

## BÁO CÁO LAB04

**Tên: Phùng Nghĩa Phúc**

**MSV: 20127284**

**Môn: Xử lý ảnh số và video số**

**Lớp: 20TMGT01**

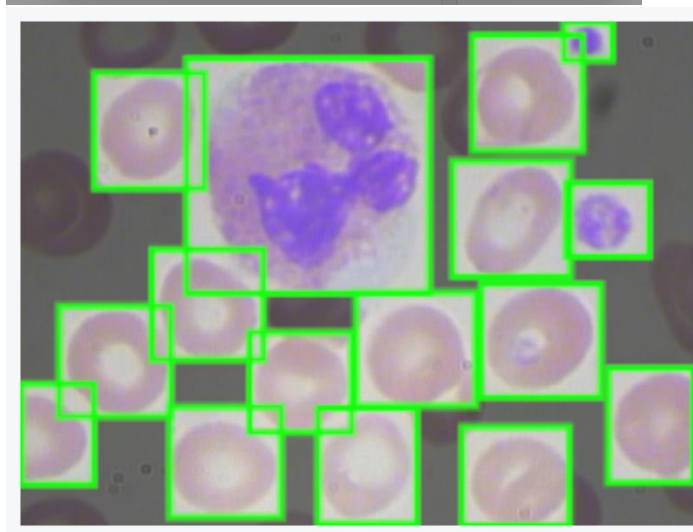
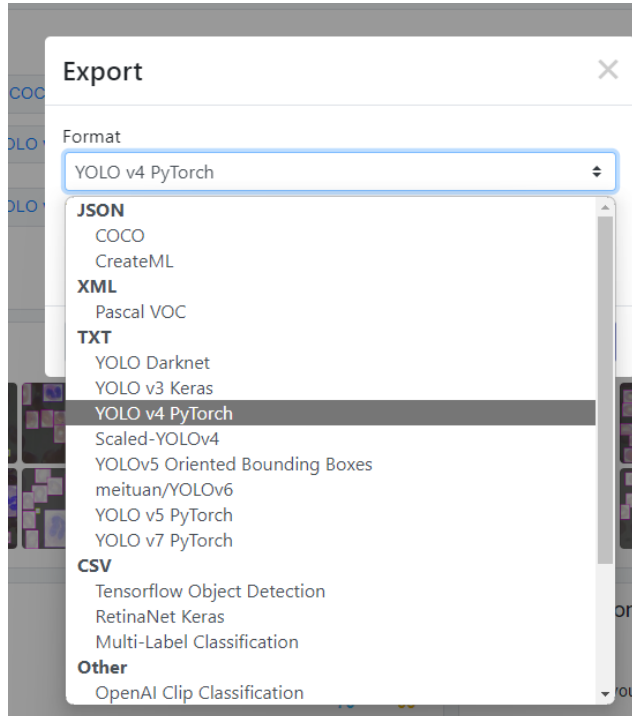
### MỤC LỤC

1. YOLOv4 Pytorch .....	2
2. YOLOv5 Pytorch .....	5
3. So sánh YOLOv4 Pytorch và YOLOv5 Pytorch.....	9

## 1. YOLOv4 Pytorch

### 1.1. Data

- Sử dụng data của roboflow: <https://public.roboflow.com/object-detection/bccd>



1.2. Repo: <https://github.com/roboflow-ai/pytorch-YOLOv4>

1.3. Clone Repo và cài đặt các thư viện trong requirements.txt

```
%cd /content/gdrive/My Drive
%mkdir colab
%cd /content/gdrive/My Drive/colab
!rm -rf pytorch-YOLOv4
!git clone https://github.com/roboflow-ai/pytorch-YOLOv4.git
%cd /content/gdrive/My Drive/colab/pytorch-YOLOv4

/ content/gdrive/My Drive
mkdir: cannot create directory 'colab': File exists
/content/gdrive/My Drive/colab
Cloning into 'pytorch-YOLOv4'...
remote: Enumerating objects: 413, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 413 (delta 4), reused 4 (delta 1), pack-reused 401
Receiving objects: 100% (413/413), 932.49 KiB | 6.86 MiB/s, done.
Resolving deltas: 100% (231/231), done.
/content/gdrive/My Drive/colab/pytorch-YOLOv4

[ ] %cd /content/gdrive/My Drive/colab/pytorch-YOLOv4
!pip install -r requirements.txt
```

#### 1.4. Weights

- Link tải file: <https://drive.google.com/u/0/uc?id=1fcbR0bWzYfIEdLJPzOsn4R5mlvR6IQyA>

```
[ ] # copy converted pre-trained weights
%cp /content/gdrive/My Drive/colab/weights/yolov4.conv.137.pth yolov4.conv.137.pth
```

#### 1.5. Train:

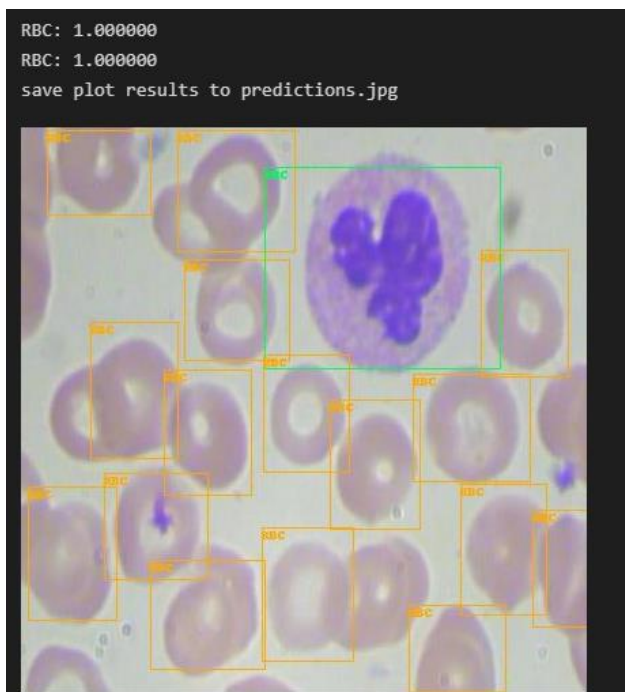
```
Epochs:      100
Batch size:   2
Subdivisions: 1
Learning rate: 0.001
Training size: 255
Validation size: 73
Checkpoints:  True
Device:      cuda
Images size: 608
Optimizer:   adam
Dataset classes: 3
Train label path: train.txt
Pretrained:
```

#### 1.6. Kiểm tra các file có trong thư mục Checkpoints:

```
Yolov4_epoch100.pth  Yolov4_epoch30.pth  Yolov4_epoch55.pth  Yolov4_epoch75.pth
Yolov4_epoch10.pth   Yolov4_epoch35.pth  Yolov4_epoch5.pth   Yolov4_epoch80.pth
Yolov4_epoch15.pth   Yolov4_epoch40.pth  Yolov4_epoch60.pth  Yolov4_epoch85.pth
Yolov4_epoch20.pth   Yolov4_epoch45.pth  Yolov4_epoch65.pth  Yolov4_epoch90.pth
Yolov4_epoch25.pth   Yolov4_epoch50.pth  Yolov4_epoch70.pth  Yolov4_epoch95.pth
```

#### 1.7. Kết quả:

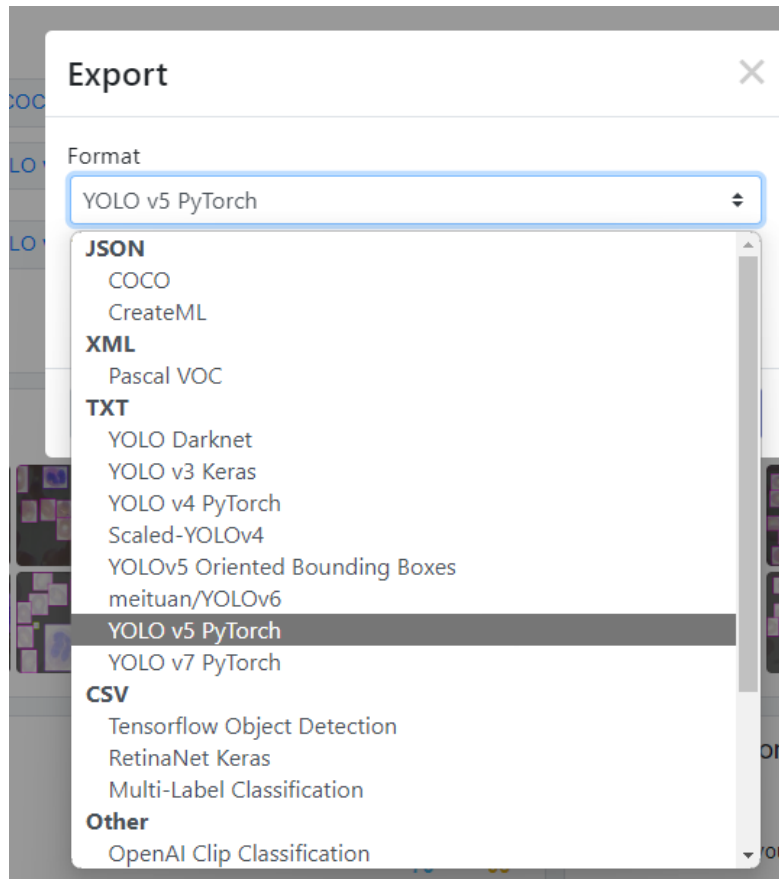
- Chọn yolov4\_epoch100.pth để tìm kiếm kết quả



## 2. YOLOv5 Pytorch

### 2.1. Data

- Sử dụng dữ liệu của roboflow: <https://public.roboflow.com/object-detection/bccd/4/download/yolov5pytorch>



```
%cd /content/yolov5
#after following the link above, recieve python code with these fields filled in
#from roboflow import Roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="umdCNCg6EExbVv6aSMpp")
project = rf.workspace("joseph-nelson").project("bccd")
dataset = project.version(4).download("yolov5")

/content/yolov5
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in BCCD-4 to yolov5pytorch: 100% [13320855 / 13320855] bytes
Extracting Dataset Version Zip to BCCD-4 in yolov5pytorch:: 100% [1760/1760 [00:00<00:00, 2404.55it/s]
```

### 2.2. Repo

- Sử dụng repo: <https://github.com/ultralytics/yolov5>

```
# clone YOLOv5 repository
!git clone https://github.com/ultralytics/yolov5 # clone repo
%cd yolov5
!git reset --hard fbe67e465375231474a2ad80a4389efc77ecff99

Cloning into 'yolov5'...
remote: Enumerating objects: 14513, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 14513 (delta 19), reused 25 (delta 14), pack-reused 14477
Receiving objects: 100% (14513/14513), 13.60 MiB | 33.09 MiB/s, done.
Resolving deltas: 100% (9998/9998), done.
/content/yolov5
HEAD is now at fbe67e4 Fix `OMP_NUM_THREADS=1` for macOS (#8624)
```

### 2.3. Cài đặt các thư viện cần thiết:

```
# install dependencies as necessary
!pip install -qr requirements.txt # install dependencies (ignore errors)
import torch

from IPython.display import Image, clear_output # to display images
from utils.downloads import attempt_download # to download models/datasets

# clear_output()
print('Setup complete. Using torch %s %s' % (torch.__version__, torch.cuda.get_device_properties(0) if torch.cuda.is_available() else 'CPU'))

1.6 MB 5.1 MB/s
Setup complete. Using torch 1.13.0+cu116 _CudaDeviceProperties(name='Tesla T4', major=7, minor=5, total_memory=15109MB, multi_processor_count=40)
```

### 2.4. Cài đặt roboflow và tải xuống dataset

```
#follow the link below to get your download code from from Roboflow
!pip install -q roboflow
from roboflow import Roboflow
rf = Roboflow(model_format="yolov5", notebook="roboflow-yolov5")

42 kB 835 kB/s
67 kB 3.7 MB/s
145 kB 30.5 MB/s
138 kB 73.4 MB/s
178 kB 73.5 MB/s
54 kB 3.4 MB/s
62 kB 948 kB/s

Building wheel for wget (setup.py) ... done
upload and label your dataset, and get an API KEY here: https://app.roboflow.com/?model=yolov5&ref=roboflow-yolov5

[5] %cd /content/yolov5
#after following the link above, recieve python code with these fields filled in
#from roboflow import Roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="umdcNCg6EExbVv6aSMpp")
project = rf.workspace("joseph-nelson").project("bccd")
dataset = project.version(4).download("yolov5")

/content/yolov5
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in BCCD-4 to yolov5pytorch: 100% [13320855 / 13320855] bytes
Extracting Dataset Version Zip to BCCD-4 in yolov5pytorch:: 100% [1760/1760 [00:00<00:00, 2404.55it/s]
```

## 2.5. Train

- Epochs: 100
- Batch: 2

```

%# time its performance
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 2 --epochs 100 --data {dataset.location}/data.yaml --cfg ./models/custom_yolov5s.yaml --weights '' --name

```

Epoch	gpu_mem	box	obj	cls	labels	img_size	416: 100%	383/383	[00:39<00:00, 9.80it/s]
37/99	0.361G	0.03739	0.1228	0.001984	15				
	Class	Images	Labels	P	R		mAP@.5	mAP@.5:.95:	100% 19/19 [00:01<00:00, 16.45it/s]
	all	73	967	0.84	0.905		0.906	0.562	
Epoch	gpu_mem	box	obj	cls	labels	img_size	416: 100%	383/383	[00:40<00:00, 9.49it/s]
38/99	0.361G	0.03691	0.1208	0.001637	30				
	Class	Images	Labels	P	R		mAP@.5	mAP@.5:.95:	100% 19/19 [00:01<00:00, 16.89it/s]
	all	73	967	0.835	0.897		0.907	0.552	

Epoch	gpu_mem	box	obj	cls	labels	img_size	416: 100%	383/383	[00:39<00:00, 9.80it/s]
97/99	0.361G	0.03154	0.1121	0.001122	21				
	Class	Images	Labels	P	R		mAP@.5	mAP@.5:.95:	100% 19/19 [00:01<00:00, 17.67it/s]
	all	73	967	0.849	0.9		0.918	0.615	
Epoch	gpu_mem	box	obj	cls	labels	img_size	416: 100%	383/383	[00:38<00:00, 9.93it/s]
98/99	0.361G	0.03111	0.1101	0.00093	11				
	Class	Images	Labels	P	R		mAP@.5	mAP@.5:.95:	100% 19/19 [00:01<00:00, 17.71it/s]
	all	73	967	0.847	0.902		0.916	0.617	
Epoch	gpu_mem	box	obj	cls	labels	img_size	416: 100%	383/383	[00:38<00:00, 9.94it/s]
99/99	0.361G	0.03126	0.1107	0.000908	21				
	Class	Images	Labels	P	R		mAP@.5	mAP@.5:.95:	100% 19/19 [00:01<00:00, 17.13it/s]
	all	73	967	0.855	0.895		0.915	0.615	

100 epochs completed in 1.139 hours.  
Optimizer stripped from runs/train/yolov5s\_results/weights/last.pt, 14.8MB  
Optimizer stripped from runs/train/yolov5s\_results/weights/best.pt, 14.8MB

Validating runs/train/yolov5s\_results/weights/best.pt...  
Fusing layers...

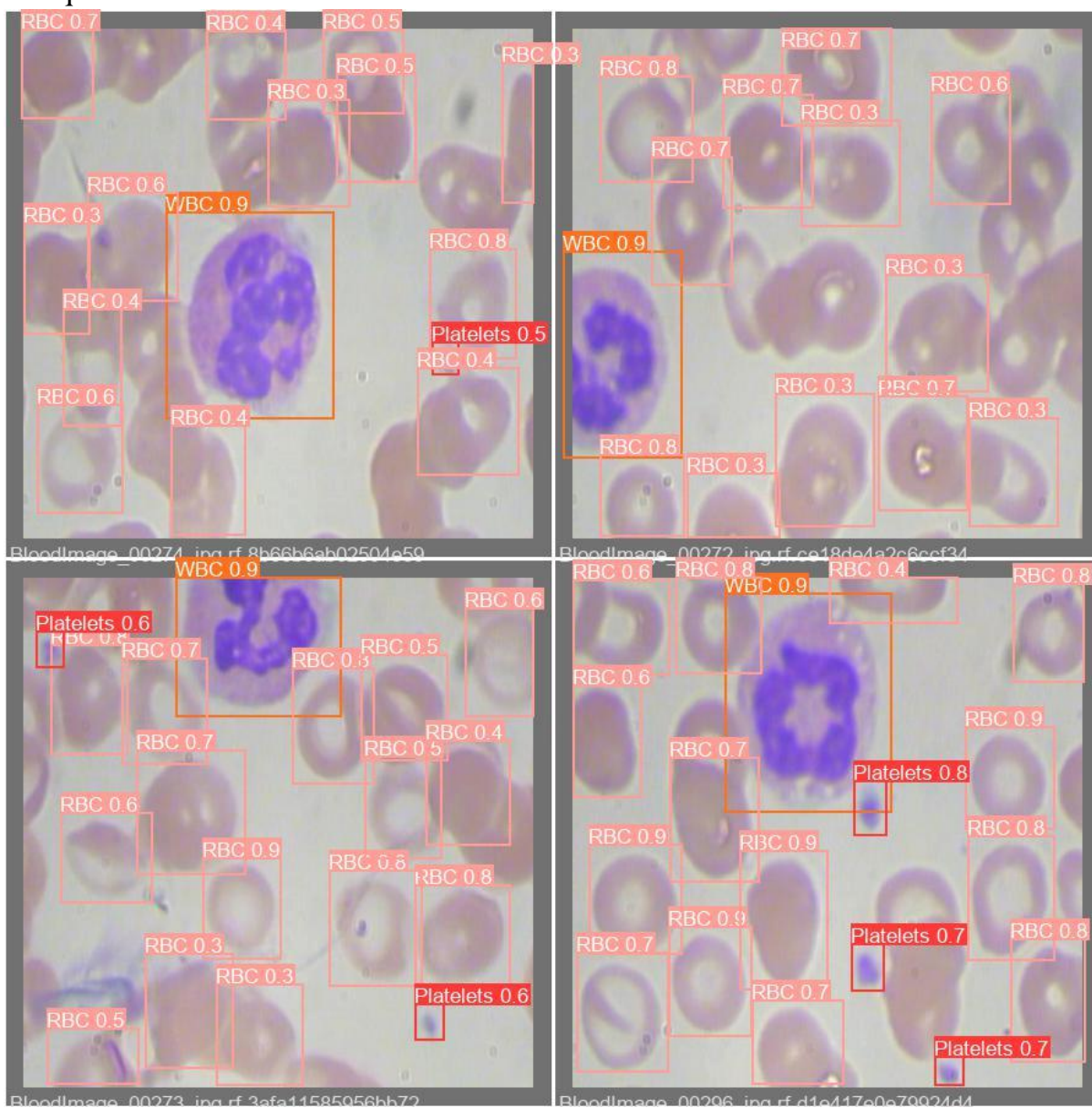
custom\_YOLOv5s summary: 232 layers, 7251912 parameters, 0 gradients, 16.8 GFLOPs

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:	100%	19/19	[00:01<00:00, 10.56it/s]
all	73	967	0.86	0.891	0.918	0.618			
Platelets	73	76	0.784	0.947	0.897	0.485			
RBC	73	819	0.83	0.725	0.871	0.606			
WBC	73	72	0.966	1	0.985	0.763			

Results saved to runs/train/yolov5s\_results  
CPU times: user 1min 1s, sys: 6.86 s, total: 1min 8s  
Wall time: 1h 8min 58s



## 2.6. Kết quả





### 3. So sánh YOLOv4 Pytorch và YOLOv5 Pytorch

- Quá trình train của yolov5 nhanh hơn yolov4
- Kết quả predict ở yolov5 có kết quả chính xác hơn so với yolov4: Các đối tượng ít bị miss hơn và xác suất predict cao hơn.
- Ở yolov5 và yolov4 việc thay đổi epochs cho kết quả chính xác hơn nhưng việc xử lý sẽ lâu hơn
- Ở 2 mô hình yolov4 và yolov5 còn hạn chế ở một số mô hình không detect được. Có thể là do dữ liệu và các tham số sử dụng chưa hợp lý.