Sólo miras una vez (YOLO) es un sistema de detección de objetos en tiempo real de última generación. Yolo es extremadamente rápido y preciso.

Cómo funciona

Aplican el modelo a una imagen en múltiples ubicaciones y escalas. Las regiones de la imagen con puntuación alta se consideran detecciones.

Utilizamos un enfoque totalmente diferente. Aplicamos una única red neuronal a la imagen completa. Esta red divide la imagen en regiones y predice cuadros delimitadores y probabilidades para cada región.

YOLO es un popular algoritmo de detección de objetos en imágenes y videos que ha sido ampliamente utilizado en aplicaciones de visión por computadora. La versión 3 de YOLO es una mejora significativa con respecto a sus predecesoras y se lanzó en 2018. A continuación, se presenta un informe sobre YOLOv3:

YOLOv3

Introducción:

YOLO (Yo Only Look Once) es un enfoque de vanguardia en detección de objetos en tiempo real que se centra en la eficiencia y precisión. La tercera versión, YOLOv3, es una mejora significativa en comparación con las versiones anteriores, y ha sido ampliamente adoptada en aplicaciones de visión por computadora y en el campo de la inteligencia artificial.

Características Clave:

1. \*\*Detección en Tiempo Real: YOLOv3 es conocido por su capacidad para detectar objetos en imágenes y videos en tiempo real. Esto lo hace ideal para aplicaciones en tiempo real, como sistemas de seguridad, vehículos autónomos y más.

2. \*\*Detección de Múltiples Clases: Puede detectar una amplia variedad de objetos en una sola pasada, lo que lo convierte en una excelente opción para tareas de detección de múltiples clases.

3. \*\*Mejora de la Precisión: YOLOv3 mejoró significativamente la precisión en comparación con versiones anteriores, lo que lo hace más confiable para aplicaciones críticas.

4. \*\*Detección de Escala Múltiple: YOLOv3 utiliza tres escalas de detección diferentes en lugar de una única escala, lo que mejora la detección de objetos de diferentes tamaños.

5. \*\*Backbone Network: YOLOv3 utiliza una red de convolución más grande como su "backbone" (Darknet-53) para extraer características de las imágenes de entrada.

6. \*\*Uso de Anclajes: La técnica de anclajes se utiliza para mejorar la precisión de la detección al considerar diferentes tamaños y relaciones de aspecto de objetos.

7. \*\*Postprocesamiento: La supresión de no máximos se utiliza para eliminar detecciones duplicadas y mantener solo las detecciones más confiables.

Rendimiento y Uso:

YOLOv3 es ampliamente utilizado en aplicaciones de visión por computadora y ha demostrado un excelente rendimiento en tareas de detección de objetos. Algunos de los casos de uso comunes incluyen:

-Detección de Peatones y Vehículos en Vehículos Autónomos: YOLOv3 se utiliza para detectar peatones, vehículos y otros objetos en tiempo real, lo que es fundamental para la seguridad en vehículos autónomos.

-Vigilancia y Seguridad: En sistemas de vigilancia y seguridad, YOLOv3 se utiliza para detectar actividades sospechosas o personas no autorizadas en áreas vigiladas.

-Clasificación de Objetos en Imágenes y Videos: Se utiliza en aplicaciones que requieren la identificación y seguimiento de objetos específicos en medios visuales, como en la clasificación de productos en cadenas de producción.

- Reconocimiento de Objetos en Drones: Drones equipados con YOLOv3 pueden identificar objetos de interés en tiempo real, lo que es útil en aplicaciones de búsqueda y rescate, agricultura de precisión y más.

Desafíos y Limitaciones:

- YOLOv3, si bien es muy eficiente y preciso, puede ser intensivo en recursos computacionales, lo que puede ser un desafío en entornos con recursos limitados.

- La precisión de la detección puede verse afectada en situaciones de baja iluminación o con objetos parcialmente ocultos.

Código

Programa: Pycharm.

Avance del código para el uso correcto de yolov3.

import cv2  
import numpy  
cap = cv2.VideoCapture(0)  
classesFile = ""  
classesNames = []  
with open(classesFile, "rt")as f:  
 classNames = f.read().rstrip("\n").split("\n")  
while True:  
 success, img = cap.read()  
 cv2.imshow("image",img)  
 cv2.waitKey(1)

con este código logre usar la cámara de mi profesora la cual en el siguiente código ocupare para detectar objetos con yolov3.