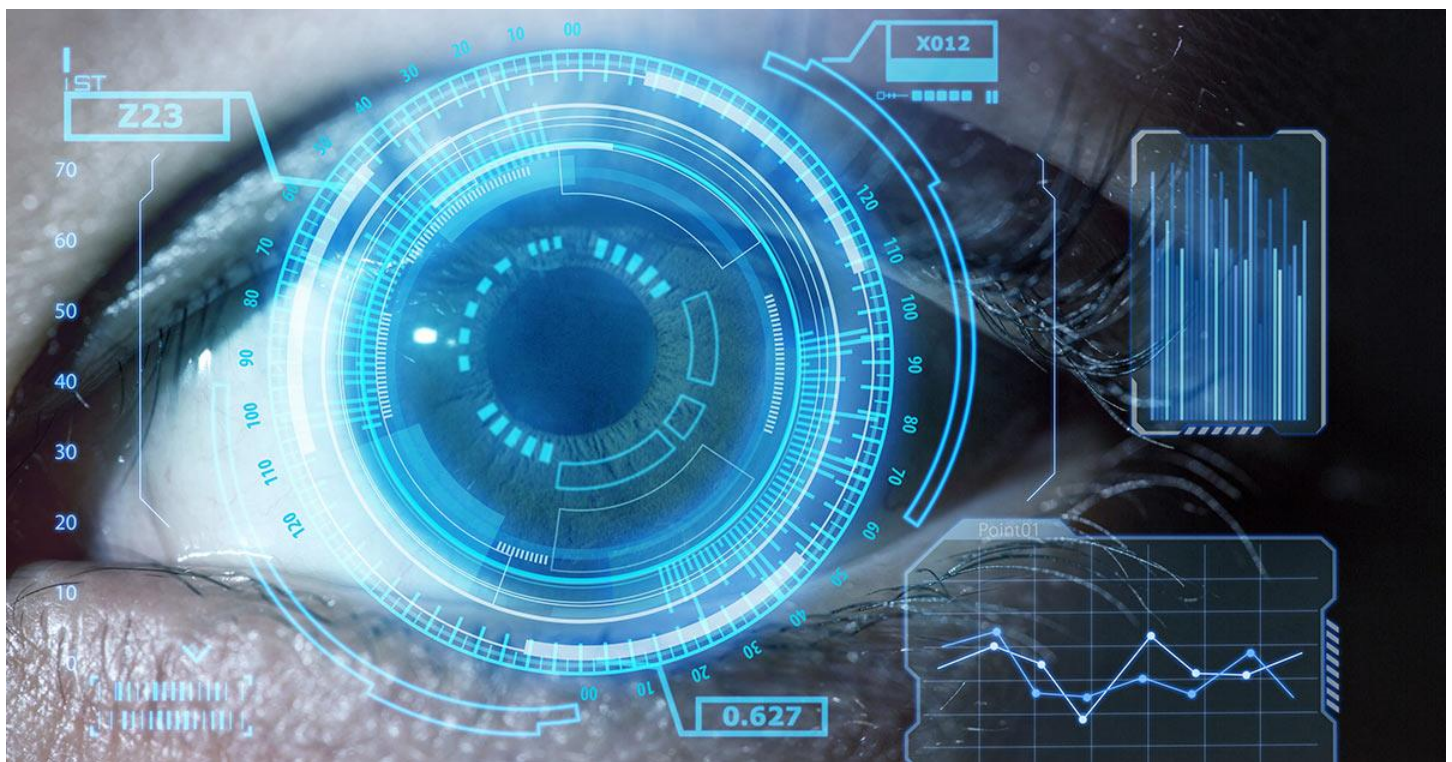


“COMPUTER VISION”

ENG: AHMED MUBARAK

01020451375



SESSION NO. “9”

- PROJECTS

1. Rendering object detection

2. Yolo object detection

ENG.AHMED MUBARAK

01020451375

1. Rendering object detection

```
import cv2
import cvlib
from cvlib.object_detection import draw_bbox

# الفيديو الأصلي ؟
video = cv2.VideoCapture("videos\highway-civil.mp4")
fps = video.get(cv2.CAP_PROP_FPS)
frame_width = int(video.get(cv2.CAP_PROP_FRAME_WIDTH))
frame_height = int(video.get(cv2.CAP_PROP_FRAME_HEIGHT))

# الفيديو الناتج ؟
fourcc = cv2.VideoWriter_fourcc(*'MP4V')
output_video = cv2.VideoWriter(
    'output.mp4', fourcc, fps, (frame_width, frame_height))

# البدء في إنتاج الفيديو ؟
print('Writting...')

while True:
    ret, frame = video.read()
    # frame = cv2.resize(frame, (800, 500))
    if not ret:
        break

    #! لا تحول الصورة الى ابيض و اسود
    # grey = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    #! استخدم نموذج اصغر بدلا من ولكن الدقة ستصبح اقل بكثير yolov3
    #! yolov3 : 10 دقائق
    #! yolov3-tiny : دقيقة

    bounding_box, label, conf = cvlib.detect_common_objects(
        frame, model='yolov3-tiny') # , enable_gpu=True

    # 3 انظر السطر رقم ؟
    frame = draw_bbox(frame, bounding_box, label, conf, write_conf=True)

    # المدة المستغرقة كبيرة جدا لأنى لا امالك كارت شاشة ؟
    # أضعاف 10 الى 5 بوجود كارت شاشة ستضعاف قوة المعالجة بمقدار من ؟
    output_video.write(frame)

print('Done.')
```

2. Yolo object detection from image

```
import cv2
import numpy as np
# Load Yolo
net = cv2.dnn.readNet("yolov3.weights",
                     "cfg/yolov3.cfg")

classes = []
with open("coco.names", "r") as f:
    classes = [line.strip() for line in f.readlines()]
layer_names = net.getLayerNames()
output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]
colors = np.random.uniform(0, 255, size=(len(classes), 3))

# Loading image
img = cv2.imread("mm.jpg")
img = cv2.resize(img, (800, 800))
height, width, channels = img.shape
cv2.imshow("image", img)
cv2.waitKey(0)

# Detecting objects
blob = cv2.dnn.blobFromImage(
    img, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
for b in blob:
    for n, img_blob in enumerate(b):
        cv2.imshow(str(n), img_blob)

net.setInput(blob)
outs = net.forward(output_layers)
# print(outs)

# # Showing informations on the screen
class_ids = []
confidences = []
boxes = []
for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > 0.9:
            # Object detected
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[2] * width)
            h = int(detection[3] * height)
            cv2.circle(img, (center_x, center_y), 10, (0, 255, 0), 2)

            # Rectangle coordinates
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)
            cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            class_ids.append(class_id)
print(len(boxes))
```

```

indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.4, 0.3)
print(indexes)
for i in range(len(boxes)):
    if i in indexes:
        x, y, w, h = boxes[i]
        label = str(classes[class_ids[i]])
        confidence = confidences[i]
        color = colors[class_ids[i]]
        cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
        cv2.putText(img, label + " " + str(round(confidence, 2)),
                    (x, y + 30), cv2.FONT_ITALIC, 1, color, 2)

cv2.imshow("Image", img)
cv2.waitKey(0)

```

3. Yolo object detection “real time”

```

import cv2
import numpy as np
import time

# Load Yolo
net = cv2.dnn.readNet("yolov3.weights",
                    "cfg/yolov3.cfg")

classes = []
with open("coco.names", "r") as f:
    classes = [line.strip() for line in f.readlines()]
layer_names = net.getLayerNames()
output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]
colors = np.random.uniform(0, 255, size=(len(classes), 3))

# Loading image
cap = cv2.VideoCapture(1)

font = cv2.FONT_HERSHEY_PLAIN
starting_time = time.time()
frame_id = 0
while True:
    _, frame = cap.read()
    frame_id += 1

    height, width, channels = frame.shape

    # Detecting objects
    blob = cv2.dnn.blobFromImage(
        frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
    net.setInput(blob)
    outs = net.forward(output_layers)

    # Showing informations on the screen
    class_ids = []
    confidences = []
    boxes = []
    for out in outs:
        for detection in out:
            scores = detection[5:]

```

```

class_id = np.argmax(scores)
confidence = scores[class_id]
if confidence > 0.2:
    # Object detected
    center_x = int(detection[0] * width)
    center_y = int(detection[1] * height)
    w = int(detection[2] * width)
    h = int(detection[3] * height)

    # Rectangle coordinates
    x = int(center_x - w / 2)
    y = int(center_y - h / 2)

    boxes.append([x, y, w, h])
    confidences.append(float(confidence))
    class_ids.append(class_id)

indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.8, 0.3)

for i in range(len(boxes)):
    if i in indexes:
        x, y, w, h = boxes[i]
        label = str(classes[class_ids[i]])
        confidence = confidences[i]
        color = colors[class_ids[i]]
        cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
        cv2.putText(frame, label + " " + str(round(confidence, 2)),
                    (x, y + 30), font, 3, color, 3)

elapsed_time = time.time() - starting_time
fps = frame_id / elapsed_time
cv2.putText(frame, "FPS: " + str(round(fps, 2)),
            (10, 50), font, 4, (0, 0, 0), 3)
cv2.imshow("Image", frame)
key = cv2.waitKey(1)
if key == 27:
    break

cap.release()
cv2.destroyAllWindows()
4.

```

WITH MY BEST WISHES
 ENG/AHMED MUBARAK 😊