

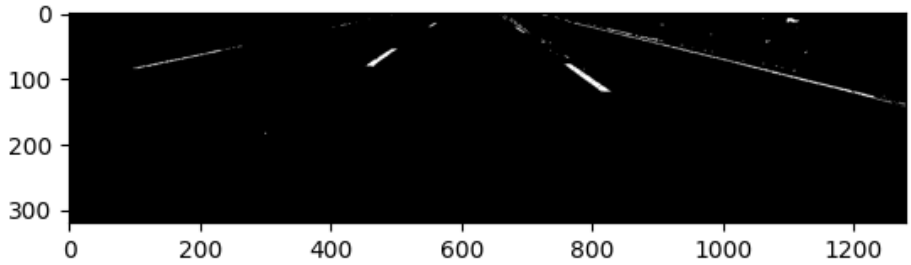

עדן בן זמרה 209877125
נמרוד גולדברגר 316169036

ראייה ממוחשבת פרוייקט 1:

Lane detection

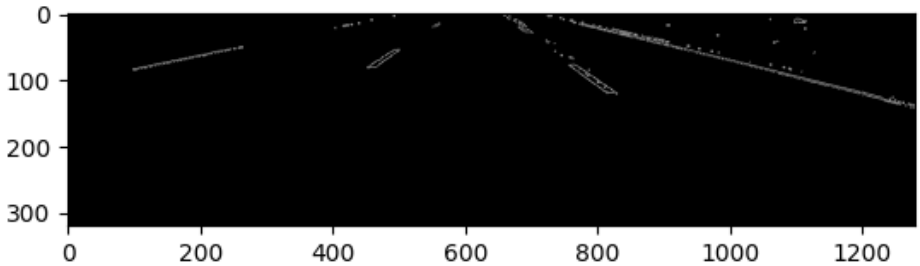
בתרגיל זה היינו צריכים לבנות אלגוריתם ראייה ממוחשבת שיקבל סרטון של כביש ויזהה את נתיבי הכביש. בנוסף נדרשנו לזהות מעברי נתיבים; לזהות לאיזה נתיב יש מעבר (שמאל או ימין).

לקחנו סרטון של כביש עם תנועות רכבים ומעברי נתיבים.
יצרנו לולאה שקוראת בסרטון זה frame אחר frame.
תחילה המרנו את הframe לgray scale.
למניעת רעשים בזיהוי הנתיבים חתכנו את החלק העליון של הframe שאינו שימושי לנו לזיהוי הנתיבים.
יצרנו מסיכה שתהפוך ללבנים את כל הפיקסלים שערכם הוא בתווך 'הבהיר יותר' של התמונה.
הגבול התחתון של התווך הוא הוספת 200 לערך המינימלי כלומר החשוך ביותר של התמונה.
הגבול העליון הוא 255 – הלבן ביותר. מסיכה זו עזרה לנו להמיר ל 255 (לבן) רק פיקסלים שהם בתווך זה וכל השאר הומר ל 0 (שחור).



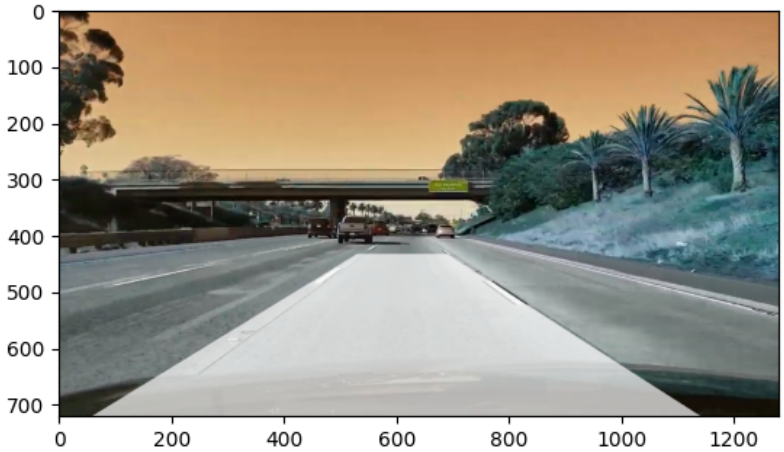
תמונה לאחר חיתוך אזור עליון והפעלת המסיכה

כעת נשתמש בתוצר זה ונבצע עליו זיהוי קצוות באמצעות canny.

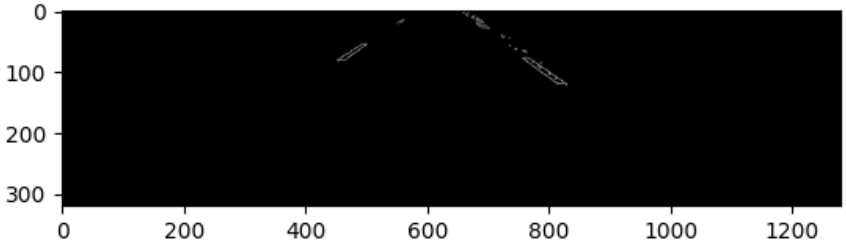


תוצאת הcanny על התוצר הקודם

נרצה לזהות רק את הנתיבים הרלוונטיים לרכב שלנו ולשם כך ניצור מסיכה נוספת בצורת טרפז ונשאיר בלבן רק את הפיקסלים שהם לבנים ובתוך הטרפז. את הטרפז יצרנו פעמיים , פעם אחת לצורך חישובים להתאמה ל frame החתוך ופעם שניה להתאמה ל frame המקורי לצורך תצוגה בסרטון.

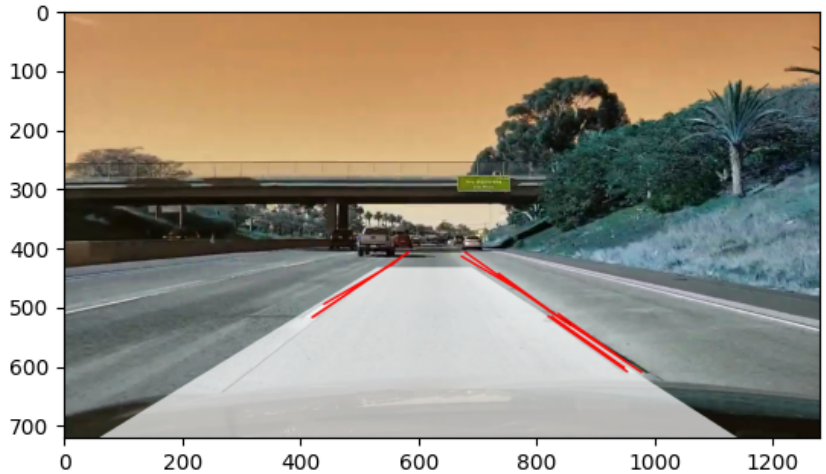


frame המקורי עם תצוגת הטרפז שבתווכו נחפש נתיבים



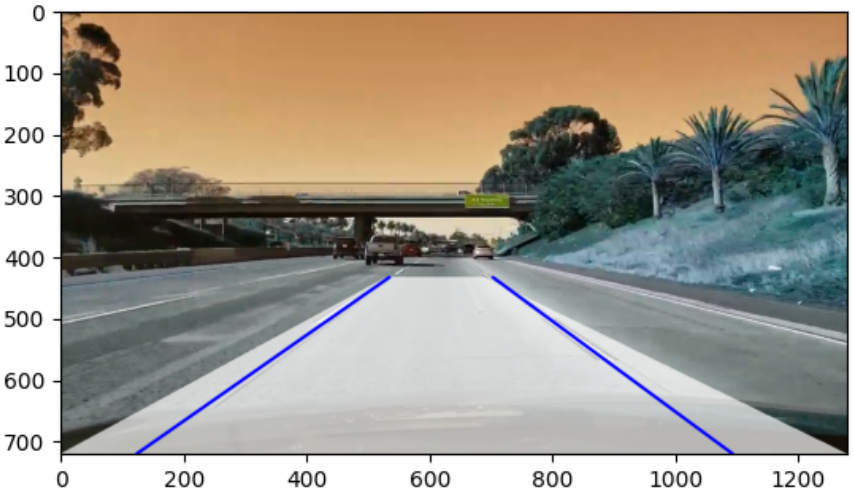
התוצר לאחר הפעלת מסיכת הטרפז שבתווכו נחפש נתיבים

נרצה ליצור מהנתיבים שזיהינו שורות. לשם כך נשתמש בHough transform לזיהוי שורות.



frame עם השורות שאותרו בHough transform

נרצה ליצור שורה שתהווה ממוצע של השורות שמצאנו. לשם כך כתבנו פונקציה בשם lane_lines שמקבלת את הframe ואת מערך השורות שנמצאו כשכל שורה מיוצגת על ידי 2 נקודות ומפרידה את הנתיבים הימניים מהשמאליים. היא יוצרת ממוצע של כל הקווים הליניאריים שנוצרים מכל 2 נקודות של שורה מאותה הקטגוריה (ימין או שמאל).
בנוסף הפונקציה lane_lines לאחר שמוצאת את הנתיב הימני והשמאלי תבדוק האם מיקומן מרמז על מעבר נתיב ולאיזה נתיב יש חשש שיש מעבר.
עשינו את בדיקה זו לפי ערך ה x של הנקודה העליונה של כל שורה ומצאנו תווך קרוב לאמצע הפוליגון שאם ערך ה x באזור זה, קיים חשש למעבר נתיב.
הממוצע מחושב בעזרת הפונקציה average_slope_intercept ויצירת הפונקציה הלינארית בעזרת pixel_points.
draw_lane_lines היא הפונקציה שמציירת את השורות שמתקבלות מlane_line על הframe.



הframe הסופי : עם הממוצע של השורות שמצאנו כקו לינארי

כל תוצר סופי כזה נכניס למערך של frame ים עד לסיום קריאת כל הסרטון.
בסוף קריאת הסרטון נשתמש ב cv2.VideoWriter ליצירת וידאו חדש ונעבור על מערך ה frames ים שיצרנו ונכתוב אותם לווידאו.

הסרטון הסופי נמצא בקובץ הזיפ.

ניתן לראות שהאלגוריתם לא מדויק, הוא לא תמיד מזהה את כל הנתיבים והם יכולים להתפספס מכל מיני סיבות.
יתכן שהם לא יהיו לבנים מספיק והם יושחרו במסכה , כנראה שהגדלת התווך של המסכה תכניס אותם אך היא עלולה גם להכניס רעשים נוספים.
נתיבים שנצבעו באזור קצת מחוץ לתחום שהגדרנו גם עלולים להתפספס אבל הגדלת התחום גם עלולה להוסיף רעשים.
ההגדרות של גודל מינימלי של שורה בHough transform גם עלול לגרום לפספוסים של נתיבים או בכללי כל הגדרות הערכים לפונקציות של open cv כי לצערנו, כל frame שונה ואין הגדרות אידאליות שיתנו תוצאה מרבית לכל הframe ים של הסרטון.
בנוסף מעבר מתחת לגשרים וחושך גם יגרמו לפספוסים מהסיבה שהבהירות של הנתיבים נמוכה יותר והם עלולים להתפספס במסכה. בניסיון לפתור את זה ניסינו להגדיר את המסכה לפי בהירות התמונה אך זה לא תמיד שיפר.

בנוסף גם צל יכול להוות רעשים, במעבר מתחת לגשר בסרטון למשל הצל של הגשר מזוהה בקצוות וכנתיב. לצורך מניעת רעש זה הוספנו בדיקה שמונעת קווים המזוהים כאופקיים כי נתיב לא יהיה אופקי.
בעיה נוספת שנתקלנו בה הייתה זיהוי נתיב אליו הרכב עובר. כאשר יש מעבר נתיב שמאלה למשל , הנתיב שהיה השמאלי, בסוף המעבר הופך להיות הנתיב הימני. זה קורה כאשר הנתיב קצת אחרי האמצע של התמונה וזה יביא לזיהוי שגוי של המעבר כמעבר לנתיב ימין.
למניעת בעיה זו הגדרנו אזור קטן של 40 פיקסלים שזיהוי הנקודה העליונה של שורה בתווך זה יהווה חשש למעבר נתיב. הקטנת האזור גרמה לפספוסים של frame ים שמעבר הנתיב כבר התחיל בהם אך לא זוהה באלגוריתם. אם היינו מגדילים את האזור האלגוריתם היה מזהה יותר מוקדם את המעבר אך גם היה מתריע בסוף המעבר למעבר לנתיב שגוי (ימני במקום שמאלי למשל).