Parking Protector

מערכת לזיהוי מצב חניה באמצעות עיבוד תמונה

GitHub link
Slides link

מטרת הפרויקט

זיהוי מצב חניה מסוימת ברגע נתון ועדכון הלקוח.
באיטרציה הראשונה, מטרת המערכת היא לאפשר לבעלי חניות פרטיות (ביניהם, בעלי מוגבלויות) לקבל אינדיקציה על מצב החניה בזמן נתון (האם החניה פנויה או תפוסה).
בהמשך, צפויה המערכת לזהות את מצב החנייה בזמן אמת ולדווח למערכת התראות על מנת לאפשר לנהגים למצוא חניה פנויה או לעדכן על חניות פנויות בחניונים מסחריים.

משתמשים פוטנציאלים

- בעלי חניות פרטיות- המערכת מזהה שחנייה נתפסה ושולחת התראה בדוא"ל, מסרון, או בעזרת אפליקציה לטלפון.
- חניונים מסחריים- מערכת מנורות, כך שמעל כל חניה יש מנורה אחת. כאשר החניה פנויה ידלק אור ירוק וכאשר החניה תפוסה, ידלק אור אדום.
- אפליקציות ניווט - כדי לכוון נהגים לחניות פנויות (חניות רחוב או חניונים פרטיים)

איך זה עובד?

- אתחול המערכת
  - הכנסת תמונה של החניה הריקה
  - סימון של שטח החניה המדויק
- בכל איטרציה
  - המערכת מקבלת תמונה של מצב החניה הנוכחי
  - השוואה של התמונה הנוכחית לתמונה הקודמת, אם יש (אם אין, אז ההשוואה היא לתמונה המקורית)
  - זיהוי מצב החניה הנוכחי
  - דיווח למשתמש על מצב החניה (עוד לא מומש)

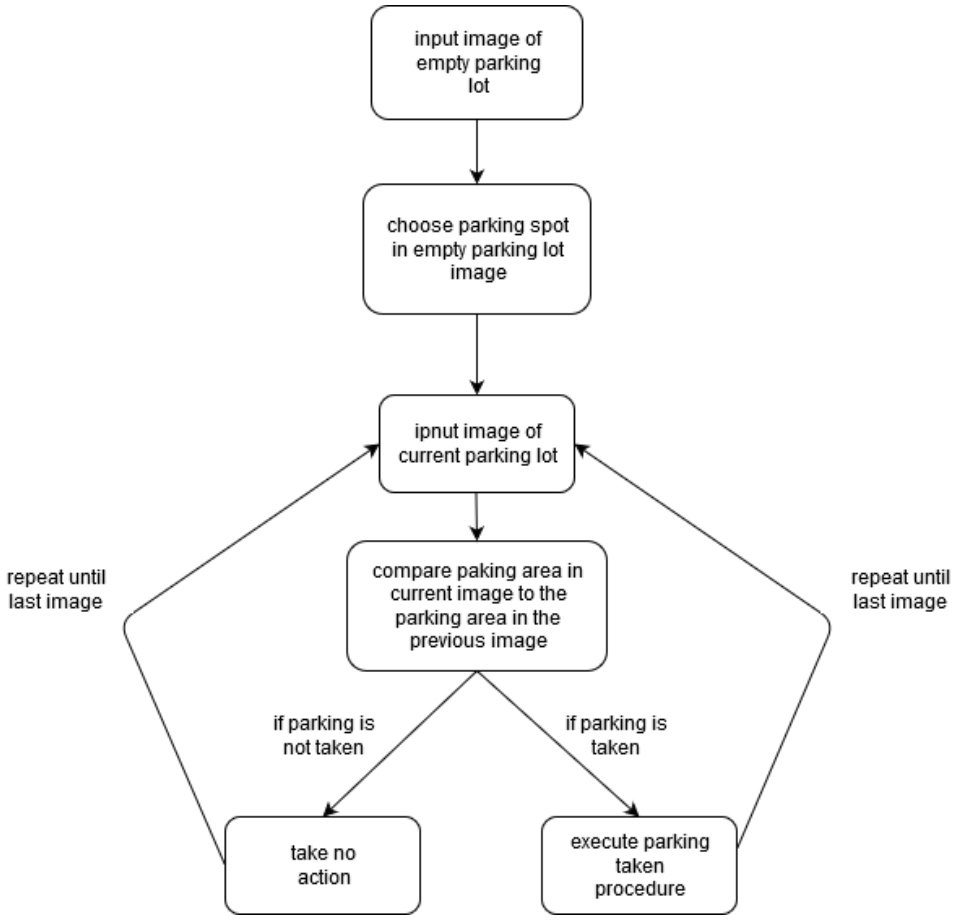
דרישות המערכת

- שרת ועליו מותקנים -
  - Python3.6 ומעלה.
  - Git
  - חבילת הפייתון openCV-python בגרסא 4.1 ומעלה.
  - חבילת matplotlib מעודכנת.
- מקור קלט לתמונות- מצלמה קבועה בחניון [יתרון משמעותי למצלמה שמאפשרת ראיית לילה] ששולחת תמונות לתיקייה מסוימת בשרת בכל פרק זמן קבוע.

רכיבי המערכת

פונקציית ה-main:

בחרתי לכתוב את ה-main לפי ה-flow הרצוי שהגדרתי (פירוט לאיך הגעתי ל-flow הזה נמצא ב- design review שבהמשך המסמך):




```
def main():
    num_of_images = get_num_of_images(IMG_FILE_PATH)
    parking_mark = mark_parking_spot()

    previous_image = read_image(IMAGE_PATH_TEMPLATE.format(1))
    previous_image_crop = crop_image(previous_image, parking_mark)

    for image_index in range(2, num_of_images):
        current_image = read_image(IMAGE_PATH_TEMPLATE.format(image_index))
        current_image_crop = crop_image(current_image, parking_mark)
        biggest_contour = compare_images(previous_image_crop, current_image_crop, image_index)
        is_parking_taken(biggest_contour, current_image_crop, image_index)
        previous_image_crop = current_image_crop
```

בנוסף יצרתי את האובייקט ParkingMark שמייצג את סימון החניה הרצויה.
השדות של האובייקט הם:

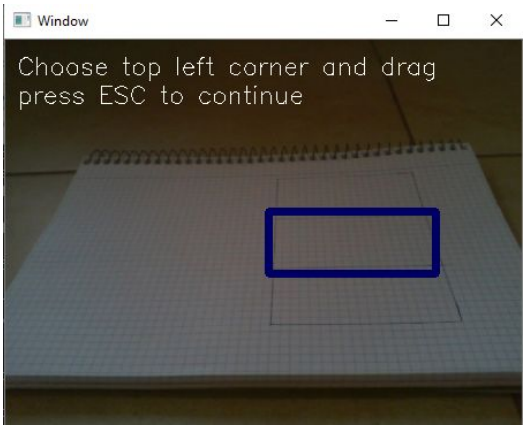
- top_left_corner - מייצג הפינה השמאלית העליונה של המלבן המסמן.
- bottom_right_corner - מייצג את הפינה הימנית התחתונה של המלבן המסמן.
- Crop_top_row - מייצג את השורה העליונה של המלבן המסמן.
- Crop_bottom_row - מייצג את השורה התחתונה של המלבן המסמן.
- Crop_left_colomn - מייצג את העמודה השמאלית של המלבן המסמן.
- Crop_right_colomn - מייצג את העמודה הימנית של המלבן המסמן.
- Base_image - מייצג את התמונה הראשונית שבה כל החניות ריקות.

הפונקציות של האובייקט הן:

- Draw_rectangle - מצייר את הסימון על גבי התמונה.
- Put_white_text - כותב את ההוראות על גבי התמונה.
- Mark_parking - מטפל בקלט שמגיע מהעכבר בזמן הסימון.

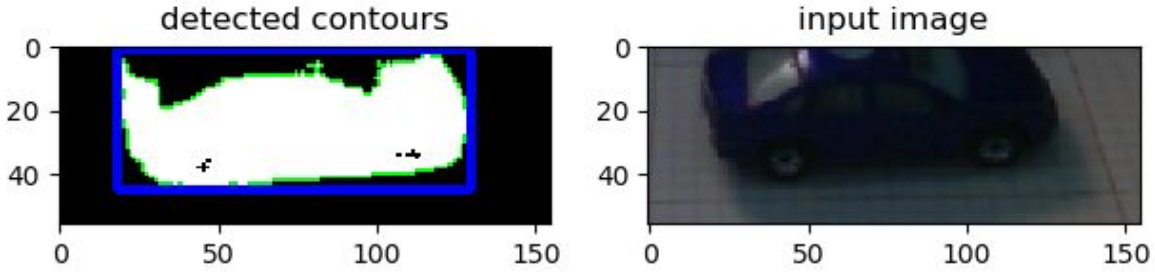
הוראות הרצה

1. אם לא מותקן, התקנת openCV-python על השרת על ידי הפקודה:
```
pip install opencv-python
```
2. אם לא מותקן, התקנת matplotlib על השרת ע"י הפקודות:
```
cd matplotlib
python -mpip instal
```
3. הורדת הפרוייקט מ-gitHub ע"י הפקודה:
```
git clone
https://github.com/NoamSchwarz/Parking-Protector.git
```
4. באמצעות CMD נכנסים לתיקיה שבה נמצא הקובץ parking_protector
5. עושים unzip לקובץ input_images.zip
6. מריצים את המערכת באמצעות הפקודה:
```
python parkingMVP.py
```
7. מסמנים באמצעות העכבר את החניה שנרצה לקבל עליה אינדיקציה ולוחצים על Esc



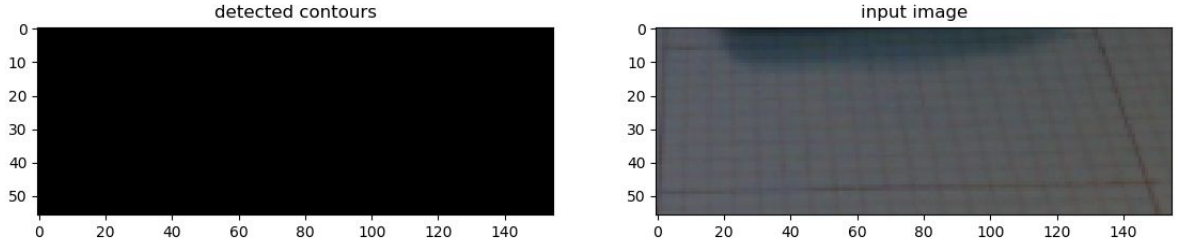
8. התכנית תציג, אחת אחרי השניה, את התמונות ממכשיר הקלט ולצידן את האובייקיטים שנמצאו. במקביל, לכל תמונה שהתקבלה, מודפס מצב החניה כפי שזיהתה המערכת.

דוגמה לפלט כאשר החניה תפוסה:




```
C:\Users\noamn\Documents\shecodes\parking_project\parking_proj_git>python parkingMVP.py
Small object found in image 2
In image 3 parking is taken
```

דוגמה לפלט כאשר החניה פנויה:




```
C:\Users\noamn\Documents\shecodes\parking_project\parking_proj_git>python parkingMVP.py
Small object found in image 2
In image 3 parking is taken
Small object found in image 4
Small object found in image 5
Small object found in image 6
Small object found in image 7
Small object found in image 8
In image 9 parking is taken
In image 10 parking is free
In image 11 parking is free
```

Design review

הרעיון והמבנה של הפרויקט תוכננו סביב הרצון שלי ליישם את מה שלמדתי על עיבוד תמונה, מהפרקים הראשונים של קורס בראייה ממוחשבת.
בהתחלה, בניתי תוכנית בסיסית שמשווה בין 2 תמונות, וקובעת אם בשנייה יש חפץ שלא היה בראשונה.

ה-POC שלי כלל:

1) קביעה ידנית של מיקום התמונות.
2) קביעה ידנית של האינקדסים עבור סימון החניה הפנויה בתמונה.
3) השוואה של התמונות של האזורים שסומנו ומציאת צללים בעלי שטח פנים גדול מספיק ביחס לשטח החניה (ה-threshold נקבע מראש על פי הגדרה).
4) סימון האובייקט שנמצא בתמונה עם תוצאת ההשוואה.

כשראיתי שהדרך שמצאתי אכן עובדת, תכננתי את ה-MVP בתור הרחבה של ה-POC, באופן הבא:

1) שימוש בתמונת בסיס של חניון ריק כדי לסמן עליו במלבן את שטח החניה הרצוי.
2) שימוש בסימון שטח החניה כדי לקבוע את הקורדינטות לסימון של אזור החניה מתמונות הקלט.
3) קריאה אוטומטית של התמונות מתיקיה.
4) השוואה של כל אזור החניה בכל תמונה, לאזור החניה בתמונה בה החניה ריקה.
    a) מאוחר יותר - השוואת כל תמונה לזו שבאה מיד לפניה, כדי לנסות למנוע שגיאות שנעשות עקב שינוי בתנאי תאורה
5) השוואת תמונות כדי לקבוע עם השטח נתפס
    a) התחשבתי במצב בו כלל לא נמצאים אובייקטים בתמונה.

את שלב 1 ביצעתי בעזרת תוכנית קצרה שכבר כתבתי עבור תרגיל בית מהקורס CV, שבו היה צריך לסמן ריבוע על גבי תמונה בעזרת העכבר. השתמשתי בתוכנית הזו כדי לסמן בעזרת מלבן את האזור הספציפי בתמונות שמעניין אותי לבדוק.

כשסיימתי את יישום כל השלבים בתוכנית הראשונית, התחלתי לחפש דרכים לשפר וליעל את התוכנית.

- הפרדתי את התוכנית לסימון אזור בתמונה לקובץ אחר, והמרתי אותה ל-class כדי שיהיה אפשר להשתמש באובייקט ובמשתנים שלו, במקום להשתמש ישירות במשתנים שחזרו מפונקציות הסימון.
- הוצאתי את רוב קוד מה-main לפונקציות נפרדות, והגדרתי משתנים קבועים כדי להקל על תחזוק המערכת.

מה ה-MVP כולל

ה-MVP מזהה מתי חניה שהייתה פנויה, נתפסת.
כדי לפשט את תהליך הצילום ולשלוט בתנאי תאורה, השתמשתי במכוניות צעצוע עבור התמונות שצילמתי כקלט לתוכנית.

תוכנית ה-MVP מורכבת מ-2 קבצים -

- parkingMVP.py שבו פונקציית ה-main והפונקציות שמבצעות את עיבוד התמונות.
- Parking_class.py מכיל את ParkingMark class, *שמשמש* לסימון החניה הרצויה על גבי התמונה שבה כל החניות ריקות (התמונה הראשונה של הקלט).

מה רציתי לכלול ב-MVP אבל לא הספקתי:

- זיהוי למתי חניה תפוסה, מתפנית.
- מערכת דיווח למשתמש