תדריך קדם ראיה

בתדריך זה תבנו את ספריית הקוד שתעזור לכם בבצוע המטלות של תדריך הראייה.
כאשר תבצעו את תדריך ראייה תשתמשו בספרייה שתבנו בעזרת תדריך זה.
מומלץ לשמור את כל התוכניות שלכם בקובץ יחיד, בעל שם משמעותי (myVisionLib.py למשל).
לביצוע הקוד בבית עליכם להצטייד במחשב עם מצלמה, ולהוריד את cv2.
המטלות כולן ניתנות לביצוע בבית ודורשות רק מצלמת רשת (מה שלרוב יש מובנה במחשבים ניידים).
כלל המטלות וכמו גם הסברים על התקנות נלוות ומטרות המעבדה מפורטים בסרטונים המצורפים כקישורים לאורך המסמך.

הסבר אודות מעבדה זו – כאן.
הסבר אודות התקנת חבילת OPEN-CV – כאן.

נתון קוד להצגת תמונה במחשב.

```
22  pic=cv2.imread('path')
23  cv2.imshow('gray',pic)
24  cv2.waitKey(0)
25  cv2.destroyAllWindows()
26
```

הסבר ע"פ מס' שורה:
22 קריאת תמונה לזכרון המחשב התמונה נמצאת בנתיב המסופק כפרמטר למתודה.
23 הצגת התמונה בחלון שנקרא gray.
24 כל עוד לא נלחץ מקש התמונה מוצגת.
25 פינוי זכרון המחשב.

- הצגת וידאו בזמן אמת בחלון על המחשב:

```
1   import cv2
2
3   cap = cv2.VideoCapture(0)
4
5   while(True):
6       # Capture frame-by-frame
7       ret, frame = cap.read()
8       # Display the resulting frame
9       cv2.imshow('frame',frame)
10      if cv2.waitKey(1) & 0xFF == ord('q'):
11          break
12
13  # When everything done, release the capture
14  cap.release()
15  cv2.destroyAllWindows()
```

הסבר ע"פ מספר השורה:

נכתב ע"י גד הלוי.

3. יצירת אובייקט המצלמה בהנחה שהמצלמה נמצאת בפורט 0, אם הקוד לא עובד צריך לשנות את המספר עד 9.
אם מציגים וידאו קיים, במקום מספר הפורט יש לכתוב את כתובת הוידאו.
7. ret מחזיר האם הפעולה הצליחה, frame תמונה בודדת, רצף של frames מרכיב את הסרט.
9. הצגת הסרט בחלון בשם frame על המחשב.
10. waitKey מציגה את התמונה על המסך למס' המילי שניות שנשלחות אליה כפרמטר (אצלנו 1 מ"ש).
אלא אם נלחץ המקש q.

הסבר המרת פורמט לתמונה:
משתמשים בפונקציה `cv2.cvtColor(input_image, flag)`.
ב-`opencv` הצבע מוצג בפורמט של BGR כאשר הצבע הכחול ראשון, והאדום אחרון.
ב-`opencv` יש מעל `150` פורמטים. הרבה פונקציות ב-`opencv` דורשות שקלט התמונה יהיה בסולם אפור או אפילו בשחור לבן.

מטלה1:
כתוב תכנית:
- תציג וידאו בזמן אמת על המסך.
- בלחיצה על מקש s תשמר תמונה במחשב בנתיב שיוגדר לפונקציה, ובפורמט שיוגדר לה.(השתמש בפונקציה imwrie).
- בלחיצה על מקש q התכנית תסתיים וזיכרון המחשב ינוקה.

התכנית תקבל כפרמטרים:
1. נתיב.
2. פורמט כלשהו שבו התמונה תשמר.
לנוחותכם נתון שלד התכנית (מטלה 1 במודל) עליכם להשלים במקומות המסומנים.
הסבר אודות מטלה זו – כאן.

מטלה 2:
נתון קוד מה-`tutorial` של `openCv` שמטרתו לצייר ע"י העכבר.
העתק את הקוד במחשב והרץ אותו.
שנה קוד זה כך שניתן יהיה לבצע חיתוך אובייקט מתמונה ע"י העכבר.
עזרה:
בלחיצה על לחצן שמאל של העכבר, המשתמש עובר למצב סימון מלבן, הסימון מוצג על המסך כאשר המשתמש עוזב את הלחצן השמאלי.
בלחיצה על הלחצן הימני של העכבר התמונה נחתכת ונשמרת על הדיסק. השתמש לחיתוך התמונה ב-slice של פייטון תחת numpy. דוגמה לשמירת 100*100 פיקסלים של תמונה pic[:100,:100].
שים לב שהקורדינטות של numpy הפוכות מקורדינטות הסימון של opencv.
לחיצה כפולה על הלחצן הימני מוחקת את הסימנים הישנים של החיתוך.
לנוחותכם נתון שלד התכנית (מטלה 2 במודל) עליכם להשלים במקומות המסומנים.

הסבר אודות מטלה זו – כאן.

מטלה 3:
נתון קוד מה-`tutorial` שמטרתו להרכיב צבע על המסך בהתאם לבחירת המשתמש.
העתק את הקוד והפעל אותו.
על סמך קוד זה בנה יישום שבאמצעותו תוכל לקנפג את המצלמה בזמן אמת.
הפרמטרים לקינפוג: `contrast, saturation, brightness`.
קינפוג המצלמה למרכיב ה-`saturation` לערך `0.2`:

```
cam=cv2.VideoCapture(0)
cam.set(cv2.CAP_PROP_SATURATION,0.2)
```

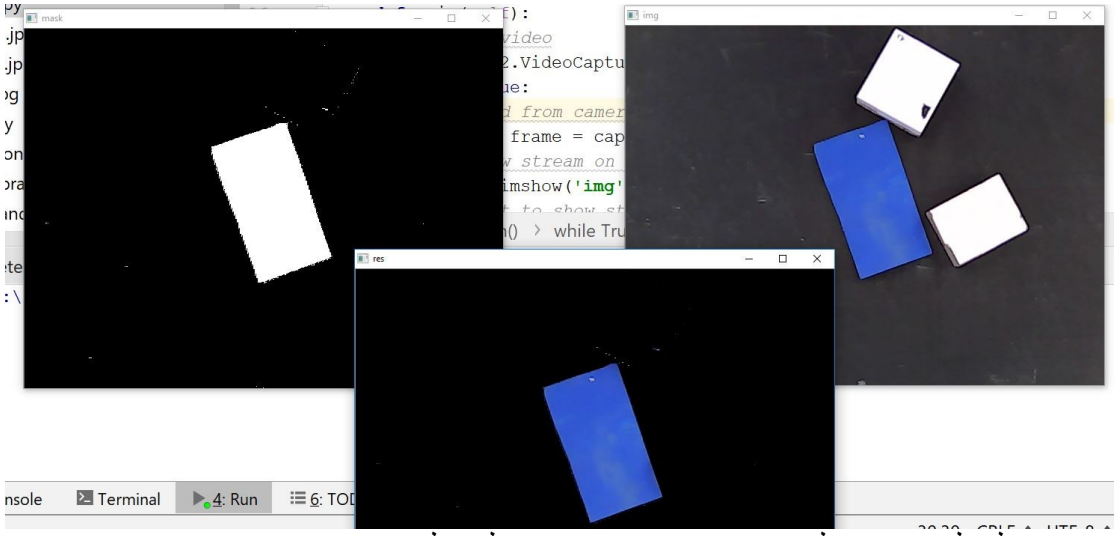

נכתב ע"י גד הלוי.

לנוחותכם נתון שלד התכנית (מטלה 3 במודל) עליכם להשלים במקומות המסומנים.

הסבר אודות מטלה זו – כאן.

מטלה 4:

זיהוי ע"פ צבע: נתון קוד תכנית של זיהוי אובייקט ע"פ צבע.
צור מערכת עקיבה דינמית לצבע, שמבוססת על הקוד לעיל. בחירת הצבע תתבצע בעזרת הלחצן השמאלי של העכבר בחלון של התמונה המקורית.



בתמונה לעיל מוצגים שלושת מסכי הוידאו שיפתחו לכם לאחר בחירת הצבע.
המסך הראשון מימין הינו המסך המקורי, המשתמש בחר בעזרת הלחצן השמאלי של העכבר את הצבע הכחול במסך המקורי, לאחר הבחירה נפתחו שני המסכים הנוספים, השמאלי שהינו מסך ה"מסכה" mask,
והימני שהינו מסך התוצאה של מעקב הצבע הדנמי שבו רואים רק את העצם הכחול.
לנוחותכם נתון שלד התכנית (מטלה 4 במודל) עליכם להשלים במקומות המסומנים.
הסבר אודות מטלה זו – כאן.

מטלה 5:

זיהוי ע"פ התאמת תכונות (feature matching): נתון קוד של זיהוי אובייקט ע"י התאמת תכונות.
השלם את הקוד בקובץ מטלה 5 כך שהתוכנה תזהה ותסמן במלבן ירוק את האובייקט שהוגדר מראש (בתמונה croped.jpg) בתמונה המקורית (בצבע), מרכז הכבד של האובייקט שהוגדר מראש יסומן בעיגול כחול.
בנוסף התוכנה תקבל כפרמטר רשימה עם שתי שיטות זיהוי (surf,sift) ותבצע את הזיהוי בשתי השיטות, ותחזיר את נקודות מרכז הכובד של האובייקט, בשתי השיטות, עליך לבחור את השיטה העדיפה משתי השיטות, ולהשתמש בה. (ראה עדכון למטה).
היתרון של שיטת feature-match על פני זיהוי צבע רגיל – בשיטה זאת ניתן לזהות חלקים מקושקשים בעלי צבעים שונים (כלומר לא רק חלקים בעלי צבע אחיד כמו שרואים בסרטון!).
הסבר אודות מטלה זו – כאן.
עדכון: שתי השיטות (surf,sift) נכון לעכשיו לא זמינות (זכויות יוצרים), יש להשתמש בשיטות אחרות. אנא השתמשו בשיטות Orb, Kaze, Akaze, Brisk קובץ מטלה 5 שונה בהתאם.

נכתב ע"י גד הלוי.

מטלה 6:

זיהוי חלק ע"פ קונטורים (קווי מתאר).

מצ"ב ה-tutorial בנושא קונטורים.

השלם את הקוד בקובץ מטלה 6, הפוך קוד זה למחלקה שמקבלת כקלט את הנתיב של התמונה, ו-n מספר חלקים.

המחלקה תחזיר תמונה עם קווי מתאר כחולים, ועיגולים אדומים במקומות של מרכזי הכובד, ותחזיר את ערכי מרכזי הכובד של n החלקים בעלי ההיקף הגדול ביותר.

לנוחותכם מצ"ב הקוד הבסיס מה-Tutorial עם הסברים, בנוסף להסברים המצויים בקובץ המטלה.

הסבר אודות מטלה זו – כאן.

myWork [~/tensorflow/myWork] - ~/.PyCharmCE2019.1/config/scratches/scratch_3.py - PyCharm

Help

scratch_3

myTf.py | res.html | ex.html | scratch_3.py

```
218 # detect(['sift','surf'])
219
220 import cv2
221 im=cv2.imread('img1.jpg')
222 imgray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
223 ret, thresh = cv2.threshold(imgray, 127, 255, 0)
224 image, contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
225 mmuyan=sorted(contours, key= lambda c : cv2.arcLength(c,True),reverse=True )
226 for i in range(3):
227     rect = cv2.minAreaRect(mmuyan[i])
228     box = cv2.boxPoints(rect)
229     box = np.int0(box)
230     pic = cv2.drawContours(im, [box], 0, (0, 0, 255), 2)
231     M=cv2.moments(mmuyan[i])
232     cx=int(M['m10']/M['m00'])
233     cy=int(M['m01']/M['m00'])
234     pic=cv2.circle(pic,(cx,cy),3,(0,255,0),2)
235 # cv2.imshow('img',img)
236 cv2.imshow('pic',pic)
237 cv2.waitKey(0)
238 cv2.destroyAllWindows()
```

הסבר:

222 המרת התמונה לאפור.

223 המרה לתמונה בינארית ערך מעל 127 יצבע בלבן (255) מתחת בשחור (0), thresh היא תמונת התוצאה.

224 מציאת קווי המתאר. הפרמטר הראשון מקצה שיטה מסוימת לתיאור ההיררכיה (לא מעניין אותנו), השני מקצה שיטה לחסוך בזכרון, מקצה 4 נקודות לקונטור בצורת מלבן במקום הקצעת נקודות רצף נקודות לכל קו של המלבן.

225 מיון הקונטורים בסדר יורד מבעל ההיקף הגדול ביותר לבעל ההיקף הקטן ביותר.

226-234 לכל קונטור מוצאים את המלבן המינימלי שמקיף אותו, מציירים מלבן זה, מחשבים מומנטים, בעזרת המומנטים מחשבים מרכז כובד, מציירים את מרכז כובד זה.

מטלה 7:

השלם את הקוד בקובץ מטלה 7, הוסף למחלקה זאת יכולת להחזיר את זוית החלק כאשר הזוית מוגדרת כזוית של המימד הצר של המלבן, בנוסף זוית החלק תופיע בכיתוב על התמונה.

מטרת חישוב הזוית שהצלע הקצרה יוצרת עם הציר החיובי של ציר ה-X – כדי שהגריפר של הרובוט יוכל לתפוס את החלק המזוהה.
הסבר אודות מטלה זו – כאן.

מטלה 8:
אגד את כל התוכניות שכתבת לקובץ יחיד, ושמור קובץ זה בתוך תקייה.
בקובץ זה תשתמש במעבדה הבאה שלך.

להגשה ב-Verifier:
- סרטים של מטלות 1-7.
- קוד של מטלה 8.

בהצלחה!!!