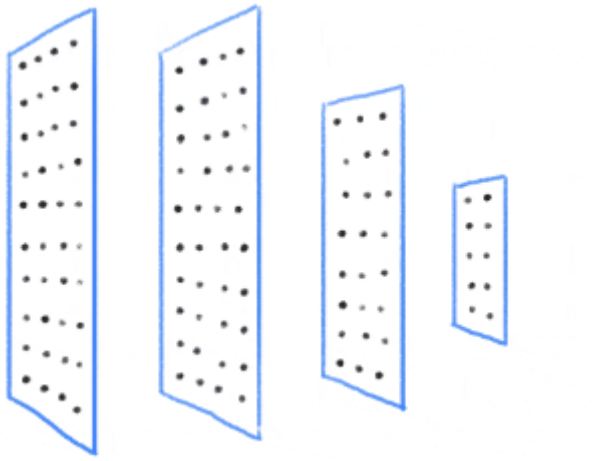
TP Final

Inteligencia Artificial 2023

Red convolucional para detección de objetos con bounding boxes



Asignatura

Inteligencia Artificial

Jefe de cátedra

Gustavo Javier Meschino

JTP

Matías Yerro

Integrantes del grupo:

- Alan Gutiérrez
- Francisco Stimmler

Introducción

En este trabajo se decidió crear un sistema con YOLO versión 7 para detectar objetos tanto en vídeos, como imágenes.

Se decidió usar la versión 7 porque a pesar de no ser la más reciente hoy en día, siendo la versión 8, al tener un año de antigüedad se puede encontrar más información de ella online.

¿Qué es el algoritmo YOLO?

El **algoritmo YOLO** (acrónimo de **You Only Look Once**) es un popular y eficiente algoritmo de detección de objetos en imágenes y videos en tiempo real de código abierto. Utiliza una única red neuronal convolucional para detectar objetos en imágenes, lo que permite una detección de objetos en tiempo real a alta velocidad. Esta red neuronal divide la imagen en regiones, prediciendo cuadros de identificación y probabilidades por cada región; las cajas son ponderadas a partir de las probabilidades predichas y el algoritmo aprende representaciones generalizables de los objetos, permitiendo un bajo error de detección para entradas nuevas, diferentes al conjunto de datos de entrenamiento.

El algoritmo YOLO predice múltiples cuadros de identificación por cuadrícula de celdas. En tiempo de entrenamiento se busca tener un solo cuadro de identificación por objeto, lo cual se consigue a partir de las probabilidades predichas para cada cuadro, manteniendo el de mayor probabilidad.

¿Qué novedades trae YOLOv7?

Optimización del proceso de formación

El algoritmo YOLOv7 no solo intenta optimizar la arquitectura del modelo, sino que también tiene como objetivo optimizar el proceso de entrenamiento. Utiliza una arquitectura de *bag-of-freebies* (bolsa de regalos) entrenable y su objetivo es utilizar módulos y métodos de optimización para mejorar la precisión de la detección de objetos.

Asignación guiada de etiquetas de plomo grueso a fino

YOLOv7 planea usar una nueva asignación de etiqueta guiada de plomo grueso a fino en lugar de la asignación dinámica de etiquetas. Es así porque con la asignación de etiquetas dinámicas, entrenar un modelo con varias capas de salida provoca algunos problemas. Uno de ellos es la asignación de objetivos dinámicos para diferentes ramas y sus salidas.

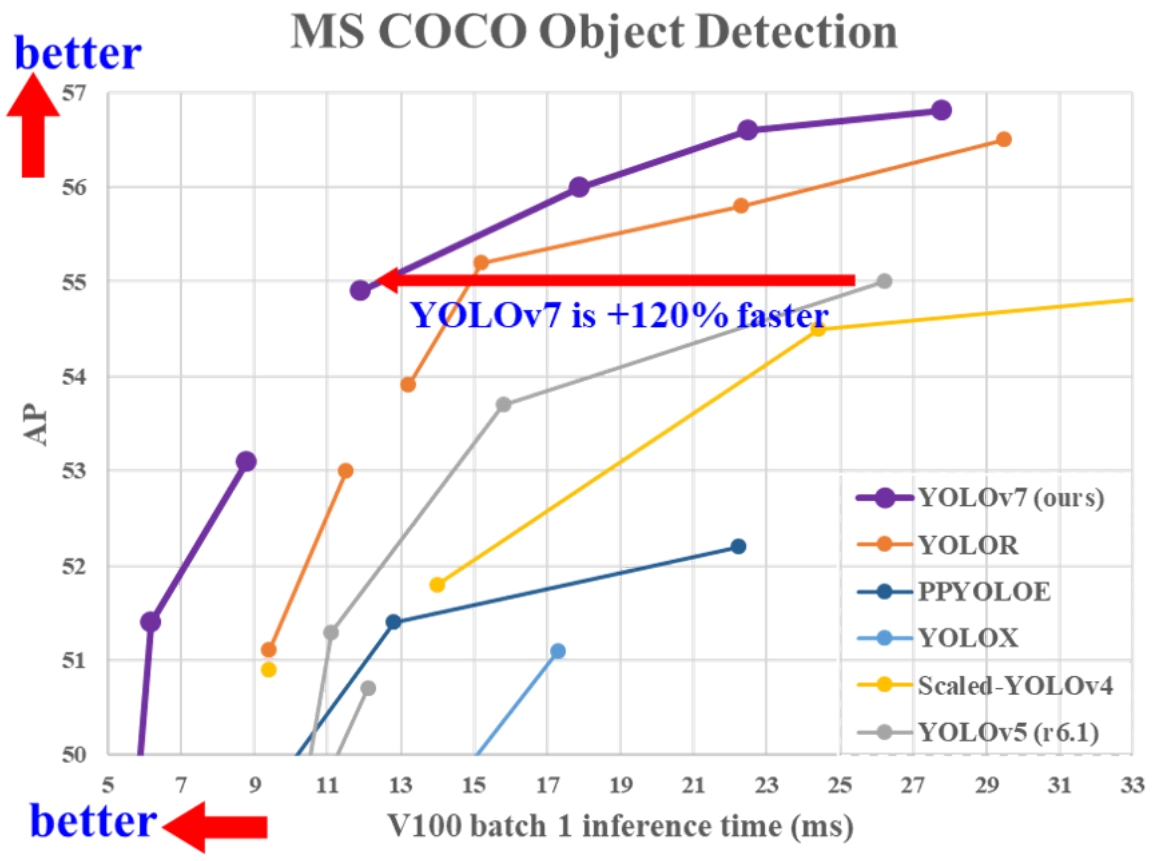
Reparametrización del modelo

La reparametrización del modelo es un concepto importante en la detección de objetos, y su uso generalmente se sigue con algunos problemas durante el entrenamiento. El algoritmo

planea usar el concepto de trayectoria de propagación del gradiente para analizar las políticas de reparametrización del modelo aplicable a las diferentes capas de la red.

Escala extendida y compuesta

YOLOv7 introduce métodos de escalamiento compuestos y extendidos para usar de manera efectiva los parámetros y cálculos para la detección de objetos en tiempo real.



¿Cómo funciona YOLO?

1. Preprocesamiento de la imagen:

YOLO toma una imagen de entrada y la procesa para adaptarla al modelo. Por lo general, se redimensiona para que se ajuste a las dimensiones que la red neuronal espera como entrada.

2. División en cuadrícula:

YOLO divide la imagen de entrada en una cuadrícula de celdas.
Cada celda en la cuadrícula es responsable de predecir un conjunto fijo de bounding boxes y sus correspondientes clases de objetos.

Cada celda en la cuadrícula es responsable de predecir un conjunto fijo de bounding boxes y sus correspondientes clases de objetos.

3. Predicción de bounding box

Para cada celda de la cuadrícula, YOLO predice un número fijo de bounding boxes.
Cada bounding box está representado por un vector que contiene información sobre la posición (coordenadas x, y, ancho y alto) y la confianza de la detección.
Además de las coordenadas de los bounding boxes, YOLO predice las probabilidades de las clases de objetos presentes en cada bounding box.

4. Cálculo de confianza y umbral de detección:

La confianza de detección se refiere a cuán seguro está el modelo sobre la presencia de un objeto en un bounding box específico.
Se aplica un umbral de confianza para aceptar o rechazar detecciones basadas en la confianza de la red en la precisión de la detección.
Las detecciones que no superan este umbral de confianza se descartan.

5. Supresión de no máximos (Non-Maximum Suppression, NMS):

YOLO utiliza el algoritmo de NMS para eliminar las detecciones redundantes y mantener solo las detecciones más confiables.
Durante este proceso, se consideran las superposiciones entre bounding boxes y se retiene sólo la detección con la mayor confianza.

6. Clasificación y salida final:

Después de aplicar filtros y umbrales, se obtiene la salida final de YOLO que consiste en las bounding boxes con sus coordenadas, las clases detectadas y las probabilidades asociadas.
Estos resultados representan los objetos detectados en la imagen original junto con la confianza en esas detecciones.

¿Qué se hizo?

El objetivo del grupo era implementar el sistema de YOLO v7 visto en clase, originalmente se quería hacer usando código local, pero debido a que no teníamos computadoras lo suficientemente potentes se optó por usar Collab.
En Collab se creó un notebook que se conecta a Google drive y clona un repositorio en una carpeta creada en la unidad de drive, ese repositorio es una versión modificada del repositorio original de YOLO v7.
Luego descarga dos modelos de pesos de sistemas ya entrenados para su uso.
Luego probamos con distintas imágenes y videos, las imágenes y los videos de prueba, tanto como sus resultados se pueden ver en la presentación, así como en el link de drive del repositorio.

¿Qué se modificó del YOLO v7 original?

Se le agregó:
-Contador de FPS para videos
-Discriminador para filtrar por clase o clases
-Texto en el video o imagen del tipo de clases y cantidad detectado
-Cambiado el valor de los parámetros por defecto para mejores resultados

¿Qué puede detectar el algoritmo y qué no puede detectar?

Como se ve en la presentación, puede detectar personas, autos, bicicletas, semáforos, pelotas de deportes, naranjas y sillas, pero no puede detectar palos de hockey, frutillas, lobos marinos e instrumentos musicales.

Conclusión

Tras la ejecución del algoritmo y la evaluación a través de múltiples pruebas con diversas imágenes y videos, se infiere que el incremento en los valores de los umbrales conlleva a una menor detección de objetos, no obstante, con una precisión sustancialmente mayor. En contraste, la utilización de un umbral más bajo propicia la identificación de más objetos, aunque con una propensión a la identificación errónea.

Asimismo, se pudo establecer que los modelos preentrenados sometidos a prueba exhiben un nivel de entrenamiento adecuado, evidenciando su capacidad para detectar objetos con un margen de error reducido.

También cabe aclarar que el sistema preentrenado usa el dataset COCO el cual solamente dispone de los elementos más comunes.

Bibliografía

https://es.wikipedia.org/wiki/Algoritmo_You_Only_Look_Once_(YOLO)

https://ddd.uab.cat/pub/tfg/2017/tfg_71066/paper.pdf

https://www.unite.ai/es/yolov7/

https://github.com/WongKinYiu/yolov7