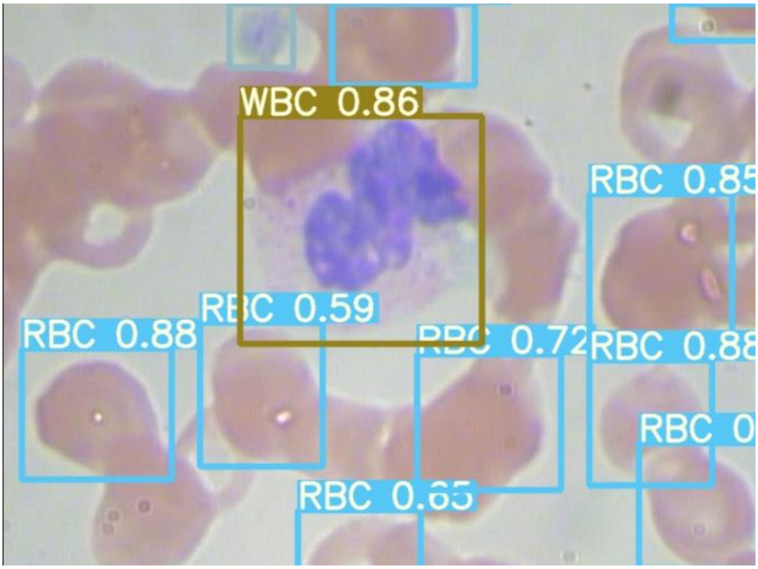


Fig. 4.: Blood Cells Prediction using YOLO V5 Pre-trained model.

Applied to a test set of 72 BCCD images, YOLO V5 successfully detected Platelets, RBCs, and WBCs, providing valuable insights into the model's localization and classification abilities. Visualizing an example test image ('BloodImage_00007.jpg') highlighted the model's proficiency in delineating blood cell types through bounding boxes. The YOLO V5 architecture demonstrated its effectiveness in medical image analysis, suggesting potential for further refinement based on specific metrics and objectives.

5 Conclusion

This project focuses on blood cell detection using three distinct models: linear regression, support vector machine (SVM), and the YOLO V5 object detection model. Each model is built to solve the particular issues of detecting Platelets, Red Blood Cells (RBC), and White Blood Cells (WBC) by leveraging the BCCD dataset with 410 pictures and related annotations. The models are assessed using a variety of measures, such as mean accuracy, confusion matrix, and classification report.

A single-layer architecture is used in the Linear Regression model to predict blood cell types directly from variables such as bounding box dimensions, colour intensities, and Histogram of Oriented Gradients (HOG). It is trained and evaluated, demonstrating its performance on the supplied dataset.

As a discriminative model, the Support Vector Machine (SVM) model employs a kernel function (Radial Basis Function - RBF) for effective feature mapping. The SVM model performs well in blood cell classification because to an emphasis on class balance and thorough assessment measures.

The YOLO V5 architecture, a state-of-the-art object detection model, is adapted for blood cell detection. With input preprocessing, training, and inference phases, the YOLO V5 model excels in real-time object detection tasks. The model's architecture, YAML configuration, and training and inference execution are thoroughly explained, showcasing its proficiency in accurately identifying blood cell types.

Results from Linear Regression, SVM, and YOLO V5 models are presented, offering insights into their respective performances. The SVM model exhibits high accuracy, supported by a detailed classification report, while the YOLO V5 architecture excels in real-time detection tasks, making it a powerful solution for medical image analysis.

References

[1] Cheung C, Thomas G, Tay W, Ikram M, Hsu W, Lee M, et al. Retinal vascular fractal dimension and its relationship with cardiovascular and ocular risk factors. Am J Ophthalmol. 2012;154(4):663–74. doi: 10.1016/j.ajo.2012.04.016 [PubMed] [CrossRef] [Google Scholar]

[2] Popovic N, Vujosevic S, Popovic T. Regional patterns in retinal microvascular network geometry in health and disease. Sci Rep. 2019;9(1). doi: 10.1038/s41598-019-52659-8 [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[3] Cheung C, Tay W, Mitchell P, Wang J, Hsu W, Lee M, et al. Quantitative and qualitative retinal microvascular characteristics and blood pressure. J Hypertens. 2011;29(7):1380–91. doi: 10.1097/HJH.0b013e328347266c [PubMed] [CrossRef] [Google Scholar]

[4] Popovic N, Lipovac M, Radunovic M, Ugarte J, Isusquiza E, Beristain A, et al. Fractal characterization of retinal microvascular network morphology Microcirculation. 2019;26(4):e12531. doi: 10.1111/micc.12531 [PubMed] [CrossRef] [Google Scholar]

[5] Kramer C, Rodrigues T, Canani L, Gross J, Azevedo M. Diabetic retinopathy predicts all-cause mortality and cardiovascular events in both type 1 and 2 diabetes: meta-analysis of observational studies. Diabetes Care. 2011;34(5):1238–44. doi: 10.2337/dc11-0079 [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[6] Brownrigg J, Hughes C, Burleigh D, Karthikesalingam A, Patterson B, Holt P, et al. Microvascular disease and risk of cardiovascular events among individuals with type 2 diabetes: a population-level cohort study. The Lancet Diabetes Endocrinol. 2016;4(7):588–97. doi: 10.1016/S2213-8587(16)30057-2 [PubMed] [CrossRef] [Google Scholar]