מעבדת בראייה ממוחשבת

סמסטר ב' 2017

מגישים:

לוי גלעד

אפראימוב אורן

סביבת העבודה:

הפרויקט נכתב בפייתון ונתמך הן בפייתון 2.7 והן ב3.6.

מטרת הפרויקט:

בפרוייקט שלנו בחרנו לבנות כלי אשר יוכל לזהות ולעקוב אחרי כלי רכב (מוכניות, אופנועים, משאיות וכו') על גבי כביש מהיר ולהחזיר סטטיסטיקות ונתונים על רמת השימוש בכביש ומספר כלי הרכב שעוברים בו. את ההשראה לפרוייקט קיבלנו לאחר שראינו אדם אשר מצלם, מודד וסופר בעזרת מערכות של משרד התחבורה את מספר הרכבים שנוסעים בכביש החדש שנסלל באיזור השרון המחבר בין איילון צפון לכביש 531. רצינו לבנות גרסה ביתית של אותה מערכת על מנת להבין את מורכבות הדבר ולהבין כיצד אותן מערכות בנויות.

תיאור האלגוריתם:

כדי לעקוב אחרי כלי הרכב בחרנו להשתמש באלגוריתם קלמן פילטר (או בשמו האחר 'מסנן קלמן'). קלמן פילטר הינו אלגוריתם, המקבל כקלט סדרה של מדידות מורעשות של ערכים שונים המייצגים מצב של מערכת בזמן מסוים, ומייצר הערכה, בדרך כלל מדויקת יותר, של אותם הערכים על סמך תוצאת המסנן בעבר. המסנן עובד בצורה רקורסיבית על קלט מורעש ומייצר הערכה סטטיסטית של מצב המערכת.

הרצת התוכנית:

את התוכנית ניתן להריץ על הרצה מ-command line, על ידי הזנת השורה הבאה:

python run.py -v dataset/Highway8.mp4 -d 0 -s 30 -p 50

פירוט על הארגומנטים:

* -v – ארגומנט זה מקבל path של הסרטון (עדיפות שפורמט הסרטון יהיה מסוג avi).
* -d – ארגומנט זה מקבל את כיוון תנועת כלי הרכב. 0 מסמן תנועה מלמטה למעלה, 1 מסמן תנועה מלמעלה למטה וכל ערך אחר אינו חוקי.
* -s – ארגומנט זה מקבל את מהירות קריאת הסרטון והצגתו על פני המסך. ערכים שיכול לקבל ארגומנט זה הם 1-100, 1- מסמן קריאה מהירה מאוד ו-100 קריאה איטית.
* -p – ארגומנט זה מקבל את מיקום הקו (באחוזים) על פי הסרטון. הערכים שיכול לקבל ארגומנט זה הם 0-100.

ממשק משתמש:

הגדרנו כמה מקשי עזר:

* מקש P/p – מקש זה 'מקפיא' את הסרטון ואת זיהוי כל הרכב.
* מקש Q/q/esc – מקשים אלו יוצאים מן התוכנית.

בזמן הרצה:

אנו פותחים מספר חלונות אשר מראים כיצד אנו מזהים את כלי הרכב בכל שלב במערכת שלנו. כל חלון מראה פעולה אחרת במערכת וכיצד לבסוף אנו מזהים רק את כלי הרכב.

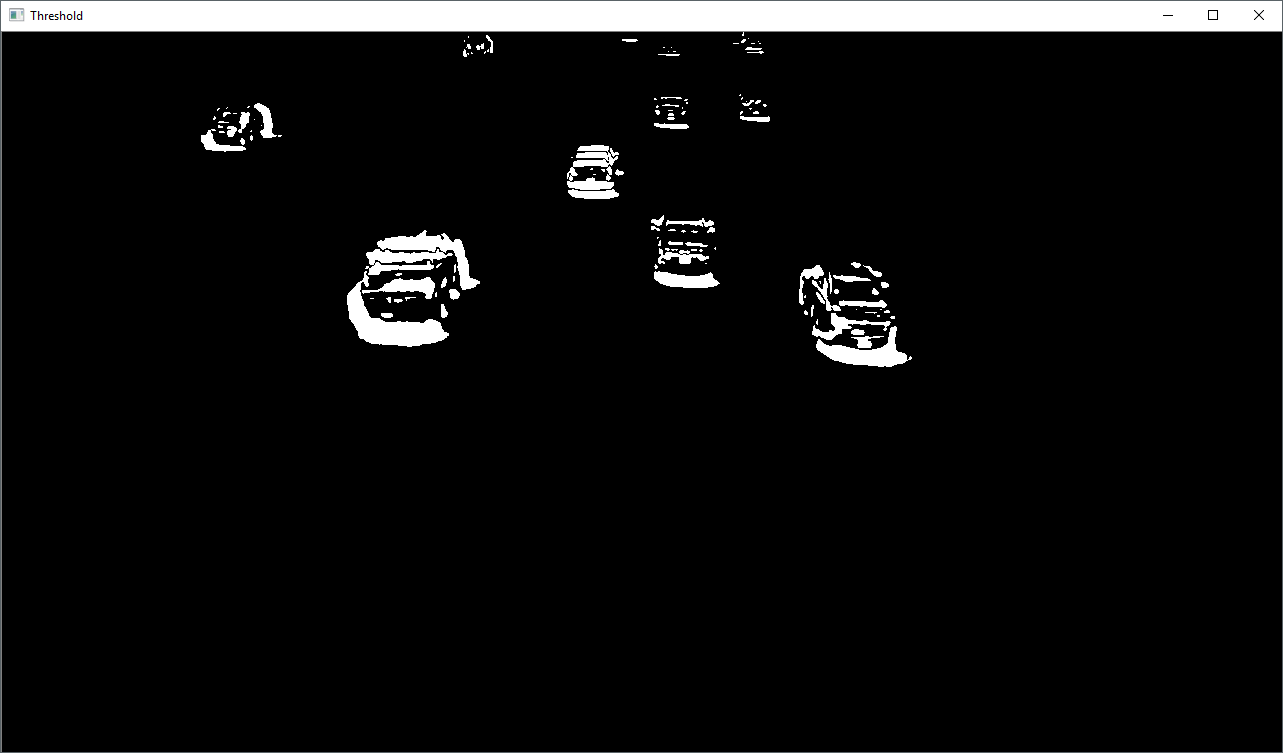
החלונות מסייעים בהבנה ודיבוג המערכת.

השלבים במערכת לצורך זיהוי כלי הרכב בסרטון:

1. בכל פעם עוברים על שני 'פריימים' עוקבים בו זמנית וממירים את צבעי התמונה לצבעי אפור. לאחר מכן, עבור כל שתי תמונות הפריים מפעילים את Gaussian Blur בחלון 5\*5 (על מנת להפחית את הרעש בתמונה וככה להוריד רעשים אקראיים בסרטון), "מחסרים" בין שתי התמונות ומציגים רק פריטים שעומדים ב'סף' מסויים וקבוע (30).

נשים לב כי בחלק העליון של התמונה מקבלים כתמים לבנים שבשלב זה לא ידוע האם הם כלי רכב או לא.

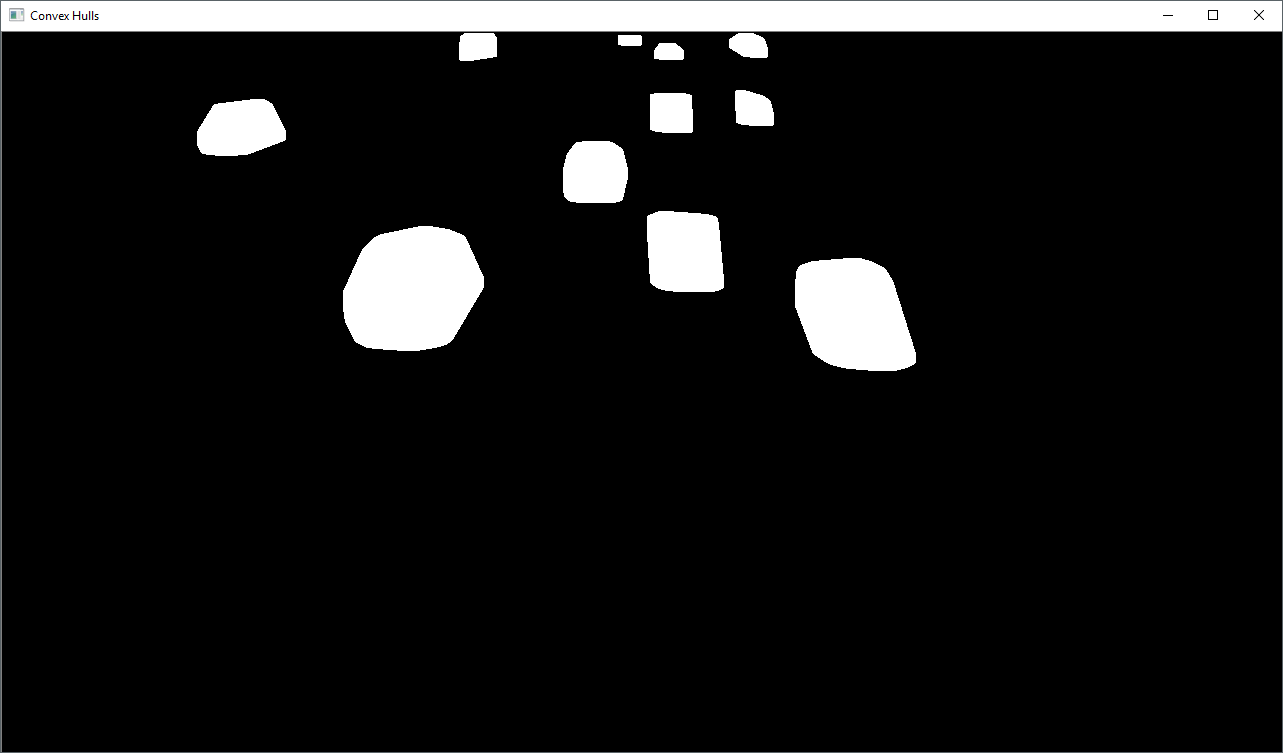
חלון הפלט:



1. בשלב השני, השתמשנו ב-api של OpevCV (dilate ו-erode) על מנת לזהות את קווי המתאר של כל כלי רכב וכך למצוא את Contours:

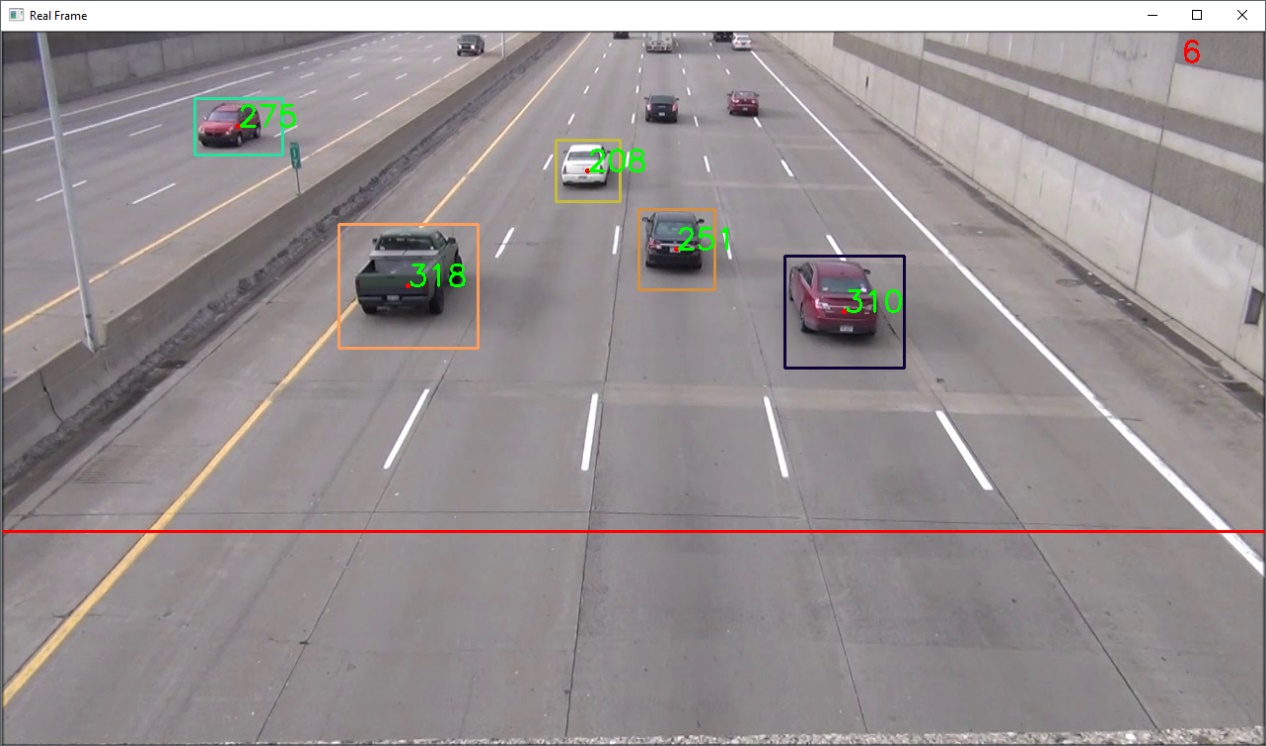


1. בשלב השלישי, מימשנו את הפונקציה convexHull על מנת לקבל את כלי הרכב כ'גוף אחד'. כלומר, בשלבים הקודמים זיהנו את כלי הרכב ואת קווי המתאר, בשלב זה אנו יכולים לראות איפה מתחיל ומסתיים כלי הרכב על גבי התמונה.

חלון Convex Hulls: 

1. בשלב זה, אנו מציגים את כלי הרכב של החלון הנוכחי (Current Frame Blobs) אל מול החלון של זיהוי כלי הרכב בסרטון (All Blobs). All Blobs זהה לחלון Real Frame שבו מוצג בזמן אמת זיהוי כלי הרכב.
2. לאחר שהצלחנו לזהות את כלי הרכב בהצלחה ולעקוב אחריו, אנו סופרים את מספר כלי הרכב שעברו את הקו המונה שצבוע בצבע אדום (ניתן לראות בחלון Real Frame), וברגע שכלי רכב עובר את קו זה אנו מוסיפים למונה. בפועל, אנו סופרים בכל פעם שמרכז המסה של blob רכב נפגש עם הקו המונה (הנקודה האדומה בכל מרובע), הקו מחליף את צבעו לצבע ירוק ומעלה את המונה בצד ימין של המסך.

התוצאה הסופית:



זיהוי וספירת מכוניות

כל כלי רכב מקבל מספר id וצבע אשר מלווה אותו מתחילת הזיהוי ועד סוף המסלול.שני ערכים אלו הינם קבועים לכל אורך המעקב. במידה והרכב סטה מידי וקלמן פילטר לא חזה נכון את המיקום שלו, התוכנה מאבדת את הזיהוי עבור הכלי רכב ולאחר מכאן מצליחה לזהותו מחדש עם ערכים id וצבע חדשים.

חסרונות

למרות שאלגוריתם יודע להתמודד עם מספר שונה של סרטונים, הוא אינו עובד על גבי סרטונים בהם יש פקק או אין מרווח בין המכוניות. כתוצאה מכך המערכת מזהה מספר כלי רכב כ'גוש' אחד שעונה על הקריטריונים שקבענו לכלי רכב (גודל שטח, גודל האלכסון, גודל הגובה, גודל האורך וכד').

עקב כך שמדובר בכביש מהיר, מקרים אלו נדירים וכמעט ולא קורים.

ניתוח

מיקום הקו יכול להשתנות בין סרטון לסרטון וניתן למקם רק לאחר הסתכלות על תנועת כלי הרכב בסרטון. עבור הסרטונים בדקנו:

* Highway1.mp4 – עבור סרטון זה, בכול מיקום הקו בין 15-70 הצלחנו לספור את כמות המכוניות בצורה נכונה.
* Highway2.mp4 – עבור סרטון זה, רק שמיקום הקו היה בין 10-15 הצלחנו לספור את כמות המכוניות בצורה טובה.