



COMPLETE BLOOD CELL

DETECTION AND CLASSIFICATION

Table of contents

01	PROBLEM OVERVIEW
02	OBJECTIVE
03	DATASET
04	CELLS OVERVIEW
05	FASTER R-CNN MODEL
06	PROCESS OVERVIEW
07	MODEL TRAINING
08	EVALUATION METRICS
09	MODEL PERFORMANCE
10	SUMMARY

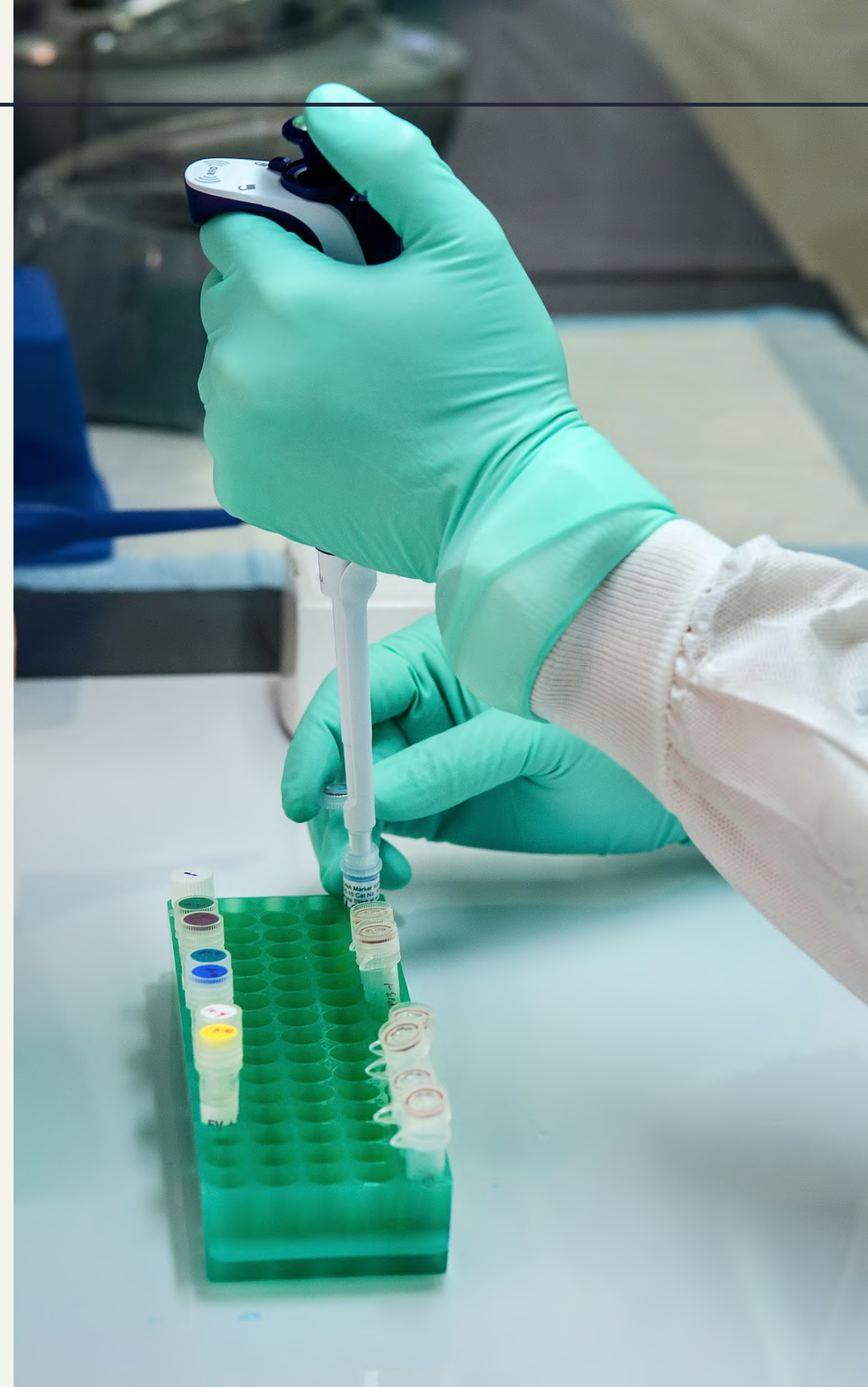
01

PROBLEM OVERVIEW

- Complete Blood Count, is a common blood test used to evaluate overall health and detect a wide range of medical conditions. The test involves drawing a sample of blood from a vein in the arm using a needle and syringe.
- CBC test measures various components of the blood including red blood cells, white blood cells, and platelets.
- Red blood cells carry oxygen throughout the body, and their number, size, and shape can provide important clues about health status.
- White blood cells are important for fighting infection, and their number and types can indicate whether there is an infection or inflammation present.
- Platelets play a crucial role in blood clotting, and abnormalities in their number or function can cause bleeding disorders or increase the risk of clotting disorders.
- Labs use automated hematology analyzer - quite expensive equipment that requires regular maintenance and renewal.
- Manual examination of blood sample smears, provided by lab technician is required to confirm or clarify the results of an automated blood cell count, and to provide additional information about the overall health status of the patient.

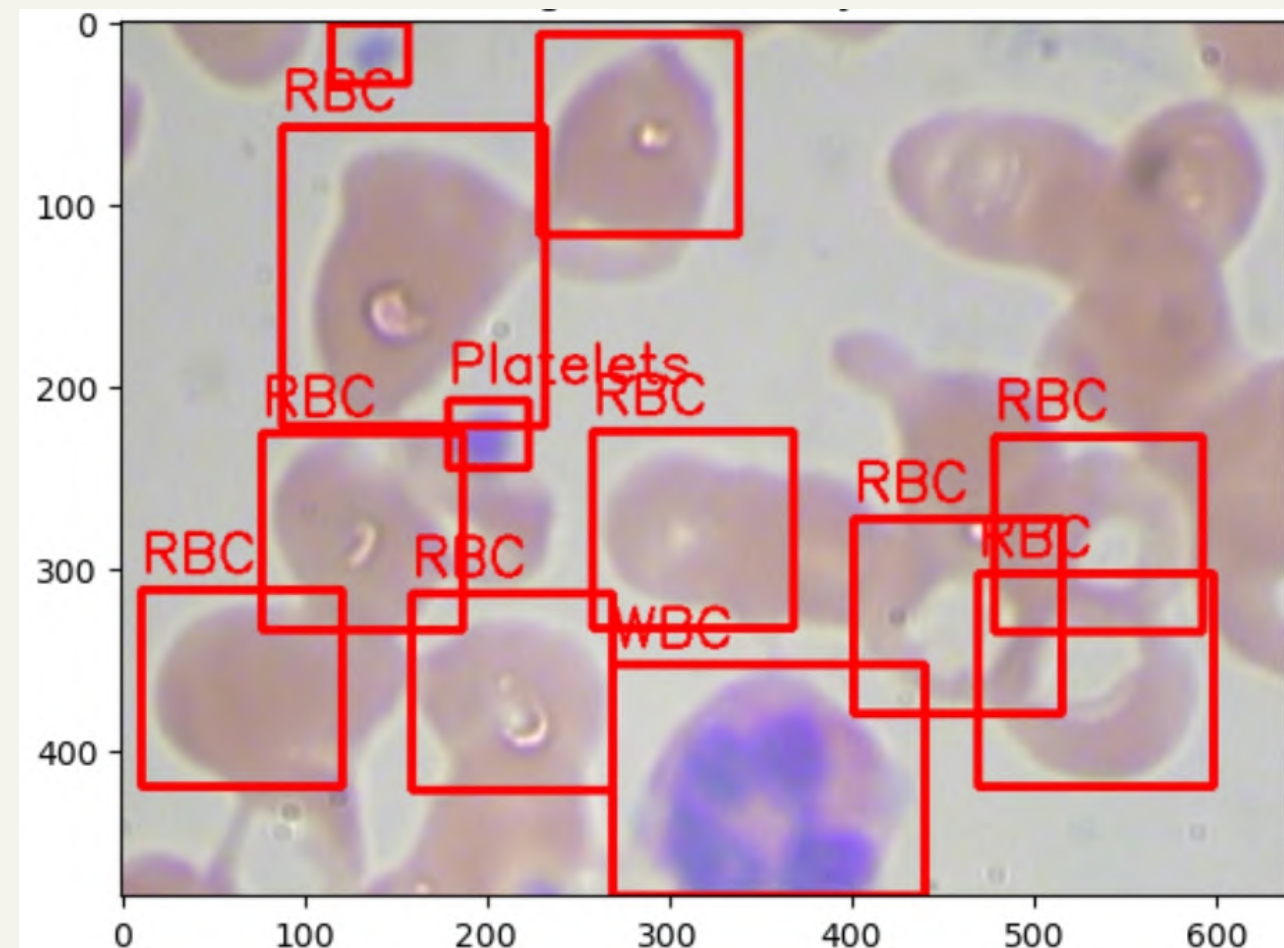


- To provide an alternative diagnostic tool that may be more cost-effective or accessible than automated hematology analyzers or manual examination in certain settings or for certain applications.
- To reduce the burden on laboratory technologists by automating certain aspects of blood cell analysis, freeing up their time for other important tasks.
- To improve the standardization and consistency of blood cell analysis across different laboratories and settings, potentially reducing variability and improving the quality of diagnostic results.

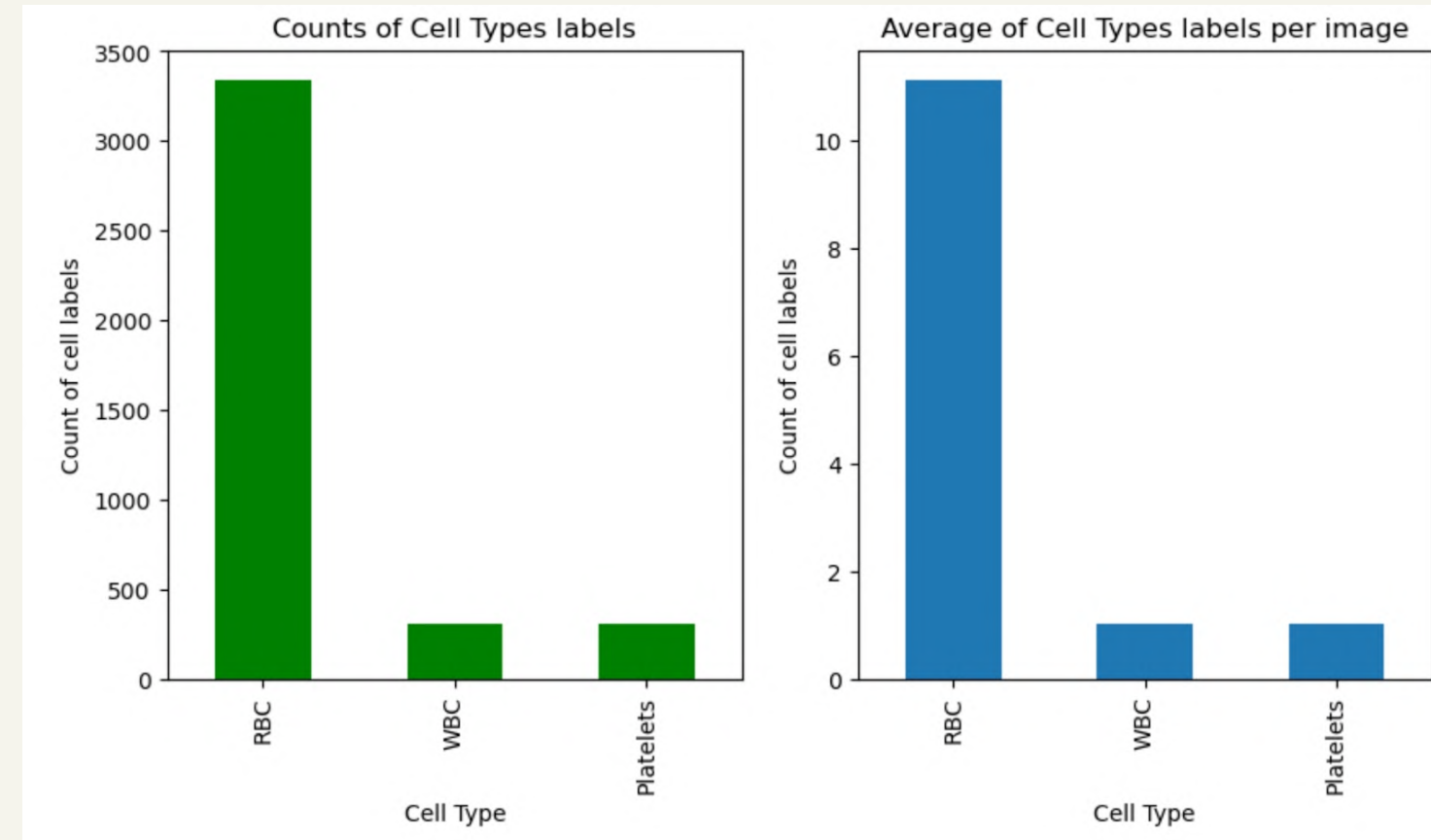


Current work provides object detection and classification for the dataset provided by MIT ([CBC](#)) which contains :

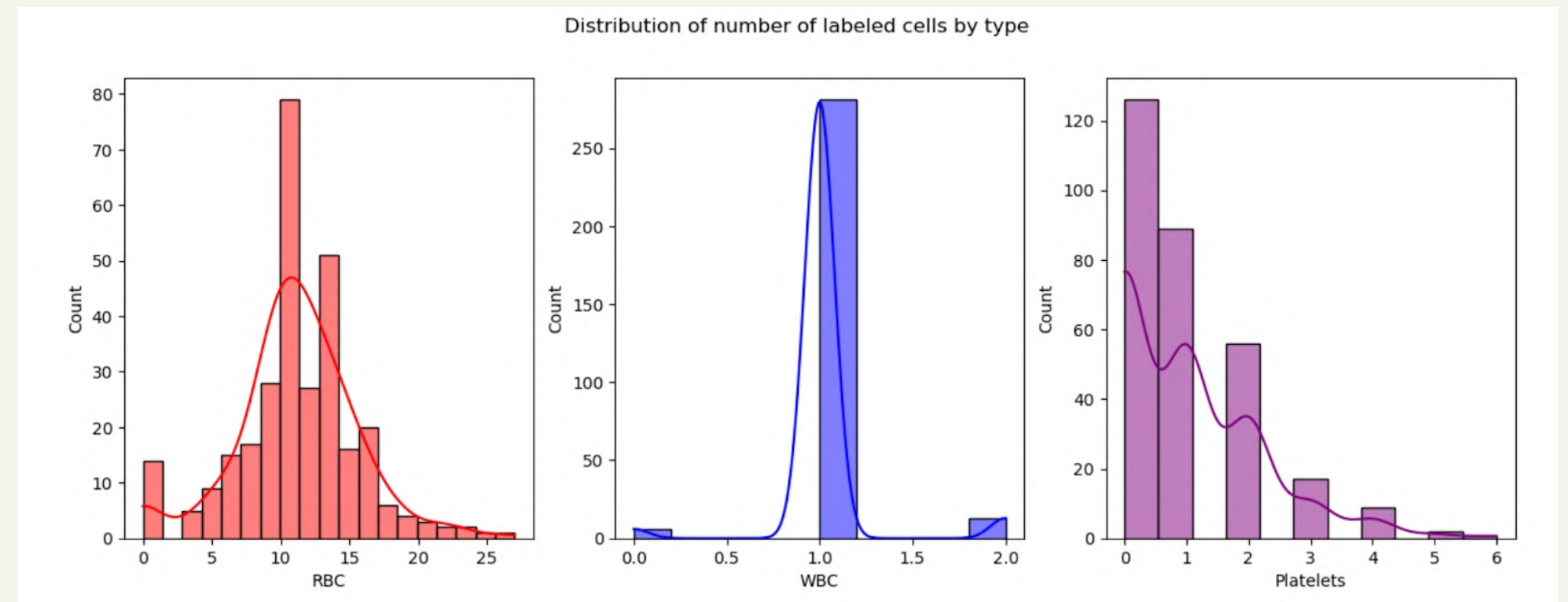
- 360 images in total, divided to 300 for training and 60 for testing. Size of all images is 640 x 480
- All images have labels for detected cells with bounding boxes coordinates and cell type name



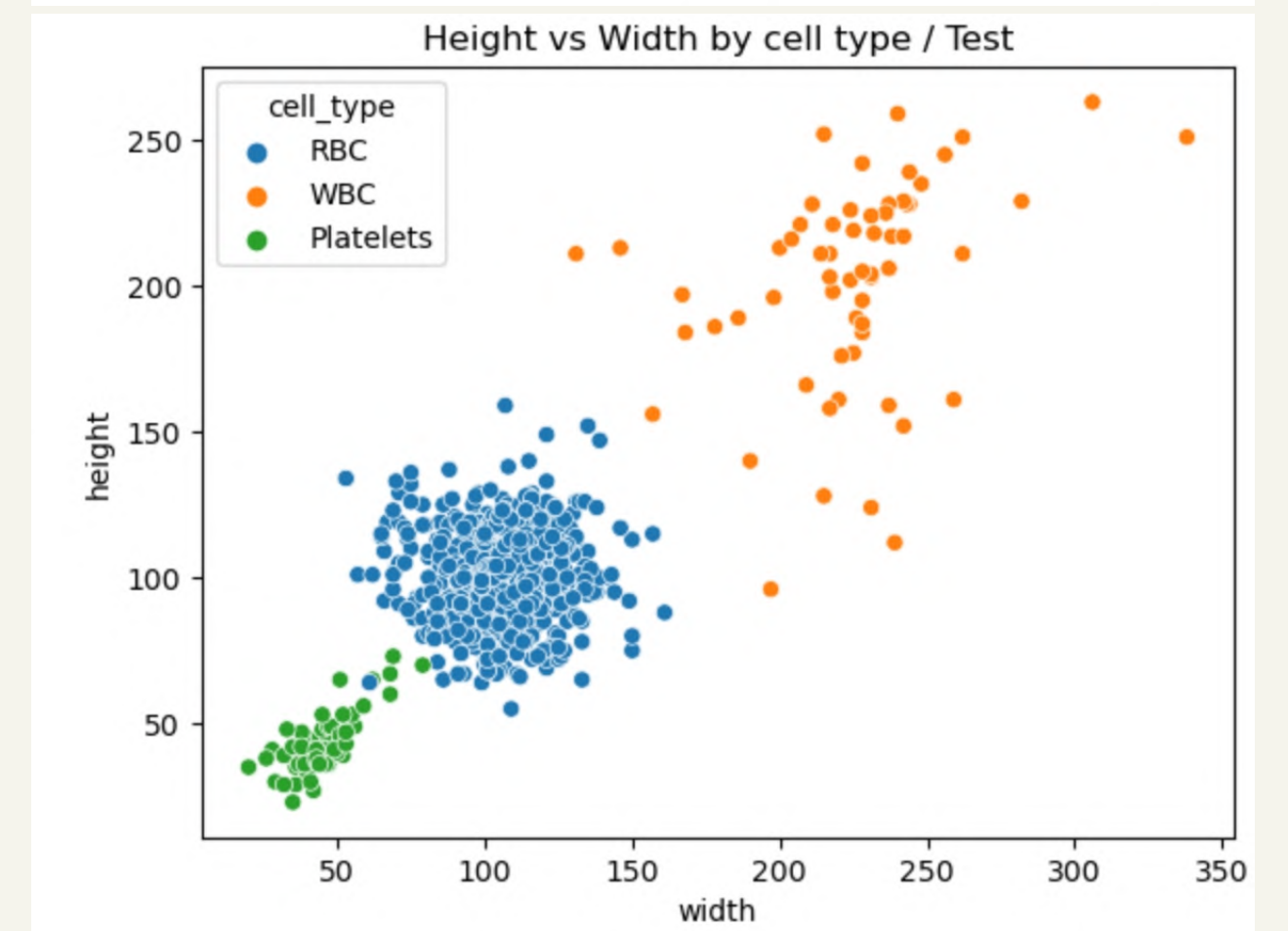
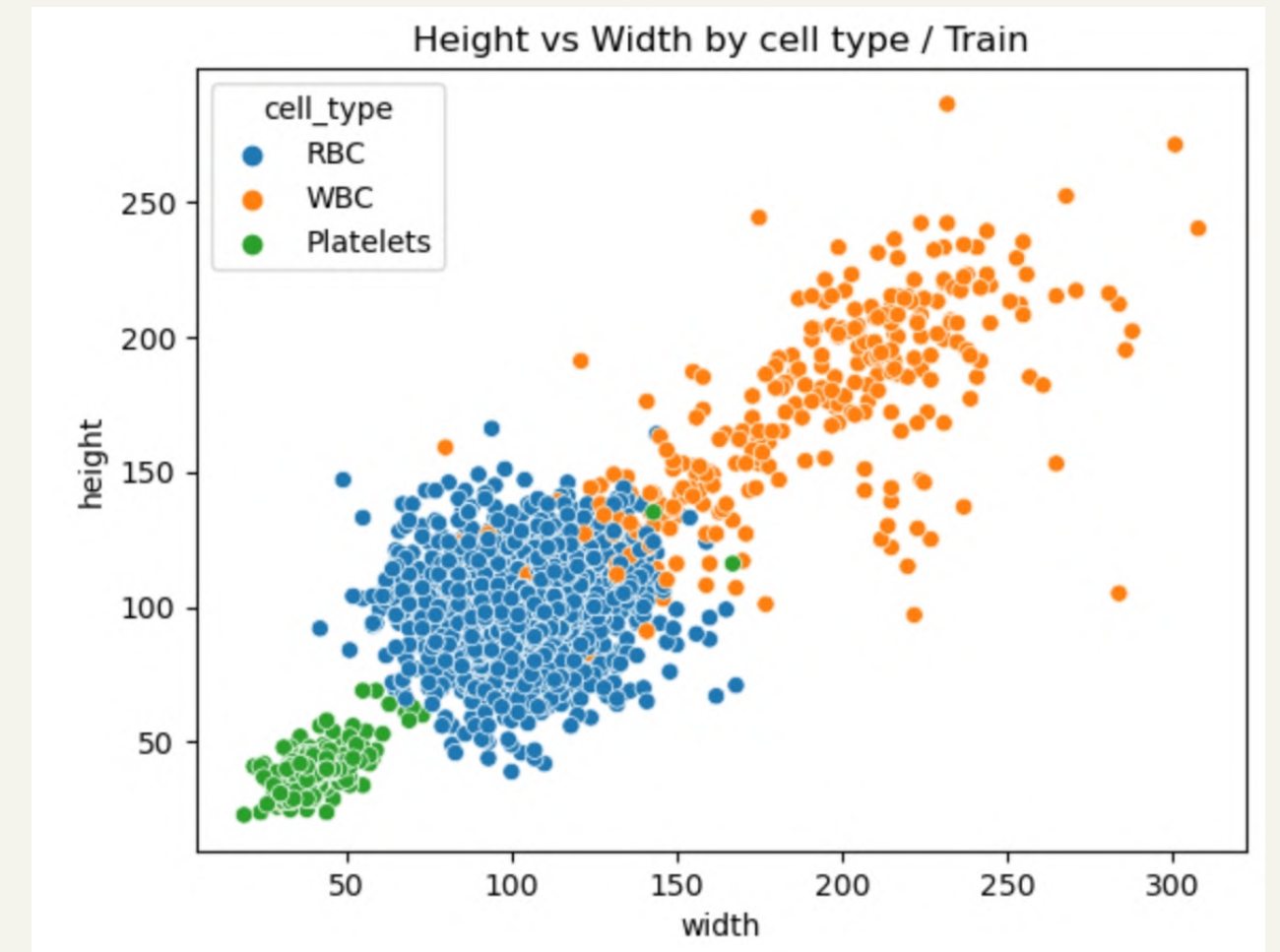
- Distribution of labels by cell types:



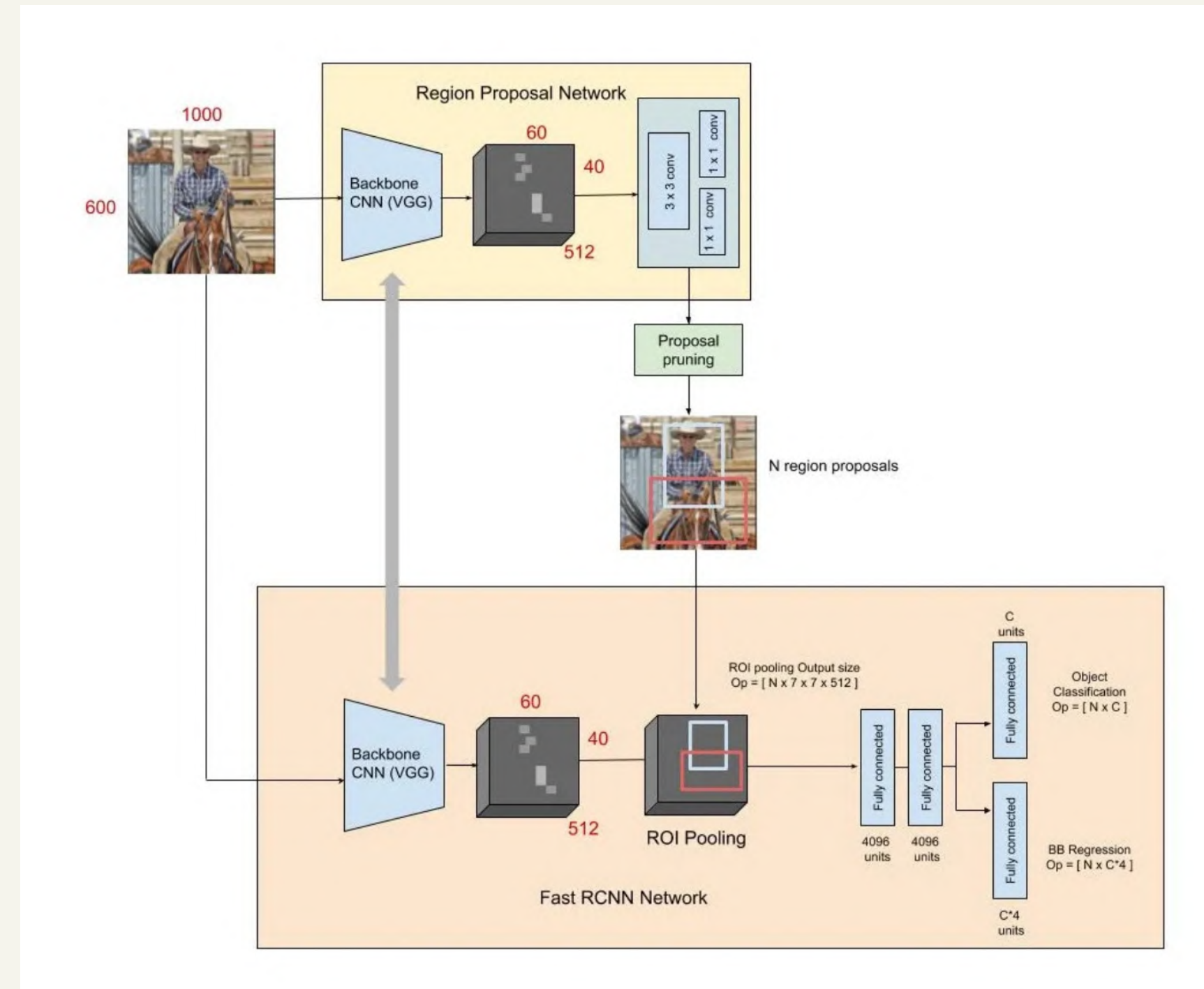
- Distribution of number of cell type per image:



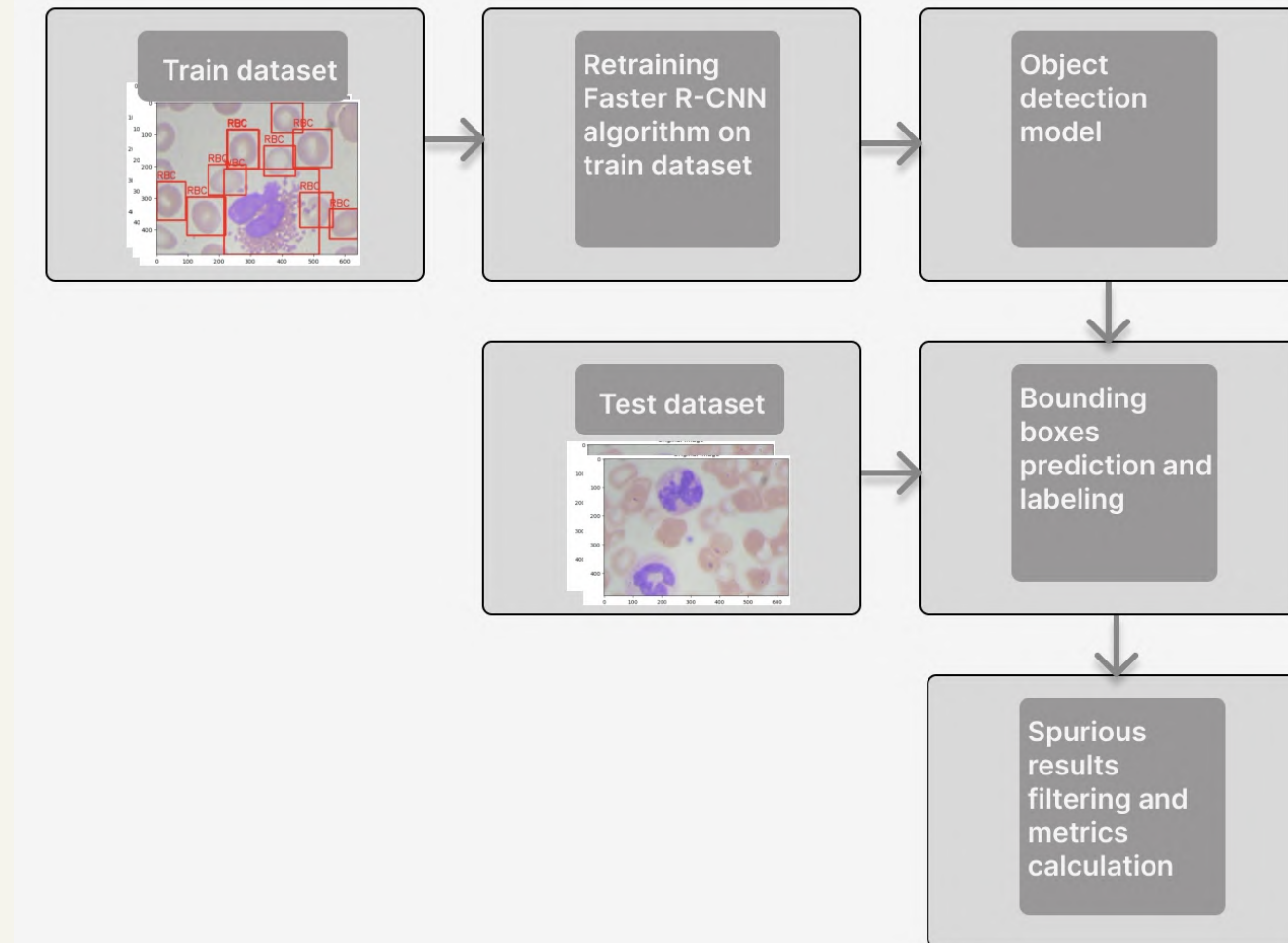
- Size of cells by type in Train set
- Size of cells by type in Test set



The Faster R-CNN architecture consists of two main components: a region proposal network (RPN) and a Fast R-CNN network. These components work together to generate region proposals and perform object detection and classification.



1. Dataset preparations. Wrapping in a custom PyTorch dataset class. This class should define how to load the images and annotations, apply any necessary transformations (in our case converting the image to a tensor ensures that the image data is normalized and scaled), and return the data as PyTorch tensors.
2. Pre-trained model will be used with ResNet50 backbone trained on COCO image dataset.
3. Pre-trained model should be fine-tuned on the dataset using the PyTorch training pipeline.
4. Save fine-tuned model for further usage with all trained parameters.
5. Run PyTorch evaluation step of the model on the prepared test dataset. The output is a list of dictionaries with PyTorch tensors for bounding boxes, labels and confidence scores for each of them.
6. Define metric for model evaluation and ways to eliminate wrong predictions. mAP will be the main metric we'd like to maximize. For results filtering we will use different levels of score thresholding depending on cell type and also we will use non-maximum suppression function to remove redundant detections of the same object.



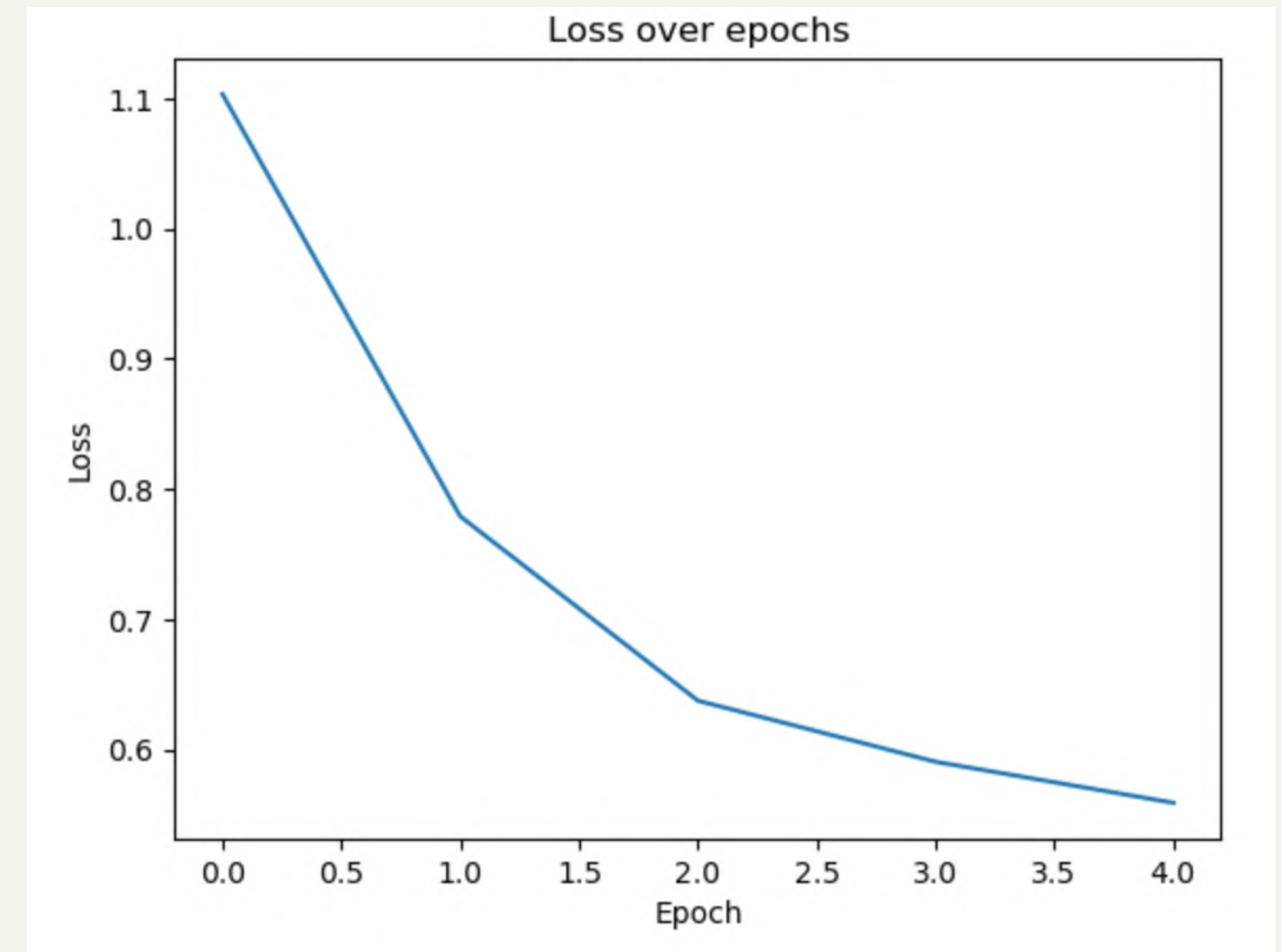
Model was trained over **5 epochs**.

The **batch size** was **4** images.

Total training time is **676.31 min**

Loss is a sum of 4 losses calculated by model:

- `loss_classifier`
- `loss_box_reg`
- `loss_objectness`
- `loss_rpn_box_reg`

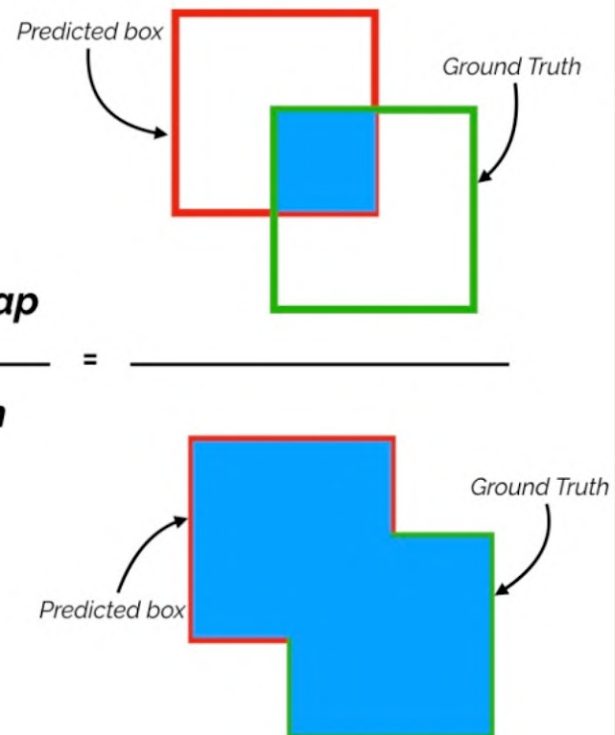


mAP - mean **average precision** was chosen as the main metric for the model performance analysis. In order to understand mAP, we need to first understand some of the key concepts that it builds upon:

- Confusion Matrix
- IoU
- $\text{Precision} = \text{True Positive} / (\text{True Positive} + \text{False Positive})$
- $\text{Recall} = \text{True Positive} / (\text{True Positive} + \text{False Negative})$

IoU threshold for mAP is set to **0.5**

		Actual Values	
		Positive	Negative
Predicted Values	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

$$\text{Intersection over Union (IoU)} = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$


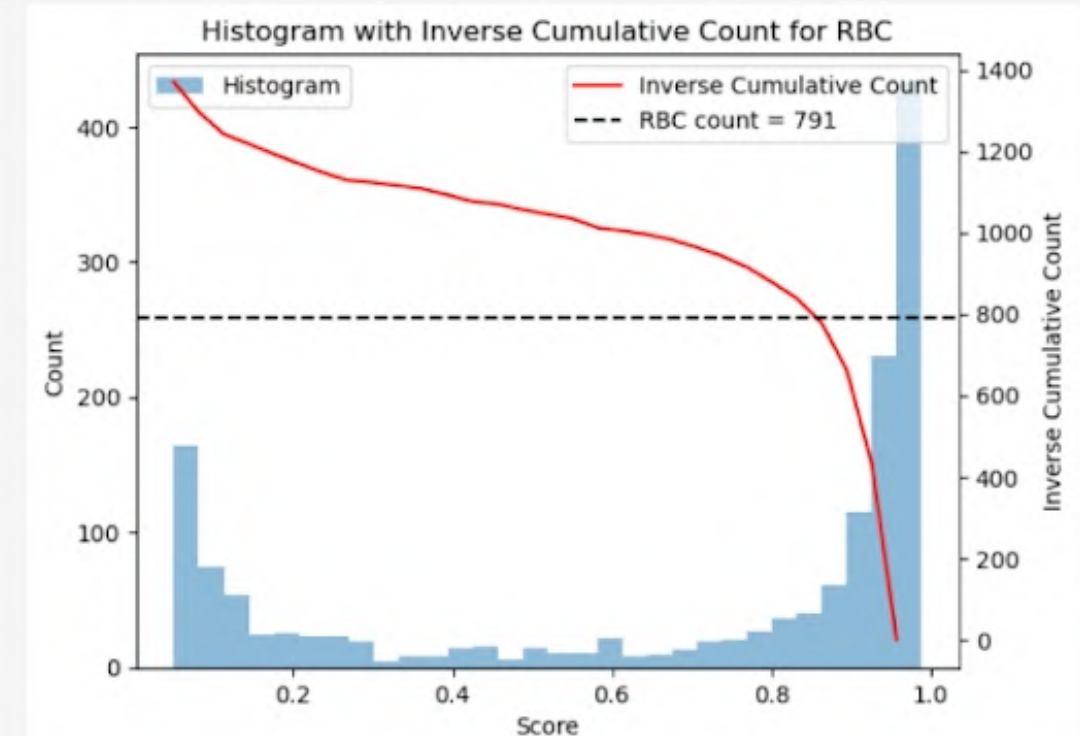
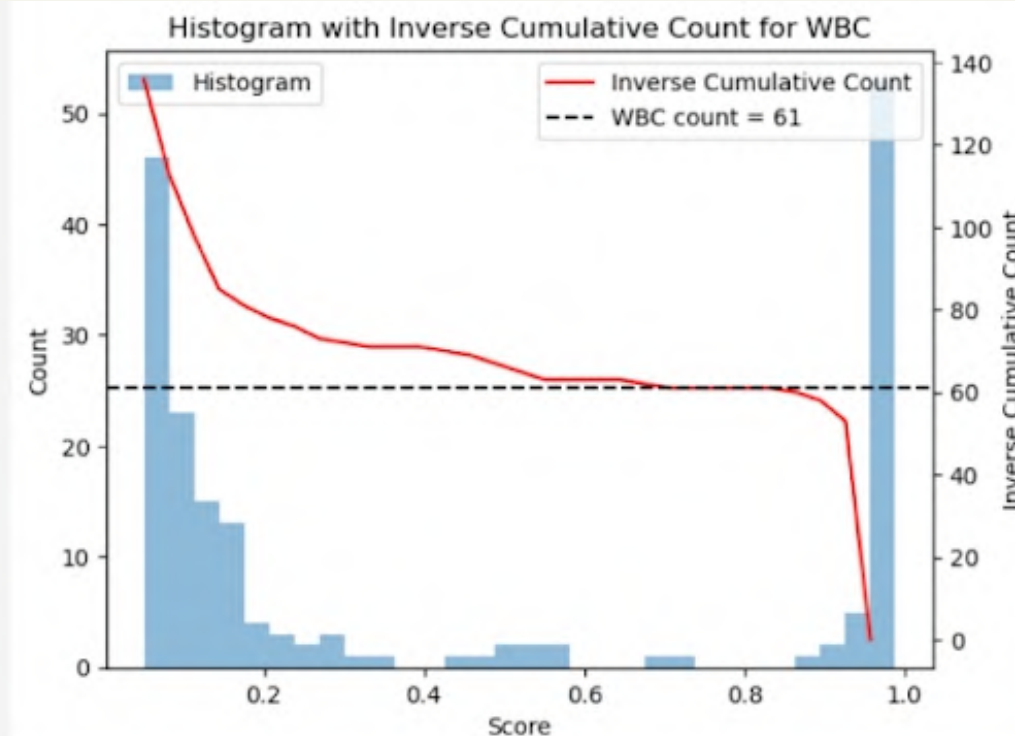
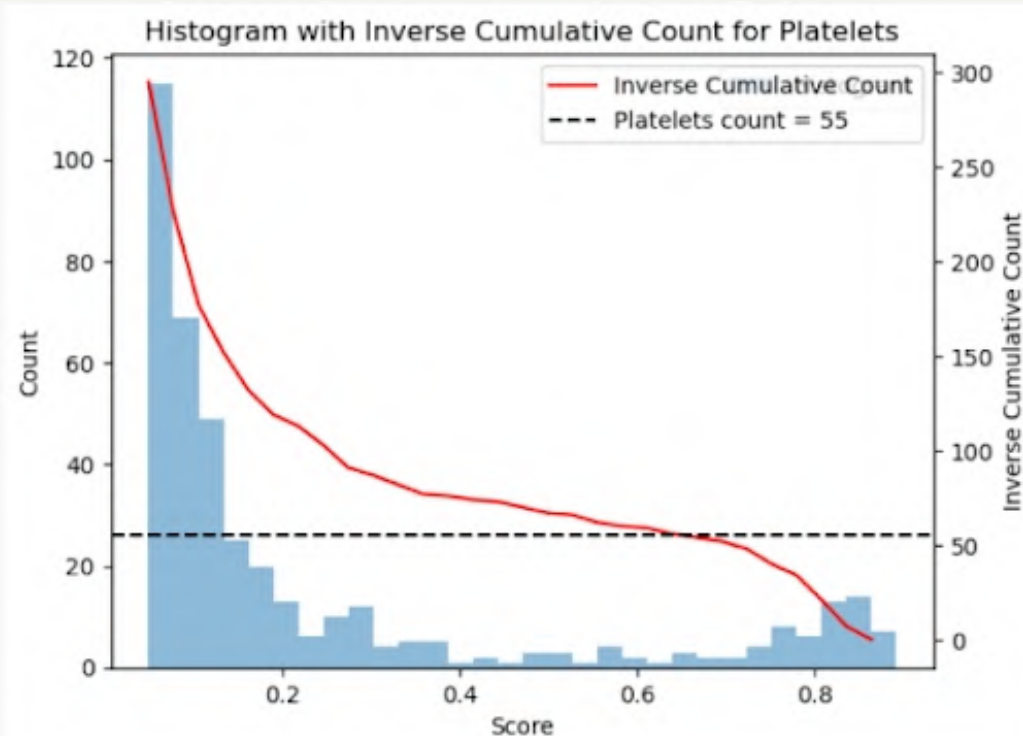
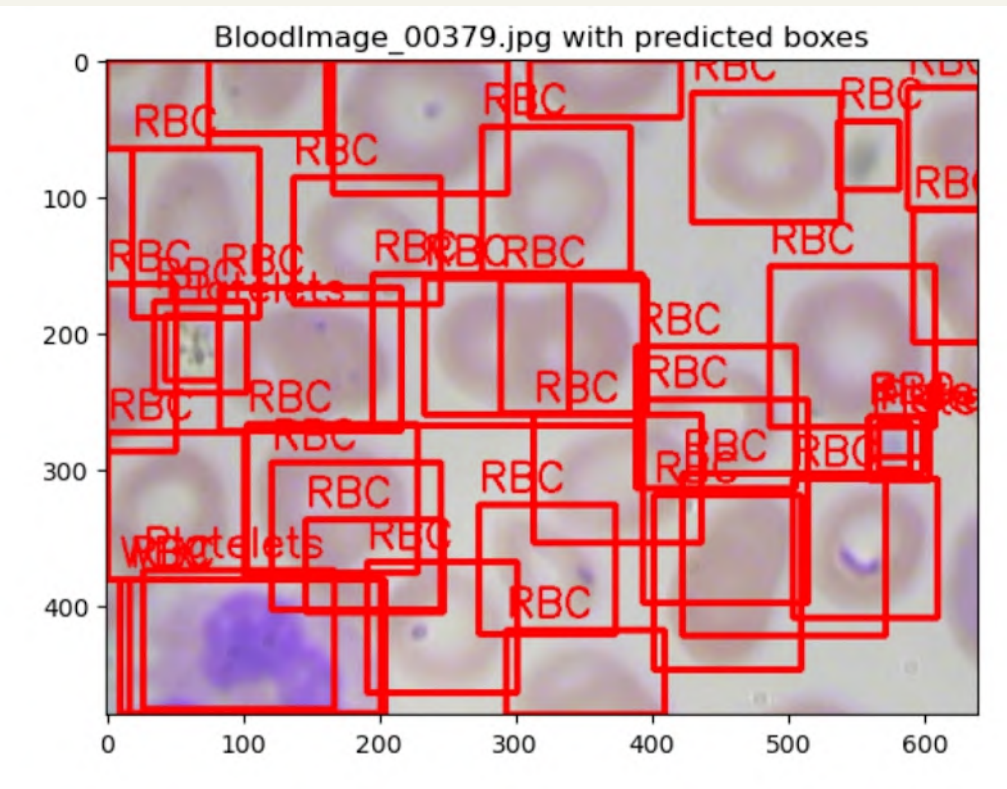
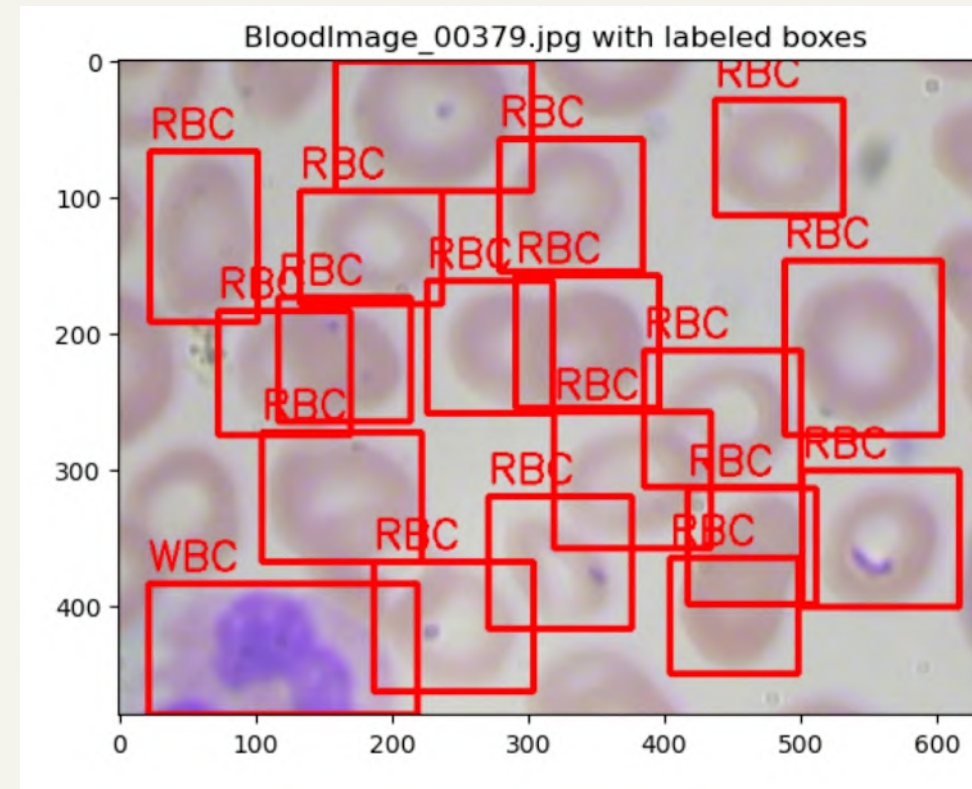
Unfiltered predictions

mAP=**0.35**

Average precision (AP) 'RBC' = **0.5**,

AP 'WBC' = **0.35**,

AP 'Platelets' = **0.2**



Filtered predictions

mAP=**0.84**

AP 'RBC' = **0.7**,

AP 'WBC' = **1.0**,

AP 'Platelets' = **0.78**

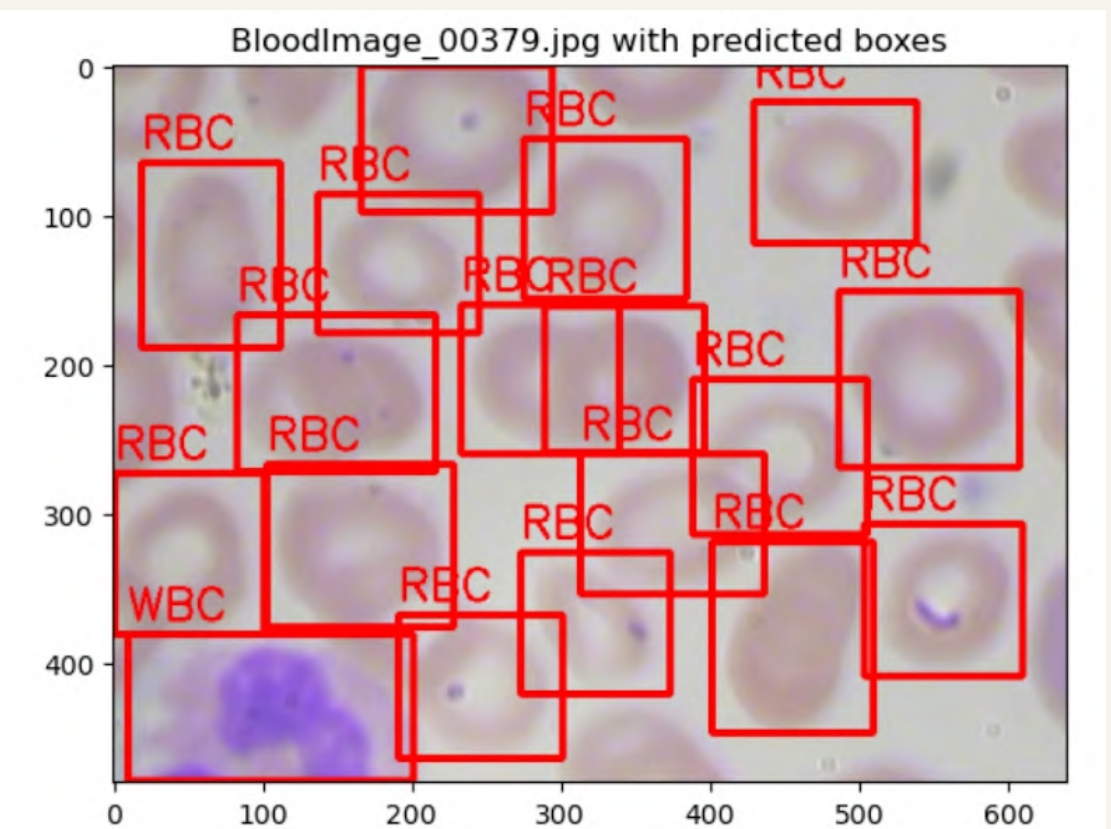
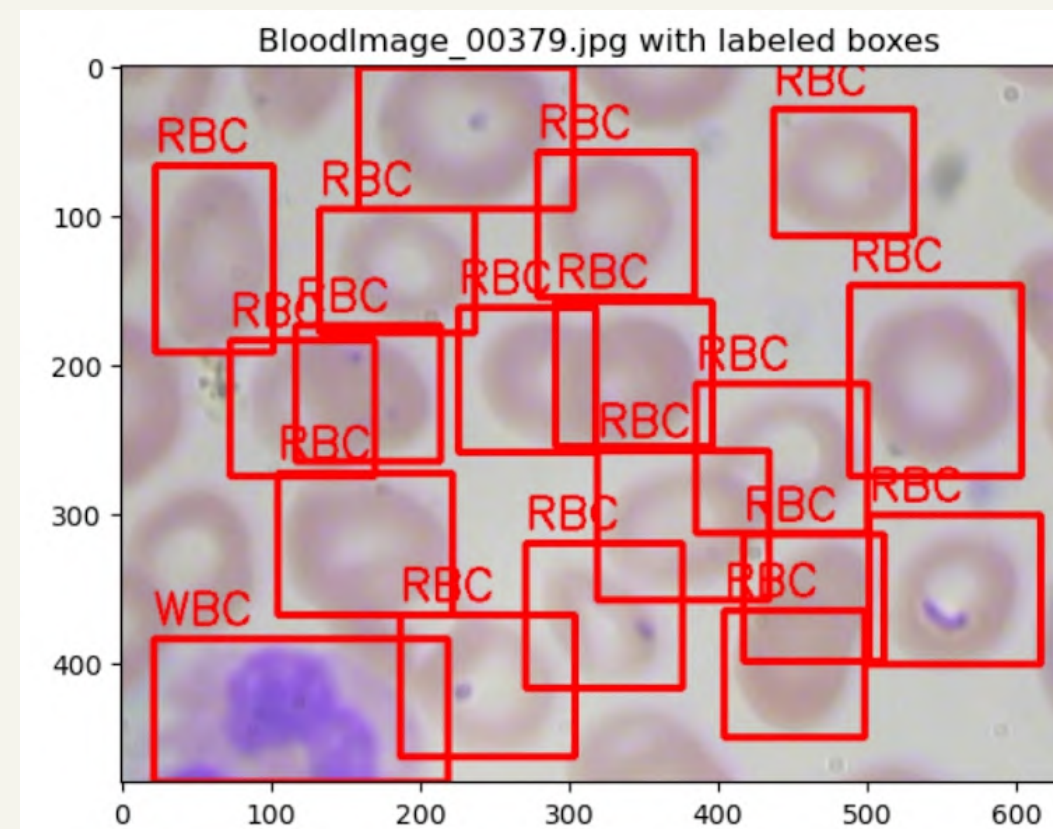
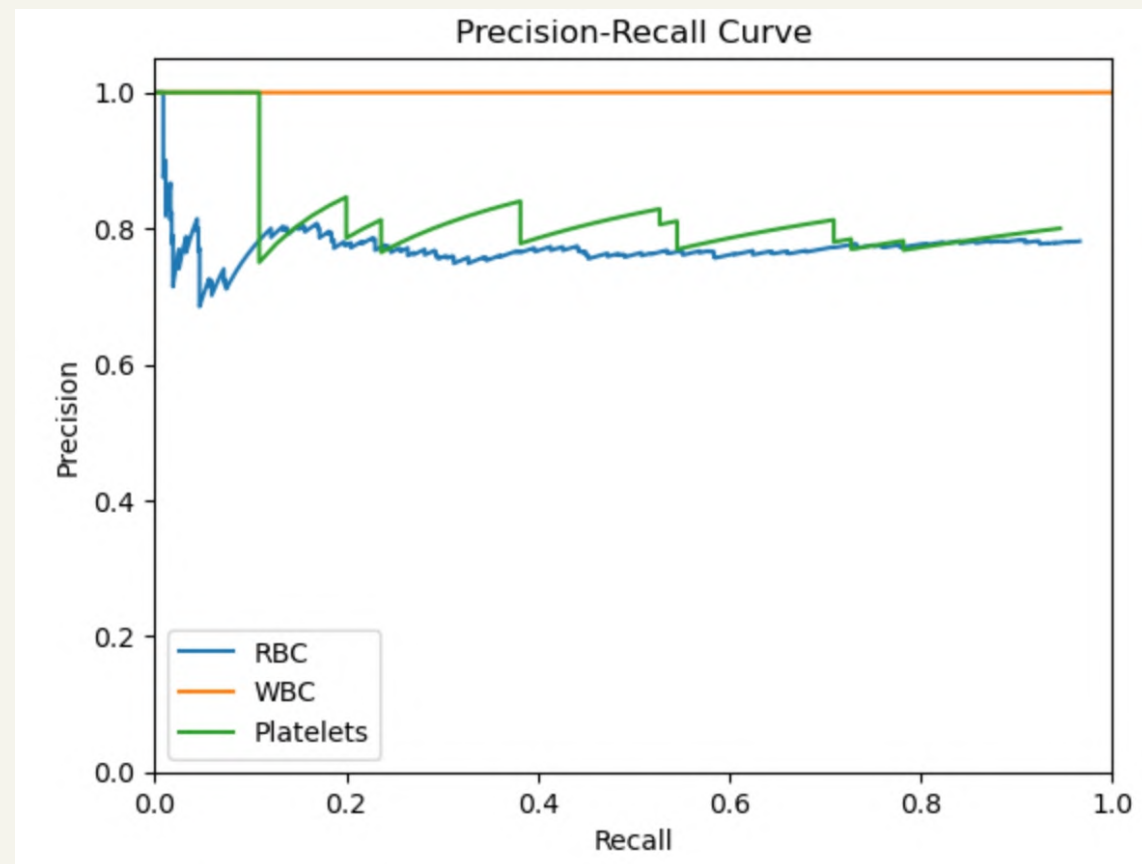
Thresholds to filter predictions by score:

Platelets: **0.55**

RBC: **0.70**

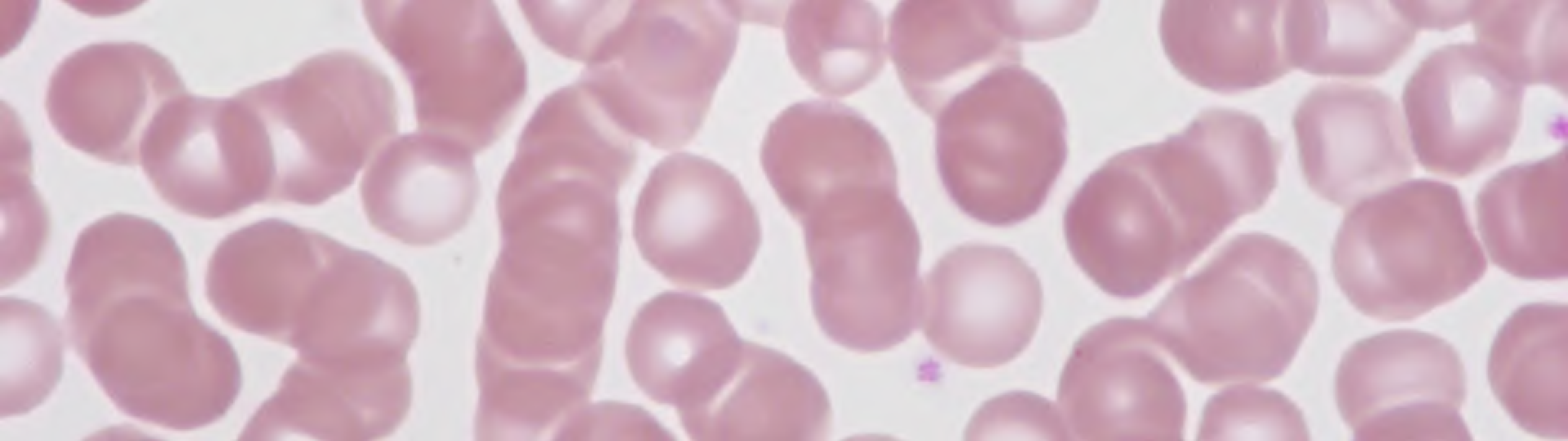
WBC: **0.80**

Non-maximum suppression threshold: 0.45



The Faster R-CNN model was trained on a dataset of 300 blood smear images, and the model achieved an **mAP** of **0.84** on a test set. This result suggests that the model is performing reasonably well in detecting blood cells and their classification in the images.

- The model has an AP of **1.0** for the '**WBC**' class, indicating high accuracy in detecting white blood cells
- The AP for the '**RBC**' class is **0.7**, suggesting the model is less accurate at detecting red blood cells, which may be due to overlapping cells and different IoU thresholding.
- The AP for the '**Platelets**' class is **0.78**, indicating reasonable accuracy in detecting platelets, but prediction filtering by thresholding may be necessary.
- To improve the model's performance, increasing the size of the training dataset, training for more epochs, and applying data augmentation techniques such as rotation and flipping can be used.
- Analyzing false positives and false negatives for RBC and Platelets can help identify areas where the model needs further improvement
- There is still room for improvement in the model's performance, but it shows promise in detecting blood cells accurately



THANK

YOU