

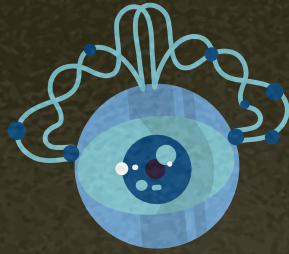


Prueba de concepto: Detección de armas sobre video



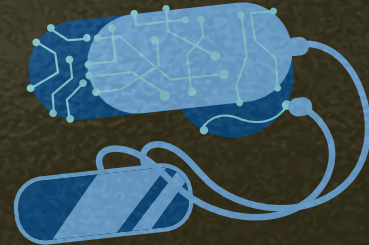
Florencia Priscilla Vela





Contexto del problema

Utilizamos sistemas de videovigilancia para reducir riesgos y minimizar impactos negativos en la seguridad de las personas. Actualmente estas tareas las llevan a cabo **operadores humanos** y esto presenta **diversos inconvenientes**.

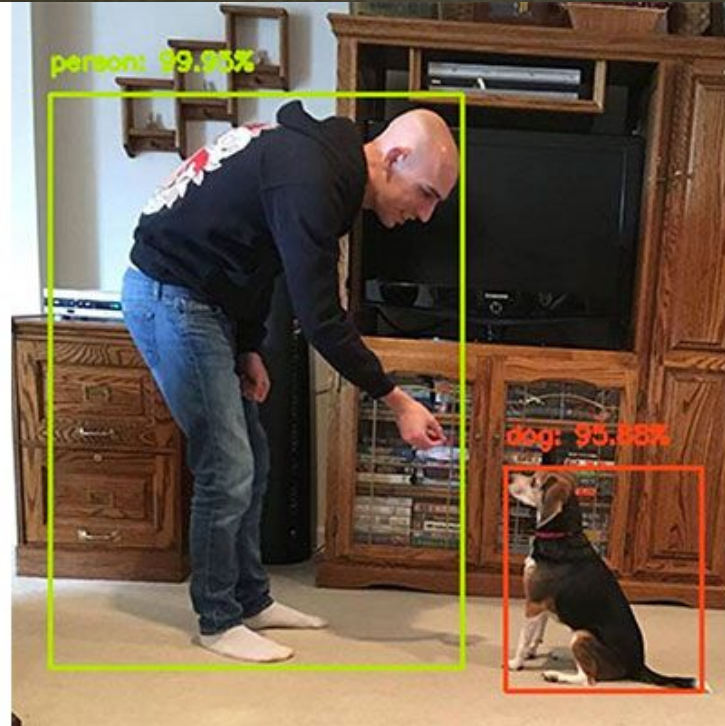




Deep Learning

- Se volvió muy popular para la detección y clasificación de comportamiento humano.
- Se está revolucionando el estado del arte.

Clasificación de Imágenes y Detección de Objetos



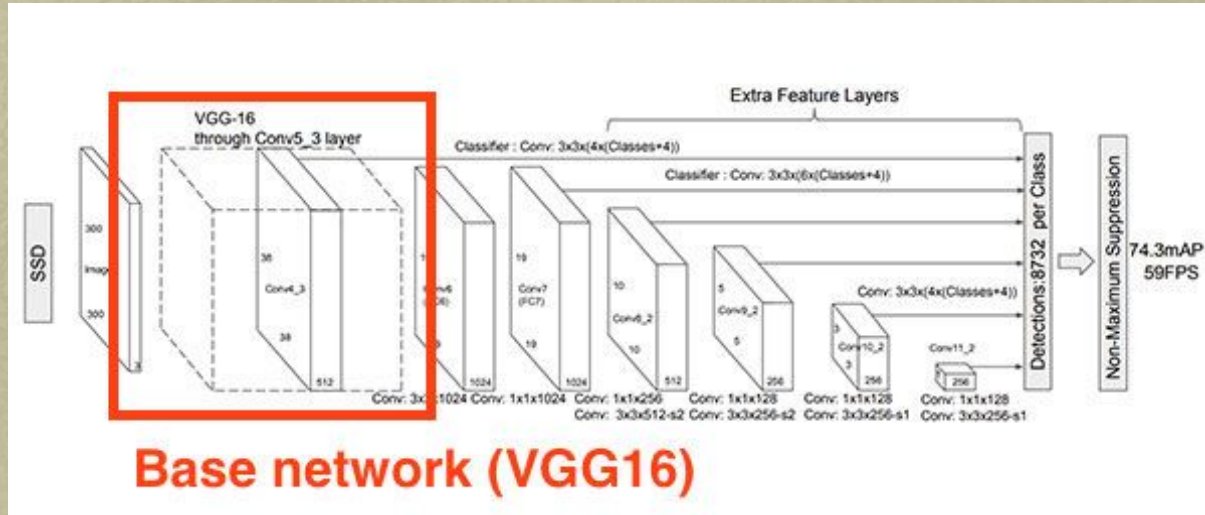
Solución del PROBLEMA



Pipeline con una Red Base

En la arquitectura de los detectores, las redes base ("base network") son arquitecturas de Redes Neuronales Convolucionales que incluyen:

- VGGNet
- ResNet
- MobileNet
- DenseNet





Faster RCNN

Con Detectron2

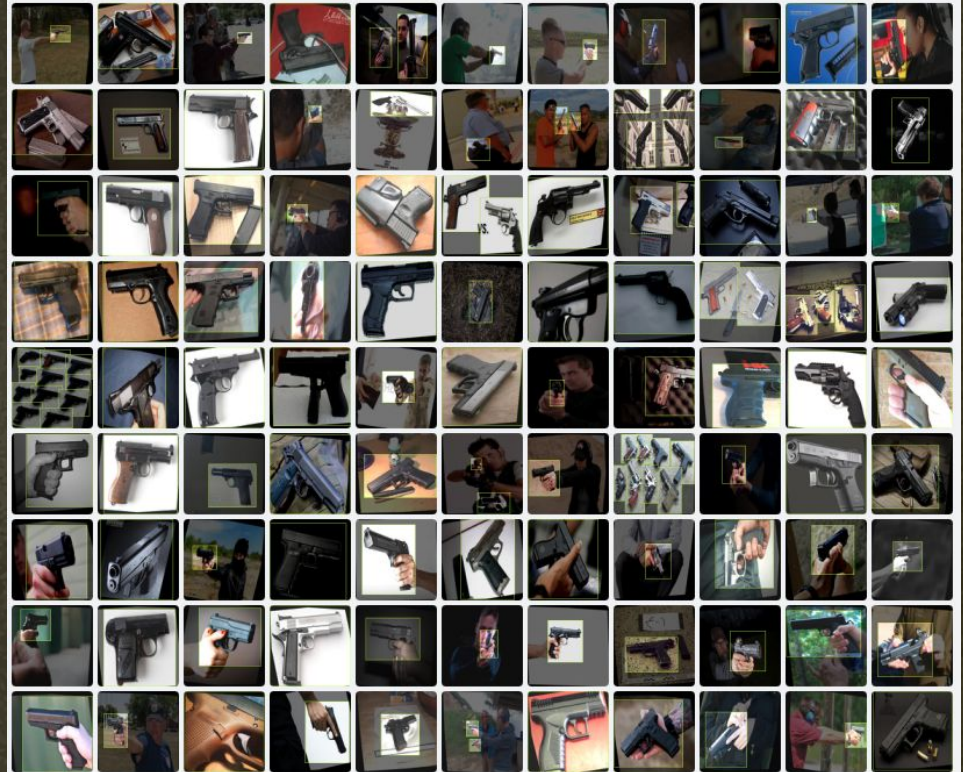


YOLO

Con CSP Darnket

Modelo de Datos

- Imágenes RGB cuadradas
- Una sola clase: "Handgun"
- Resolución 416 px x 416 px
 - Train: 1920 imágenes
 - Test: 183 imágenes
 - Valid: 91 imágenes



Modelo de Datos

Outputs per training example: 3

Flip: Horizontal

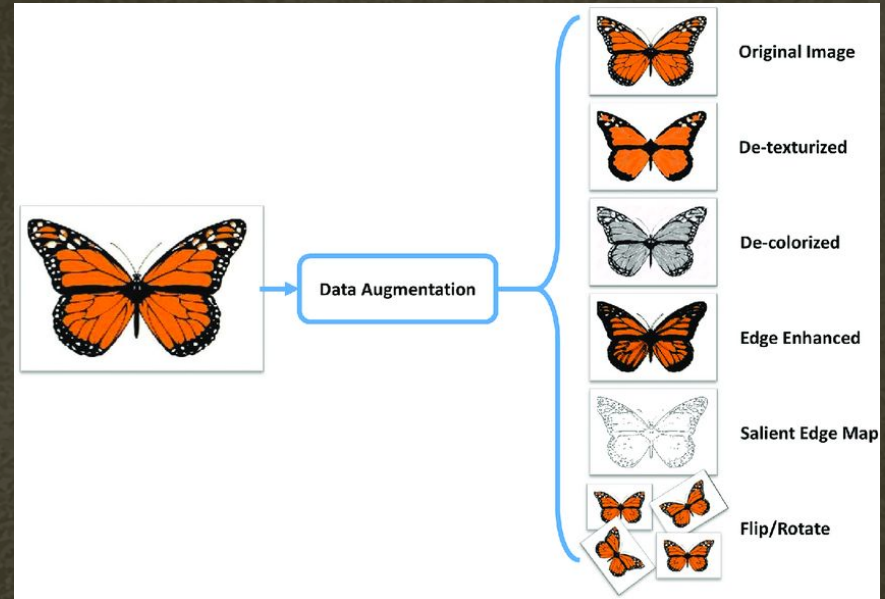
Rotation: Between -10° and $+10^\circ$

Shear: $\pm 15^\circ$ Horizontal, $\pm 15^\circ$ Vertical

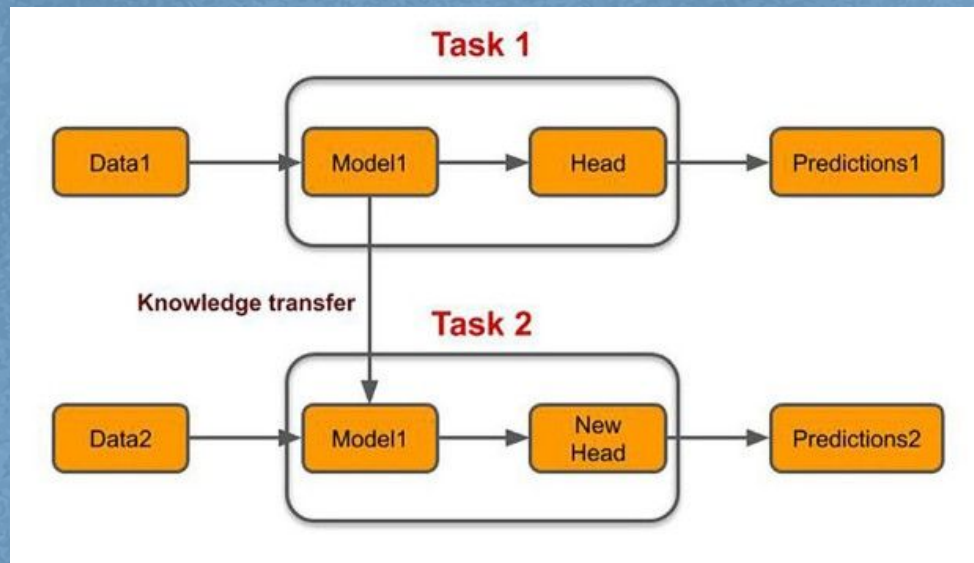
Saturation: Between -25% and $+25\%$

Brightness: Between -20% and $+20\%$

Blur: Up to 1px



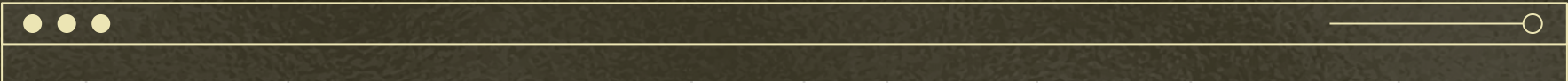
Transfer Learning



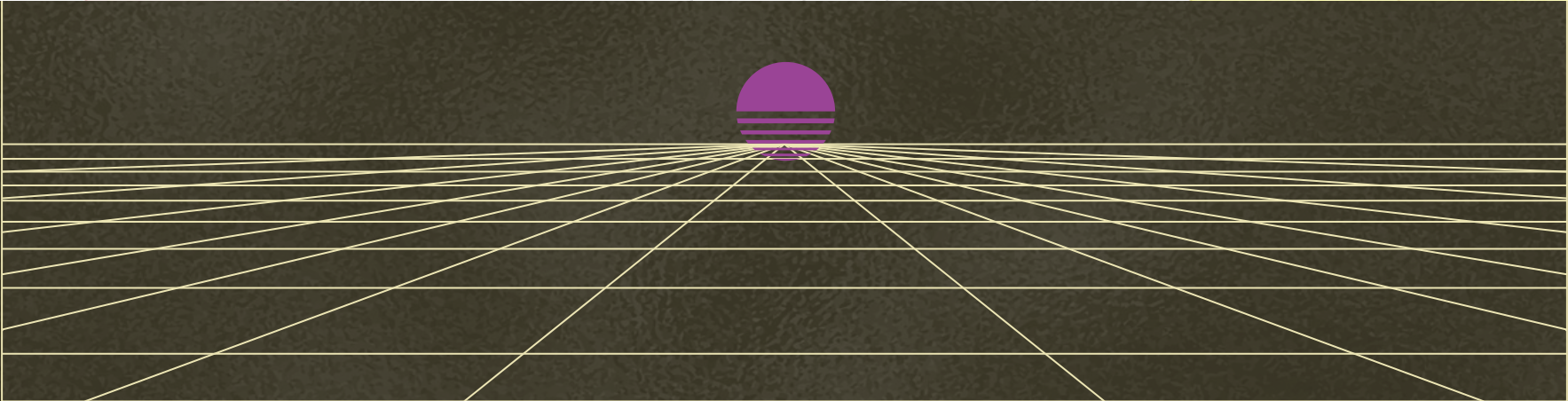


#05

PRUEBAS Y RESULTADOS

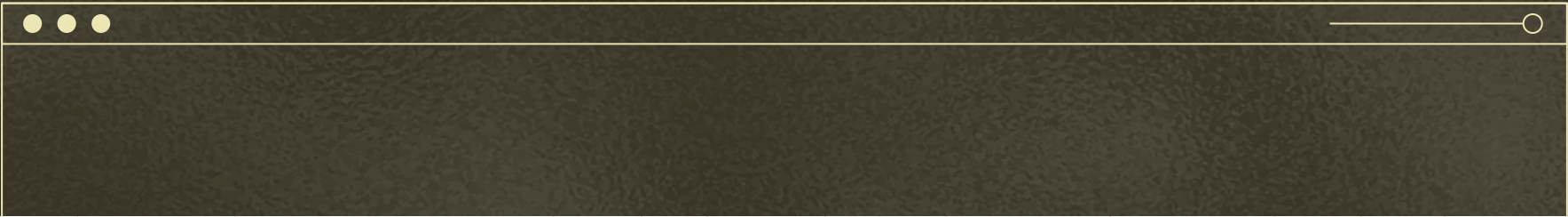


| modelo | total training time | initial_weights | transfer learning frozen layers | batch | metrics/ precision | metrics/ recall | metrics/ mAP_0.5 | metrics/ mAP_0.5:0.95 |
|--------|---------------------|--------------------|---------------------------------|-------|--------------------|-----------------|------------------|-----------------------|
| yolov5 | 0.67 hours | yolov5l.pt | 0 | 32 | 0,9424 | 0,76119 | 0,88523 | 0,54002 |
| yolov5 | 0.54 hours | yolov5l.pt | 10 | 32 | 0,55767 | 0,53317 | 0,54675 | 0,22334 |
| yolov5 | 0.42 hours | yolov5l.pt | 10 | 16 | 0,55894 | 0,58635 | 0,53923 | 0,21019 |
| yolov5 | 0.387 hours | yolov5s.pt | 24 | 32 | 0,058534 | 0,054726 | 0,016555 | 0,0034362 |
| yolov7 | 2.800 hours | yolov7_training.pt | 0 | 16 | 0,8333 | 0,8205 | 0,851 | 0,4458 |

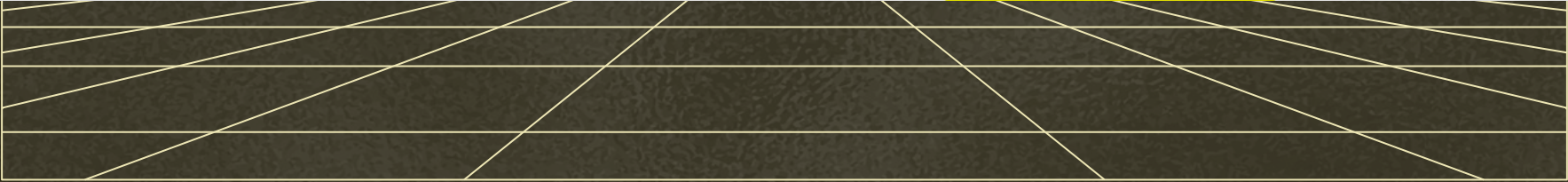


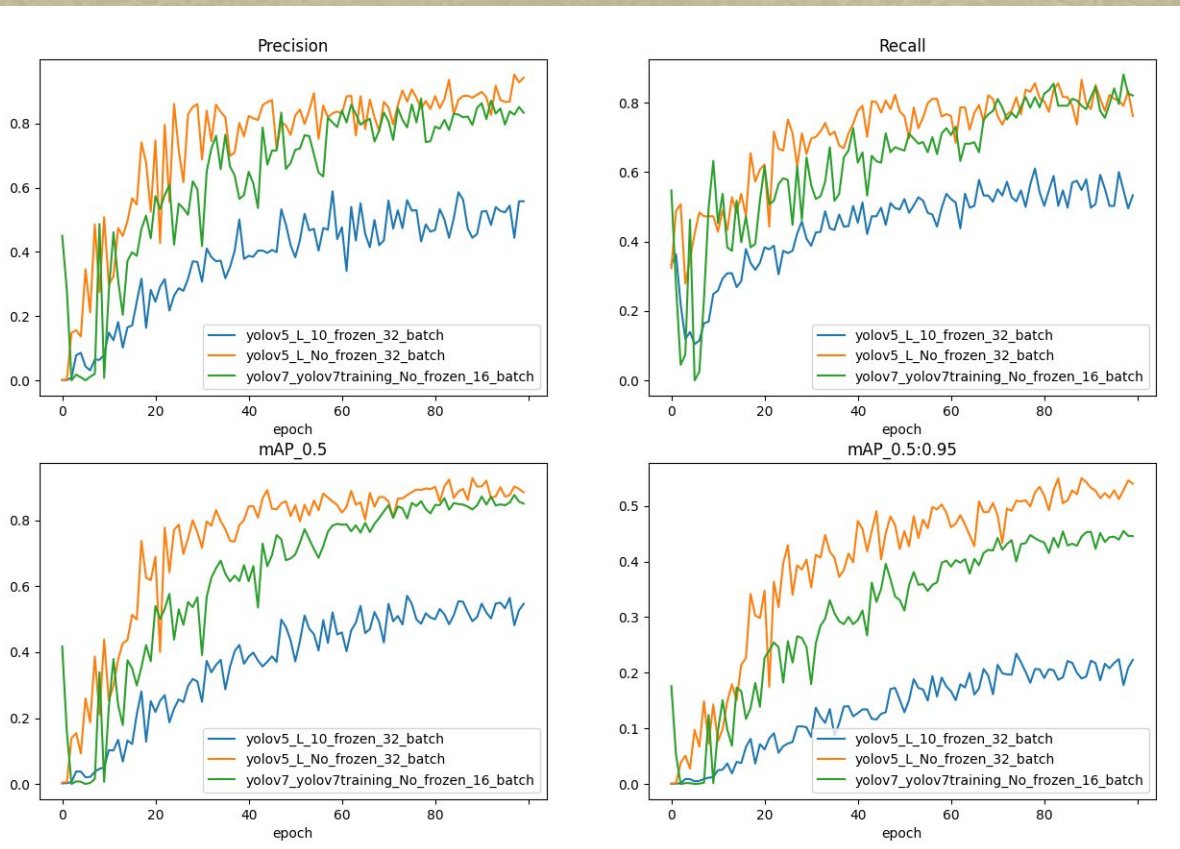
| | | | transfer learning | | | | | |
|--------|------------------------|--------------------|----------------------|-------|-----------------------|--------------------|---------------------|--------------------------|
| modelo | total training time | initial_weights | frozen layers | batch | metrics/ precision | metrics/ recall | metrics/ mAP_0.5 | metrics/ mAP_0.5:0.95 |
| yolov5 | 0.67 hours | yolov5l.pt | 0 | 32 | 0,9424 | 0,76119 | 0,88523 | 0,54002 |
| yolov5 | 0.54 hours | yolov5l.pt | 10 | 32 | 0,55767 | 0,53317 | 0,54675 | 0,22334 |
| yolov5 | 0.42 hours | yolov5l.pt | 10 | 16 | 0,55894 | 0,58635 | 0,53923 | 0,21019 |
| yolov5 | 0.387 hours | yolov5s.pt | 24 | 32 | 0,058534 | 0,054726 | 0,016555 | 0,0034362 |
| yolov7 | 2.800 hours | yolov7_training.pt | 0 | 16 | 0,8333 | 0,8205 | 0,851 | 0,4458 |

| modelo | total training time | initial_weights | transfer learning frozen layers | batch | metrics/ precision | metrics/ recall | metrics/ mAP_0.5 | metrics/ mAP_0.5:0.95 | epoch |
|--------|---------------------|-----------------|---------------------------------|-------|--------------------|-----------------|------------------|-----------------------|-------|
| yolov5 | 0.67 hours | yolov5l.pt | 0 | 32 | 0,9424 | 0,76119 | 0,88523 | 0,54002 | 100 |
| yolov5 | 1.83 hours | yolov5l.pt | 0 | 32 | 0,93763 | 0,88557 | 0,95886 | 0,62906 | 384 |
| yolov5 | 1.462 hours | yolov5l.pt | 10 | 32 | 0,64576 | 0,63184 | 0,71028 | 0,32273 | 377 |

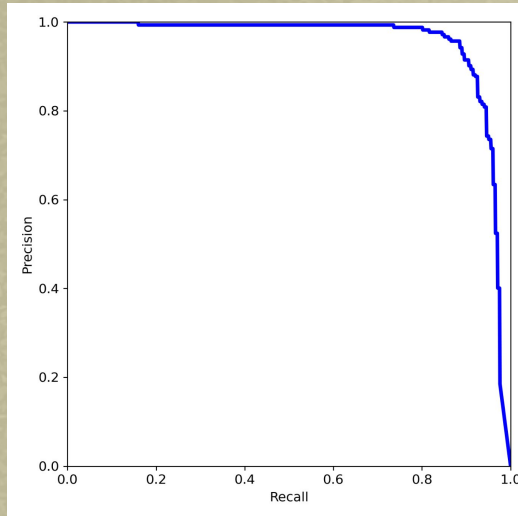


| modelo | total training time | initial_weights | transfer learning frozen layers | batch | metrics/ precision | metrics/ recall | metrics/ mAP_0.5 | metrics/ mAP_0.5:0.95 | lr0 | gamma | epoch |
|------------|---------------------|--|---------------------------------|-------|--------------------|-----------------|------------------|-----------------------|-------|-------|-------|
| yolov5 | 1.83 hours | yolov5l.pt | 0 | 32 | 0,9424 | 0,76119 | 0,88523 | 0,54002 | 0,01 | 0 | 100 |
| yolov5 | 1.83 hours | yolov5l.pt | 0 | 32 | 0,93763 | 0,88557 | 0,95886 | 0,62906 | 0,01 | 0 | 384 |
| yolov5 | 1.462 hours | yolov5l.pt | 10 | 32 | 0,64576 | 0,63184 | 0,71028 | 0,32273 | 0,01 | 0 | 377 |
| detectron2 | 0.08333 hours | COCO-Detection/faster_rcnn_X_101_32x8d_FPN_3x.yaml | 0 | 32 | | | 0,65903 | 0,3523 | 0,001 | 0,05 | 100 |
| detectron2 | 0.9 hours | COCO-Detection/faster_rcnn_X_101_32x8d_FPN_3x.yaml | 0 | 64 | 0,57984 | | 0,95069 | 0,57984 | 0,001 | 0,05 | 1500 |
| yolov5 | | yolov5l.pt | 10 | 32 | 0,51 | 0,562 | 0,56000 | 0,22400 | 0,001 | 0,05 | 100 |
| yolov5 | | yolov5l.pt | 10 | 32 | 0,598 | 0,651 | 0,66700 | 0,29800 | 0,001 | 0,05 | 300 |
| yolov5 | | yolov5l.pt | 0 | 32 | 0,89818 | 0,82587 | 0,90395 | 0,55128 | 0,001 | 0,05 | 100 |

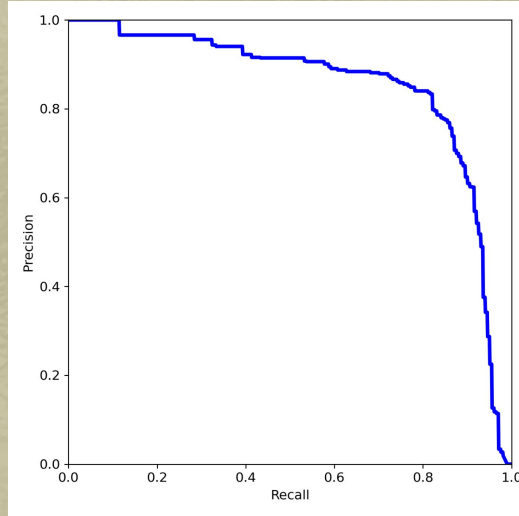




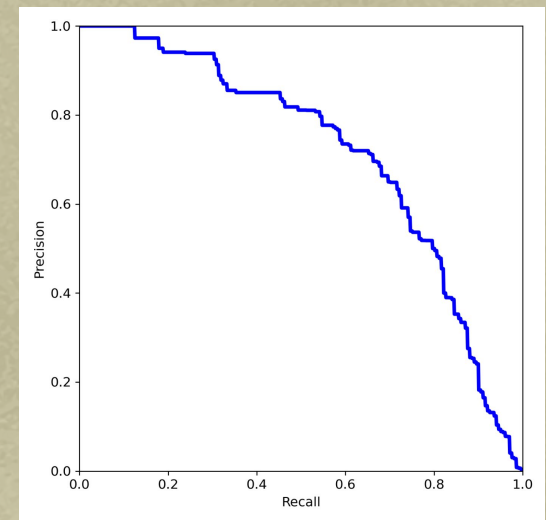
Precisión-Recall Curve



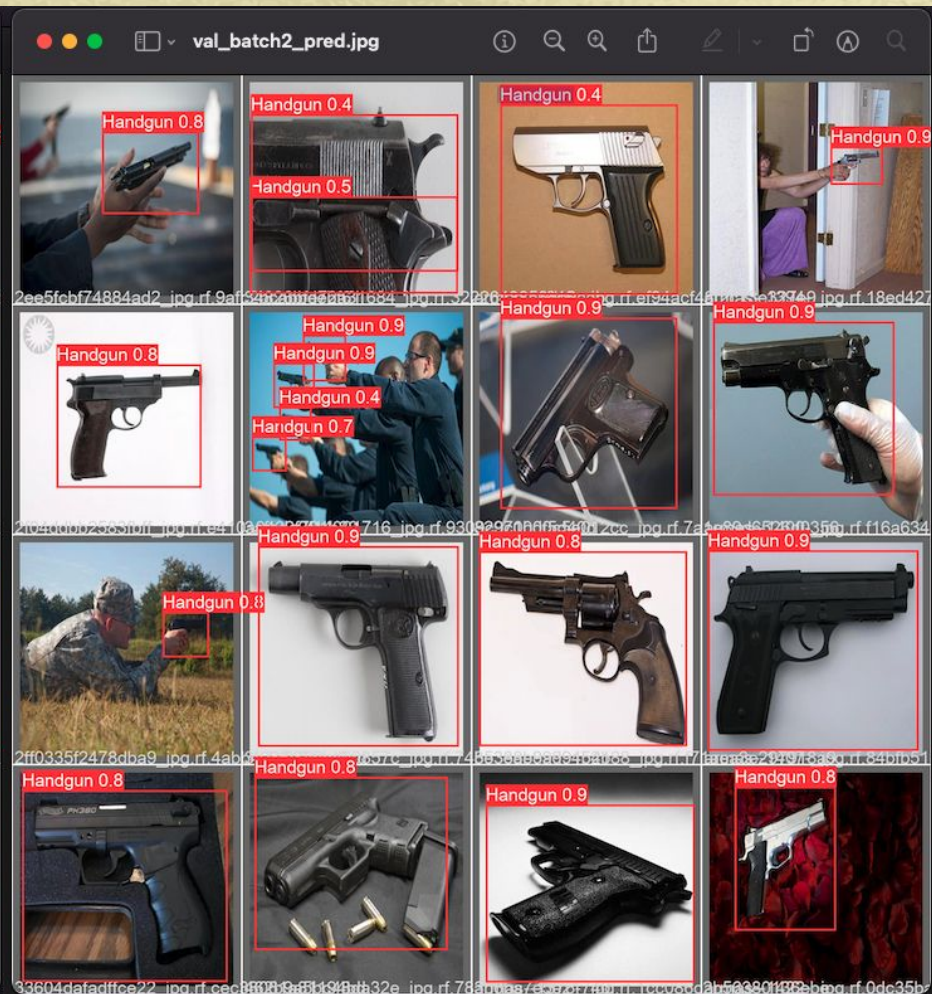
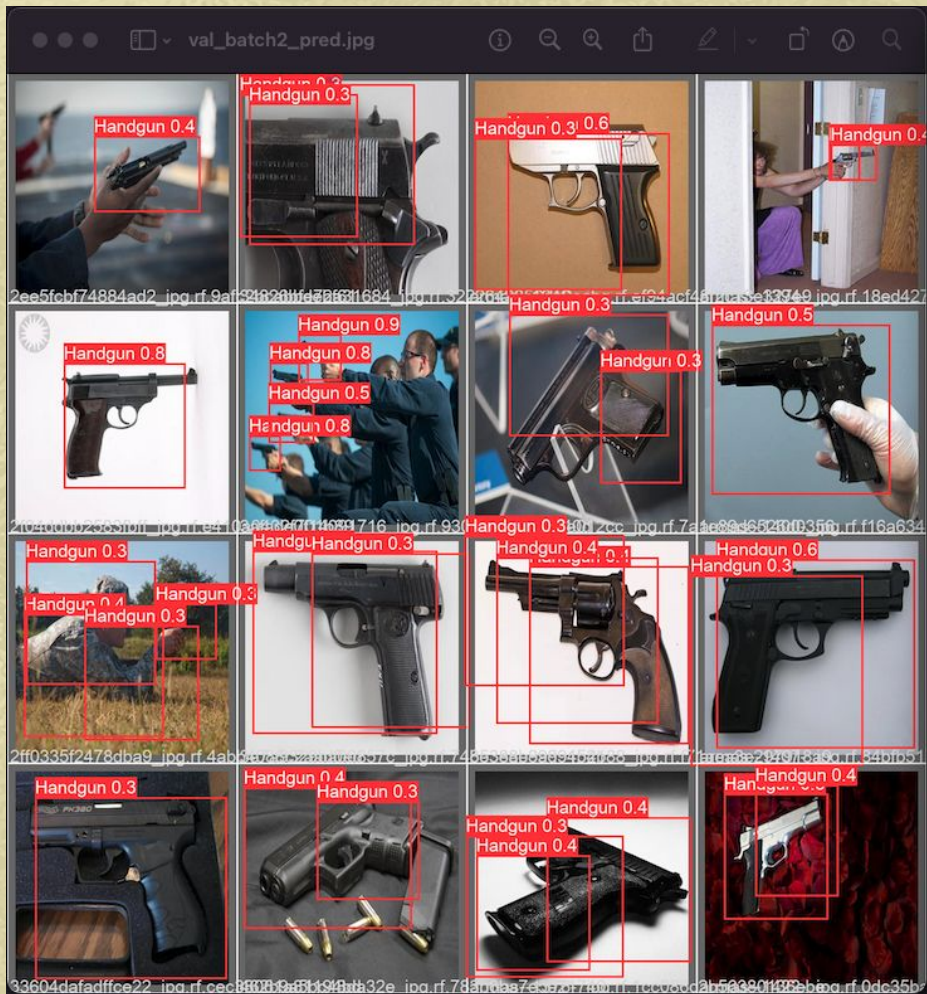
Yolov5 con 384 iteraciones
No frozen layers

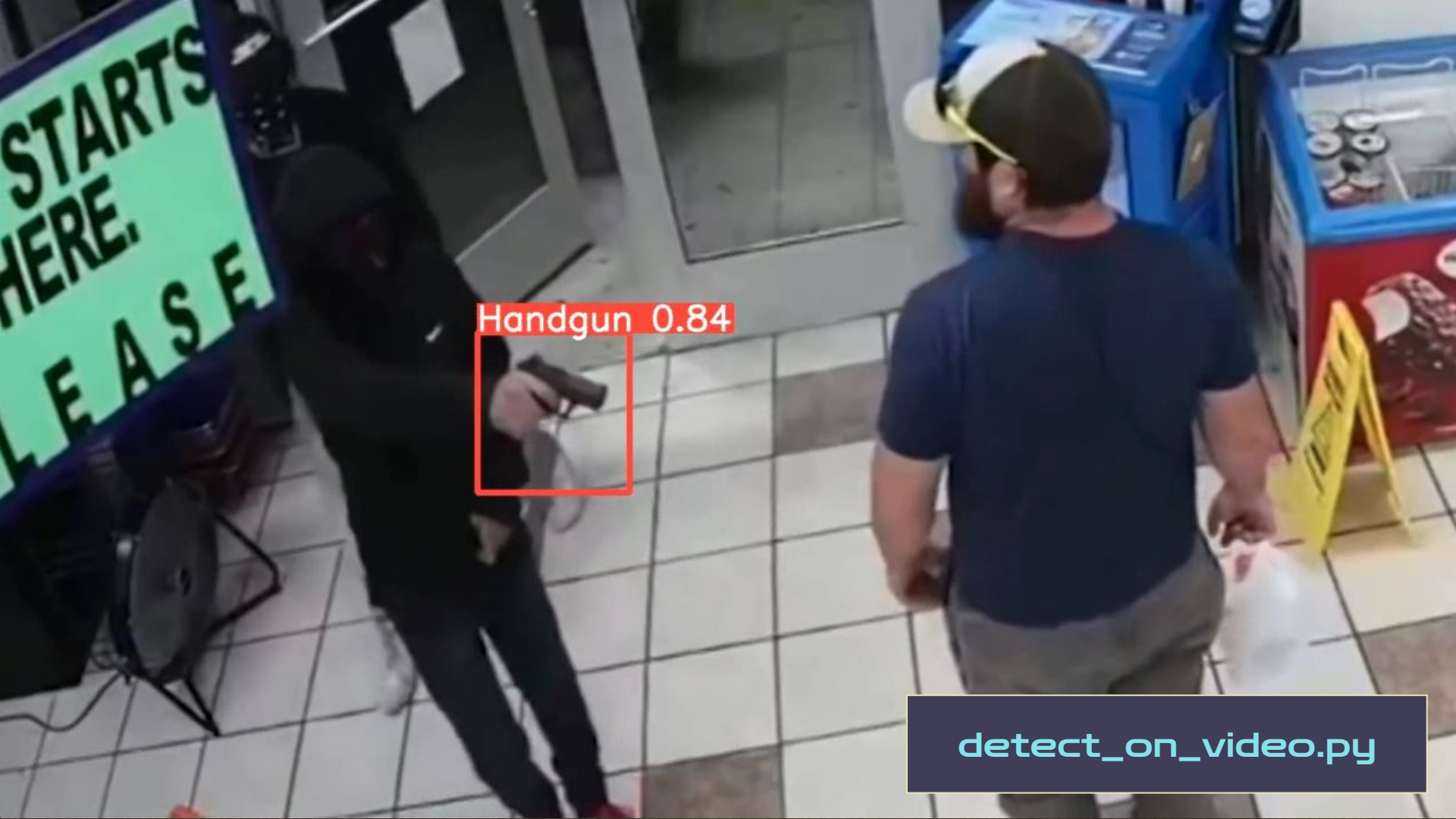


Yolov7 con 100 iteraciones
No frozen layers



Yolov5 con 377 iteraciones
10 frozen layers





Handgun 0.84

detect_on_video.py



#06

CONCLUSIONES

Conclusiones de las pruebas

- YOLOv7 tarda mucho en entrenar.
- Detectron2 tarda mucho en inferir.
- Un learning rate grande permite que el modelo aprenda más rápido, a costa de llegar a un conjunto final de pesos subóptimo.
- Se logró aplicar Transfer Learning.

Conceptos Aplicados

- Deep Learning
- Data Augmentation
- IoU
- detección de objetos
- Modelos de Detección Rápida



Líneas Futuras de Investigación



GRACIAS!

flor.p.vela@gmail.com