הקוד שכתבת מתבסס על מודל **Haar Cascade** שמובנה בספריית OpenCV, עם הקובץ haarcascade\_frontalface\_default.xmlלזיהוי פנים. אם הוא מזהה רכב במקום אדם, זה נובע ככל הנראה אחת הסיבות הבאות:

**סיבות אפשריות:**

1. **זיהוי שגוי** (False Positive):
   * המודל לא מספיק מדוייק ויכולות אובייקטים ספציפיות פנים כפנים, במיוחד אם יש דמיון מסוים בצורות שבתמונה (כמו פנסים של רכב).
2. **תמונה עם רזולוציה נמוכה או תאורה לקויה** :
   * רזולוציה או תאורה לא טובותות להקשות על המודל לזהות פנים.
3. **אזור זהה דומה לפנים** :
   * אם חלק מהרכב בתמונה (לדוגמה, פנס קדמי) דומה למבנה של פנים, המודל עשוי לטעות.

**פתרונות ושיפורים:**

1. **שימוש במודל מתקדם יותר** :
   * נסי להשתמש ב-YOLO או ב-Deep Learning במקום ב-Haar Cascade. זה עשוי לתת זיהוי מדויק יותר, במיוחד אם אתה כבר משתמש ב-YOLO ב שלך.
2. **שימוש ב-Haar Cascade מתאים** :
   * בדקי אם המודל המתאים ביותר נטען. אפשר שכדאי לבדוק את המודל haarcascade\_upperbody.xmlאם מדובר בתמונה עם בני אדם בת נוחות שונות.
3. **כונון הפרמטרים** :
   * אתה יכול לשחק עם הפרמטרים scaleFactorו- minNeighborsכדי להפחית זיהויים שגויים. לדוגמה:

פִּיתוֹן

העתק קוד

faces = face\_cascade.detectMultiScale(

gray,

scaleFactor=1.2, # מגדיל את הרגישות

minNeighbors=8, # מפחית זיהויים שגויים

minSize=(50, 50) # גודל מינימלי גדול יותר לפנים

)

1. **בדיקת המיקום הזהה** :
   * הדפיסי את המיקום והגודל של האזורים שזוהו. כך תוכל לבדוק אם מדובר באזור הגיוני או לא:

פִּיתוֹן

העתק קוד

print("Faces found:", faces)

**דוגמה שימוש ב-YOLO לזיהוי פנים:**

אם את כבר עובדת עם YOLO, ניתן לאמן או להשתמש במודל שהוא מראש לזיהוי פנים.

רוצה שאכין לך קוד לדוגמה עם YOLO?

4o

**אמרת:**

אז תביא לי קוד שמזהה דמות

**ChatGPT אמר:**

ChatGPT

כדי לזהות דמות (בן אדם) בצורה מדויקת, עדיף להשתמש במודל כמו **YOLOv8** שמסוגל לזהות דמויות ואובייקטים באופן כללי. הנה דוגמה שימוש ב-YOLOv8 לזיהוי דמויות:

**התקנת הדרישות:**

לַחֲבוֹט

העתק קוד

pip install ultralytics opencv-python

**קוד לזיהוי דמות עם YOLOv8:**

פִּיתוֹן

העתק קוד

from ultralytics import YOLO

import cv2

# טוענים את מודל YOLOv8 שהוכן מראש (coco dataset)

model = YOLO('yolov8n.pt') # ניתן להחליף ל-'yolov8s.pt' או גרסאות אחרות למודלים חזקים יותר

# טוענים את התמונה

image\_path = "C:/Users/WIN 10/Desktop/HadasProject/image/2.jpg"

img = cv2.imread(image\_path)

# מבצעים ניתוח (inference) על התמונה

results = model(img)

# לולאה על כל הדמויות שנמצאו

for result in results[0].boxes:

cls = int(result.cls[0]) # הקטגוריה שזוהתה

conf = result.conf[0] # מידת הביטחון בזיהוי

if cls == 0: # קטגוריית "אדם" (class 0 בקוקו)

# מציירים מלבן סביב הדמות

x1, y1, x2, y2 = map(int, result.xyxy[0])

cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)

cv2.putText(img, f"Person {conf:.2f}", (x1, y1 - 10),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 255, 0), 2)

# מציגים את התוצאה

cv2.imshow('Detected Persons', img)

cv2.waitKey(0)

cv2.destroyAllWindows()

**מה עושה הקוד:**

1. **המודל** : הקוד משתמש ב- yolov8n.ptשהוא מודל YOLOv8 קטן ומהיר יכול להחלפה לדגם מדויק יותר כמו yolov8m.pt.
2. **זיהוי בני אדם** : הוא מסנן זיהויים ומשאיר רק את הקטגוריה של בני אדם (מחלקה מזהה 0).
3. **ציור תיבה ומידע** : הקוד מצייר מלבן סביב כל דמות ומוסיף את אחוז הביטחון בזיהוי.

**תוצאות טובות יותר:**

* אם התמונה ברזולוציה נמוכה, שקלי להגדיל אותה לפני הניתוח.
* אם יש עומס בזיהויים, ניתן לשנות את רמת הביטחון באמצעות model.predict(source=image, conf=0.5).

רוצה עזרה נוספת או הדגמה לסרטון בזמן אמת?

4o

**אמרת:**

האם הקוד הזה מתאים גם לחושך?

**ChatGPT אמר:**

ChatGPT

הקוד שהצעתי, המבוסס על **YOLOv8** , יכול לזהות דמויות גם בתנאים של תאורה נמוכה (חושך), אך הביצועים תלויים בכמה גורמים חשובים:

**גורמים המשפיעים על זיהוי בתנאי חושך :**

1. **תמונה באיכות** :
   * אם התמונה מטושטשת או חשוכה מדי, תהיה קשה למודל לזהות דמויות.
   * מצלמות עם חיישנים איכותיים יותר משפרות את הביצועים.
2. **תאורה חלשה** :
   * תאורה חלשה עלולה להפחית את האדם של המודל לזהות דמויות. כדאי להוסיף עיבוד, כמו הגברת בהירות או שיפור ניגודיות.
3. **שימוש במצלמות לראיית לילה (אינפרא אדום)** :
   * אם יש גישה למצלמות לראיית לילה, אפשר להשתמש. הן מספקות תמונות ברורות יותר בחושך.
4. **אימון מודל מותאם** :
   * YOLO מאומן על מערך נתונים (COCO) מכיל תמונות עם תאורה רגילה. אם זיהוי בתנאי חושך הוא קריטי, אפשר לאמן מחדש את המודל עם דוגמאות מתאימות לתאורה נמוכה.

**שיפור הביצועים בתנאי חושך :**

תוספת עיבוד לתמונה יכולה לשפר את התוצאה:

פִּיתוֹן

העתק קוד

import cv2

from ultralytics import YOLO

# טוענים את התמונה

image\_path = "C:/Users/WIN 10/Desktop/HadasProject/image/2.jpg"

img = cv2.imread(image\_path)

# שיפור תאורה וניגודיות

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

enhanced = cv2.equalizeHist(gray) # שיפור ניגודיות באמצעות histogram equalization

enhanced\_colored = cv2.cvtColor(enhanced, cv2.COLOR\_GRAY2BGR) # המרה בחזרה ל-BGR

# טוענים את YOLO ומבצעים ניתוח על התמונה המשופרת

model = YOLO('yolov8n.pt')

results = model(enhanced\_colored)

# לולאה על תוצאות ומציירים זיהויים

for result in results[0].boxes:

cls = int(result.cls[0])

conf = result.conf[0]

if cls == 0: # זיהוי אדם

x1, y1, x2, y2 = map(int, result.xyxy[0])

cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)

cv2.putText(img, f"Person {conf:.2f}", (x1, y1 - 10),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 255, 0), 2)

# מציגים את התוצאה

cv2.imshow('Detected Persons', img)

cv2.waitKey(0)

cv2.destroyAllWindows()

**הסברים נוספים על שיפורים בקוד :**

1. **שיפור ניגודיות עם היסטוגרמה שוויון** :
   * שימוש ב- cv2.equalizeHistמשפר את הנראות בתנאים של תאורה חלשה.
2. **שימוש באינפרא אדום או ראיית לילה** :
   * אם התמונות מתקבלות מצלמות ייעודיות לתאורה נמוכה, ניתן לעבד אותן למעשה.

**שיפור נוסף - מודל המותאם לתאורה נמוכה :**

* קיימים מודלים מתקדמים אחרים שמתמחים בזיהוי בתנאים קשים, לדוגמה **YOLO מאומן על מערך נתונים Dark COCO** .
* אפשר גם לשלב רשתות שיפור תמונה (רשתות לשיפור תמונה) לפני ניתוח התמונות.

רוצה עזרה באימון מודל מותאם או עבודה עם מצלמות לראיית לילה?

4o

**אמרת:**

יש לי פרויקט על רובוט צבאי המאתר מטענים ודמויות המפה את המנהרה בזמן אמת יש כאלה פרוייקטים שאנשים עשו? תביא לי קישור לגיטהב

**ChatGPT אמר:**

ChatGPT

רובוטים צבאיים לאיתור מטענים, זיהוי דמויות ומיפוי מנהרות בזמן אמת הוא תחום מתקדם ומורכב, ולעיתים פיתוח קרובים פרויקטים מסוג זה הם זמינים לציבור הרחב בשל שיקולי ביטחון. עם זאת, קיימים פרויקטים בקוד פתוח המבקרים בטכנולוגיות דומות, כמו רובוטיקה, זיהוי אובייקטים ומיפוי, מקומות לשמש כבסיס או כהשראה לפיתוח פרויקט משלך.

**להלן מספר פרויקטים רלוונטיים מ-GitHub:**

1. **SLAM (לוקליזציה ומיפוי סימולטני):**
   * פרויקטים בתחום ה-SLAM עוסקים במיפוי סביבות לא ידועות תוך כדי ניווט. טכנולוגיה זו יכולה להיות שימושית למיפוי מנהרות.
   * דוגמה: [RTAB -Map](https://github.com/introlab/rtabmap) - ספרייה בקוד פתוח למיפוי בזמן אמת.
2. **זיהוי אובייקטים באמצעות למכונה:**
   * פרויקטים משתמשים במודלים כמו YOLO לזיהוי אובייקטים בתמונות או וידאו. ניתן להתאים מודלים אלו לזיהוי מנטים או דמויות.
   * דוגמה: [YOLOv5](https://github.com/ultralytics/yolov5) - גרסה מתקדמת של מודל YOLO לזיהוי אובייקטים.
3. **רובוטיקה עם ROS (מערכת הפעלה רובוט):**
   * פרויקטים משתמשים ב-ROS לפיתוח רובוטים עם אפשרות ניווט, זיהוי ומיפוי.
   * דוגמה: [TurtleBot3](https://github.com/ROBOTIS-GIT/turtlebot3) - פלטפורמת רובוטיקה בקוד פתוח עם תיעוד נוסף.
4. **מערכות רובוטיות לאיתור מוקשים:**
   * בעוד שמוקשים אינם מנטנים, טכנולוגיות המשמשות לאיתורם עשויות להיות דומות.
   * דוגמה: [Mine Detector Robot](https://github.com/RoboticsClubIITJ/Mine-Detector-Robot) - פרויקט של רובוט לאיטור מוקשים.

**הערות חשובות:**

* **התאמה לצרכים ספציפיים:** ייתכן שתתאים להתאים את הפרויקטים הללו לצרכים המדויקים של הפרויקט שלך, במיוחד בכל הנוגע לזיהוי מטענים או מיפוי מנהרות.
* **שיקולי בטיחות וביטחון:** בעת פיתוח רובוטים לצרכים צבאיים או ביטחוניים, יש לקחת בחשבון שיקולי בטיחות, אמינות ואתיקה.
* **למידה נוספת:** מומלץ לעיין בתיעוד של הפרויקטים הללו, להשתתף בקהילות המקוונות שלהם, וללמוד מהניסיון של מפתחים אחרים בתחום.

שילוב של אלו טכנולוגיות, יחד עם התאמות ספציפיות, לעזור בפיתוח רובוט המסוגל לאתר מנטים, לזהות דמויות ולמפות מנהרות בזמן אמת.

מקורות

4o

**אמרת:**

תביא לי קוד לזיהוי מטעני חבלה

**ChatGPT אמר:**

ChatGPT

זיהוי מטעני חבלה (IEDs) הוא אתגר טכנולוגי מורכב שדורש פתרון מותאם, והוא לרוב מערב שילוב של טכנולוגיות כמו ראייה ממוחשבת, חיישנים פיזיים, ובינה מלאכותית. בעוד שאין "פתרון קסם" בזיהוי מטענים, ניתן לבנות מערכת אב-טיפוס המבוססת על זיהוי אובייקטים כמו YOLOv8.

**פתרון אפשרי: זיהוי מנטים באמצעות למכונה מכונה**

נתחיל בהבנה כיצד ניתן להשתמש ב-YOLO לזיהוי מטענים:

1. **איסוף ואימון נתונים** : יש לאסוף תמונות של מטעני חבלה (IEDs) ולתייג אותם.
2. **אימון מודל YOLO** : המודל מאומן לזהות את המאפיינים הייחודיים של מנטים.
3. **שימוש במודל המותאם** : המודל ישמש לניתוח בזמן אמת.

**קוד דוגמה - שימוש במודל מוכן**

אם עדיין אין לך מודל מותאם אישית, אפשר להתחיל עם מודל מוכן (כמו COCO), אתה יכול להתאים אותו.

פִּיתוֹן

העתק קוד

from ultralytics import YOLO

import cv2

# טוענים את YOLOv8 עם מודל מאומן מראש

model = YOLO('yolov8n.pt') # ניתן להחליף למודל שאומן למטענים

# נתיב התמונה לניתוח

image\_path = "C:/path\_to\_your\_image.jpg"

img = cv2.imread(image\_path)

# הרצת ניתוח (inference)

results = model(img)

# לולאה לזיהוי אובייקטים בתמונה

for result in results[0].boxes:

cls = int(result.cls[0]) # מזהה הקטגוריה

conf = result.conf[0] # רמת הביטחון

if cls == TARGET\_CLASS\_ID: # החלף ל-ID של מטענים (בהתאמה לאימון שלך)

# מקבלים את הקואורדינטות של תיבת הזיהוי

x1, y1, x2, y2 = map(int, result.xyxy[0])

# מציירים תיבה סביב הזיהוי

cv2.rectangle(img, (x1, y1), (x2, y2), (0, 0, 255), 2)

cv2.putText(img, f"IED {conf:.2f}", (x1, y1 - 10),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 0, 255), 2)

# מציגים את התוצאה

cv2.imshow('IED Detection', img)

cv2.waitKey(0)

cv2.destroyAllWindows()

**שלבים להתאמה אישית**

1. **איסוף תמונות והכנה** :
   * אסוף תמונות של מטעני חבלה ומגוון תמונות אחרות למניעת זיהויים שגויים.
   * תייג את התמונות באמצעות כלים כמו [LabelImg](https://github.com/heartexlabs/labelImg) .
2. **אימון מודל YOLO** :
   * השתמש בנתונים מתויגים כדי לאמן את YOLO.
   * אפשר למצוא מדריכים לאימון YOLO [כאן](https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data) .
3. **פריסה ושימוש** :
   * לאחר אימון המודל שלך, החלף את הקובץ yolov8n.ptבקובץ המודל המאומן.

**הערות חשובות:**

* **אתיקה ואחריות** :
  + פרויקט כזה דורש רגישות ואחריות, במיוחד אם יש לו ישומים בטחוניים.
* **מגבלות** :
  + אם אין נתונים איכותיים או אם התמונות לא מתארות מטענים בפירוט מספיק, המודל עשוי לא להצליח לזהות אותם.

רוצה עזרה באימון מודל מותאם אישית?

4o

ראש הטופס



O

תחתית הטופס