

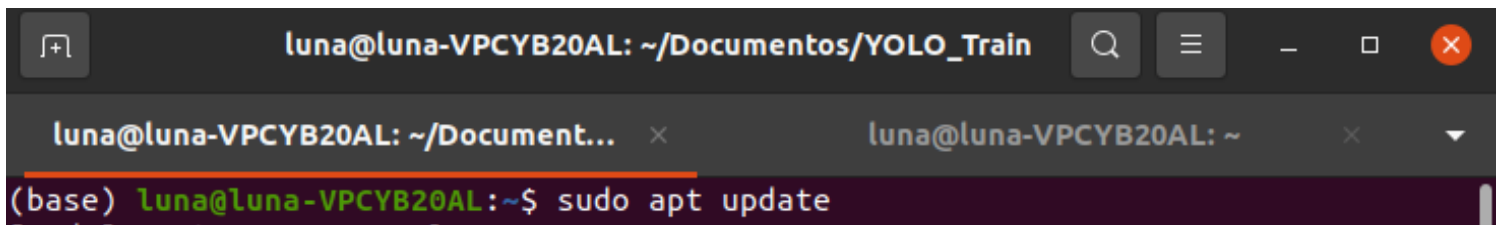
ENTRENAMIENTO PARA LA DETECCIÓN DE PIEDRAS USANDO YOLOV3

Para empezar nuestro entrenamiento de detección de las piedras vamos a clonar y guiarnos del siguiente repositorio

<https://github.com/puigalex/deteccion-objetos-video.git>

Pero antes de clonar nuestro repositorio, tendremos que actualizar nuestra maquina usando el siguiente comando

```
sudo apt update
```

A screenshot of a terminal window. The title bar shows 'luna@luna-VPCYB20AL: ~/Documentos/YOLO_Train'. The terminal has two tabs: 'luna@luna-VPCYB20AL: ~/Document...' and 'luna@luna-VPCYB20AL: ~'. The active tab shows the command '(base) luna@luna-VPCYB20AL:~\$ sudo apt update' being entered.

Posterior a ello, abrimos nuestra carpeta de documento y creamos una nueva carpeta llamada “YOLO_Train” y la abrimos

```
mkdir YOLO_Train
```

y posterior a ello procedemos a clonar nuestro repositorio

```
git clone https://github.com/puigalex/deteccion-objetos-video.git
```

una vez creado eso, abrimos nuestra carpeta de detección de objetos

```
cd deteccion-objetos-video/
```

una vez abierta la carpeta, procedemos a crear un ambiente de python(en anaconda)

```
conda create -n yolotrain python=3.6
```

y una vez generado, activamos el ambiente

```
conda activate yolotrain
```

y posterior a ello metemos el siguiente comando

```
pip install -r requirements.txt
```

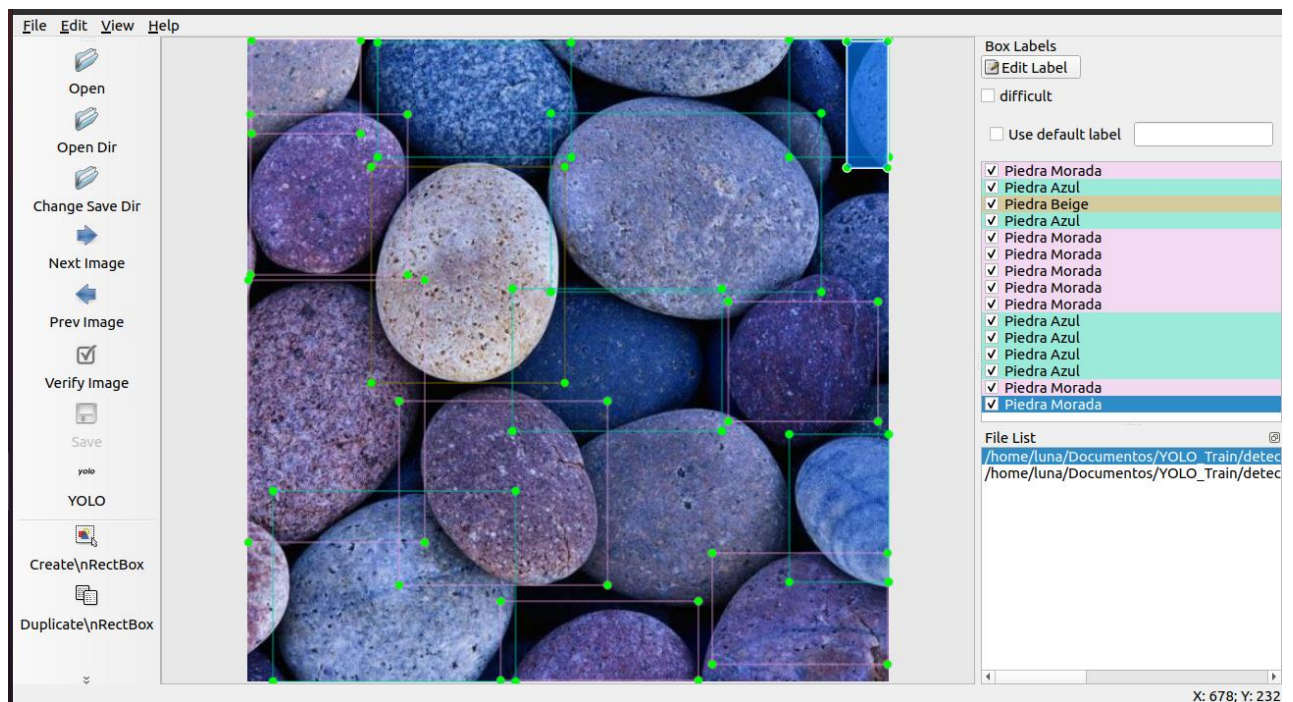
```
(base) luna@luna-VPCYB20AL:~/Documentos$ cd YOLO_Train
(base) luna@luna-VPCYB20AL:~/Documentos/YOLO_Train$ ls
(base) luna@luna-VPCYB20AL:~/Documentos/YOLO_Train$ git clone https://github.com
/puigalex/deteccion-objetos-video
Clonando en 'deteccion-objetos-video'...
remote: Enumerating objects: 115, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 115 (delta 16), reused 1 (delta 0), pack-reused 88
Recibiendo objetos: 100% (115/115), 211.69 KiB | 1.12 MiB/s, listo.
Resolviendo deltas: 100% (50/50), listo.
(base) luna@luna-VPCYB20AL:~/Documentos/YOLO_Train$ ls
deteccion-objetos-video
(base) luna@luna-VPCYB20AL:~/Documentos/YOLO_Train$ cd deteccion-objetos-video/
(base) luna@luna-VPCYB20AL:~/Documentos/YOLO_Train/deteccion-objetos-video$ ls
config  deteccion_video.py  models.py  requirements.txt  test.py  utils
data    detect.py           README.md  split_train_val.py  train.py  weights
(base) luna@luna-VPCYB20AL:~/Documentos/YOLO_Train/deteccion-objetos-video$ cond
a create -n yolotrain python=3.6
```

```
(base) luna@luna-VPCYB20AL:~/Documentos/YOLO_Train/deteccion-objetos-video$ cond
a activate yolotrain
(yolotrain) luna@luna-VPCYB20AL:~/Documentos/YOLO_Train/deteccion-objetos-video$
pip install -r requirements.txt
```

Una vez que se instale, metemos el siguiente comando

labellmg

esto es para que podamos etiquetar todas nuestras imágenes



Posterior a ello, abrimos nuestra carpeta

```
cd config/
```

```
bash create_custom_model.sh 4
```

```
cd ..
```

```
(yolotrain) Luna@Luna-VPCYB20AL:~/Documentos/YOLO_Train/deteccion-objetos-video$ cd config/
(yolotrain) Luna@Luna-VPCYB20AL:~/Documentos/YOLO_Train/deteccion-objetos-video/config$ bash create_custom_model.sh 4
(yolotrain) Luna@Luna-VPCYB20AL:~/Documentos/YOLO_Train/deteccion-objetos-video/config$ cd ..
```

Y de ahí nos vamos a la carpeta de weights

```
cd weights/
```

```
bash download_darknet.sh
```

```
cd ..
```

```
(yolotrain) Luna@Luna-VPCYB20AL:~/Documentos/YOLO_Train/deteccion-objetos-video$ cd weights/
(yolotrain) Luna@Luna-VPCYB20AL:~/Documentos/YOLO_Train/deteccion-objetos-video/weights$ bash download_darknet.sh
--2023-07-15 01:56:29-- https://pjreddie.com/media/files/darknet53.conv.74
Resolviendo pjreddie.com (pjreddie.com)... 128.208.4.108
Conectando con pjreddie.com (pjreddie.com)[128.208.4.108]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 162482580 (155M) [application/octet-stream]
Guardando como: "darknet53.conv.74"

darknet53.conv.74      100%[=====>] 154.96M  864KB/s   en 82s

2023-07-15 01:57:52 (1.90 MB/s) - "darknet53.conv.74" guardado [162482580/162482580]

(yolotrain) Luna@Luna-VPCYB20AL:~/Documentos/YOLO_Train/deteccion-objetos-video/weights$ cd ..
```

El archivo *data/custom/classes.names* debe contener el nombre de las clases, como fueron etiquetadas, un renglón por clase (en este caso vamos a sustituir las clases que vienen en ese archivo, por las creadas por nosotros) Posterior a ello, validamos nuestras clases con el siguiente comando

```
python split_train_val.py
```

y nos va a generar dos archivos *data/custom/valid.txt* y *data/custom/train.txt*

y en la carpeta de *config/custom.data* vamos a abrir el archivo y vamos a cambiar el número de clases poniendo el numero 4(que es el numero de clases a entrenar)

una vez hecho eso, vamos a meter el siguiente comando para que se empiece a entrenar nuestro modelo

```
python train.py --model_def config/yolov3-custom.cfg --
data_config config/custom.data --pretrained_weights
weights/darknet53.conv.74 --batch_size 1
```

una vez ya entrenado el modelo, metemos el siguiente comando, esto con la finalidad de poder correr nuestro modelo entrenado

```
python deteccion_video.py --model_def config/yolov3-custom.cfg
--checkpoint_model checkpoints/yolov3_ckpt_99.pth --class_path
data/custom/classes.names --weights_path
checkpoints/yolov3_ckpt_99.pth --conf_thres 0.85
```