Ciencia y Tecnología

Secretaría de Ciencia, Humanidades, Tecnología e Innovación

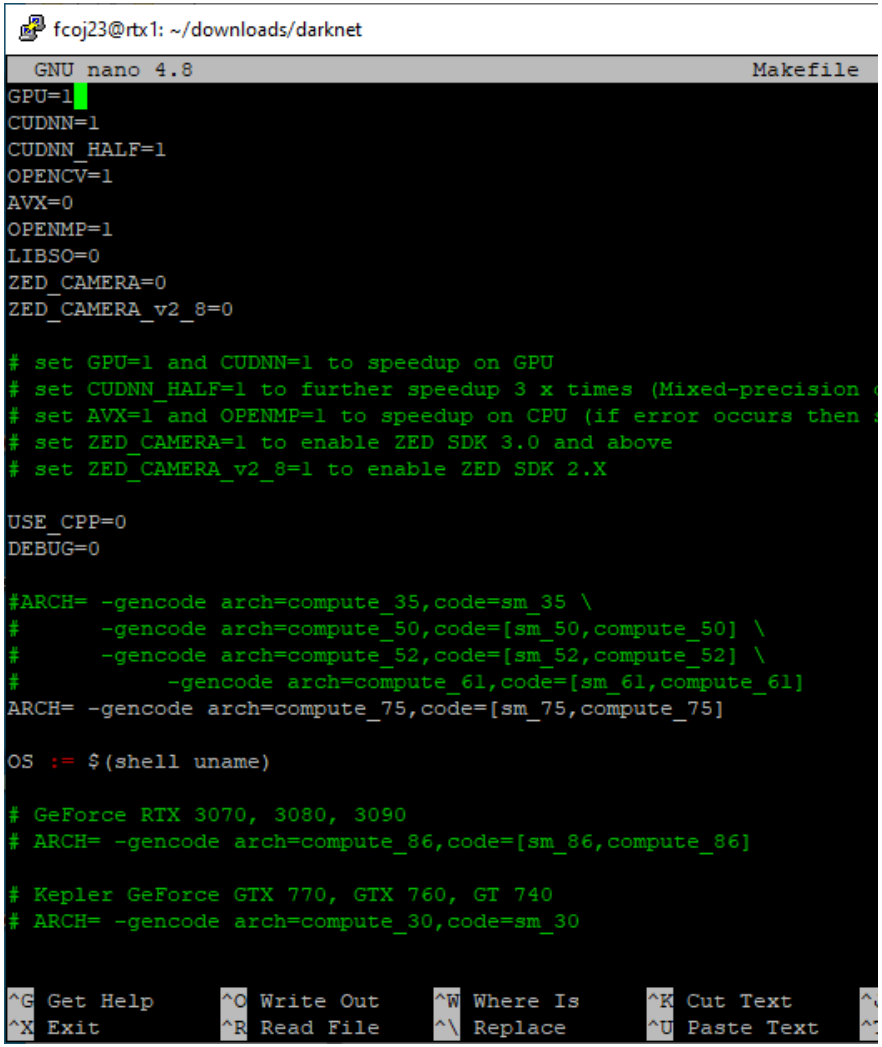
CIMAT
UNIDAD MÉRIDA

YOLO DARKNET & ULTRALYTICS

Dr. Francisco J. Hernández López
SECIHTI – CIMAT-Mérida
fcoj23@cimat.mx, www.cimat.mx/~fcoj23

INSTALAR DARKNET

- git clone https://github.com/AlexeyAB/darknet.git
- cd darknet/
- Modificar el archivo "Makefile"
- make



FACE-DETECTION-DATASET (KAGGLE)

https://www.kaggle.com/datasets/fareselmenshawii/face-detection-dataset

- El conjunto de datos se obtuvo empleando el kit de herramientas OIDv4, para extraer datos de Google Open Images
- Aprox. 4.43 GB
- Contiene 3 carpetas:
  - Images: Contiene las imágenes para entren. (13386) y valid. (3347)
  - Labels: Contiene las etiquetas en formato YOLO para las carpetas de entrenamiento y validación
  - Labels2: Contiene las etiquetas en otro formato

SUBCONJUNTO DEL FASE-DETECT-DATASET

- Vamos a tomar solo un subconjunto de este dataset:
  - Entrenamiento: 800 imágenes (80%)
  - Validación: 200 imágenes (20%)

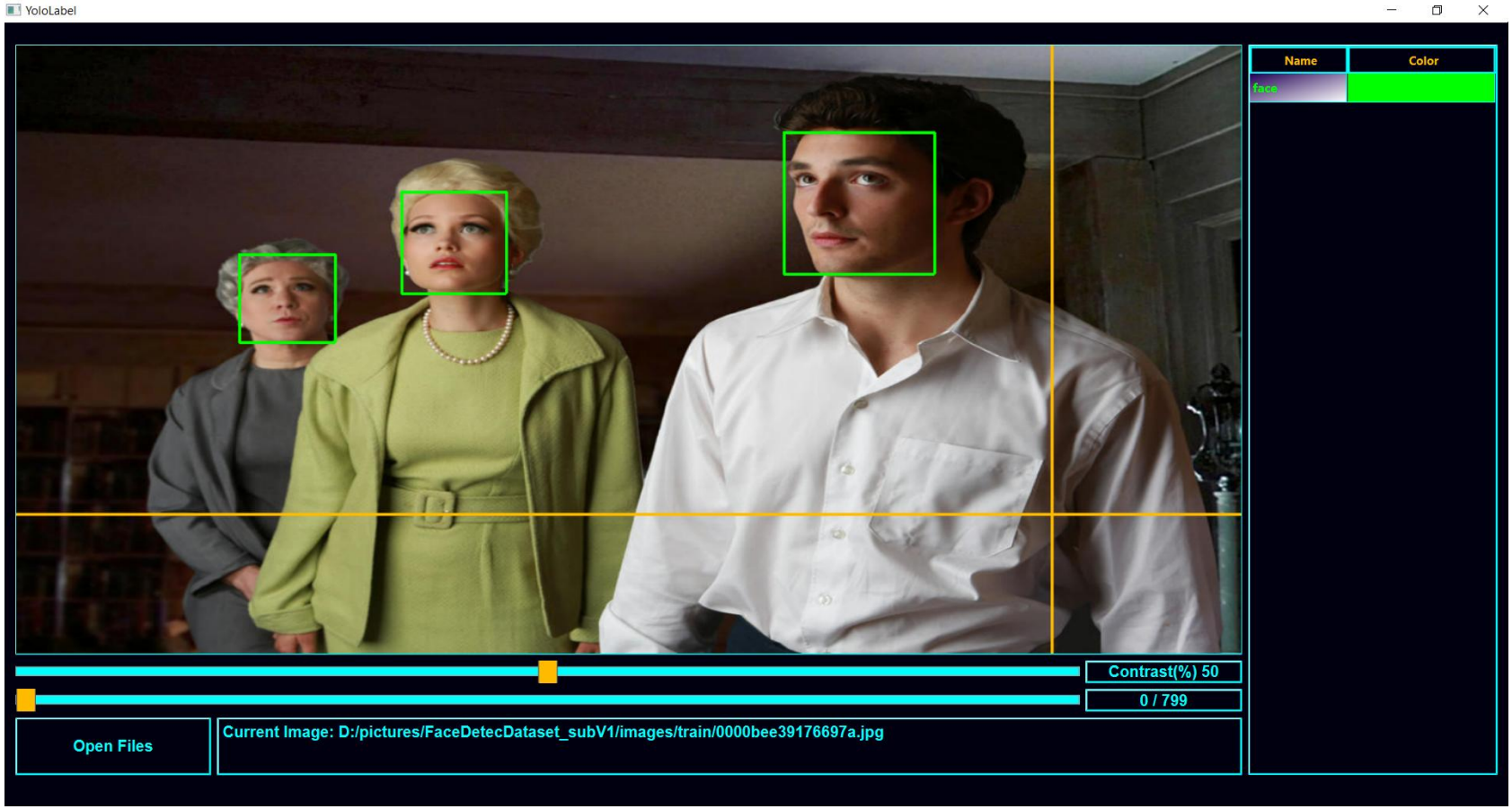
0000bee39176697a.txt - Notepad

File Edit Format View Help

```
0 0.220938 0.415139 0.078125 0.144494
0 0.356992 0.323941 0.085625 0.167401
0 0.687773 0.259155 0.123125 0.232599
```

YOLO-LABEL

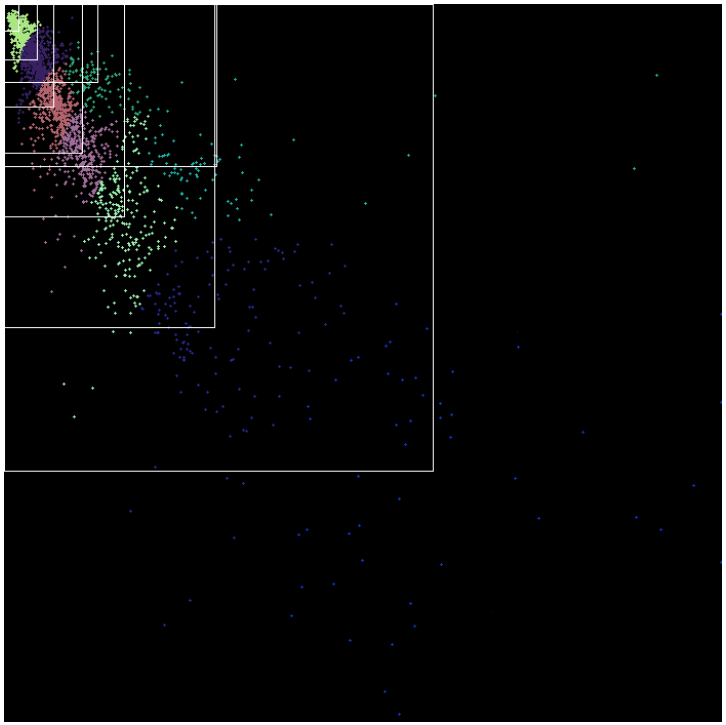
https://github.com/developer0hye/Yolo_Label



CALCULAR LAS CAJAS DE ANCLAJE

- **./darknet** detector **calc_anchors** FaceDetecDataset_subV1/obj.data -num_of_clusters 9 -width 608 -height 608 -show

num_of_clusters = 9, width = 608, height = 608
 read labels from 800 images
 loaded image: 800 box: 2327
 all loaded.
 calculating k-means++ ...
 iterations = 94
counters_per_class = 2327
 avg IoU = 69.79 %
Saving anchors to the file: anchors.txt
anchors = 13, 23, 28, 48, 42, 87, 80, 67, 66,126, 102,180, 180,138, 179,274, 364,396



MODIFICAR ARCHIVO .CFG DE YOLO

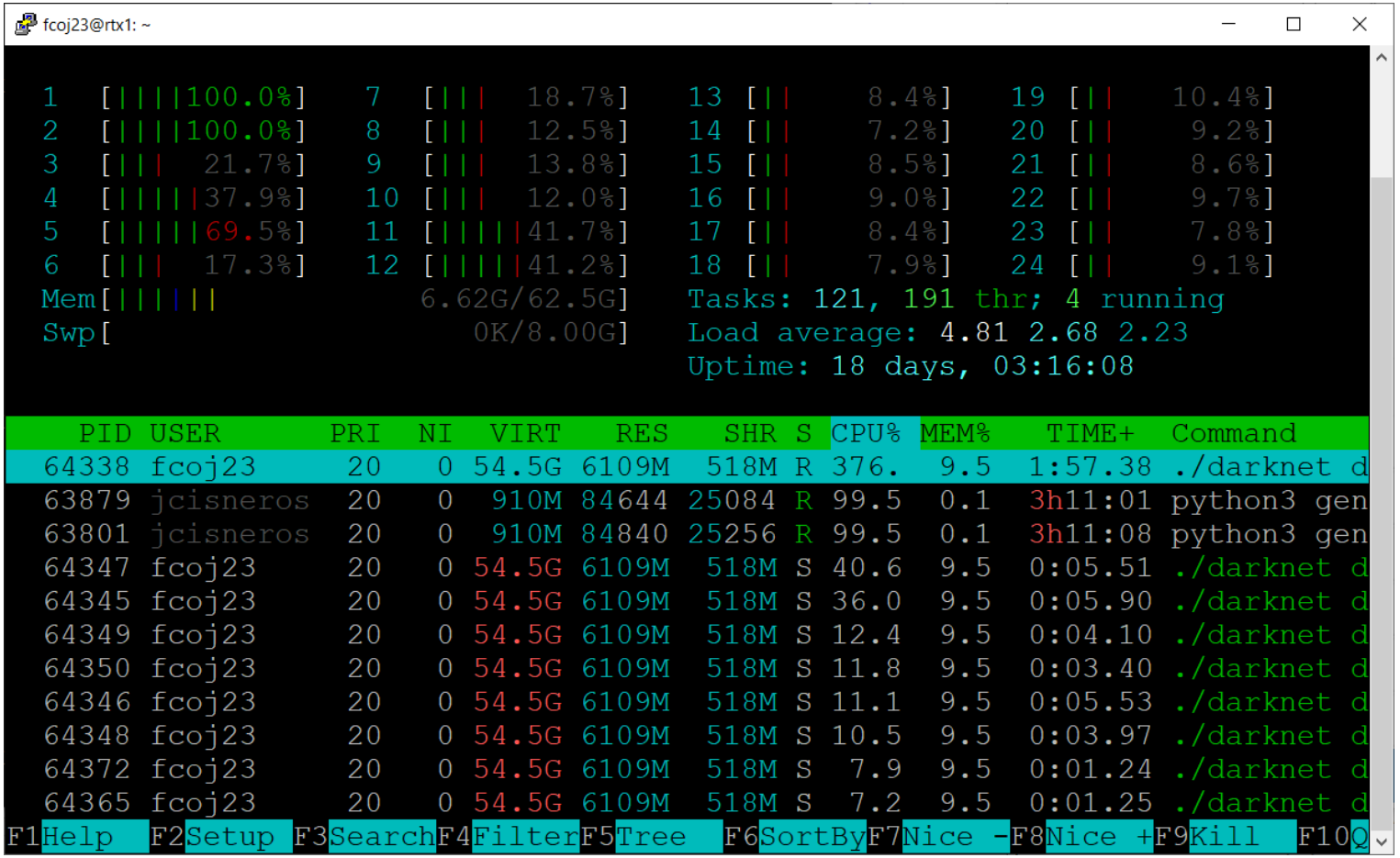
- **Batch=64** (Cantidad de imágenes utilizadas en el *Forward-Propag.* para calcular un gradiente y actualizar los pesos mediante *Back-Propag.*)
- **Subdivisions=16** (Depende de la RAM de la GPU, Batch/Subdivisons es el núm. de imágenes que se procesan en paralelo en la GPU)
- **max_batches=6000** (classes*2000 pero no menor que el num. de imag. de entrenam. y no menor que 6000), ej. max_batches=6000 si classes=3
- **Steps=4800,5400** es el 80% y el 90% de max_batches
- **Tamaño de entrada de la red (width=608, height=608)** debe ser múltiplo de 32
- **Classes=1** número de clases que hay que cambiar en cada uno de los tres [yolo]-layers del archivo
- **anchor=** 13, 23, 28, 48, 42, 87, 80, 67, 66,126, 102,180, 180,138, 179,274, 364,396
- [**filters=18**] (classes + 5)x3 en los tres [convolutional] antes de cada [yolo]
- Si classes==1 entonces filters=18. Si classes==2 entonces filters=21.

ENTRENAMIENTO

- Descargar los pesos pre-entrenados de YOLOv4:
  - yolov4.conv.137: https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137
- ./darknet detector **train** FaceDetecDataset_subV1/obj.data yolov4-FaceDetec_train.cfg yolov4.conv.137 -map -dont_show > FaceDetecDataset_subV1/train.log

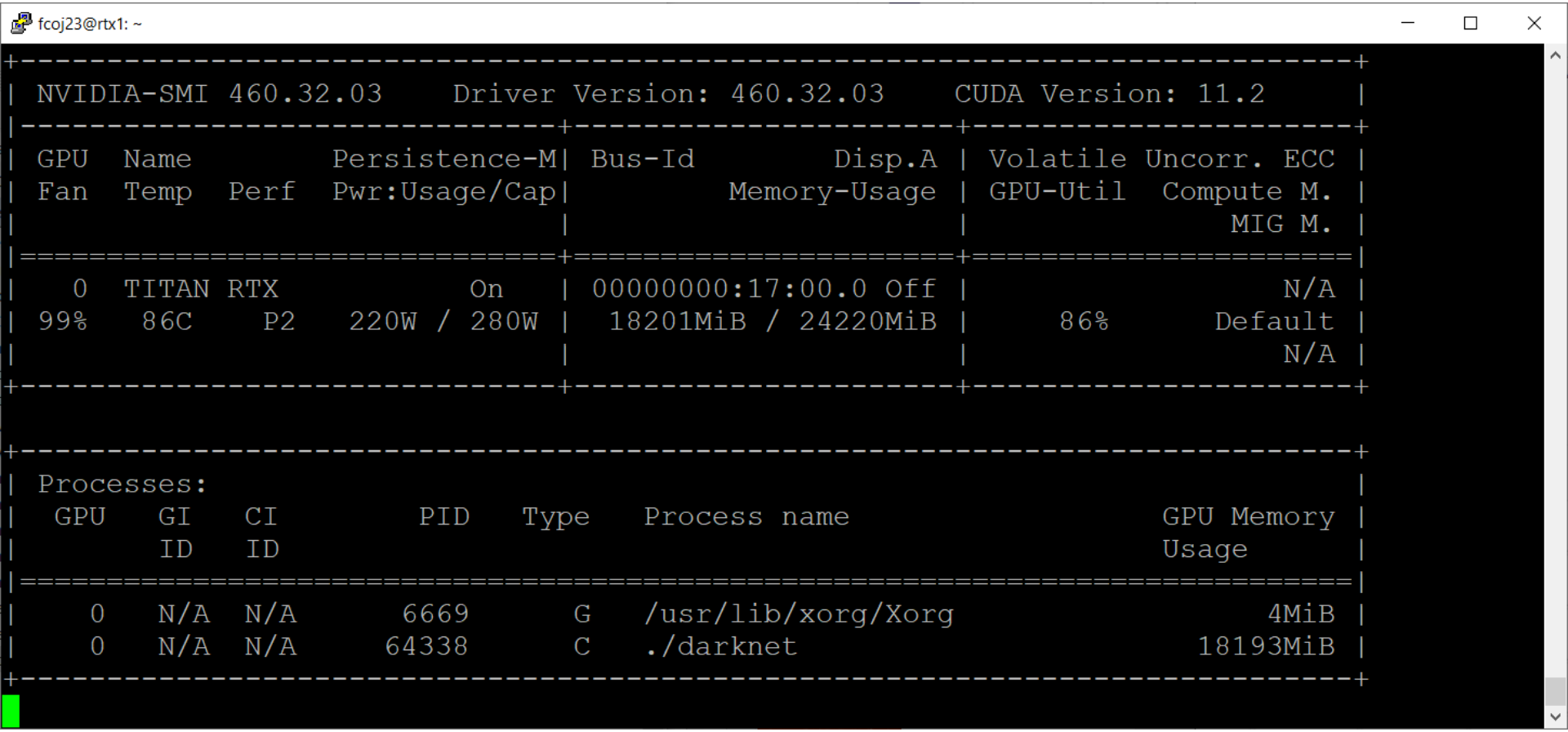
```
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.3
74654), count: 21, class_loss = 478.241974, iou_loss = 0.534027, total_loss = 478.77
6001
 total_bbox = 61836, rewritten_bbox = 0.029109 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.3
57523), count: 4, class_loss = 7308.992188, iou_loss = 1.535645, total_loss = 7310.5
27832
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.3
80599), count: 19, class_loss = 1607.462280, iou_loss = 2.047607, total_loss = 1609.
509888
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.3
58603), count: 24, class_loss = 476.473541, iou_loss = 0.424438, total_loss = 476.89
7980
```

HTOP


fcoj23@rtx1: ~
1 [||||100.0%] 7 [||| 18.7%] 13 [|| 8.4%] 19 [|| 10.4%]
2 [||||100.0%] 8 [||| 12.5%] 14 [|| 7.2%] 20 [|| 9.2%]
3 [||| 21.7%] 9 [||| 13.8%] 15 [|| 8.5%] 21 [|| 8.6%]
4 [|||||37.9%] 10 [||| 12.0%] 16 [|| 9.0%] 22 [|| 9.7%]
5 [|||||69.5%] 11 [|||||41.7%] 17 [|| 8.4%] 23 [|| 7.8%]
6 [||| 17.3%] 12 [|||||41.2%] 18 [|| 7.9%] 24 [|| 9.1%]
Mem[|||||| 6.62G/62.5G] Tasks: 121, 191 thr; 4 running
Swp[0K/8.00G] Load average: 4.81 2.68 2.23
Uptime: 18 days, 03:16:08
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
64338 fcoj23 20 0 54.5G 6109M 518M R 376. 9.5 1:57.38 ./darknet d
63879 jcisneros 20 0 910M 84644 25084 R 99.5 0.1 3h11:01 python3 gen
63801 jcisneros 20 0 910M 84840 25256 R 99.5 0.1 3h11:08 python3 gen
64347 fcoj23 20 0 54.5G 6109M 518M S 40.6 9.5 0:05.51 ./darknet d
64345 fcoj23 20 0 54.5G 6109M 518M S 36.0 9.5 0:05.90 ./darknet d
64349 fcoj23 20 0 54.5G 6109M 518M S 12.4 9.5 0:04.10 ./darknet d
64350 fcoj23 20 0 54.5G 6109M 518M S 11.8 9.5 0:03.40 ./darknet d
64346 fcoj23 20 0 54.5G 6109M 518M S 11.1 9.5 0:05.53 ./darknet d
64348 fcoj23 20 0 54.5G 6109M 518M S 10.5 9.5 0:03.97 ./darknet d
64372 fcoj23 20 0 54.5G 6109M 518M S 7.9 9.5 0:01.24 ./darknet d
64365 fcoj23 20 0 54.5G 6109M 518M S 7.2 9.5 0:01.25 ./darknet d
F1Help F2Setup F3SearchF4FilterF5Tree F6SortByF7Nice -F8Nice +F9Kill F10Q

NVIDIA-SMI

nvidia-smi –l 1



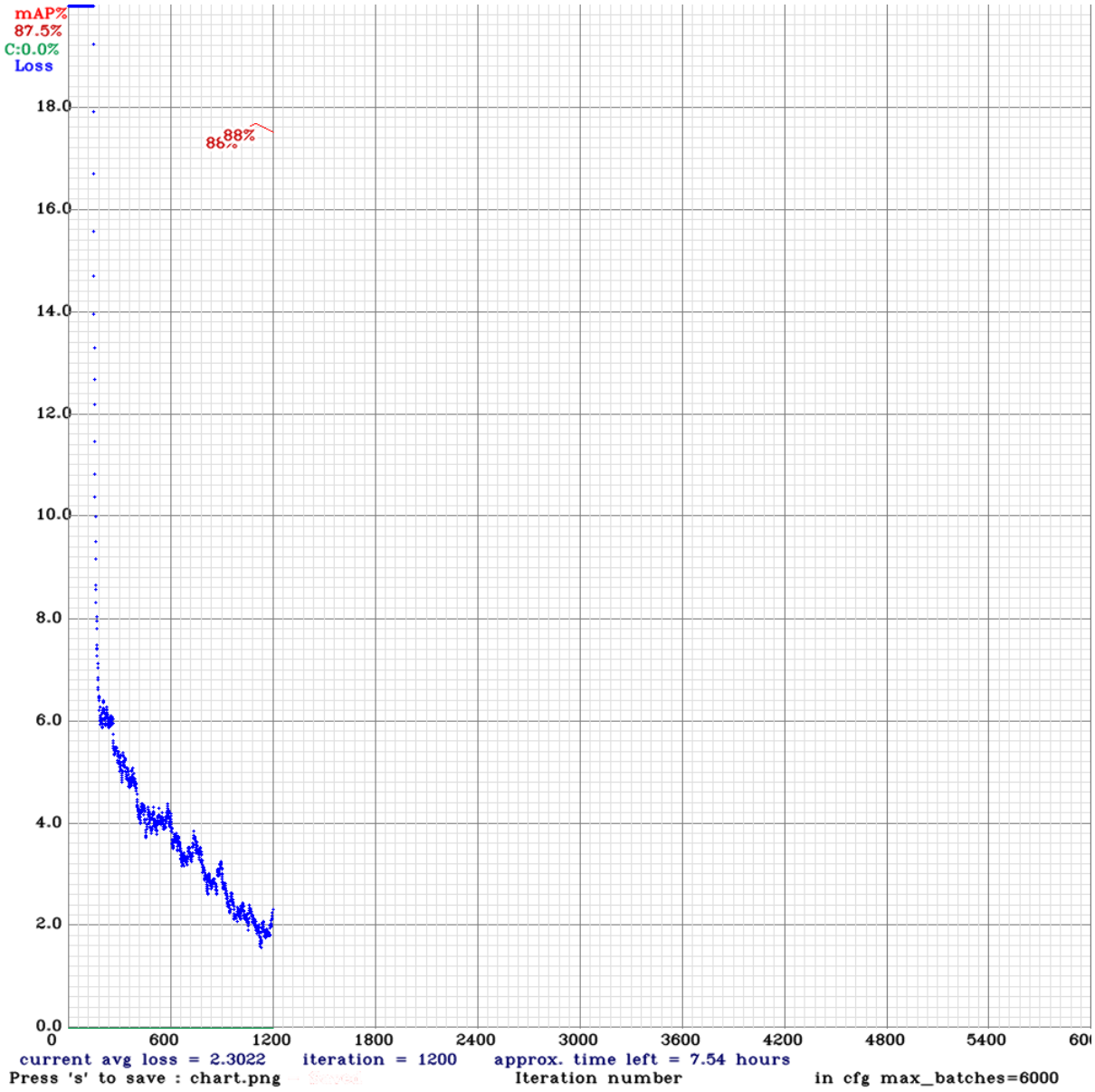
¿CÓMO VA EL ENTRENAMIENTO?

grep "avg" FaceDetecDataset_subV1/train.log

```
fcoj23@rtx1:~/downloads/darknet$ grep "avg" FaceDetecDataset_subV1/train.log
 1: 5427.008301, 5427.008301 avg loss, 0.000000 rate, 4.664203 seconds, 64 images, -1.000000 hours left
 2: 5426.323242, 5426.939941 avg loss, 0.000000 rate, 4.237279 seconds, 128 images, 648.460958 hours left
 3: 5425.117676, 5426.757812 avg loss, 0.000000 rate, 4.220216 seconds, 192 images, 647.867423 hours left
 4: 5431.368164, 5427.218750 avg loss, 0.000000 rate, 4.223999 seconds, 256 images, 647.256072 hours left
 5: 5425.723633, 5427.069336 avg loss, 0.000000 rate, 4.441825 seconds, 320 images, 646.656077 hours left
 6: 5415.039551, 5425.866211 avg loss, 0.000000 rate, 4.663491 seconds, 384 images, 646.364924 hours left
 7: 5408.686523, 5424.148438 avg loss, 0.000000 rate, 4.871707 seconds, 448 images, 646.384845 hours left
 8: 5428.757812, 5424.609375 avg loss, 0.000000 rate, 4.975479 seconds, 512 images, 646.694020 hours left
 9: 5431.627441, 5425.311035 avg loss, 0.000000 rate, 5.111208 seconds, 576 images, 647.144383 hours left
 10: 5425.229980, 5425.302734 avg loss, 0.000000 rate, 5.195512 seconds, 640 images, 647.778905 hours left
 11: 4314.066895, 5314.179199 avg loss, 0.000000 rate, 4.313348 seconds, 704 images, 648.524286 hours left
 12: 4324.183594, 5215.179688 avg loss, 0.000000 rate, 4.450318 seconds, 768 images, 648.035746 hours left
```

```
 145: 5.665501, 19.277523 avg loss, 0.000000 rate, 1.683193 seconds, 9280 images, 506.226092 hours left
 146: 5.695033, 17.919273 avg loss, 0.000000 rate, 1.637801 seconds, 9344 images, 503.503329 hours left
 147: 6.615353, 16.788881 avg loss, 0.000000 rate, 1.648340 seconds, 9408 images, 500.744705 hours left
 148: 4.688131, 15.578806 avg loss, 0.000000 rate, 1.616517 seconds, 9472 images, 498.028310 hours left
 149: 4.905978, 14.511523 avg loss, 0.000000 rate, 1.635936 seconds, 9536 images, 495.294878 hours left
 150: 6.733457, 13.733717 avg loss, 0.000001 rate, 1.655325 seconds, 9600 images, 492.615734 hours left
 151: 5.410782, 12.901423 avg loss, 0.000001 rate, 4.186024 seconds, 9664 images, 489.990336 hours left
 152: 5.572734, 12.168554 avg loss, 0.000001 rate, 4.283082 seconds, 9728 images, 490.908491 hours left
 153: 6.967654, 11.648464 avg loss, 0.000001 rate, 4.390811 seconds, 9792 images, 491.952360 hours left
 154: 8.232637, 11.306882 avg loss, 0.000001 rate, 4.421748 seconds, 9856 images, 493.135516 hours left
 155: 8.653145, 11.041508 avg loss, 0.000001 rate, 4.450928 seconds, 9920 images, 494.349814 hours left
```

LOSS Y MAP


mAP%
87.5%
C:0.0%
Loss
18.0
16.0
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0.0
88%
0
600
1200
1800
2400
3000
3600
4200
4800
5400
current avg loss = 2.3022 iteration = 1200 approx. time left = 7.54 hours
Press 's' to save : chart.png
Iteration number
in cfg max_batches=6000

ANALIZANDO EL MAP

- ./darknet detector **map** FaceDetecDataset_subV1/obj.data yolov4-FaceDetec_test.cfg FaceDetecDataset_subV1_backup/yolov4-FaceDetec_train_best.weights

```
200
 detections_count = 2835, unique_truth_count = 608
class_id = 0, name = face, ap = 88.30%           (TP = 529, FP = 133)

 for conf_thresh = 0.25, precision = 0.80, recall = 0.87, F1-score = 0.83
 for conf_thresh = 0.25, TP = 529, FP = 133, FN = 79, average IoU = 61.84 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.882951, or 88.30 %
Total Detection Time: 7 Seconds
```

PROCESANDO UNA IMAGEN

- ./darknet detector **test** FaceDetecDataset_subV1/obj.data yolov4-FaceDetec_test.cfg FaceDetecDataset_subV1_backup/yolov4-FaceDetec_train_best.weights FaceDetecDataset_subV1/images/val/0a32de8598733c17.jpg

PROCESANDO UN VIDEO

- ./darknet detector **demo** FaceDetecDataset_subV1/obj.data yolov4-FaceDetec_test.cfg FaceDetecDataset_subV1_backup/yolov4-FaceDetec_train_best.weights /home/fcoj23/videos_CD/office.avi -out_filename /home/fcoj23/videos_CD/face_detec_office.avi

YOLO - ULTRALYTICS

- Instalación en un ambiente conda:
  - conda create --name ultralytics-env python=3.11
  - conda activate ultralytics-env
  - conda install -c pytorch -c nvidia -c conda-forge pytorch torchvision pytorch-cuda=11.8 ultralytics

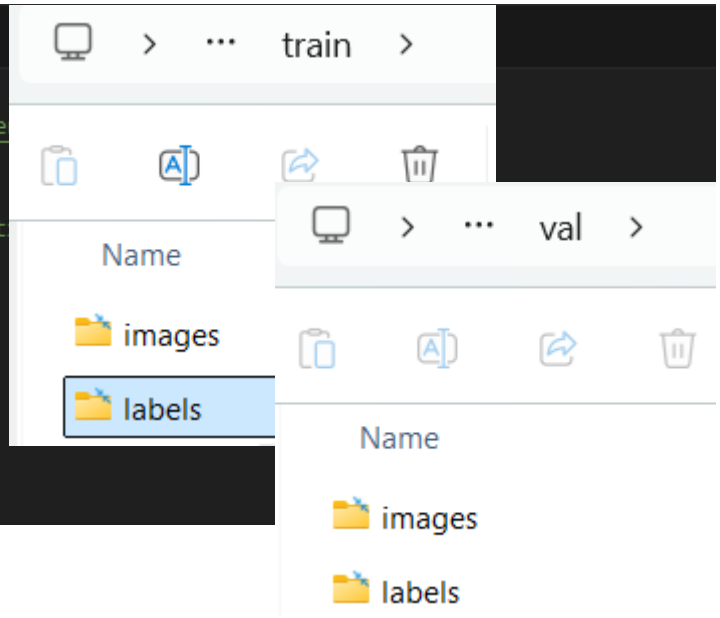
ENTRENAMIENTO DEL MODELO YOLO V12 NANO

```
model = YOLO("yolo12n.pt") # load model
# Train the model
results =
model.train(data="D:/pictures/FaceDetecDataset_subV1/faceDetect_subV1.yaml"
, epochs=5, imgsz=640)
```

YOLO_Ultralytics.ipynb | faceDetect_subV1.yaml

D: > pictures > FaceDetecDataset_subV1 > faceDetect_subV1.yaml

```
1   # Subset of FaceDetecDataset https://www.kaggle.com/datasets/fareselme
2   # Example usage: python train.py --data coco128.yaml
3
4   # Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.t
5   path: D:/pictures/FaceDetecDataset_subV1 # dataset root dir
6   train: images_labels/train # train images (relative to 'path')
7   val: images_labels/val # val images (relative to 'path')
8   test: # test images (optional)
9
10  # Classes
11  names:
12    0: face
```



```
Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
  1/5      4.99G      1.631      2.357       1.35         76        640: 100%|██████████| 50/50 [02:58<00:00,  3.56s/it]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 7/7 [00:25<00:00,  3.58s/it]
             all        200        608    0.00857      0.845     0.0218    0.00955

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
  2/5      4.98G      1.495      1.578      1.263         73        640: 100%|██████████| 50/50 [03:30<00:00,  4.21s/it]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 7/7 [00:16<00:00,  2.42s/it]
             all        200        608      0.645      0.438      0.465      0.264

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
  3/5      4.95G      1.426      1.384      1.231        124        640: 100%|██████████| 50/50 [03:02<00:00,  3.65s/it]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 7/7 [00:21<00:00,  3.06s/it]
             all        200        608      0.601      0.613      0.604      0.351

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
  4/5      4.96G       1.35      1.234      1.191         81        640: 100%|██████████| 50/50 [03:03<00:00,  3.67s/it]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 7/7 [00:20<00:00,  2.87s/it]
             all        200        608      0.875      0.689      0.797      0.472

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
  5/5      4.96G      1.267      1.135      1.167         72        640: 100%|██████████| 50/50 [03:02<00:00,  3.65s/it]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 7/7 [00:21<00:00,  3.03s/it]
             all        200        608      0.893       0.72      0.817      0.502
```

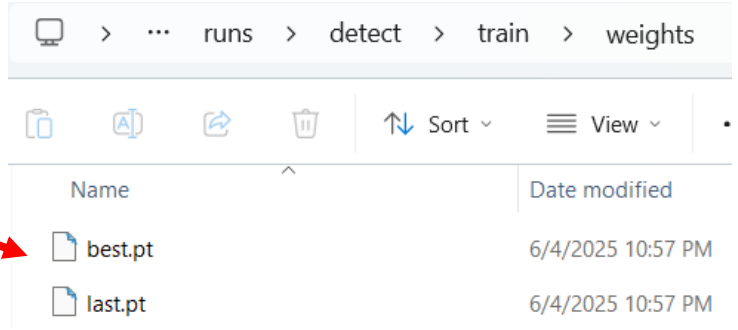
DETECCIÓN EN UNA IMAGEN USANDO EL MODELO PREVIAMENTE ENTRENADO

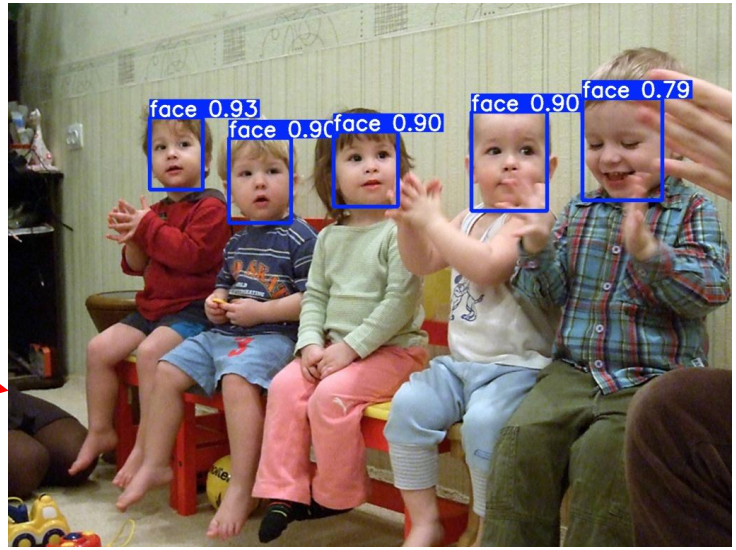
```python
# Load a pretrained model
model = YOLO("yolo12n_best_weights.pt")

# Open image
img=cv2.imread("f1072a4875cec459.jpg")

# Run YOLO inference on the image
results = model(img)
# Visualize the results on the image
annotated_img = results[0].plot()

# Display the annotated image
cv2.imshow("YOLO Inference", annotated_img)
cv2.waitKey(0)
# Save the annotated image
cv2.imwrite("f1072a4875cec459_annotated.jpg"
,annotated_img)
# Close windows
cv2.destroyAllWindows()
```





```python
# Detections from video…

# Load a pretrained model
model =
YOLO("yolo12n_best_weights.pt")

# Open the video file
video_path = "office.avi"
cap =
cv2.VideoCapture(video_path)

# Save video
fourcc =
cv2.VideoWriter_fourcc(*'mp4v')
out    =
cv2.VideoWriter('office_output.mp
4', fourcc, 20.0, (360,240))

# To adjust the window
cv2.namedWindow("YOLO
Inference",cv2.WINDOW_NORMAL)
```

```python
# Loop through the video frames
while cap.isOpened():
    # Read a frame from the video
    success, frame = cap.read()

    if success:
        # Run YOLO inference on the frame
        results = model(frame)
        #results = model(frame,conf=0.5)

        # Visualize the results on the frame
        annotated_frame = results[0].plot()

        # Display the annotated frame
        cv2.imshow("YOLO Inference",
annotated_frame)

        # save frame
        out.write(annotated_frame)

        # Break the loop if 'q' is pressed
        if cv2.waitKey(1) & 0xFF ==
ord("q"):
            break
    else:
        # Break the loop if the end of the
video is reached
        break

# Release the video capture object
cap.release()
out.release()
# close the display window
cv2.destroyAllWindows()
```

RESULTADOS

office_output

ADL-Rundle-6_output

GRACIAS POR SU ATENCIÓN

Francisco J. Hernández López

fcoj23@cimat.mx

WebPage:

www.cimat.mx/~fcoj23