CSE 472

Machine Learning Sessional

1505004 – All Rubayet Ahmed Tusher
1605026 – Mohammad Abser Uddin

PROBLEM

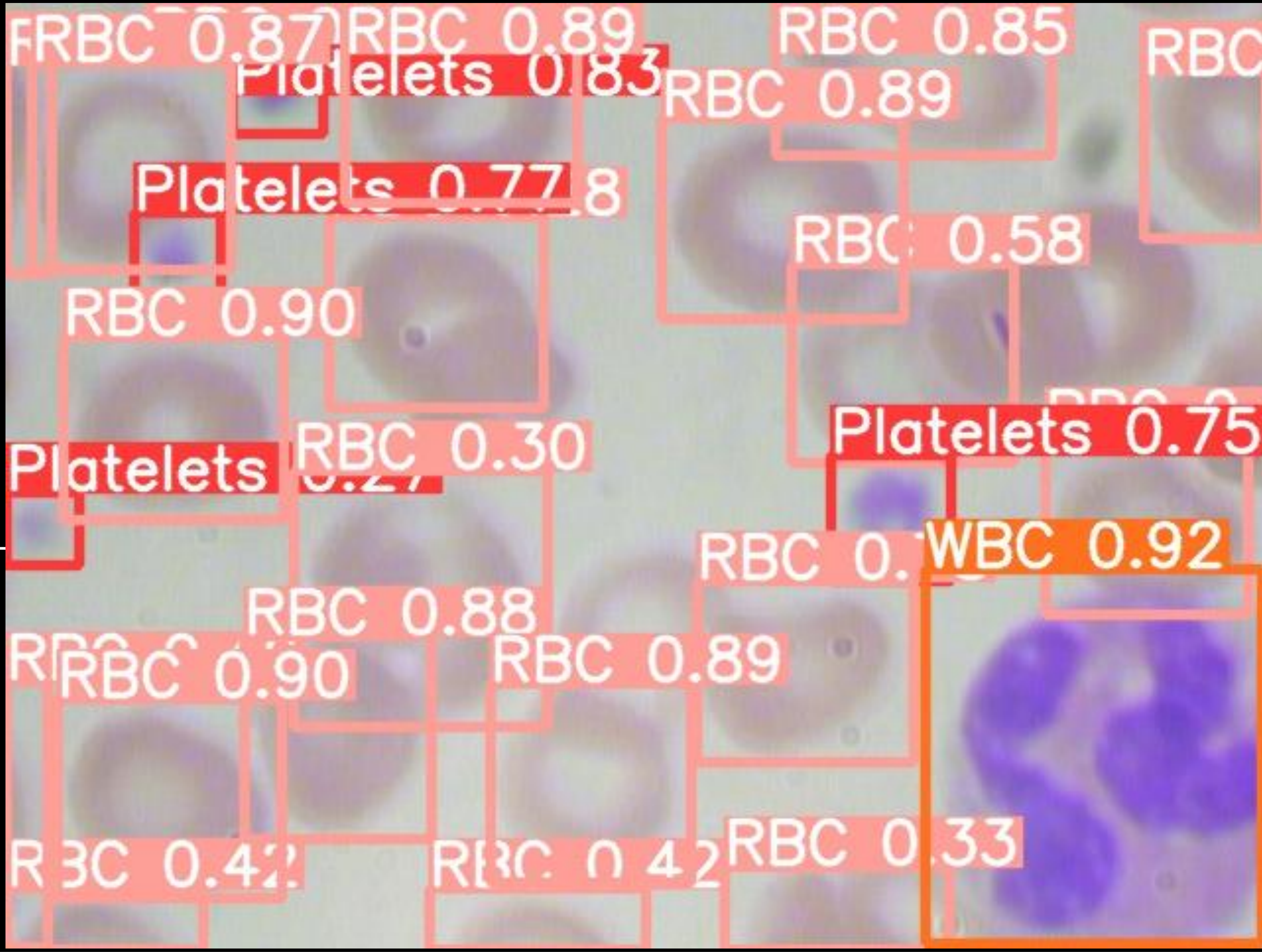
Given images of microscopic views of blood components, we need to detect basic blood cell types RBC, WBC, Platelet & their *counts* from that image

TYPE:

Image-based, Object Detection, Computer Vision

PROBLEM

Given images of microscopic views of blood components, we need to detect basic blood cell types RBC, WBC, Platelet & their *counts* from that image



MODEL

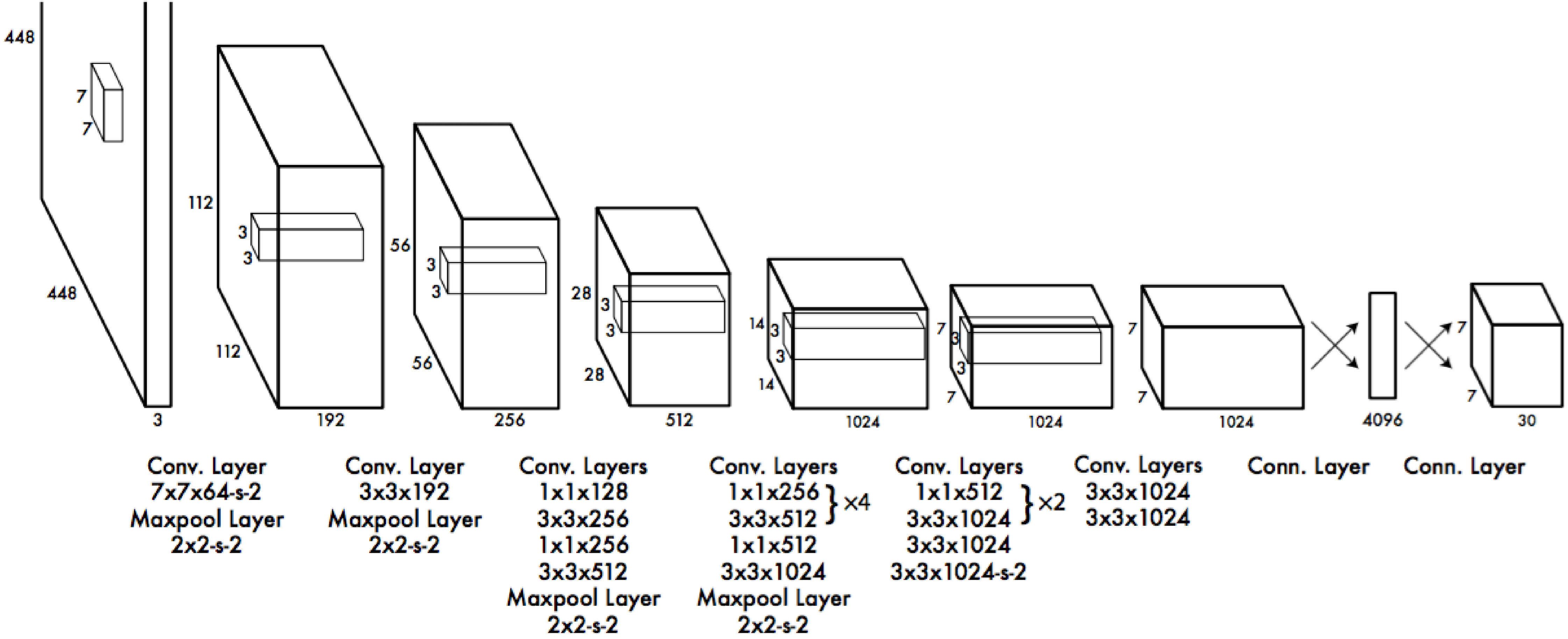
YOLO (You Only *Look* Once) v5:

- Extremely fast (**45 frames per second**) (Real Time Demo)
- Reasons globally on the entire image
- Learn **generalizable** representations (to new domains, like **art**)

YOLO ARCHITECTURE

Model Summary:

- **270** layers (original **24**)
- **7027720** parameters
- **7027720** gradients
- **15.9** GFLOPs



YOLO

ARCHITECTURE

```
                 from  n    params  module                                  arguments
  0                -1  1      3520  models.common.Conv                      [3, 32, 6, 2, 2]
  1                -1  1     18560  models.common.Conv                      [32, 64, 3, 2]
  2                -1  1     18816  models.common.C3                        [64, 64, 1]
  3                -1  1     73984  models.common.Conv                      [64, 128, 3, 2]
  4                -1  2    115712  models.common.C3                        [128, 128, 2]
  5                -1  1    295424  models.common.Conv                      [128, 256, 3, 2]
  6                -1  3    625152  models.common.C3                        [256, 256, 3]
  7                -1  1   1180672  models.common.Conv                      [256, 512, 3, 2]
  8                -1  1   1182720  models.common.C3                        [512, 512, 1]
  9                -1  1    656896  models.common.SPPF                      [512, 512, 5]
 10                -1  1    131584  models.common.Conv                      [512, 256, 1, 1]
 11                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 12           [-1, 6]  1         0  models.common.Concat                    [1]
 13                -1  1    361984  models.common.C3                        [512, 256, 1, False]
 14                -1  1     33024  models.common.Conv                      [256, 128, 1, 1]
 15                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 16           [-1, 4]  1         0  models.common.Concat                    [1]
 17                -1  1     90880  models.common.C3                        [256, 128, 1, False]
 18                -1  1    147712  models.common.Conv                      [128, 128, 3, 2]
 19          [-1, 14]  1         0  models.common.Concat                    [1]
 20                -1  1    296448  models.common.C3                        [256, 256, 1, False]
 21                -1  1    590336  models.common.Conv                      [256, 256, 3, 2]
 22          [-1, 10]  1         0  models.common.Concat                    [1]
 23                -1  1   1182720  models.common.C3                        [512, 512, 1, False]
 24      [17, 20, 23]  1     21576  models.yolo.Detect                      [3, [[10, 13, 16, 30, 33,
Model Summary: 270 layers, 7027720 parameters, 7027720 gradients, 15.9 GFLOPs
```

```
# YOLOv5 v6.0 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [64, 6, 2, 2]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, C3, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 6, C3, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, C3, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32
   [-1, 3, C3, [1024]],
   [-1, 1, SPPF, [1024, 5]],  # 9
  ]

# YOLOv5 v6.0 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 6], 1, Concat, [1]],  # cat backbone P4
   [-1, 3, C3, [512, False]],  # 13

   [-1, 1, Conv, [256, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 4], 1, Concat, [1]],  # cat backbone P3
   [-1, 3, C3, [256, False]],  # 17 (P3/8-small)

   [-1, 1, Conv, [256, 3, 2]],
   [[-1, 14], 1, Concat, [1]],  # cat head P4
   [-1, 3, C3, [512, False]],  # 20 (P4/16-medium)

   [-1, 1, Conv, [512, 3, 2]],
   [[-1, 10], 1, Concat, [1]],  # cat head P5
   [-1, 3, C3, [1024, False]],  # 23 (P5/32-large)

   [[17, 20, 23], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5)
  ]
```

YOLO ARCHITECTURE

Activation Function:

- Leaky ReLU	in middle / hidden layers
- Sigmoid	in the final detection layer

Optimization Function:

- SGD (Stochastic Gradient Descent)
- Adam

In YOLO v5, default optimization function for training is SGD.

Loss Function:

- Binary Cross-Entropy with Logits Loss (default)
- Focal Loss

YOLO

ARCHITECTURE

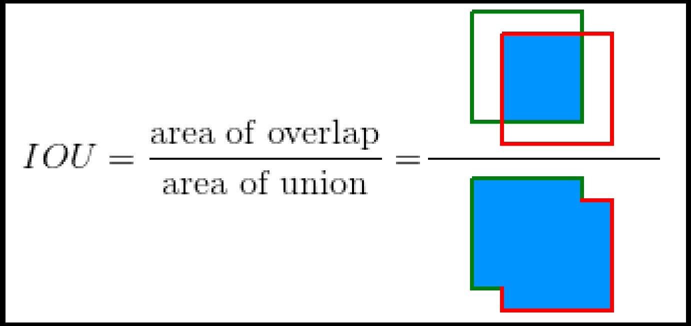
Loss Function

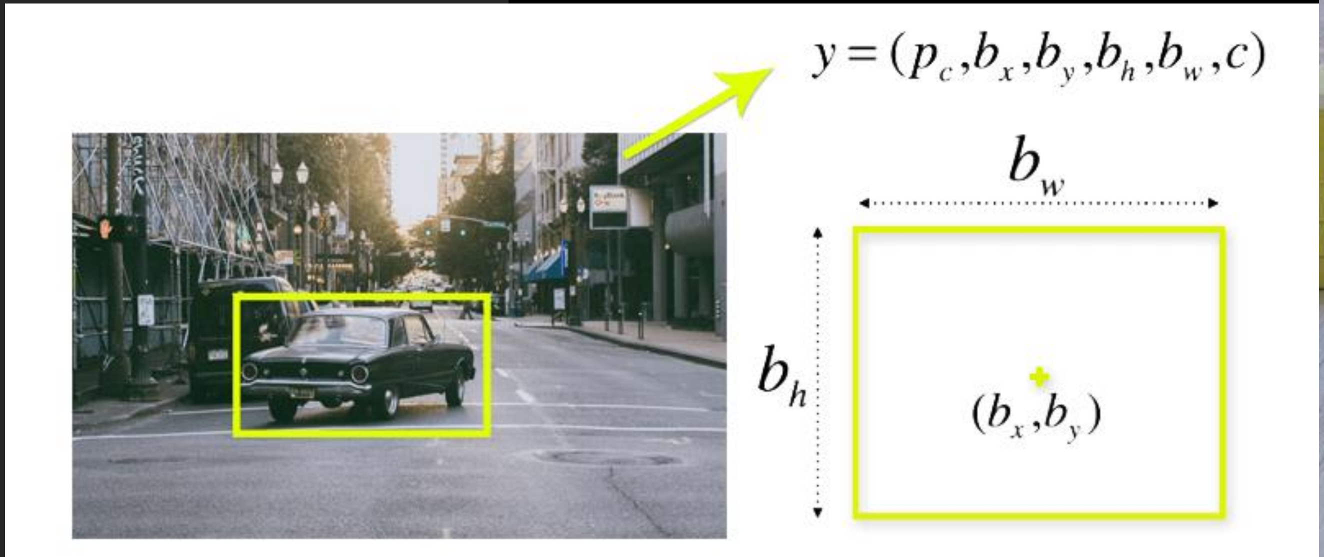
$$
\begin{aligned}
&\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \\
&+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}
$$

HOW YOLO ALGORITHM WORKS

YOLO algorithm works using the following three techniques:

- Residual Blocks
- Bounding Box Regression
- Intersection Over Union (IOU)





HOW YOLO ALGORITHM WORKS

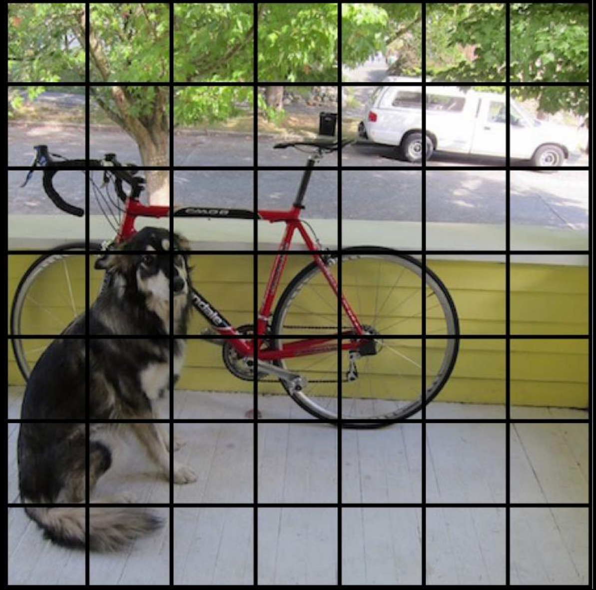
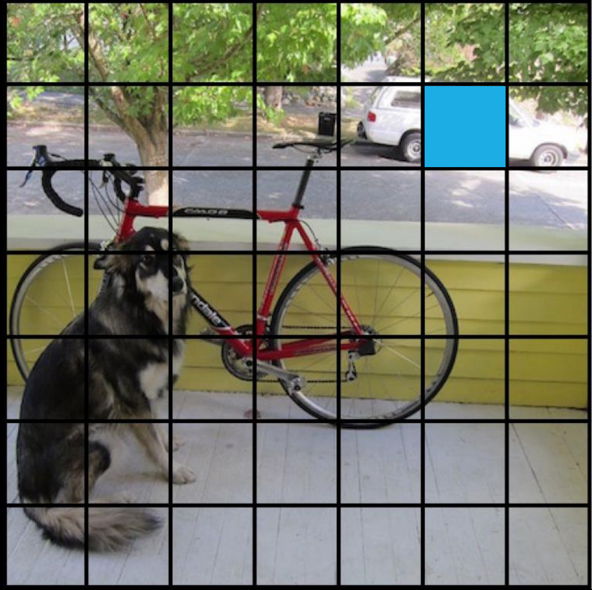
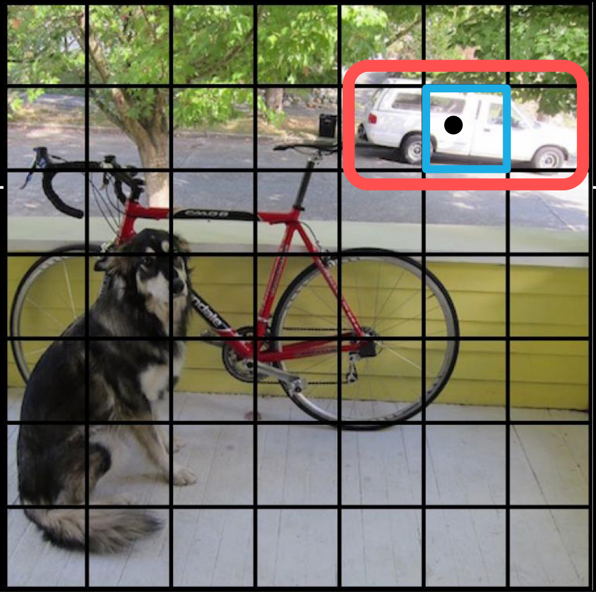


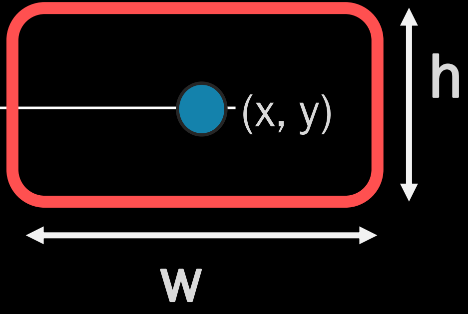
Image is divided into S*S grid



Each cell predicts B boxes (x,y,w,h) and confidence of each box: P (object)

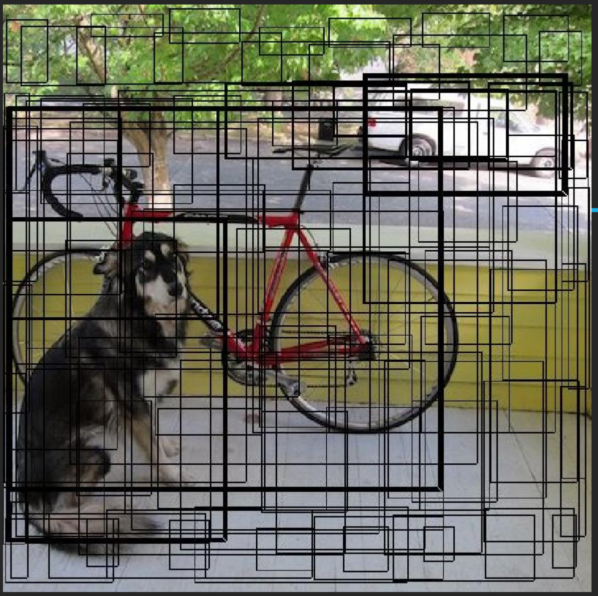


Predicted box's center is shown in black which is in the blue colored cell.

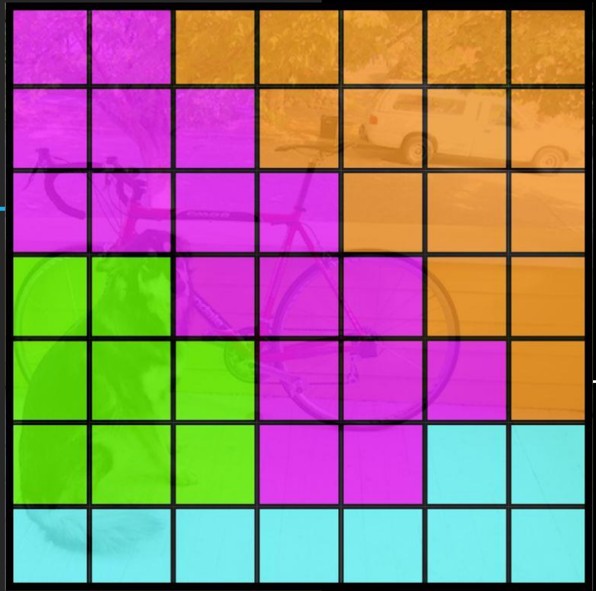


Anchor box specifying the object in a grid cell.

HOW YOLO ALGORITHM WORKS

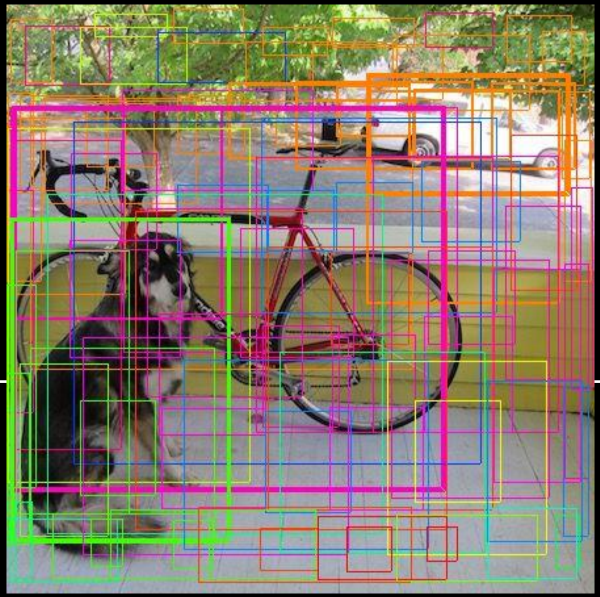


Each cell predicts boxes and confidences: P (object) Probability that the box contains an object.



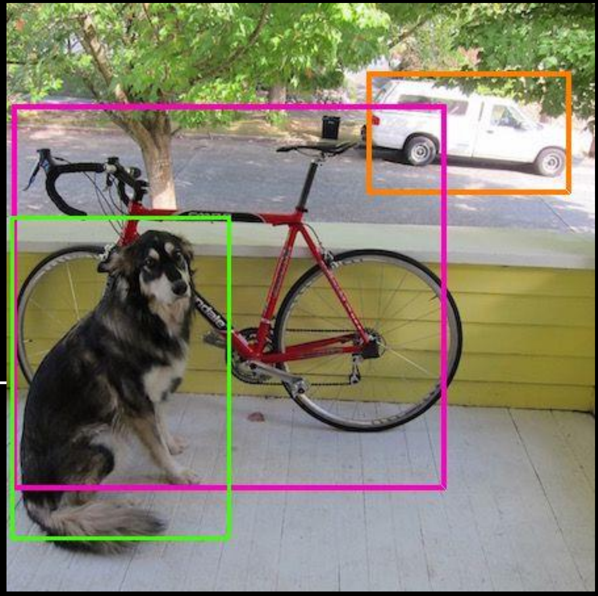
Each cell also predicts a class probability conditioning on objects:

- Green = Dog,
- Pink = Bicycle
- Orange = Car



Combining the box and class predictions.

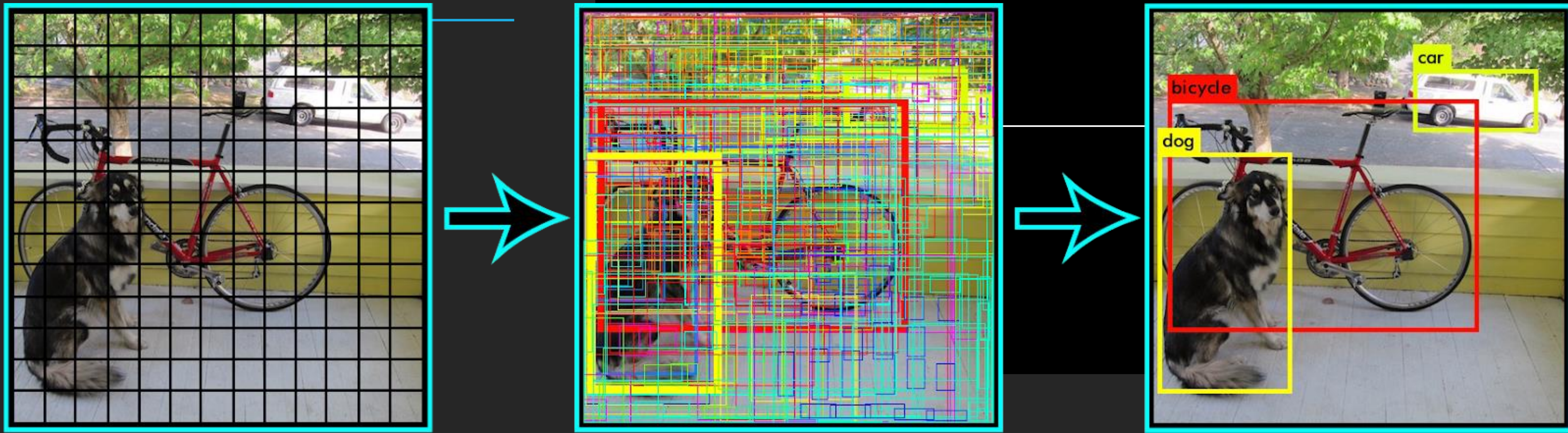
$$P(\text{class} \mid \text{object}) * P(\text{object}) = P(\text{class})$$



After performing threshold detection and Non-max suppression.

Discard all boxes with Probability ≤ 0.6

HOW
YOLO ALGORITHM
WORKS

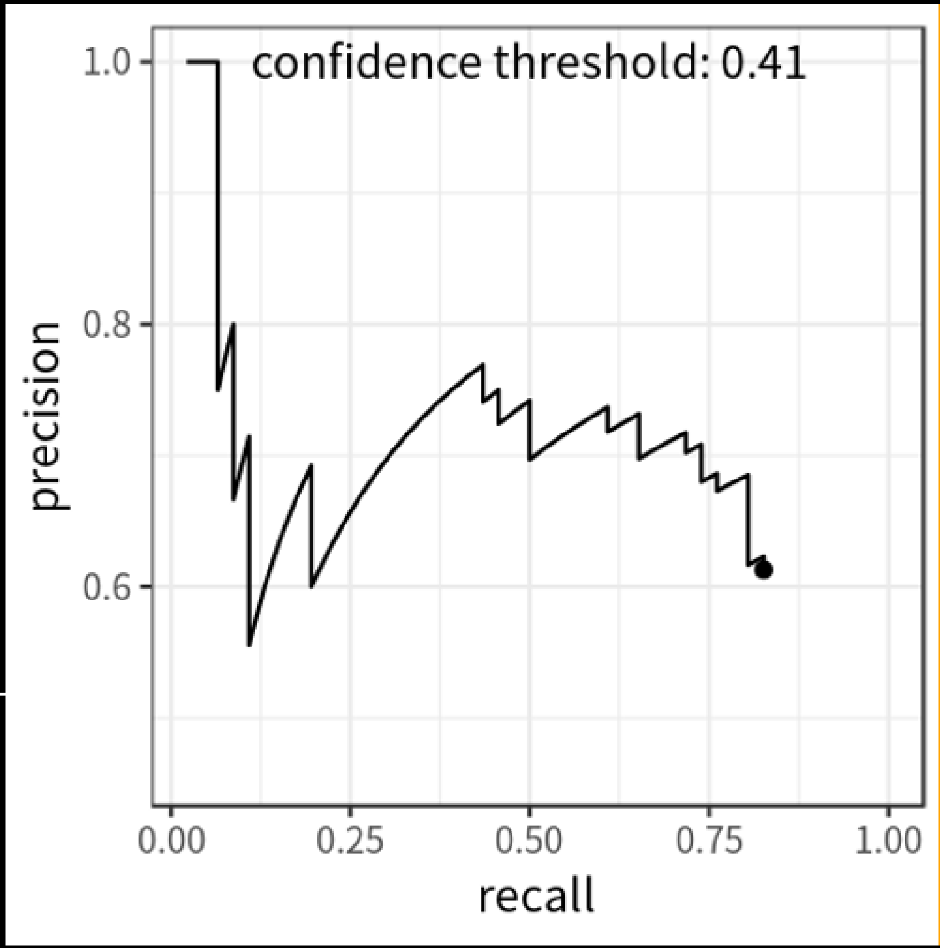


EVALUATION METRIC

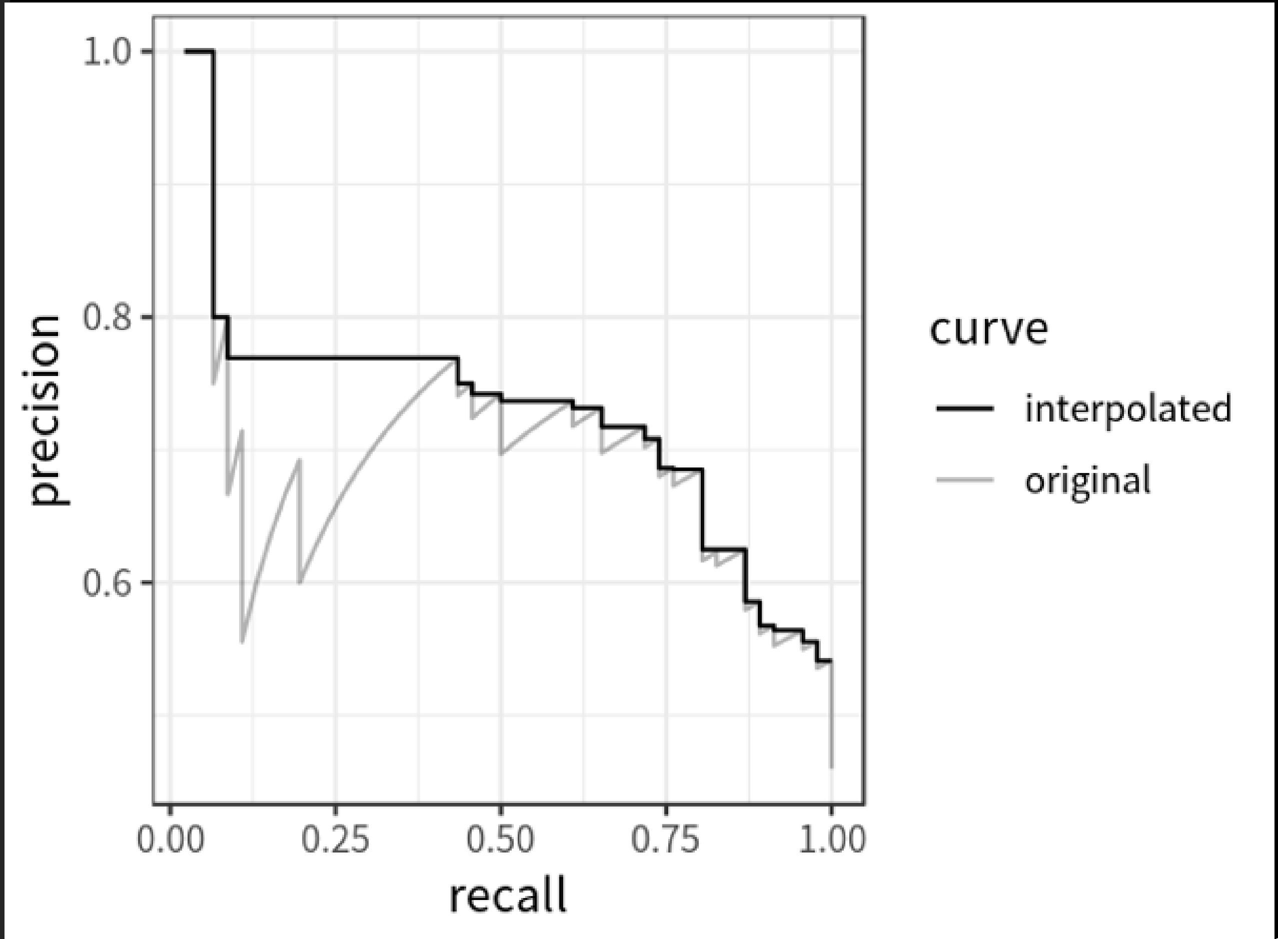
$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$



EVALUATION METRIC



AP can then be defined as the area under the interpolated precision-recall curve, which can be calculated using the following formula:

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1}) \quad (5)$$

where $r_1, r_2, \ldots, r_n$ is the recall levels (in an ascending order) at which the precision is first interpolated.

Mean Average Precision:

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K}$$

DATASET

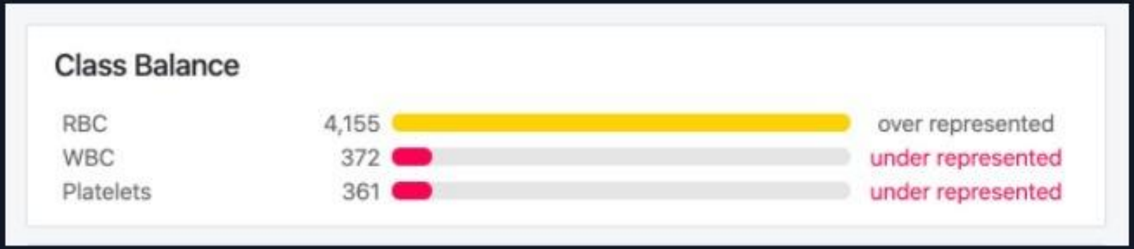
BCCD

Overview

This is a dataset of blood cells photos, originally open sourced by cosmicad and akshaylambda.

There are 364 images across three classes: `WBC` (white blood cells), `RBC` (red blood cells), and `Platelets`. There are 4888 labels across 3 classes (and 0 null examples).

Here's a class count from Roboflow's Dataset Health Check:



Data Splitting:

Partition	# of Images	%
Train	205	56.32 %
Validation	87	23.90 %
Test	72	19.78 %

TRAINING YOLO V5 100 epochs

mAP@.5:.95 → 62.5 %

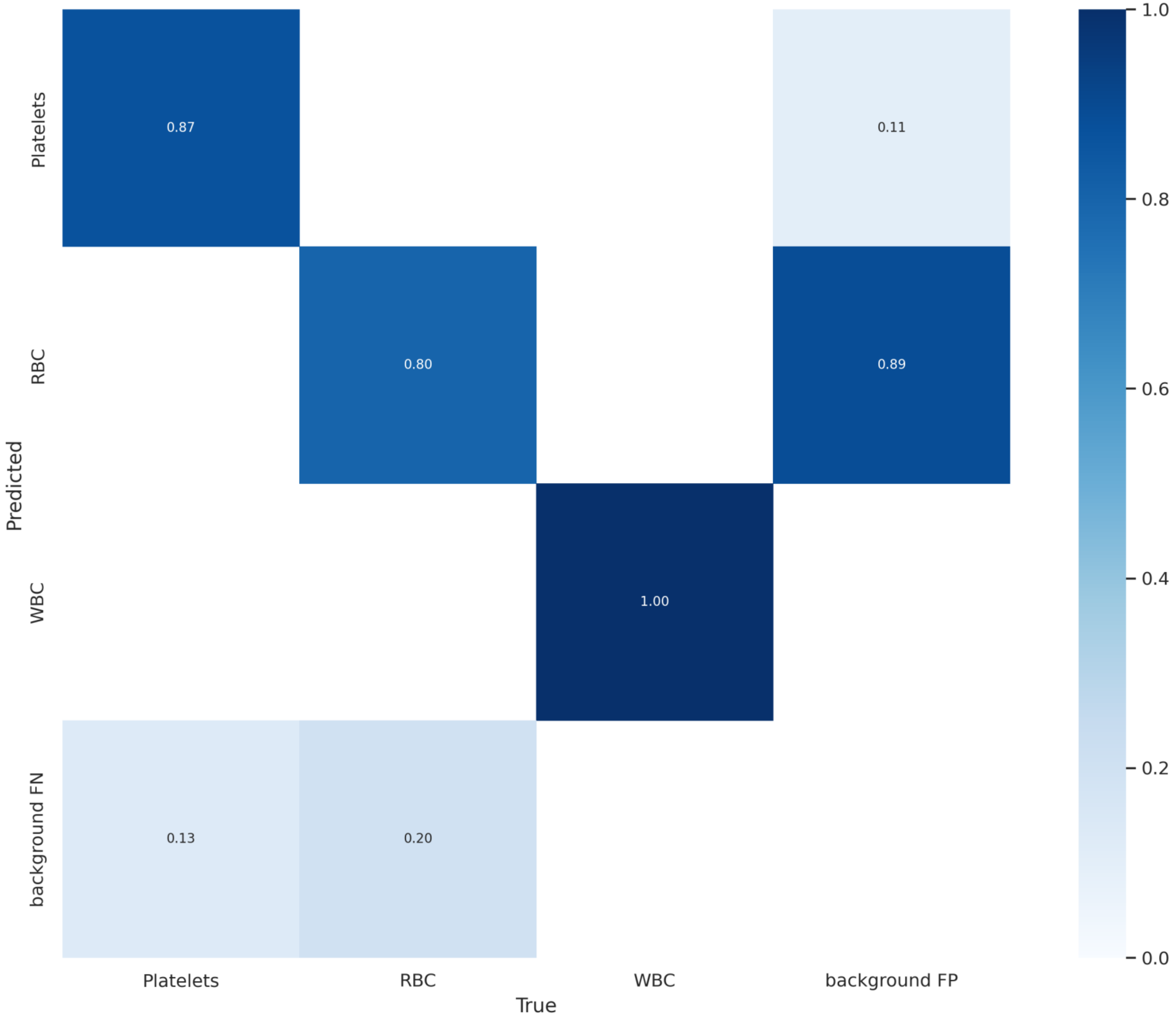
```
100 epochs completed in 0.543 hours.
Optimizer stripped from yolov5/runs/train/BCCM2/weights/last.pt, 14.4MB
Optimizer stripped from yolov5/runs/train/BCCM2/weights/best.pt, 14.4MB

Validating yolov5/runs/train/BCCM2/weights/best.pt...
Fusing layers...
Model Summary: 213 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 6/6
                 all         87       1138      0.876      0.874      0.907      0.625
           Platelets         87         83      0.818      0.865      0.879      0.454
                 RBC         87        968      0.825      0.756      0.854       0.61
                 WBC         87         87      0.984          1      0.987      0.812
Results saved to yolov5/runs/train/BCCM2
CPU times: user 18.7 s, sys: 3.08 s, total: 21.8 s
Wall time: 33min 3s
```

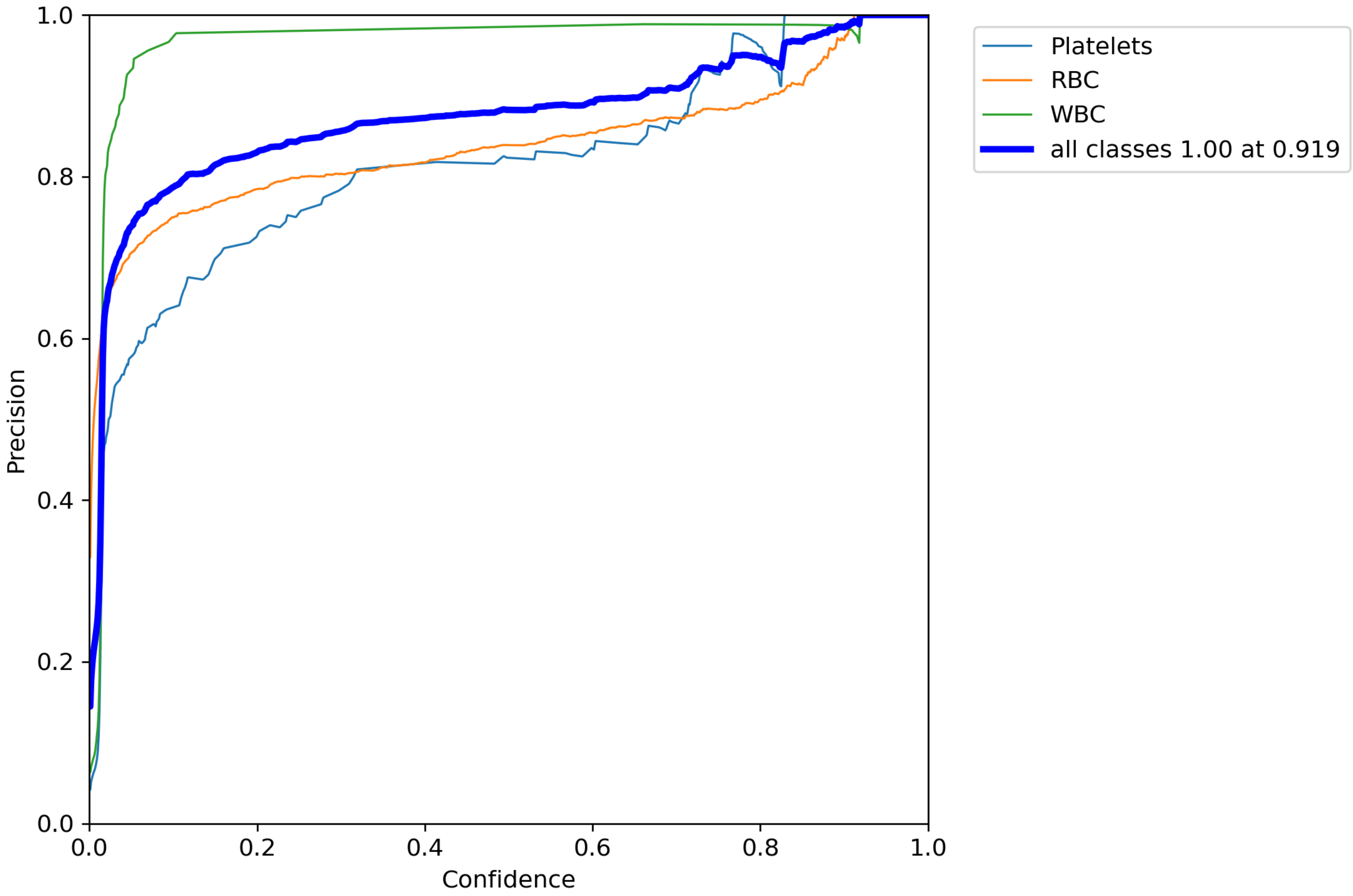
Testing Results

	Platelets	RBC	WBC
Ground Truth	69	805	71
Prediction	**85**	**1268**	**79**

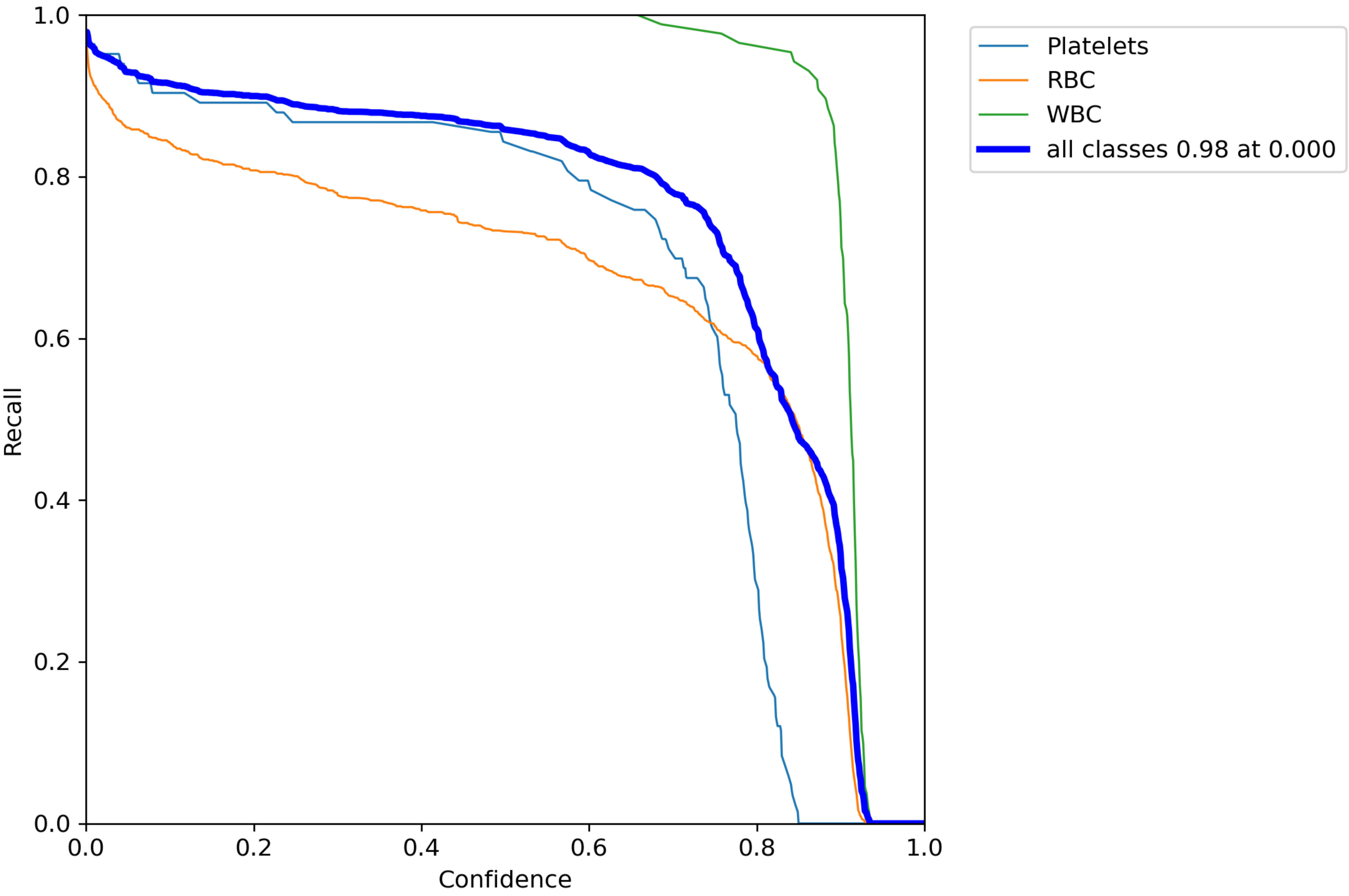
Confusion Matrix



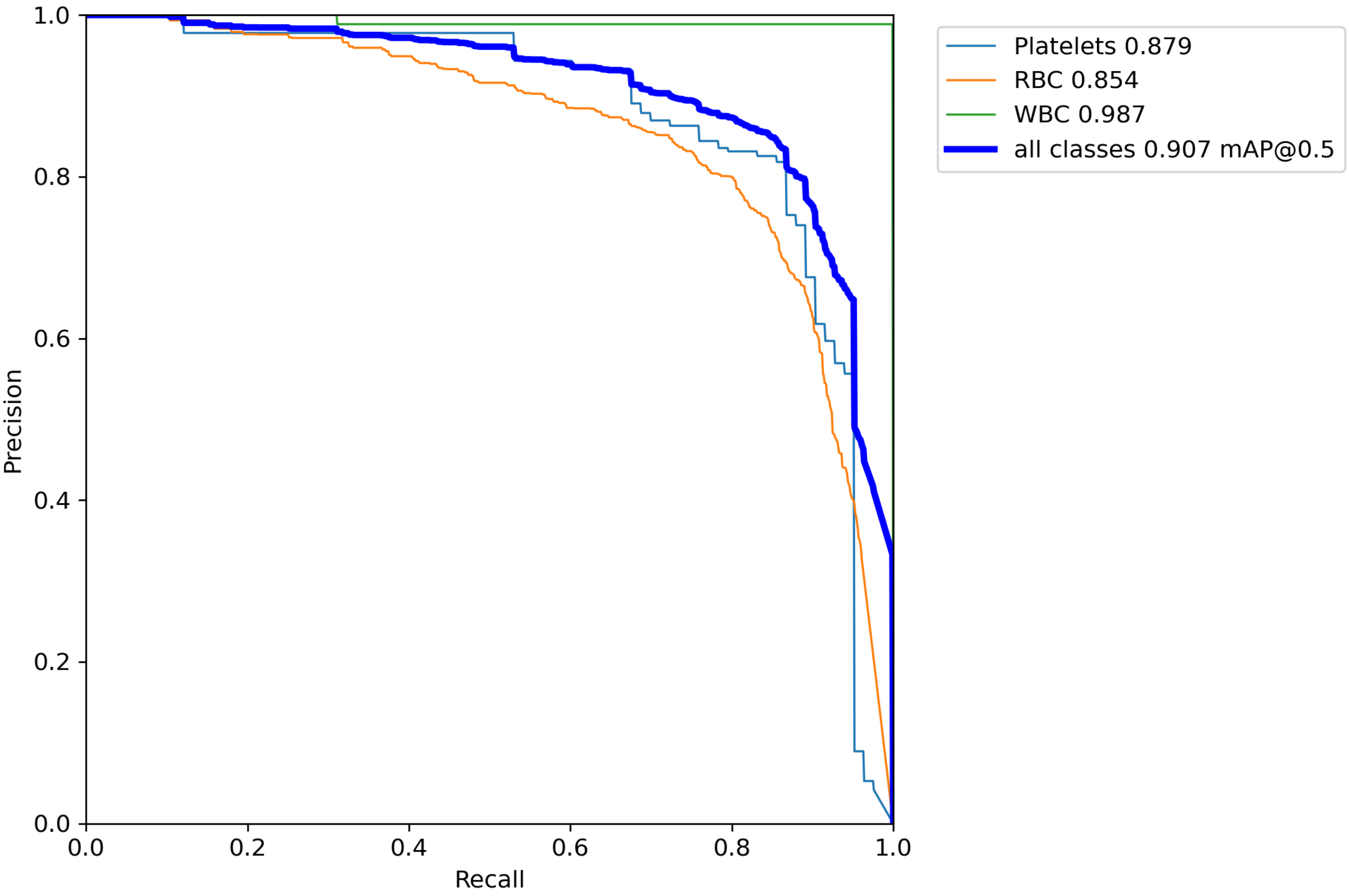
Precision (P) Curve



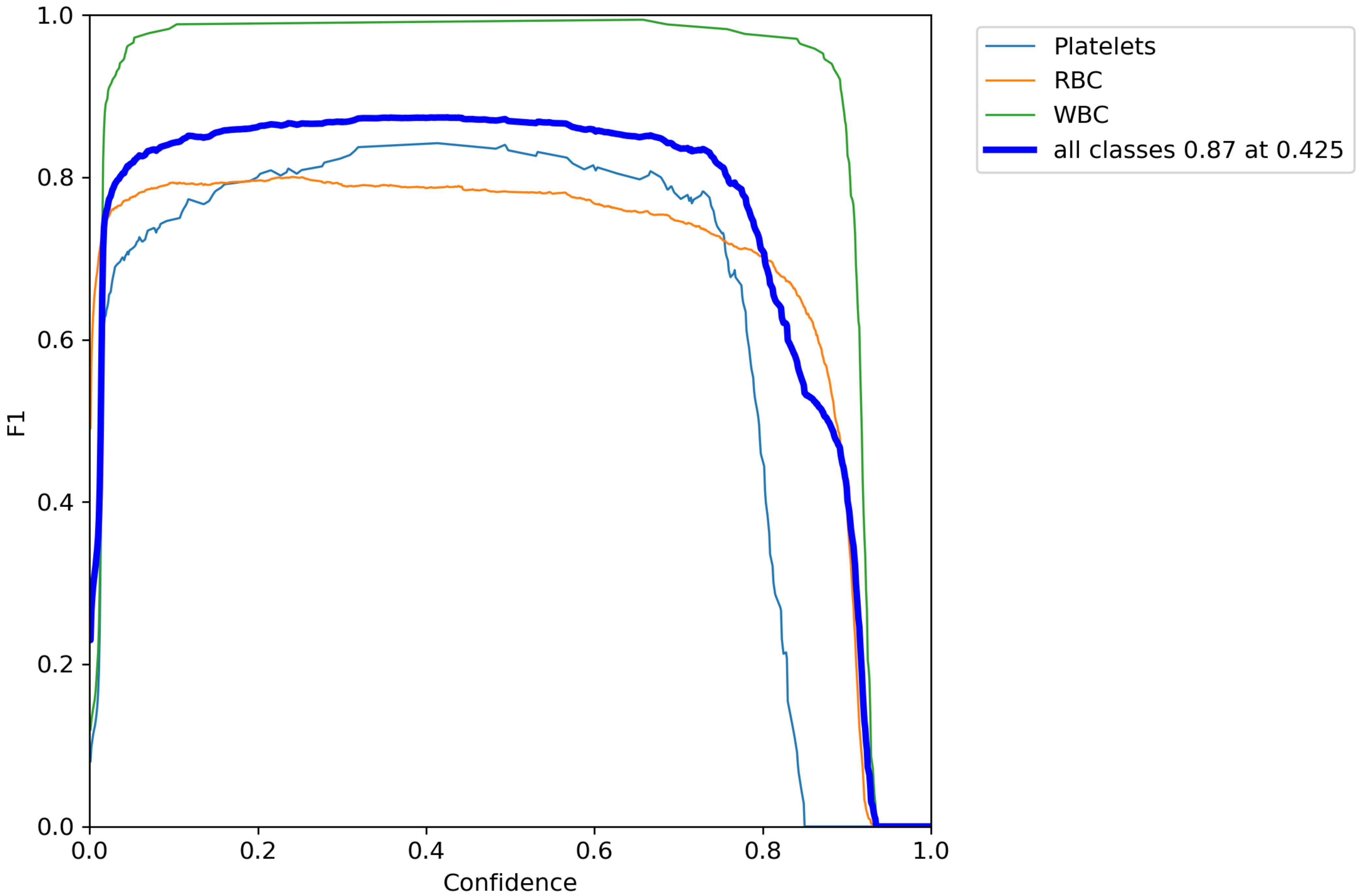
Recall (R) Curve



P-R Curve

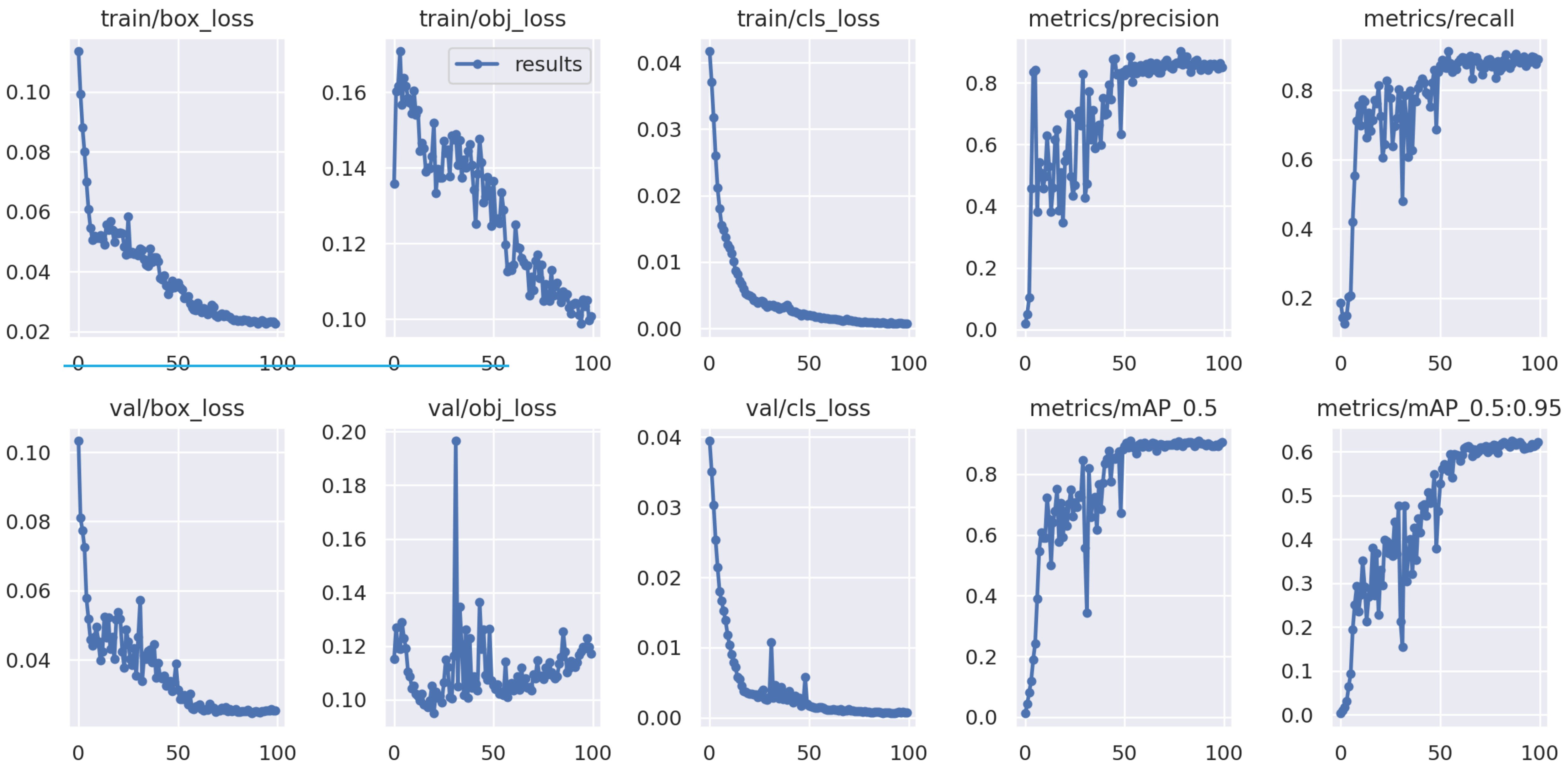

Platelets 0.879
RBC 0.854
WBC 0.987
all classes 0.907 mAP@0.5
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

F1 Curve



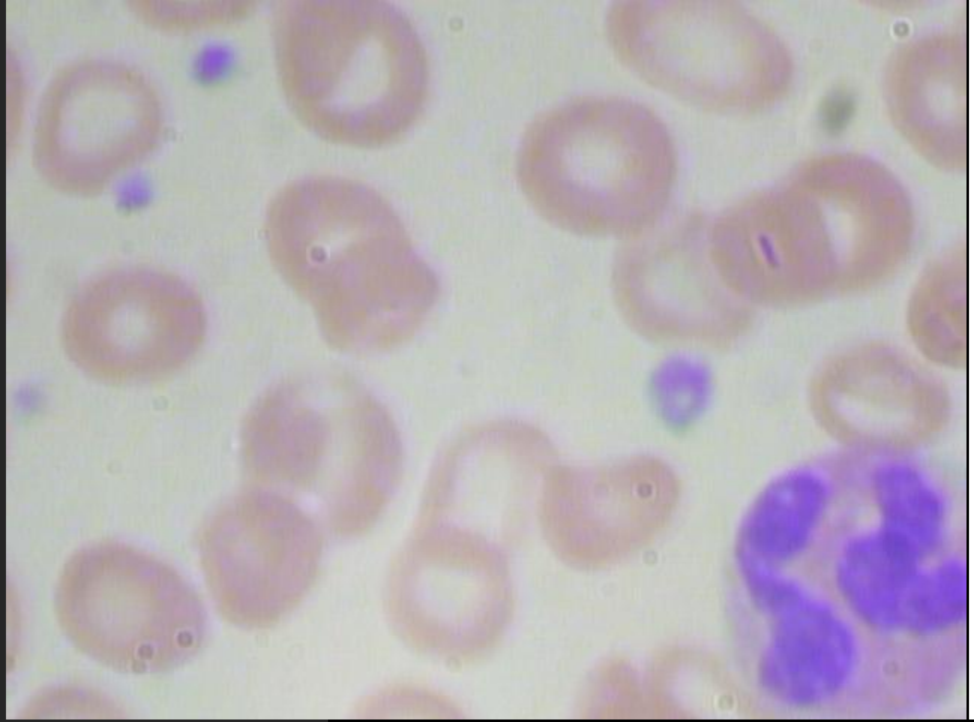
1.0
0.8
0.6
0.4
0.2
0.0
F1
0.0
0.2
0.4
0.6
0.8
1.0
Confidence
Platelets
RBC
WBC
all classes 0.87 at 0.425

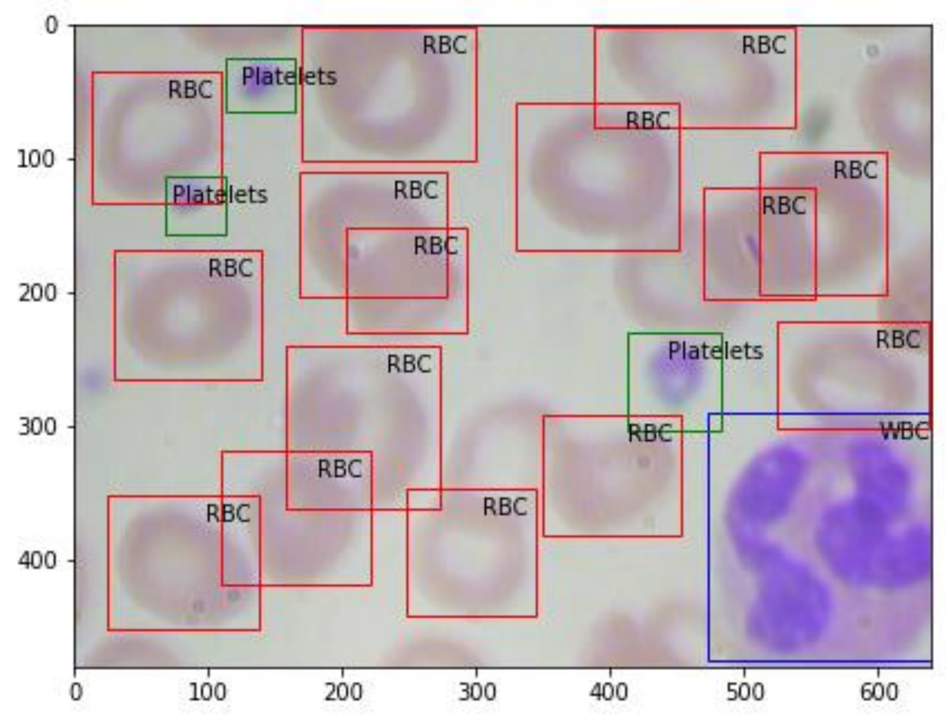
Results



train/box_loss
train/obj_loss
results
train/cls_loss
metrics/precision
metrics/recall
val/box_loss
val/obj_loss
val/cls_loss
metrics/mAP_0.5
metrics/mAP_0.5:0.95

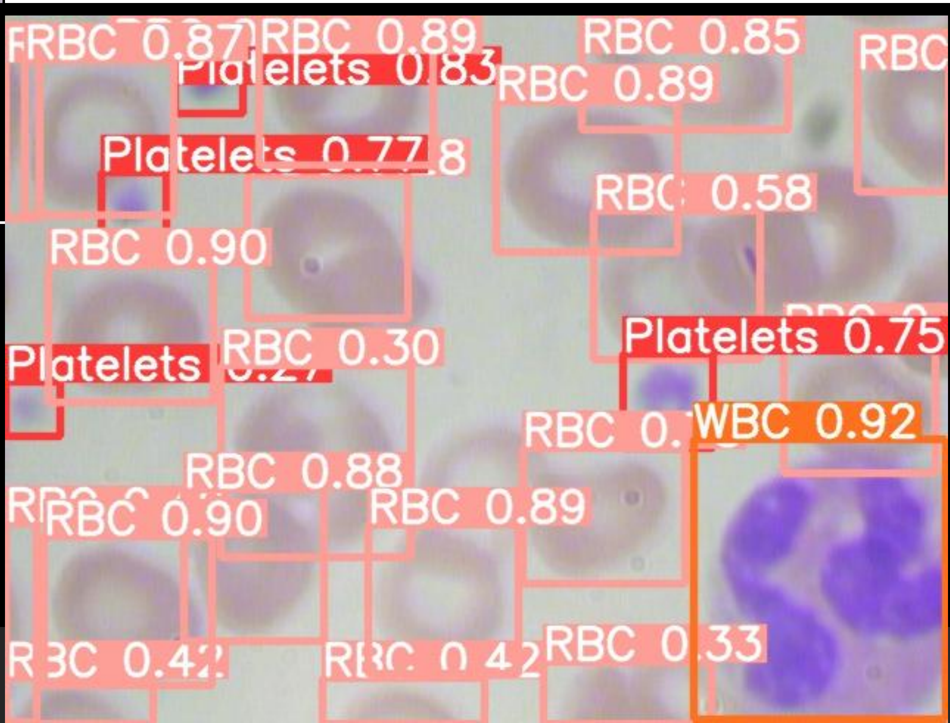
Comments on Platelets & RBC counts



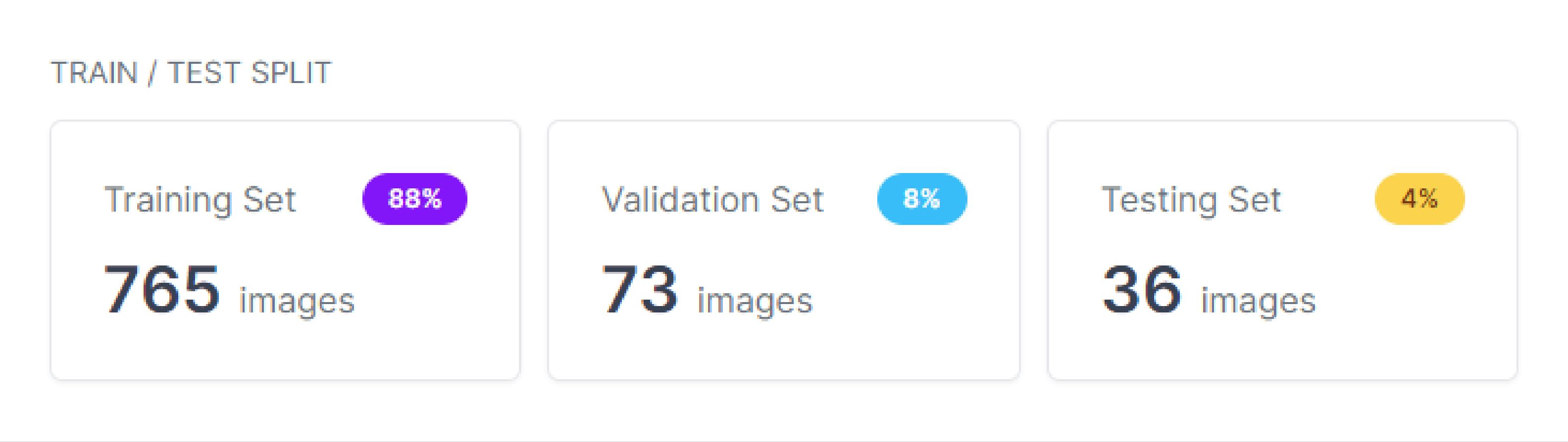


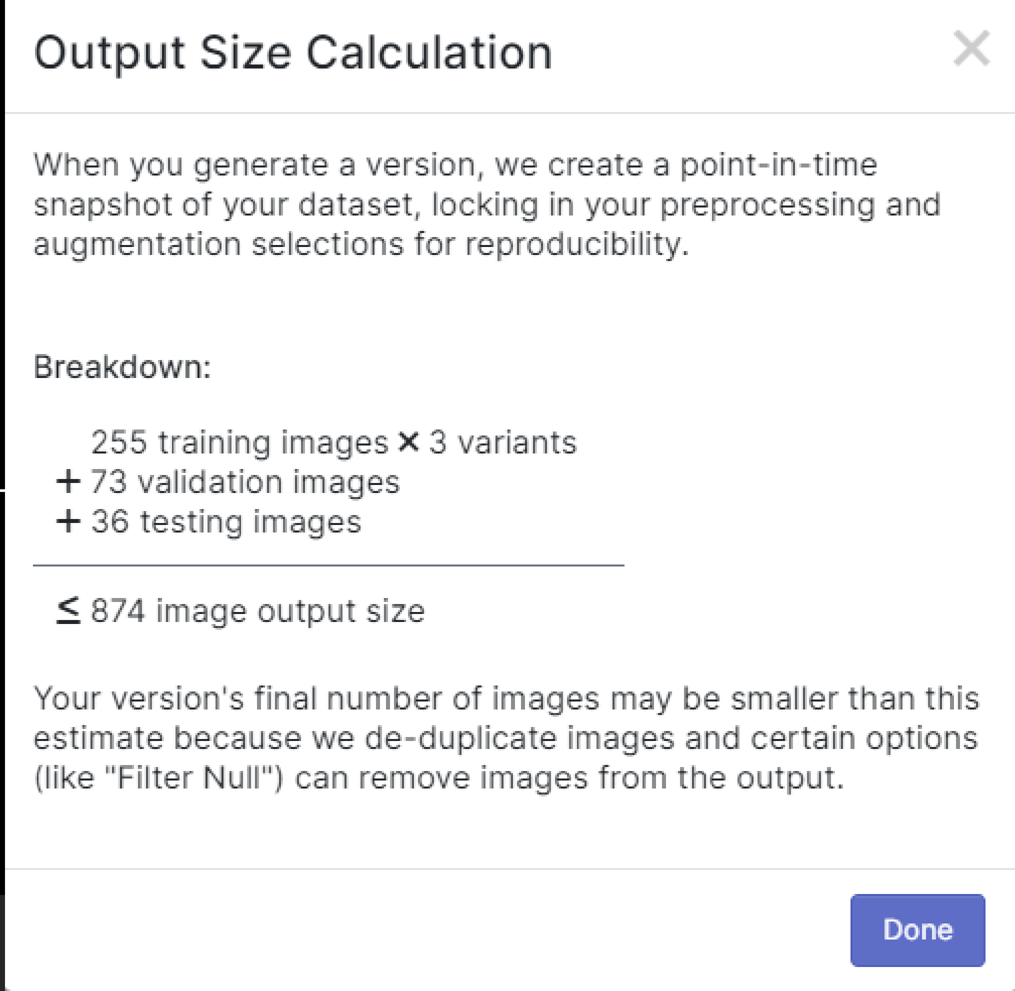
	Platelets	RBC	WBC
Ground Truth	03	15	01
Prediction	**04**	**20**	**01**

	Platelets	RBC	WBC
Ground Truth	69	805	71
Prediction	**85**	**1268**	**79**



Data Augmentation





TRAINING

YOLO V5

100 epochs

$mAP@.5:.95 \rightarrow 64.2\ \%$

```
100 epochs completed in 1.042 hours.
Optimizer stripped from yolov5/runs/train/BCCM/weights/last.pt, 14.3MB
Optimizer stripped from yolov5/runs/train/BCCM/weights/best.pt, 14.3MB

Validating yolov5/runs/train/BCCM/weights/best.pt...
Fusing layers...
Model Summary: 213 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 5/5
                 all         73        967      0.875      0.905      0.925      0.642
           Platelets         73         76      0.882      0.881      0.908      0.507
                 RBC         73        819      0.776      0.835      0.885      0.622
                 WBC         73         72      0.967          1      0.983      0.796
Results saved to yolov5/runs/train/BCCM
CPU times: user 45.1 s, sys: 6.88 s, total: 52 s
Wall time: 1h 3min 17s
```

Testing Results

	Platelets	RBC	WBC
Ground Truth	36	398	37
Prediction	45	692	40

Transfer Learning

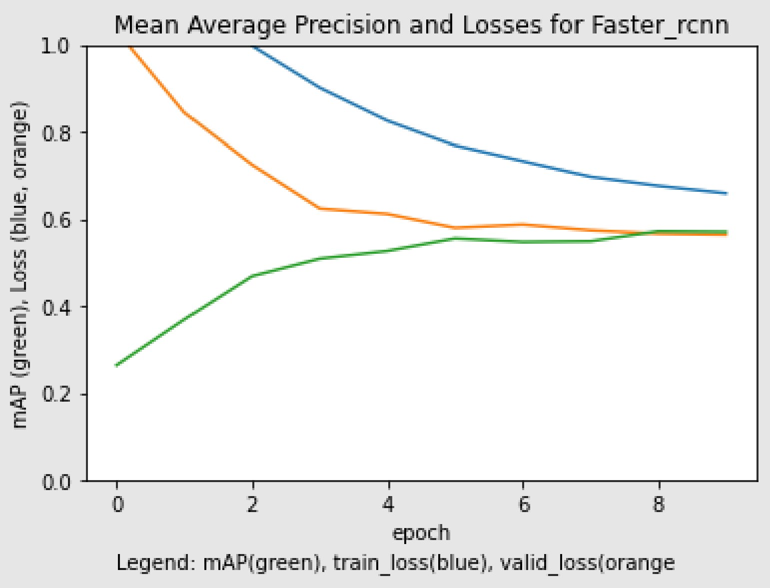
Comparison with other state-of-the-art Object Detection Models

```
model_type_yolo = models.ultralytics.yolov5
backbone_yolo = model_type_yolo.backbones.small
model_yolo = model_type_yolo.model(
             backbone = backbone_yolo(pretrained=True),
             num_classes=len(parser.class_map), img_size = size)
train_dl_yolo = model_type_yolo.train_dl(train_ds,
                      batch_size=16, num_workers=4, shuffle=True)
valid_dl_yolo = model_type_yolo.valid_dl(valid_ds,
                      batch_size=16, num_workers=4, shuffle=False)

learn_yolo = model_type_yolo.fastai.learner(
                     dls=[train_dl_yolo, valid_dl_yolo],
                     model=model_yolo, metrics=metrics)
learn_yolo.lr_find()
learn_yolo.fine_tune(10, 1e-2 , freeze_epochs=1)
plot_metrics(learn_yolo,
             'Mean Average Precision and Losses for YOLOv5')
```

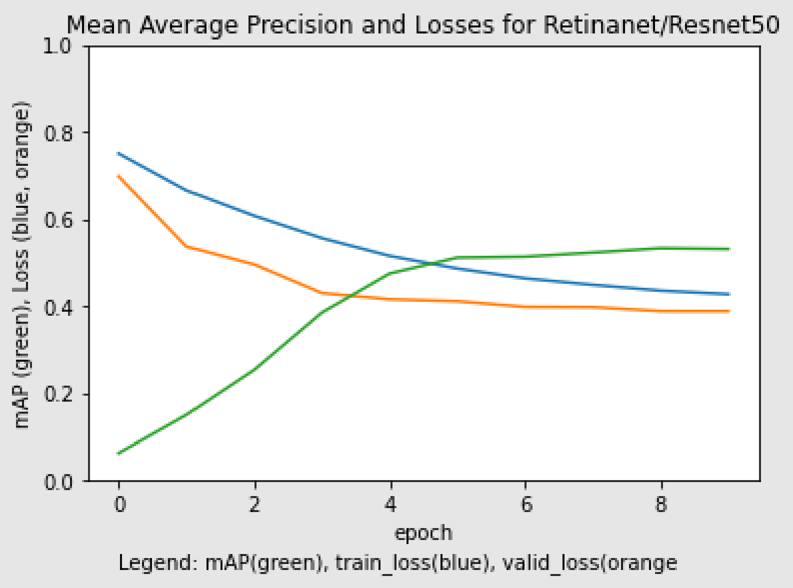
FASTER R-CNN

epoch	train_loss	valid_loss	COCOMetric	time
0	1.421260	1.033804	0.265108	01:11
1	1.162679	0.845835	0.370371	00:58
2	0.998871	0.725854	0.470032	00:58
3	0.902615	0.625586	0.510287	00:57
4	0.827155	0.613033	0.528075	00:58
5	0.770260	0.581217	0.556863	00:57
6	0.734273	0.588839	0.548724	00:57
7	0.698556	0.575590	0.550385	00:57
8	0.677905	0.567733	0.573396	00:57
9	0.660749	0.565578	0.572065	00:57



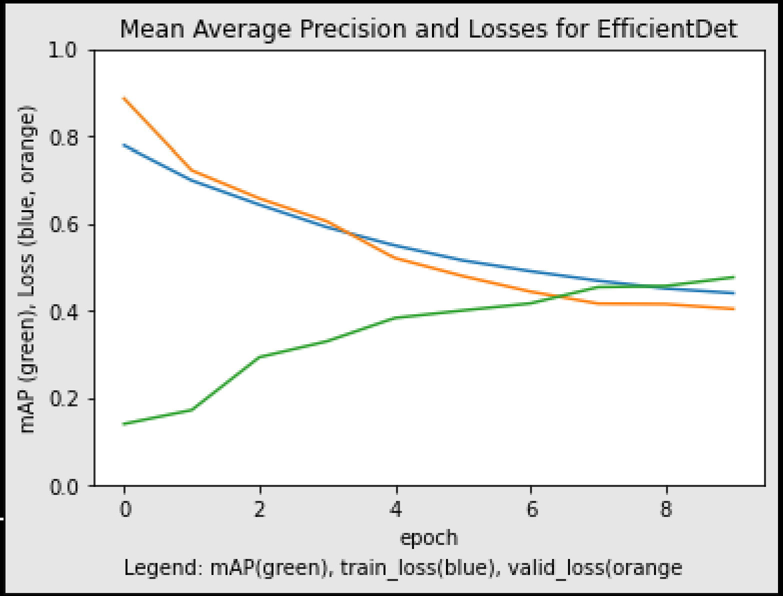
Retina Net

epoch	train_loss	valid_loss	COCOMetric	time
0	0.752289	0.699699	0.063327	00:39
1	0.667371	0.538277	0.151133	00:38
2	0.608886	0.496884	0.254696	00:38
3	0.557438	0.431085	0.386364	00:37
4	0.516701	0.416808	0.476085	00:37
5	0.487569	0.412279	0.512709	00:37
6	0.464997	0.399387	0.514859	00:37
7	0.450095	0.398680	0.524334	00:37
8	0.436650	0.389633	0.534338	00:38
9	0.428610	0.389472	0.532546	00:37



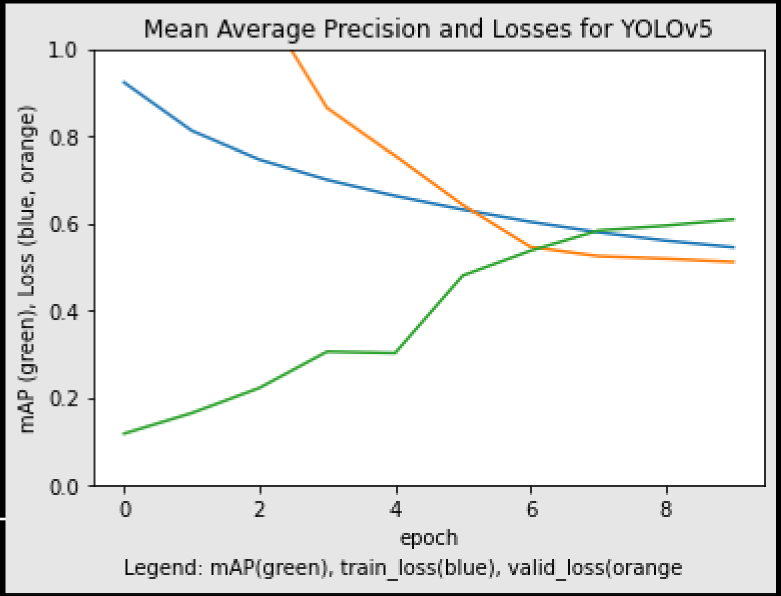
EfficientDet

epoch	train_loss	valid_loss	COCOMetric	time
0	0.779975	0.887384	0.140874	00:25
1	0.700151	0.723331	0.172714	00:23
2	0.644590	0.658918	0.294107	00:22
3	0.592794	0.606164	0.330704	00:23
4	0.550768	0.521928	0.384231	00:23
5	0.516472	0.480586	0.401486	00:23
6	0.491730	0.444813	0.417749	00:23
7	0.469823	0.417321	0.455171	00:23
8	0.451999	0.416150	0.457720	00:23
9	0.441310	0.405147	0.477708	00:23



YOLO-V5

epoch	train_loss	valid_loss	COCOMetric	time
0	0.924407	1.139100	0.118237	00:15
1	0.813875	1.455061	0.165457	00:15
2	0.747731	1.117861	0.223106	00:14
3	0.701412	0.866094	0.306402	00:14
4	0.664457	0.756519	0.303205	00:14
5	0.632931	0.643673	0.481163	00:14
6	0.604743	0.547221	0.538219	00:14
7	0.580869	0.525873	0.584507	00:14
8	0.561514	0.519982	0.595998	00:14
9	0.546258	0.512514	0.610488	00:14



Comparison

Model	LR	mAP	Avg. time
Faster R-CNN	2e-04	0.57	57 sec
YOLO-V5	**1e-02**	**0.61**	**14 sec**
Retina Net	8e-05	0.53	37 sec
EfficientDet	1e-02	0.48	23 sec

Best Model YOLO-V5

41	0.389574	0.548379	0.622749	00:13
42	0.384600	0.549746	0.621227	00:14
43	0.380539	0.559521	0.624182	00:14
44	0.376359	0.552600	0.612266	00:14
45	0.371917	0.554087	0.622479	00:14
46	0.370168	0.553329	0.615118	00:14
47	0.369631	0.555480	0.619671	00:14
48	0.368694	0.554739	0.618568	00:14
49	0.366679	0.554078	0.619776	00:14

```
Better model found at epoch 0 with COCOMetric value: 0.11712814851744503.
Better model found at epoch 1 with COCOMetric value: 0.23050699305620143.
Better model found at epoch 2 with COCOMetric value: 0.3785265946222327.
Better model found at epoch 3 with COCOMetric value: 0.4446381264126172.
Better model found at epoch 5 with COCOMetric value: 0.46801126206937094.
Better model found at epoch 8 with COCOMetric value: 0.5182532786877971.
Better model found at epoch 12 with COCOMetric value: 0.5223814597991697.
Better model found at epoch 13 with COCOMetric value: 0.5452902918600457.
Better model found at epoch 17 with COCOMetric value: 0.5495939167804712.
Better model found at epoch 22 with COCOMetric value: 0.5847185619932614.
Better model found at epoch 24 with COCOMetric value: 0.5997363300132839.
Better model found at epoch 32 with COCOMetric value: 0.6171018943293074.
Better model found at epoch 38 with COCOMetric value: 0.6210552059955694.
Better model found at epoch 41 with COCOMetric value: 0.6227487698173754.
Better model found at epoch 43 with COCOMetric value: 0.6241815784979853.
```

Challenges

- Scarcity of Computational Resources (Google Colab Free Account Restrictions)
- Small Dataset
- Error Handling
- RBC, Platelets overlapping problem

THANK YOU