Analysing YOLO variants and State-of-the-art Methods for Cell Detection and Counting Challenges

Johanna Jones

May 2024

Abstract Cell counting and detection are important tasks in the areas of medicine and aid pathologists in understanding cell behaviour in broader contexts like treatment planning and disease diagnoses. Manual methods of counting and annotations are time consuming, costly and inaccurate, which has lead to the wider adoption of Artificial Intelligence and Deep Learning techniques. Challenges like overlapped cells, weak staining and diverse cell morphologies present learning and generalising challenges for these algorithms. In this study we analyse the performance of three YOLO variants on a novel dataset as assess whether they adequately address these challenges. We also compare the top performing YOLO model to a state-of-the-art method and determine the definitive characteristics of its performance. Through a qualitative approach, the study aims to provide insights and recommendations for improving cell detection in biomedical imaging.

1 Introduction

Cell counting and detection are important and fundamental tasks in the areas of biology and medicine [Huang et al., 2020]. These tasks are done to understand cell behaviour and proliferation within contexts such as disease diagnosis and treatment planning. Cell counting refers to the number of a type of cell present within a histopathological image whereas cell detection refers to the identifying the presence of a specific cell. Traditionally, pathologists were manually counting and labelling images or, using low level machines such as Hemocytometers [Drałus et al., 2021]. However, these practices are time consuming, costly, and are highly inaccurate which in a cancer diagnosis context can have significant ramifications. Therefore, pathologists have widely adopted Deep learning (DL) techniques to count and detect cells in place of these manual methods.Despite the innovations of current DL methods, the very nature of these histopathological images presents several challenges for these algorithms to learn and make detections accurately [Drałus et al., 2021]. Overlapped and clustered cells make it difficult for algorithms to properly segment the image and distinguish cell from cell. Weakly stained cells create difficulties for the model to distinguish artefact noise and the image background from the target cells [Hagos et al., 2019]. Diverse cell morphologies prevent the algorithm from generalising more broadly and making predictions[Yellin et al., 2018].

Advanced iterations of the Convolutional Neural Networks (CNNs) and regression-based methods are already being implemented for cell detection. However, these state-of-the-art methods are often dataset-domain and do not a provide a general solution. We seek to understand how a general object detector YOLO (You Only Look Once), which has achieved state-of-the-art performance in non-biomedical imaging domains, can address the challenges mentioned above within a biomedical one. Additionally, we seek to analyse how CONCORDE-Net [Hagos et al., 2019] compares to the different YOLO variants in terms of their architectures and performance.

YOLO is a popular object detection algorithm that is applied several contexts such as traffic and crowd monitoring, security, and now medicine. Existing studies relating to YOLO detecting on histopathological images revolve around assessing different model's speed and accuracy (see Table 1). [Nair et al., 2021] deployed a YOLOv4 model for mitotic nuclei detection in breast cancer histopathological images. While a computationally efficient and reasonably fast, the study suffered in model performance when given pixel and colour normalised images and was deemed unsuitable for full scale deployment. In another study, [Topuz et al., 2023] deployed YOLO models V3, V5, V7 and V8 to compare how accurately each could detect mitosis cells as an indicator for lung cancer. Models V7 and V8 outperformed others in correctly detecting mitosis cells despite the data imbalance and low-quality input images. V7 and V8 exhibited higher performance due to a combination of higher image resolution processing, improved module innovations and new loss functions (see table 2).

Others have performed ablation studies to determine which components of YOLO's architecture yield the best detection results (table 1). Alam and Islam [Alam and Islam, 2019] deployed a Tiny YOLO model to count and detect blood cells by replacing the primary backbone of the YOLO model with other CNN architectures like VGG-16, ResNet50 and Mobile Net. Results showed that the standalone Tiny YOLO drastically outperformed the other CNN architectures and generalised well to unseen data sets. Yucel [Yücel et al., 2023] compared the performance of a YOLOv5 (baseline) to a YOLOv5 with a transformer mechanism (transformer) to detect mitosis cells in neuroendocrine tumours, with the aim of creating a high speed and accurate detector. Results showed that the transformer consistently detected more mitosis cells than the baseline and image augmentation techniques also boosted the transformer model's speed and accuracy.

While these YOLO models have been chosen for their accuracy and speed, they do not explicitly address the challenges of overlapped cells, diverse cell morphologies or weakly stained cells. In this study, we have deployed and analysed how three YOLO variants; V5, V8 and V9 perform in detecting different classes of blood cells. Through analysis of their performance metrics and architectural components, we have developed a deeper understanding of single shot detectors, their limitations and the requirements pathologists must consider before implementing YOLO. We have also analysed what components of

Study - Model	**Dataset**	**Approach**	**Result**	**Key takeaways**
Topuz (2023) V3,V5, V7,V8	MIDOG 2022 Cancer Cells	Compared performance of different YOLO models	V8 and V7 performed best	Models should be generalisable since mitotic cancer cells look similar across diferent cancer types
Yucel (2023) V5	Mitotic Endocrine Cancer Cells	Deployed V5 and V5-Transformer to detect mitotic cells	V5-Transformer outperformed V5	Attention mechanism discovered more features of mitosis cell images in weaker stained cells
Nair (2021) V4	Breast Cancer Cells	Deployed V4 to detect mitotic cells	Lower performance but deployable.	Needed balanced and larger dataset
Alam and Islam (2019) Tiny YOLO	Blood Cells	Modified architecture of backbones	Tiny YOLO outperformed other backbones.	Certain architectures have better detection capabilities for specific objects

Table 1: *List of related methods where YOLO has been deployed, the dataset used, approach taken, key results and takeaways.Most studies are concerned with detection performance through ablation studies. None specifically address the challenges of cell detection.*

YOLO Model	Backbone	Features	Key improvements
V1		Trained on the PASCAL VOC dataset (20 Object categories)	
V2	Darknet-19	Detects range of object sizes and ratios Increased stability and accuracy through batch normalisation Multi-scale training	Anchor Boxes Batch Normalisation New Loss function
V3	Darknet-53	Detects range of object sizes and ratios Pyramid feature maps detec smaller object at different scales.	Scaled Anchor boxes Feature Pyramid Networks (FPN)
V4	CSPNet	Shallow backbone structure	Scaled Anchor boxes using K-means Clustering GHM loss improved FPNs from V3 Spatial Pyramid Pooling layers (SPP)
V5	EffcientDet	Trained on D5 dataset (600 object categories) Achieves small object detection at multiple scales Loss improves performance on unbalanced datasets	Dynamic Anchor Boxes Improved architecture to SPP CIOU loss
V6	EffcientNet-L2	Fewer parameters and higher computational efficiency than V5	Scaled Anchor boxes
V7	E-ELAN	Can detect objects of different shapes Higher resolution for image processing	Predefined anchorboxes with different aspect ratios Focal Loss
V8	CSPDarknet53	Increased detection speed and performance Achieves real time detection	Anchor box free detection head GPU requirements Spatial Attention mechanism Feature fusion Bottleneck and SPPF Distribution Focal Loss
V9	GELAN	Combats information loss which improves classification and accuracy performance	Programmable Gradient Information Reversible functions

Table 2: *Summary table of all YOLO versions at present* [Kundu, 2023, Torres and Austen, 2024]

the state-of-the-art methods are fundamentally different or similar to YOLO in order to determine how YOLO can be scaled and deployed in more complex and diverse scenarios. From this study, pathologists and researchers will be provided with the advantages and limitations of YOLO to aid their diagnostic and prognostic cell detection and counting studies. Pathologists will also benefit from knowing how different DL techniques perform on given datasets, taking the considerations and recommendations from this paper and applying it to their own scenarios.The proposed method was evaluated on the publicly available Blood-Cell-Detection [MrAnayDongre, 2021] dataset consisting of red blood, white blood cells and platelets. The results demonstrate that YOLOv8 and YOLOv9 are formidable detectors and have the potential to be used in biomedical imaging contexts on higher complexity datasets in combination with other statistical and DL techniques.

2 Methodology

For a full breakdown of all tasks and steps refer fig. 10 in Appendix A.1.

2.1 About the data

In this study, experiments were conducted the Blood-Cell-Detection dataset. The Blood-Cell-Detection dataset is a small scale, publicly available dataset of manually annotated blood cell images. The dataset has been commonly used to assess model performance [Alam and Islam, 2019]and has been sourced from [MrAnayDongre, 2021]'s GitHub repository. In total there are 874 images and annotation files, with pixel dimensions 416 x 416 and contains three object classes: Red Blood Cells (RBCs), White Blood Cells (WBCs) and Platelets. The annotation files contain the labelled classes for each cell object, their bounding box x-y coordinates and width-height dimensions. Figure 1 shows these ground truth labels for each object class and the nature of the images. The images have already been split into training (765 images), validation (73 images) and test (36 images) sets. In the human body, RBCs are the most populous and common cell type making up to 40-45% of all blood cells, this is reflected in fig. 2 and table 3 where the RBCs in the dataset drastically outnumber the other two.

	Train	**Validation**	**Test**
RBC	8814	819	398
WBC	789	72	37
Platelets	739	76	36

Table 3: *Number object instances that have been annotated per training set*

On the other hand, WBCs are larger cells and make up only 1% of blood cells. Platelets also make up a large percentage of blood cells, but this is not reflected in our dataset, leading to an imbalance across the object classes. In figure 3 it can be observed that most images have between 9 and 19 objects annotated across the training and validation

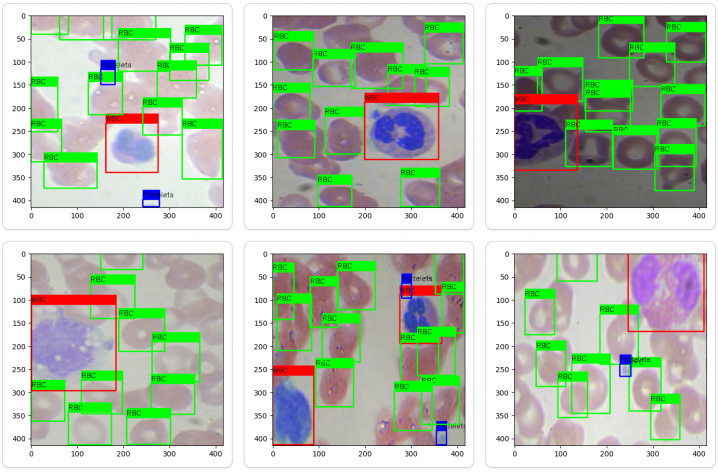

Figure 1: *Ground truth also known as the labelled data for RBCs (green), WBCs (red) and platelets (blue). Each object is surrounded by a bounding box.*

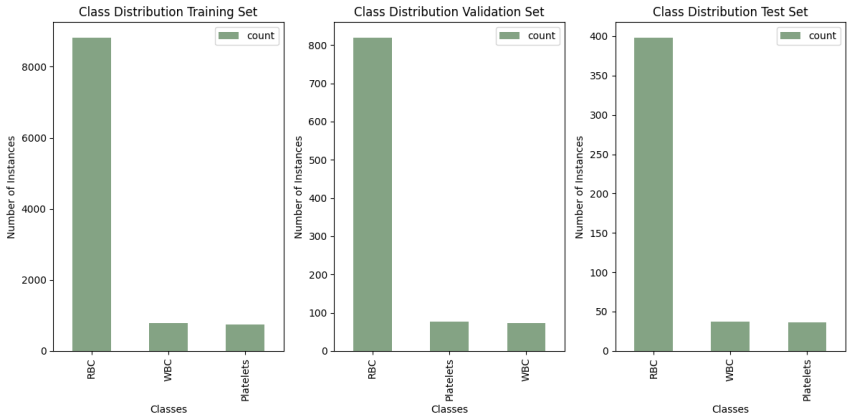

Figure 2: *Bar chart showing the object class distribution of RBCs, WBCs and Platelets per training, validation, and test sets. Each bar graph shows a clear class imbalance between the three classes*

sets. Within the test set, majority of images have between 10 and 15 objects labelled. It is worth noting, that not all cells within an image have ground truth labels assigned to them. Often it is the overlapped blood cells or cells that are cut off that are unlabelled leaving us with an incomplete dataset (see fig. 1). To introduce diversity into the data set, training and validation images have been rotated, flipped, and cropped by the owner of the GitHub repository. Since our data set exhibits all three challenges of overlapped cells,

weak staining, and diverse cell morphologies, it is of great interest to see to what extent each YOLO models V5, V8 and V9 will address these challenges.

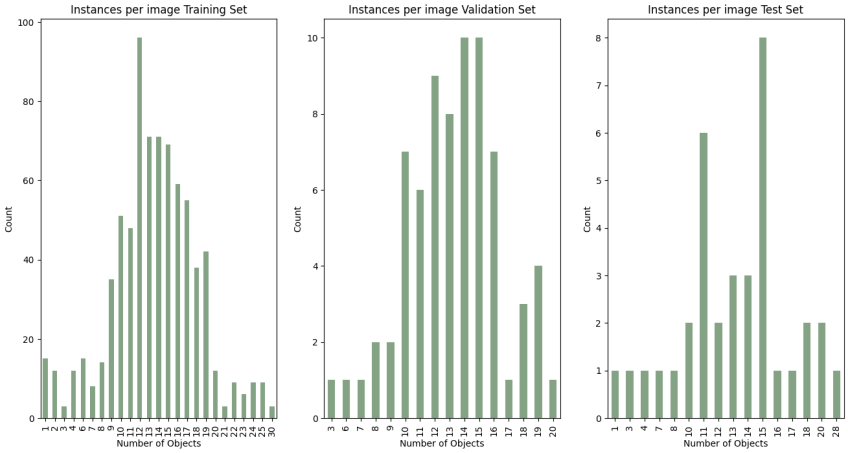

Figure 3: *The number of annotated objects by image across training, validation, and test sets.*

2.2 Implementation

The method for this study consists of two distinct components; the first involves executing three deep learning YOLO models and assessing their performance on our data set. The second component consists of conducting a qualitative study of one bespoke state-of-the-art method for cell detection in the existing literature. We have implemented YOLOv5, YOLOv8 and YOLOv9, all three of which belong to the Ultralytics open-source platform.

2.2.1 Ultralytics

Ultralytics [Ultralytics, 2024] is an open-source platform that builds and deploys several YOLO models for tasks such as segmentation, classification and pose estimation. The platform supports model versions three through to nine as well as other state-of-the-art object detection, segmentation, and transformer models.

2.2.2 YOLO

YOLO (You Only Look Once) is a state-of-the-art object detection algorithm that leverages CNNs, to detect objects in real time. It is a single shot detector that passes an image through the CNN once, producing classifications and bounding boxes of the objects within that image. YOLO has demonstrated a high level of generalisability and versatility in a variety of domains such as security, crowd and traffic monitoring, often outperforming Region-based CNNs and Single Shot detectors. [Qureshi et al., 2023]. First built in 2015, the framework has undergone continuous modifications and improvements

to achieve better performance, with V8 and V9 being the latest releases from Ultralytics and [Wang et al., 2023] in 2023 and 2024 respectively. In general, there are three main components to YOLO's architecture represented by figure 4 the backbone, neck and head.

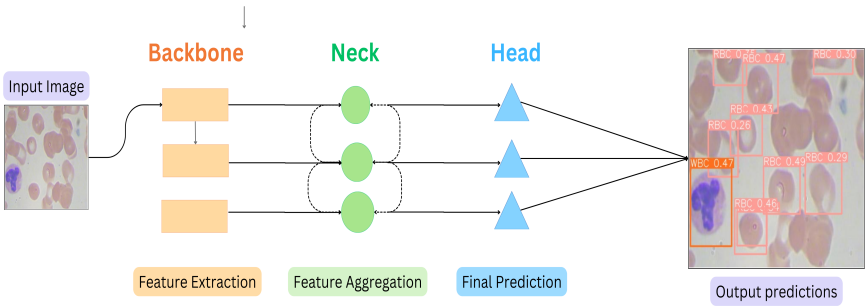

Figure 4: *The general architecture for the YOLO framework consisting of the backbone, neck and head.*

Backbone: The backbone is responsible for extracting valuable characteristics and then generating feature maps from the input image. A CNN that has been extensively pre-trained on large image datasets is commonly used for the backbone. Popular choices for backbones include VGG16, ResNet50, CSP-Darknet53 and EfficientRep [Qureshi et al., 2023].

Neck: The neck is made up of two components; the Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PAN) [Qureshi et al., 2023]. Both components work to combine the extracted feature maps generated by the backbone through a process of feature aggregation and propagate them to the head[Qureshi et al., 2023].

Head: The head handles the aggregated features from the neck and makes predictions on the image relating to the bounding boxes coordinate estimates, classification scores class probabilities [Qureshi et al., 2023, Yücel et al., 2023]. Often, studies have tried different combinations of backbones, necks, and heads to investigate model speed, accuracy, and detection performance as in the case of [Alam and Islam, 2019]. YOLO heads are also interchangeable across different YOLO models. For instance, a YOLOv3 head was chosen as the head of a YOLOv4 model used to detect mitotic nuclei in breast histopathological images [Nair et al., 2021] see table 1.

2.2.3 YOLO Variants

In this study we deploy and train three YOLO models; V5, V8 and V9. The following section goes into greater depth of the key components, artchitectural differences of these three frameworks. Please refer to table 2 for a list of all YOLO models.

YOLOv5: YOLOv5 introduces some key innovations building on top of V4. A key component of V5's backbone leverages a Cross Stage Partial Network (CSP-Net) to extract target features from the input images. The EfficientDet backbone allows V5 to achieve better generalisation and higher accuracy on small and unbalanced datasets [Kundu, 2023]. This is due to the introduction of key improvements to the SPP modules within the neck that reduces the resolution of feature maps [Kundu, 2023]. Within the head, dynamic anchor boxes are implemented that use a clustering algorithm to align the bounding boxes with object's size and shape [Kundu, 2023].

YOLOv8: YOLOv8 comes in different computational sizes, suitable for a range of different tasks and scenarios. Its backbone uses the same backbone as YOLOv5; a CSP-DarkNet53 CNN to extract the target features from the input images. The CSP module has improved the information flow between the CNN's layers boosting accuracy and gradient flow during the training and back-propagation process [Torres and Austen, 2024, Qiao et al., 2023]. This improvement can be attributed to replacing the C3 blocks found within YOLOv5 with C2f blocks. Like V5, V8's neck consists of a Path Aggregation Network (PANet) which works to capture features at multiple scales. Within the neck a key innovation of V8 is its feature fusion and SPPF layers (Spatial Pyramid Pooling Fast) where it combines high level features with lower-level spatial information improving the detection accuracy for smaller objects [Torres and Austen, 2024]. V8's head is different to its predecessors in having an anchor box free detection head, eliminating the need for predefined anchor boxes. Within the head, the IOU loss function generates bounding boxes that can handle overlapped objects with higher accuracy [Torres and Austen, 2024].

YOLOv9: YOLOv9 introduces three key innovations as it builds on the models that have come before it; reversible functions, Generalised Efficient Layer Aggregation Network (GELAN) and Programmable Gradient Information (PGI) [Wang and Liao, 2024]. As an image undergoes transformations by the deeper layers of a CNN, there is a loss of information that occurs which is known as the bottleneck principle [Wang and Liao, 2024]. PGI and reversible functions work together to preserve essential information and data across the CNN network's depth ensuring better model convergence and detection performance. PGI is the ability to compute and update the gradients during the training process which are necessary for optimising the networks parameters and improve prediction capabilities. GELAN is YOLOv9's backbone and like the name suggests, efficiently combines information from different layers within the CNN. The GELAN ensures that by preserving important information computational costs are also reduced. The backbone is generalised making it adaptable to broader detection tasks and datasets [Wang and Liao, 2024].

2.2.4 Limitations

Despite YOLO's innovations and benefits, the framework still presents some challenges. Like other deep learning algorithms, YOLO requires extensive and large datasets for training. In actuality, the Blood cell dataset used in this study consists of 364 images orig-

inally [Alam and Islam, 2019] but several images have been flipped, rotated and brightness altered to introduce variety to the training set. YOLO tends to generally be sensitive to object scale, tending to detect larger objects more efficiently than smaller one [Qureshi et al., 2023]. Since the algorithm divides the input image into a grid of cells, small objects might not be large enough to occupy the entire cell. Additionally, occluded and overlapped objects are particularly challenging to YOLO algorithms which presents a further problem within a biomedical image context as overlapped cells are a prominent challenge for cell detection and counting. YOLO is also an incredibly computationally expensive algorithm and requires sufficient GPUs (Graphical Processing Units) to operate. The model's network and architecture escalate the complexity and computational resources needed, which is a trade-off for its speed and high accuracy. These drawbacks should be taken into consideration when determining which method to adopt based on the given problem and dataset.

2.2.5 GADI

Since YOLO is a computationally expensive algorithm, it is recommended that users have access to sufficient GPUs to properly deploy the model. For this study, we were provided access to the National Computing Infrastructure's supercomputer GADI. Within GADI, we were allocated 1 Terabyte of storage and 60 KiloStore computing Units so that a miniconda environment could be set up, JupyterLab notebooks run, and data files stored. Access to powerful GPUs were also made available to execute and deploy the YOLO models as they operate on the Pytorch NVIDIA core.

2.2.6 Experimental Setup

2 sets of experiments were conducted. First, we set the baseline performance for YOLO versions V5, V8 and V9. All models used the Adam optimiser and the cosine learning rate scheduler. The cosine learning rate scheduler adjusts the learning rate over time based on the cosine function, gradually decreasing it smoothly and continuously. It helps avoids sudden changes in the learning rate that might disrupt the optimisation process, achieves a good balance between the models convergence and stability and is easy to implement. The initial learning rate was set to 0.1 and the final learning rate was set to 0.00001. Secondly, we tuned each model for 100 iterations, 25 epochs per iteration allowing all parameters to reach its optimal value including learning rates, augmentation parameters such as hue and mosaic, as well as loss. The best parameters after 100 iterations for each model can be found in table 4.

2.3 State of the Art Method: CONCORDE-Net

The second component of this study compares our YOLO results to an existing state-of-the-art method that addresses the challenges of cell counting and detection.

CONCORDE-Net (Cell Count RegulariseD Convolutional Neural Network) is a state-of-the-art method that explicitly addresses the challenges of variability in staining, differing

Hyperparameter	V5	V8	V9
lr0:	0.0520	0.0246	0.0419
lrf:	0.0001	0.0002	0.0001
momentum:	0.9445	0.8209	0.8740
weight decay:	0.0003	0.0003	0.0002
warmup epochs:	4.5063	4.8121	4.9676
warmup momentum:	0.7757	0.3435	0.5063
box:	7.5398	7.9340	11.6439
cls:	0.2739	0.3560	0.2000
dfl:	0.9206	1.3791	3.3217
hsvh:	0.0265	0.0077	0.0060
hsvs:	0.6667	0.7217	0.6505
hsvv:	0.1783	0.1670	0.1872
degrees:	0.0000	0.0000	0.0000
translate:	0.1473	0.0277	0.0703
scale:	0.3742	0.5061	0.2290
shear:	0.0000	0.0000	0.0000
perspective:	0.0000	0.0000	0.0000
flipud:	0.0000	0.0000	0.0000
fliplr:	0.6469	0.6307	0.2989
bgr:	0.0000	0.0000	0.0000
mosaic:	1.0000	0.5251	0.9910
mixup:	0.0000	0.0000	0.0000
copypaste:	0.0000	0.0000	0.0000

Table 4: *Complete list of the best hyper-parameters for each model. Each model was tuned for 100 iterations and 25 epochs per iteration.*

cell expressions and artefact noise within immunohistochemical images. [Hagos et al., 2019]'s method combined image preprocessing and a CNN framework that was itself constituted of a cell counter and cell detector. [Hagos et al., 2019] used morphological reconstruction by erosion, pseudo-segmentation and thresholding to extract the weakly stained cells prior to feeding it to their counter and detector framework. Morphological reconstruction by erosion works to construct the images by removing features from the image without altering the shape of the objects within [Instruments, 2024]. The cell counter component consists of a CNN layer with 4 layers. The output layer was a dense layer that computed an estimate of the number of cells from the input image. The cell detector component consisted of an encoder, decoder and cell counter. At a high level, the encoder-decoder section was a CNN with a U-net architecture which has parallel and varying size filters that allow the network to extract multiple scale features within a given layer. They also integrated a variety of techniques such as dice overlap, new cell count loss functions and a multi-stage CNN to separate weakly stained cells from the background so that they could be better detected and counted (refer to table 7 for summary).

3 Results

Two sets of experiments were conducted where we set the baseline performance and hyper-parameter tuned performance for each YOLO version. From table 5, it was consistently observed that all tuned models increased their Mean Average Precision (MAP) performance. For the remainder of this paper all discussions of model performance relate to the hyper-parameter tuned models.

YOLOv5, YOLOv8 and YOLOv9 were tuned for 100 iterations and 25 epochs per iteration, with validation performed during training, a feature provided by Ultralytics. At the model level, YOLO versions V8 and V9 performed the best achieving 2% and 9% increase in performance respectively, for MAP at thresholds greater than 50% (MAP50-95). V5 saw a 9% increase in performance after tuning.

	Precision	Recall	MAP50	MAP50-95
YOLOv5 Baseline	0.74	0.80	0.83	0.49
YOLOv5 Hyperparameter	0.77	0.92	0.88	**0.58**
YOLOv8 Baseline	0.83	0.88	0.90	0.60
YOLOv8 Hyperparameter	0.83	0.90	0.91	**0.63**
YOLOv9 Baseline	0.81	0.56	0.85	0.53
YOLOv9 Hyperparameter	0.83	0.90	0.91	**0.64**

Table 5: *Baseline versus hyper-parameter tuned models. Tuned models have Mean Average Precision Values (MAP50-90%) outperform the baseline models.*

		Precision	Recall	MAP50	MAP50-95
YOLOv5	Platelets	0.77	0.89	0.84	0.44
	RBC	0.62	0.87	0.83	0.58
	WBC	0.93	1.00	0.97	0.72
YOLOv8	Platelets	0.81	0.88	0.88	0.48
	RBC	0.72	0.83	0.86	0.61
	WBC	0.97	1.00	0.98	0.80
YOLOv9	Platelets	0.78	0.90	0.88	0.51
	RBC	0.76	0.81	0.87	0.63
	WBC	0.95	1.00	0.97	0.79

Table 6: *Precision, Recall, MAP for each object class across all three YOLO models. WBCs are consistently detected with high precision, recall and MAP. Platelets and RBCs are very similar in detection results.*

YOLOv5 Performance: The best F1-score for YOLOv5 is observed at 84% accuracy at an IOU threshold of 38.1% on average (see fig. 7 in Appendix A.1). Indicating that 38.1% is the optimal threshold where V5 produces the most accurate and confident predictions. This value was then set when making predictions for V5. From the F1-confidence scores and table 6, WBCs have 90% accuracy at 90% confidence on average. RBCs are peaking at 75% accuracy at 45% confidence on average whereas, platelets are achieving 80% accuracy at 30% confidence. The Precision-Recall (fig. 7 in Appendix A.1) curves for V5 demonstrate that MAP at the 50% threshold is performing well at the object level 84%, 83% and 97% for platelets, RBCs and WBCs respectively (refer to table 6).

YOLOv8 Performance: The best F1-score for YOLOv8 is observed at 86% accuracy at an IOU threshold of 41.1% on average (see fig. 8 in Appendix A.1). Indicating that 41.1% is the optimal threshold where V8 produces the most accurate and confident predictions. This value was then set when making predictions for V8.From the F1-confidence scores and table 6, WBCs have perfect accuracy at 90% confidence on average. RBCs and Platelets are performing similarly to each other peaking at 75%-79% accuracy at 45-50% accuracy on average. The Precision-Recall (fig. 8 in Appendix A.1) curves for V8 demonstrate that MAP at the 50% threshold is performing well at the object level 88%, 86% and 98% for platelets, RBCs and WBCs respectively (table 6. Due to the higher number of missing ground truth labels, RBCs generate a lot more false positives and/or false negatives which leads to a dip in the precision and recall metrics for the object class (see fig. 5).

YOLOv9 Performance: V9's performance is only marginally higher than V8, making predictions at a threshold only 0.06% higher. The best F1 score for YOLOv9 is observed at 86% accuracy at an IOU threshold of 41.7% on average (see fig 9 in Appendix A.1). Indicating that at 41.7%, V9 produces the most accurate and confident predictions for the different object classes on average. Like V8, this value was set when making predictions for V9. From the F1- confidence score and table 6, WBCs have a very high accuracy of 98% at 90% confidence on average. The Precision-Recall (fig. 9) curves for V9 demonstrate that MAP at the 50% threshold is performing well at the object level 88%, 87% and 97% for platelets, RBCs and WBCs respectively (table 6). Performance of RBCs and platelets are very similar bar the higher thresholds that V9 operates at.

CONCORDE-Net Performance: CONCORDE-Net achieved a F1-score of 87.3% overall and outperformed the other three contenders within[hagos]'s study. However, the model saw a comparatively lower precision. CONCORDE-Net also showed good performance at detecting touching cells with weak boundary gradients.

4 Discussion

This study sought to analyse the performance of YOLO variants V5, V8 and V9 on the novel Blood-Cell-Detection dataset and determine whether YOLO as a framework can adequately address the challenges of cell detection and counting. From the baseline

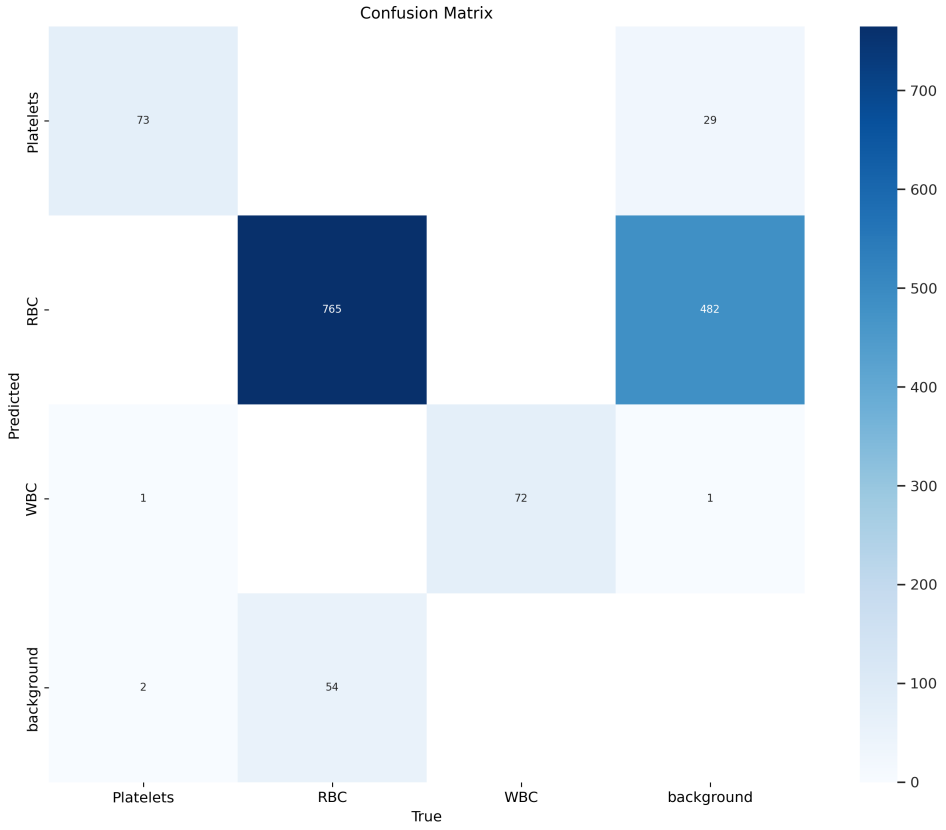

Figure 5: *Confusion Matrix for YOLOv8 demonstrates that alot of RBCs that do not have ground truth labels are being detected in addition to platelets. Which in turn impacts precision and recall even though it is a positive result.*

and tuned performance model, it was consistently observed that all YOLO versions saw an increase in MAP50 and MAP50-95 after all parameters had been tuned. This can be attributed to the data augmentation parameters such as mosaic, hue, saturation etc. adjusting and adding variety to the images. Mosaic is one of the key features of YOLO, where up to four images are patched together to make a composite image. As noted above, the dataset is incomplete in its annotations especially the overlapped cells or those that fall on the cropped border. Mosaic boosts performance by piecing together other cropped objects together during the training process as in the case of RBCs.

YOLOv8 and YOLOv9 were the top performers due to their higher accuracy at higher threshold results. In order to produce predictions, the model must exceed a certain classification confidence threshold such as the IOU threshold. For V8 and V9 the IOU threshold sits at 41-42% approximately. Whereas YOLOv5 has a threshold at 38.1%. Visually this is indicated by V5 producing more detection predictions but at a lower confidence (see fig. 6). In contrast, all bounding boxes produced by V8 and V9 are of higher accuracy

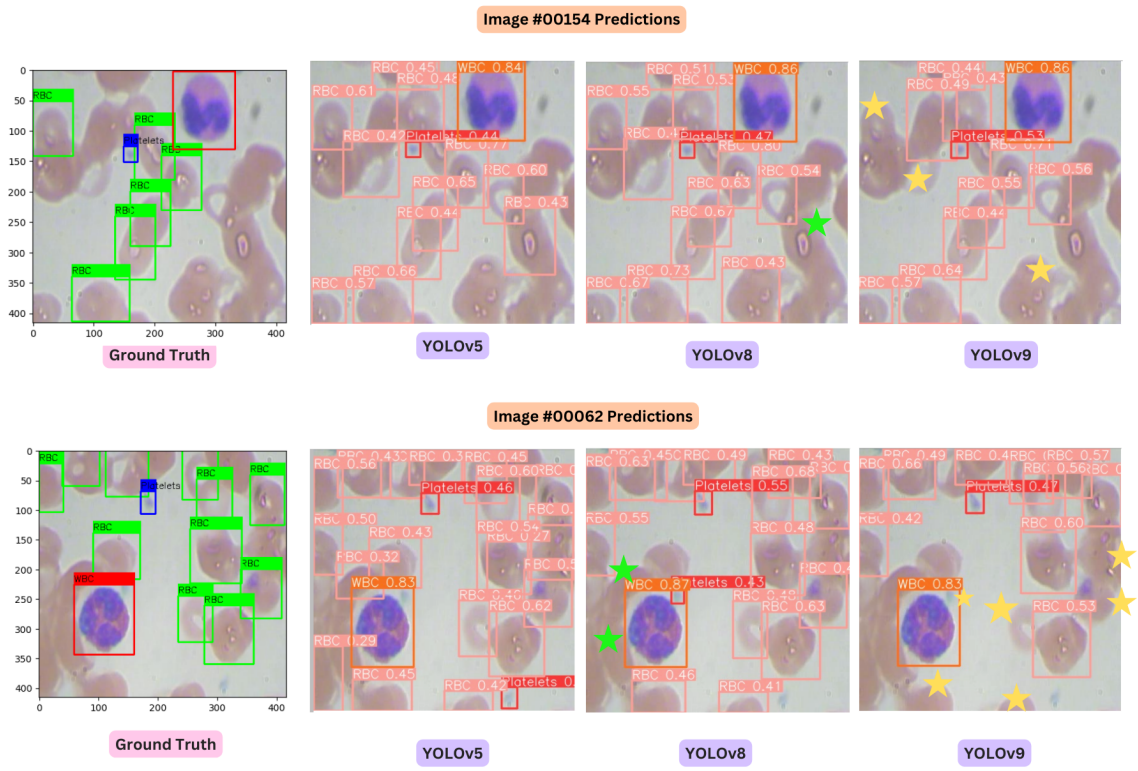

Figure 6: *Image #00154 (top) and Image #00062 (bottom). From left to right shows the ground truth, prediction results for YOLOv5, YOLOv8 and YOLOv9. The colored stars highlight the missing bounding boxes from the preceding detector.*

and higher confidence. These higher confidence results are due to the key innovations of V8 and V9's architectures. Both model preserve greater image information as the image is passed through the network allowing for higher confidence predictions. Their IOU loss functions also generate bounding boxes that can handle overlapped objects with higher accuracy and this is demonstrated in fig. 6.

Across the board, WBCs are detected and predicted with near perfect precision and recall for all models. This comes down to YOLO's general ability to detect larger objects with higher accuracy than smaller ones since they occupy most of the the grid cell and provide more information to be extracted. RBCs and platelets perform similarly to each other peaking at 80% accuracy at an IOU threshold of 40% at the object level. This is not to say that V8 and V9 are not detecting RBCs well. As seen from the confusion matrix (fig. 5), there are more 'background' objects being detected leading to a high number of false positives being classified. The false positives are generated due to the incomplete annotations given by the ground truth and these occur most often as overlapped cells or those cells cropped at the image edge. Higher false positives therefore lead to lower precision and recall values. In the case of image #000154 and #00062, these confidence

SOTA	CCD Challenges	Key components to solution	Approach	Data
CONCORDE-Net	Variability in cell staining expression intensity artefact noise	Conventional Dice overlap Cell Count loss function	Extensive image pre-processing: erosion, making, thresholding, psuedo segmentation	5 object classes 6 Whole slide Images 170 annotated images

Table 7: *Summary of CONCORDE-Net's challenges, components approach and data. SOTA (State of the Art)*

scores fall below the 41-42% threshold. Pathologists will need to consider what is the appropriate level of materiality they are comfortable with and determine the trade off between confidence and accuracy based on their dataset and context. Ideally, they should aim for high thresholds and high accuracy without the occurrence of overfitting. Data augmentations were shown to boost model performance and therefore is recommended that this step is done before passing input images to the model, especially when evaluating much more complex image datasets than in this study.

SOTA	F1-score	Precision	Recall
CONCORDE-Net	0.87	0.85	0.89
YOLOv9	0.86	0.83	0.90

Table 8: *Comparison of YOLOv9 and CONCORDE-Net's performance metrics. YOLOv9 exhibits similar performance but on a much simpler dataset.*

As it was in the case of [Hagos et al., 2019] where extensive image preprocessing was carried out to extract the cells from the background through distortion erosion and pseudo-segmentation. CONCORDE-Net achieved a F1-score of 87.3% which is not extremely different to YOLOv9's accuracy of 86% at 41.7% IOU (table 8). However, it must be emphasised that our data set was incredibly simple, clean and lacked the complexity of those images used in Hagos' study. Therefore, if YOLOv9 were to be executed on the same dataset without any image preprocessing it might suffer in performance. CONCORDE-Net performs exceptionally well in detecting cells with weak boundaries and identifying weak cells. It demonstrated the ability through its different loss functions, to learn patterns and separate those closely located and weakly stained cells. Precision and recall are not dissimilar to YOLOv9's performance as they both share a complete list of ground truth labels. Pathologists must consider the quality of labels and images they have at their disposal as this will inform the preprocessing steps they must undertake before deploying YOLOv8 or YOLOv9.

In terms of whether the YOLO variants assessed in this study have addressed the challenges of cell detection and counting, we would state not completely. V8 and V9

are able to detect overlapped cells and to an extent varied stained cells but, not at high confidence. YOLO is known for it speed and accuracy. Practitioners must determine which is valued more; if it is speed then YOLO is the ideal framework of choice. However, if accuracy and high confidence is required then other methods or combinations of techniques must be developed.

5 Conclusion and Future work

While YOLOv8 and YOLOv9 are achieving close to state-of-the-art performance, they are doing so on a novel and simple dataset. YOLO must be tested on higher complexity datasets in conjunction with other image processing, statistical and deep learning techniques. Pathologists must consider the limitations of YOLO discussed in this paper such as GPU requirements, and select the appropriate model framework best suited to their scenario and dataset. From this study, we conclude that YOLOv8 and YOLOv9 have great potential to be used within a biomedical imaging context but, must develop a robust methodology to produce high confidence and high accuracy predictions especially in prognostic and diagnostic contexts.

References

[Alam and Islam, 2019] Alam, M. M. and Islam, M. T. (2019). Machine learning approach of automatic identification and counting of blood cells. *Healthcare Technology Letters*, 6(4):103–108.

[Drałus et al., 2021] Drałus, G., Mazur, D., and Czmil, A. (2021). Automatic detection and counting of blood cells in smear images using retinanet. *Entropy*, 23(11):1522.

[Hagos et al., 2019] Hagos, Y. B., Narayanan, P. L., Akarca, A. U., Marafioti, T., and Yuan, Y. (2019). Concorde-net: Cell count regularized convolutional neural network for cell detection in multiplex immunohistochemistry images. *Lecture Notes in Computer Science*, 11764:667–675.

[Huang et al., 2020] Huang, Z., Ding, Y., Song, G., Wang, L., Geng, R., He, H., Du, S., Liu, X., Tian, Y., Liang, Y., and et al. (2020). Bcdata: A large-scale dataset and benchmark for cell detection and counting. *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, page 289–298.

[Instruments, 2024] Instruments, N. (2024). Morphological reconstruction. Accessed: 2024-03-27.

[Kundu, 2023] Kundu, R. (2023). Yolo algorithm for object detection explained [+examples].

[MrAnayDongre, 2021] MrAnayDongre (2021). Mranaydongre/bloodcell-detection-datatset: This is a dataset of blood cells photos.

[Nair et al., 2021] Nair, L. S., R, R. P., Sugathan, G., Gireesh, K. V., and Nair, A. S. (2021). Mitotic nuclei detection in breast histopathology images using yolov4. *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*.

[Qiao et al., 2023] Qiao, Y., Guo, Y., and He, D. (2023). Deep learning-based autonomous cow detection for smart livestock farming. 13744:246–258.

[Qureshi et al., 2023] Qureshi, R., RAGAB, M. G., ABDULKADER, S. J., muneer, a., ALQUSHAIB, A., SUMIEA, E. H., and Alhussian, H. (2023). A comprehensive systematic review of yolo for medical object detection (2018 to 2023). *IEEE Access*, 11:1–30.

[Topuz et al., 2023] Topuz, Y., Yıldız, S., and Varlı, S. (2023). Performance analysis of the yolo series for object detection: Detection of mitosis cells in histopathology images. *2023 Medical Technologies Congress (TIPTEKNO)*.

[Torres and Austen, 2024] Torres, J. and Austen, J. T. J. (2024). Yolov8 architecture: A deep dive into its architecture.

[Ultralytics, 2024] Ultralytics (2024). Yolov9.

[Wang et al., 2023] Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2023). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Wang and Liao, 2024] Wang, C.-Y. and Liao, H.-Y. M. (2024). YOLOv9: Learning what you want to learn using programmable gradient information.

[Yellin et al., 2018] Yellin, F., Haeffele, B. D., Roth, S., and Vidal, R. (2018). Multi-cell detection and classification using a generative convolutional model. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 8953–8961.

[Yücel et al., 2023] Yücel, Z., Akal, F., and Oltulu, P. (2023). Mitotic cell detection in histopathological images of neuroendocrine tumors using improved yolov5 by transformer mechanism. *Signal, Image and Video Processing*, 17(8):4107–4114.

A Appendix

A.1 Additional Figures and Tables

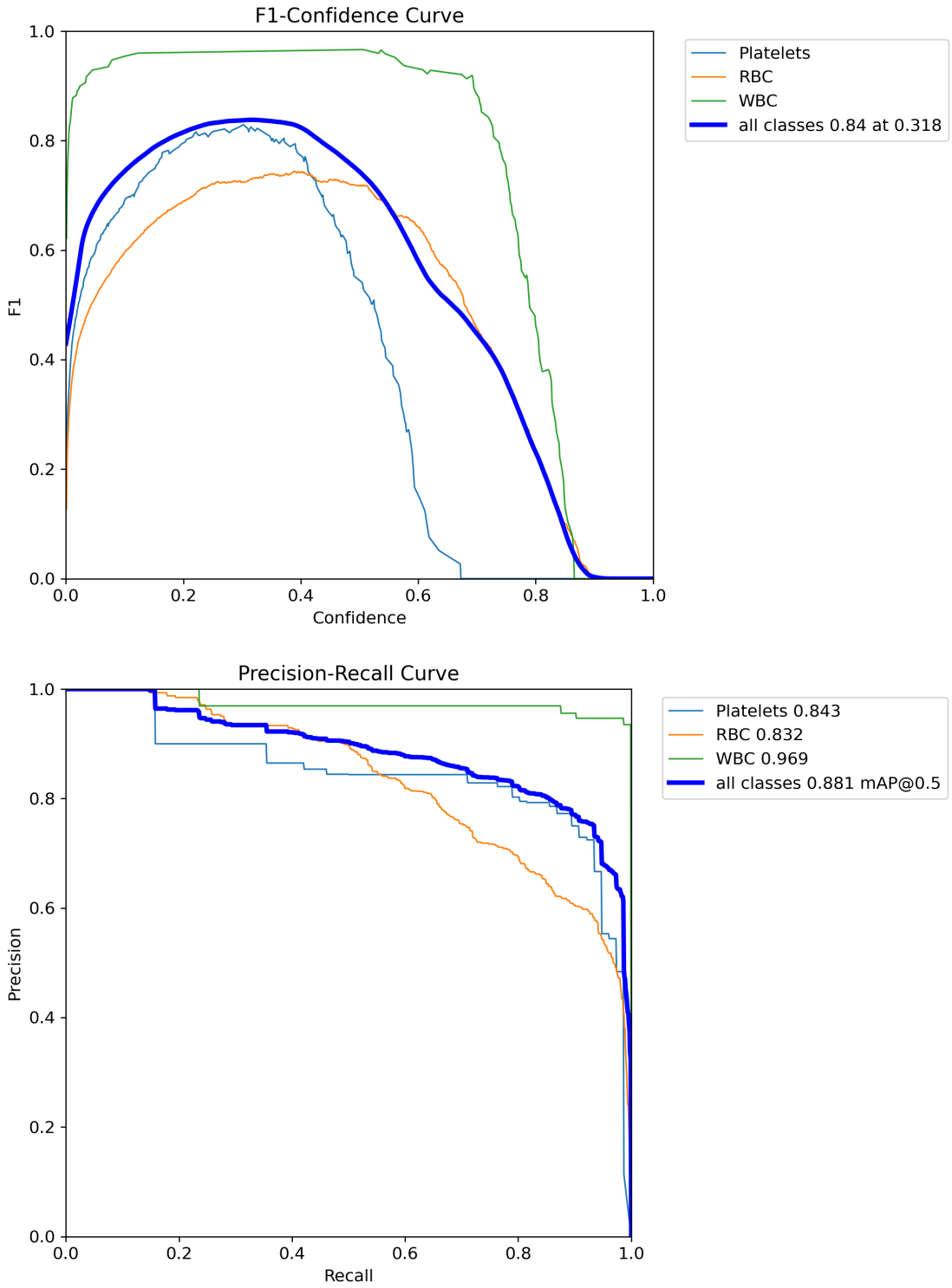

Figure 7: *F1-Confidence and Precision-Recall curves for YOLOv5*

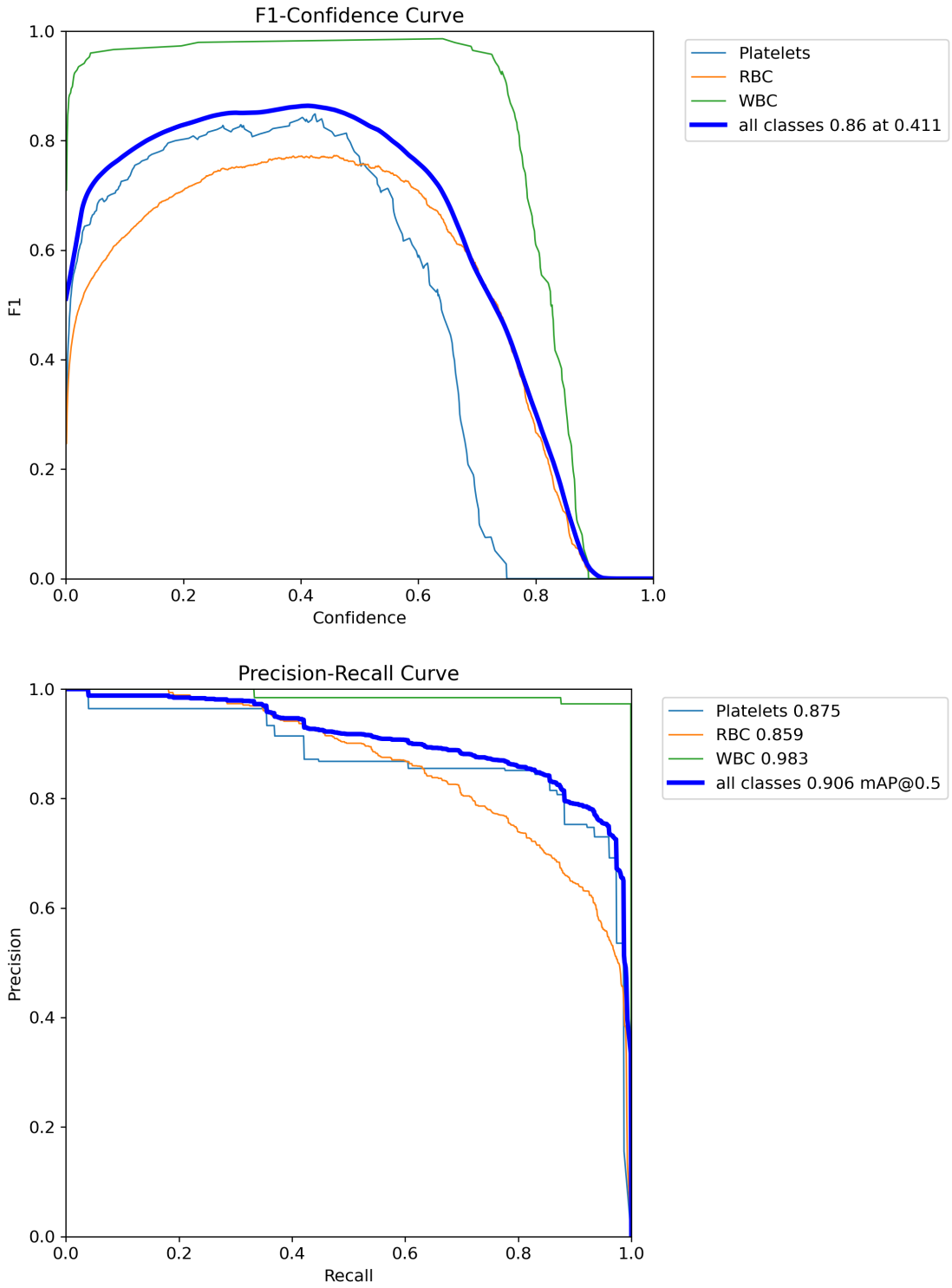

Figure 8: *F1-Confidence and Precision-Recall curves for YOLOv8*

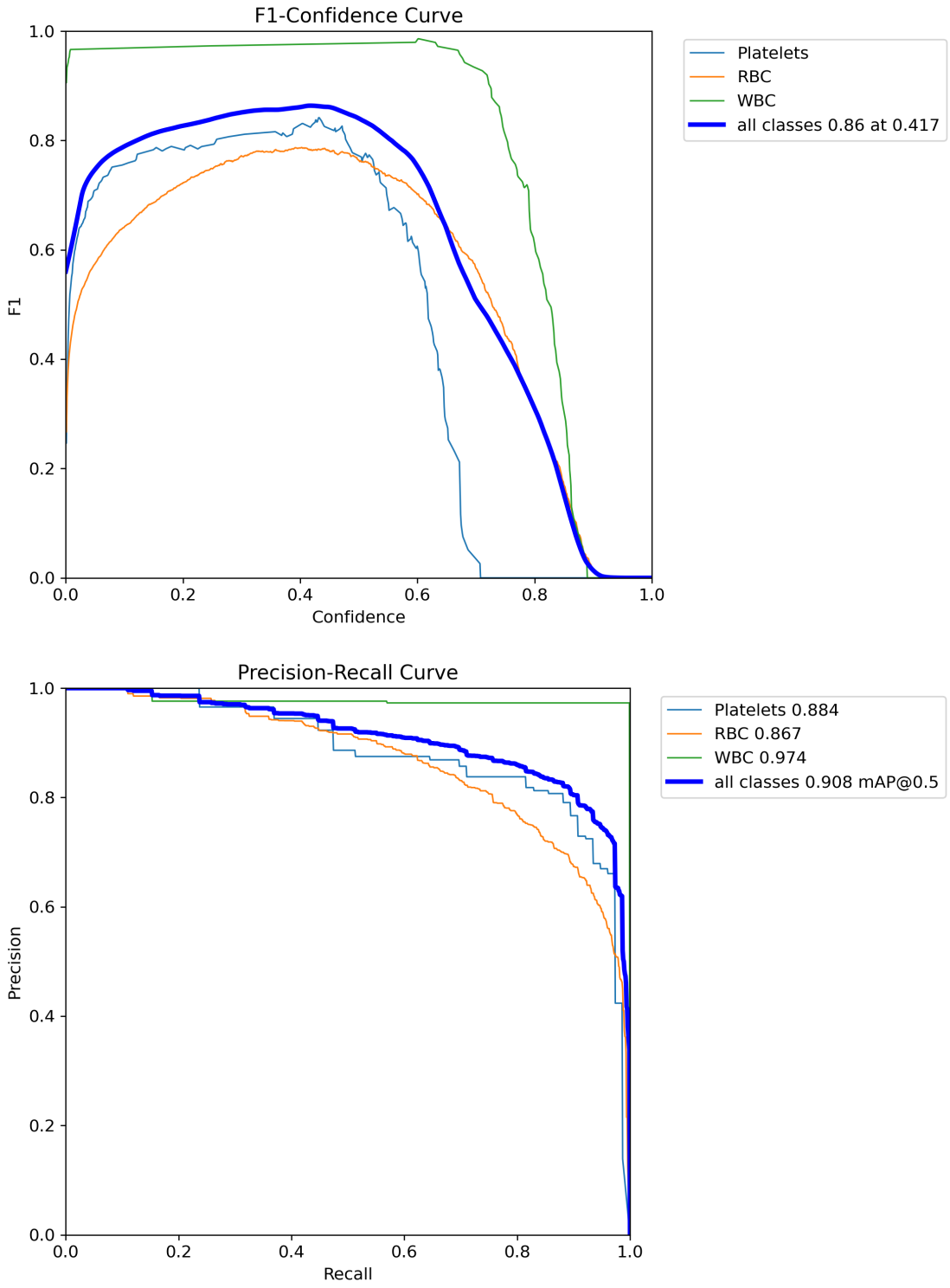

Figure 9: *F1-Confidence and Precision-Recall curves for YOLOv9*

Task	Status	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Break1	Break 2	Week 9	Week 10	Week 11	Week 12	Week 13
Technical																
T.1 GADI Installation and Set Up	C															
T.1.2 Set up virtual Environment and its dependencies (miniconda, Ultralytics, pandas, etc.) with data	C															
T.1.3Deploy Kaggle code	C															
T.1.4 Create Pipeline on train, test predict	C															
T.1.5 Baseline YOLOv8	C															
T.1.6 Baseline YOLOv9	C															
T.1.7 Baseline YOLOv5	C															
T.1.8 Hyperparam YOLOv8	C															
T.1.9 Hyperparam YOLOv9	C															
T.1.10 Hyperparam YOLOv5	C															
T1.11 Model comparison- Record results and findings, selecting baseline model, performance metrics	C															
T1.12 select best performing model	C															
T1.13 Results and discussion	C															
Research																
R1.1 Conduct research YOLO variants and architecture- differences, similarities, objectives	C															
R1.2 Read 6 papers from 2018 onwards- record methods, results and how they addressed challenges	C															
R1.3 Select 3 SOAMS to High level+ understand methods YOLO or not YOLO related	C															
Comparisons																
C1.1 YOLO variants comparison	C															
C1.2 YOLO vs SOAMs comparison	C															
Key Deliverables																
D1.1 Initial Proposal Slides	C															
D1.2 Initial Proposal	C															
D1.1 Revised Proposal Slides	C															
D1.2 Revised Proposal	C															
D1.3 Preliminary Slides	C															
D1.4 Preliminary Report	C															
D1.5 Final Slides	C															
D1.6 Final Report	C															

Figure 10: Full timeline for all methods