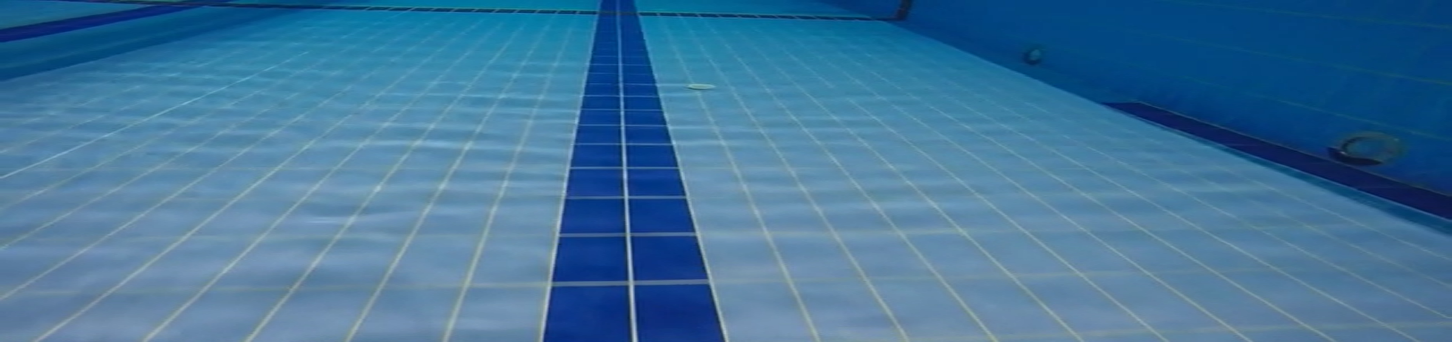
**מסמך סיכום פרויקט בראיה ממוחשבת:**

**Support for a blind swimmer**



**שם הקורס:** ראיה ממוחשבת

**המרצה:** ד"ר יונתן רובין

**מגישים:** מירה גלייזר ודבורה-דניאל בן-צבי

**שנה"ל:** תש"ף, סמסטר ב'

**סביבת העבודה:** jupyter notebook

**שפת תכנות:** Python 3.7

**מטרת הפרויקט:**

סיוע לשחיין העיוור באמצעות התרעות קוליות במצבים בהם היא מזהה סטיות שלו ימינה/שמאלה וכן כשהוא מגיע לגבול סוף הבריכה.

**תאור המערכת:**

המערכת תכלול מצלמה שתורכב על מצח השחיין ובאופן שתוכל לצלם את רצפת הבריכה במצבים בהם השחיין נמצא מתחת לפני מים

התוכנה תבחין בין מצבים בהם השחיין נמצא בתוך המים לבין מצבים בהם ראשו מורם מעל פני המים לצורך הפסקות נשימה לסירוגין.

במצב בו הוא סוטה ממסלולו (המסומן ע"י הקו האנכי האמצעי הכחול לאורך הבריכה) ימינה או שמאלה תופק הודעת צליל מתאימה

וכן ברגע שהשחיין מגיע אל הטווח בו מופיע הקו הכחול המאוזן בבריכה - המציין את ההגעה כמעט לקצה, תופק הודעת התרעה על כך.

**שלבי פיתוח:**

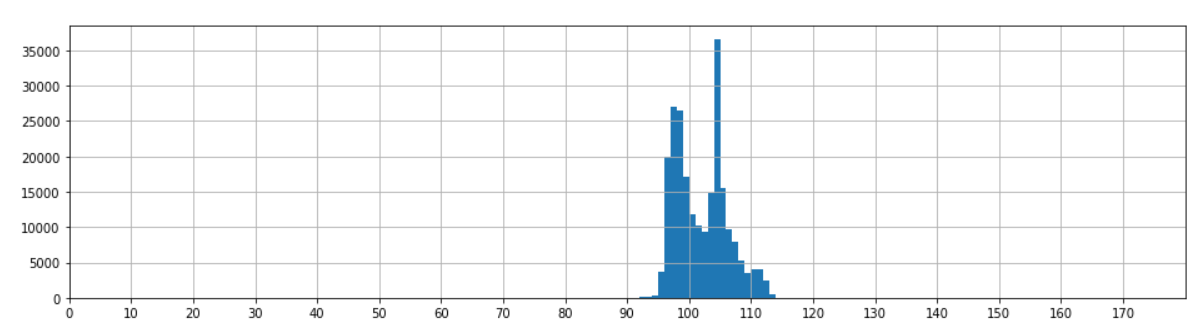
א) על מנת לבדוק האם אנחנו בפריים הרלוונטי (כשראש השחיין מתחת למים והמצלמה מופנית לקרקעית הבריכה):

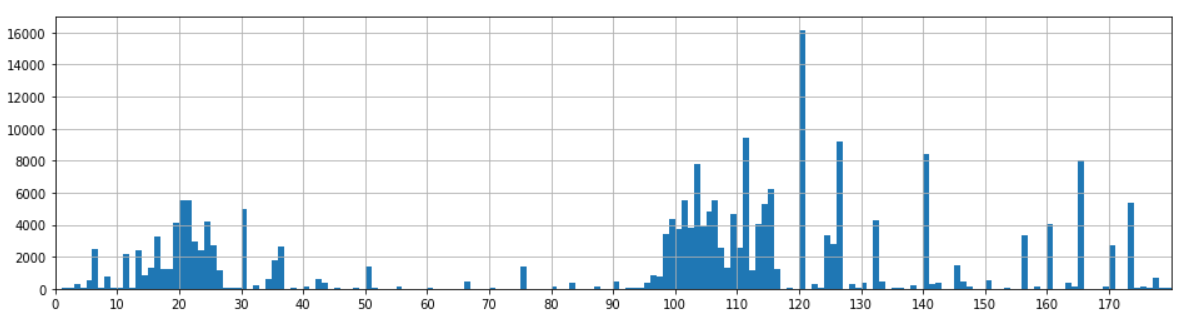
השתמשנו ב histogram לבדיקת צבעי הפיקסלים של כל תמונה: הקטנו את התמונה פי 4 ובדקנו כל פיקסל שני. ספרנו פיקסלים בגבולות הצבע הכחול, וכן פיקסלים לא כחולים (בכל גוון אחר) בנפרד, זאת על מנת להסיק מכך על מצב השחיין:

ע"י השוואה בין כמות הפיקסלים הכחולים לעומת כמות הפיקסלים בגוונים אחרים.

במידה והמצלמה בתוך המים, יוחזר true, אחרת יוחזר false, כשמתקבל true ממשיכים עיבוד תמונה, אחרת פשוט עוברים לפריים הבא.

Difference in histograms for “blue” and “not blue” images:





שלב ב':

בפונקציה השניה השתמשנו ב- cv2.canny לזיהוי פינות, אחר כך ב-cv2.houghlinesp לזיהוי קוים.

בודקים מה השיפוע של הקו, עם איתור קווים אנכיים/מאוזנים במסגרת התחום הרלוונטי בתמונה,

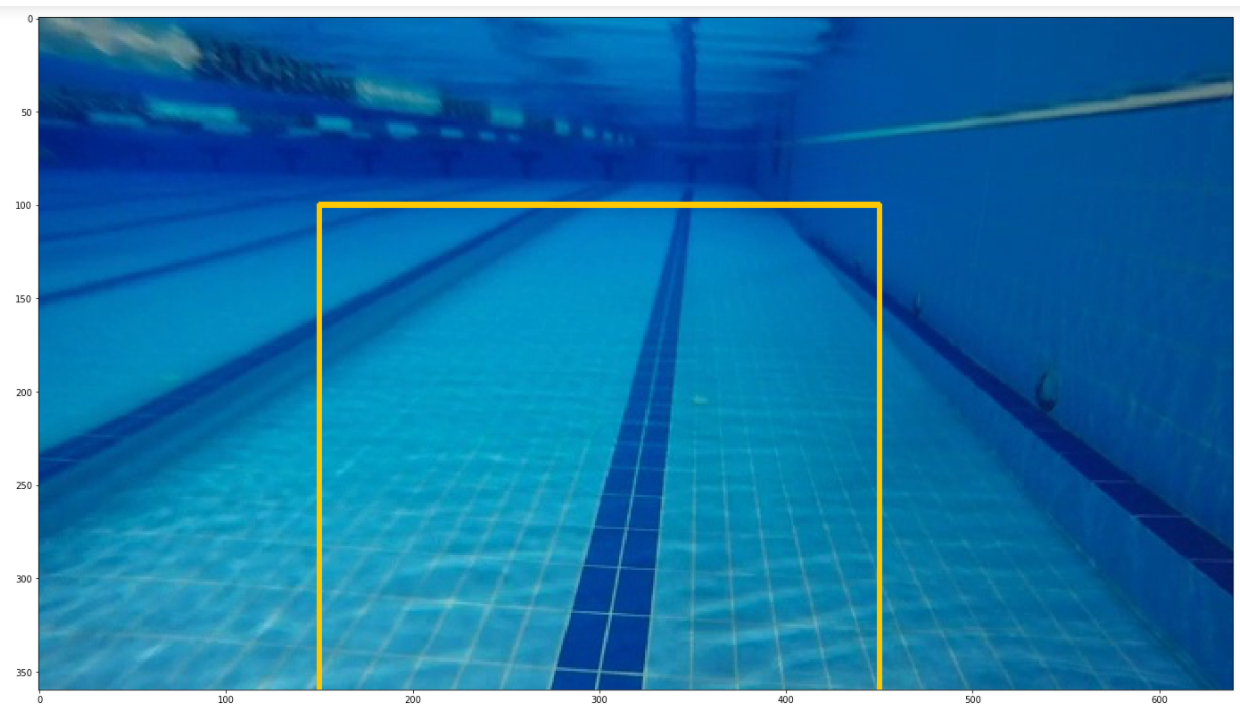
אם השיפוע של הקו קרוב למאוזן - קרוב ל-0 בטווח של -0.5- ל 0.5 פונים לפונקציה שלישית ורביעית כדי לבדוק האם הפס כחול או אדום בעזרת אלגוריתם . kmeans אלגוריתם kmeans מקבל ערכים של צבעים פיקסלים באלכסון ומחלק את התצפיות ל-2 אשכולות לפי מרכזי כובד (k-means). אם שני המרכזים נמצאים בתווך הצבע הכחול - אז הקו קרוב למאוזן הוא כחול וצריך לתת סימן לעצור, אחרת פשוט ממשיכים.

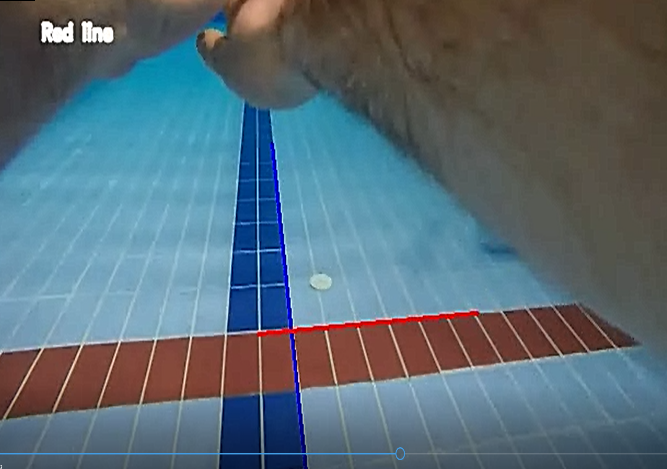
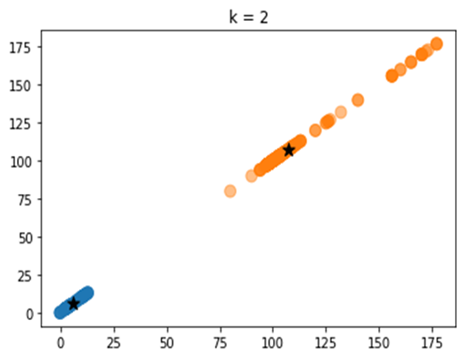
במידה וזוהה הפס המאוזן הכחול שמציין את הסוף- המערכת תרתיע על כך.

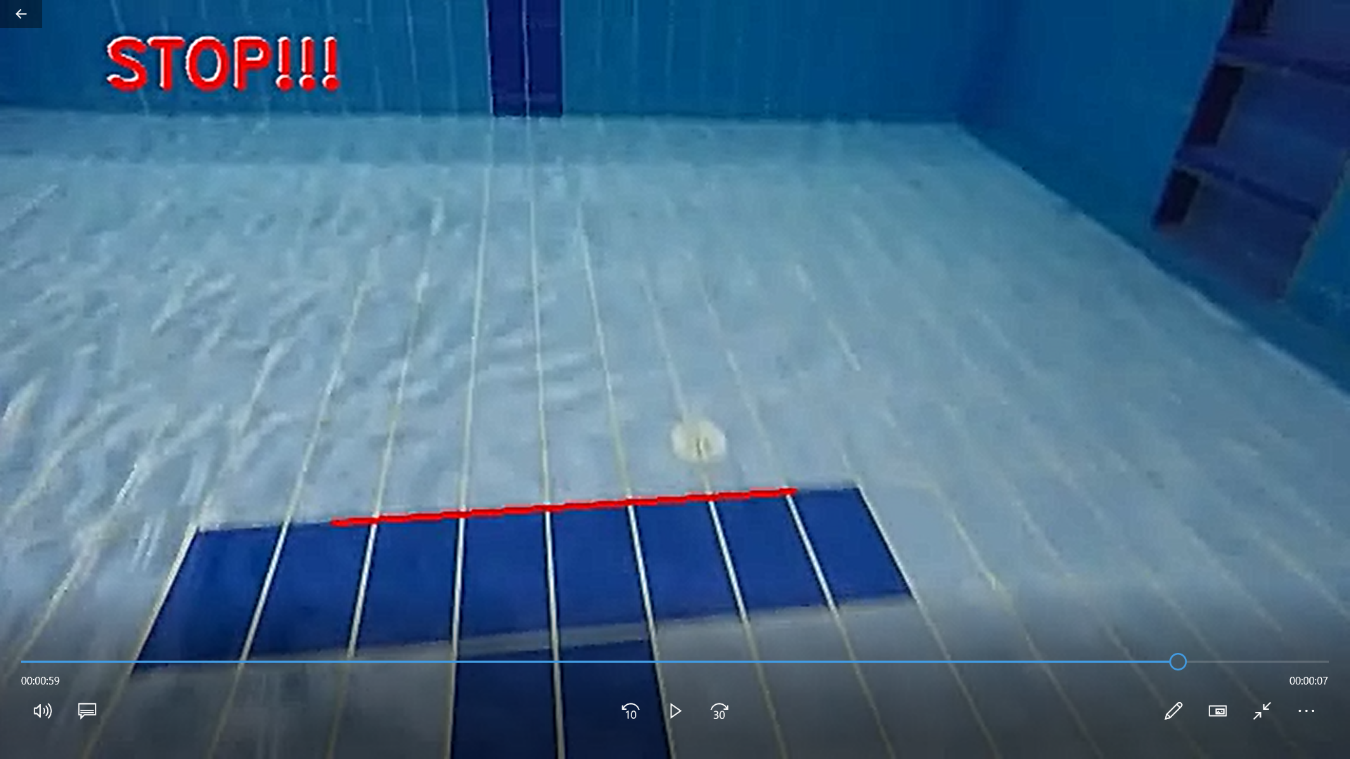
אם השיפוע של הקו המזוהה הוא בין2- ל -0.5- ובין 0.5 ל-2 זה אומר שהשחיין פנה ימינה או שמאלה

והמערכת תפיק הודעה קולית מתאימה (האם לפנות ימינה או שמלה)

התייחסנו לקטע תמונה אמצעי רלוונטי ולא לכל שטח התמונה (150 מימין ומשמאל, ו-50 מלמעלה)





**קשיים ודרכי התמודדות במהלך הפיתוח:**

בעיה ראשונה היתה כיצד מתמודדים עם המצבים בהם המצלמה מחוץ למים ואינה מצלמת את קרקעית הבריכה,

התמודדנו ע"י זיהוי מצב באמצעות בדיקת כמות הפיקסלים הכחולים בכל תמונה, מה שמעיד על מיקום המצלמה הנוכחי.

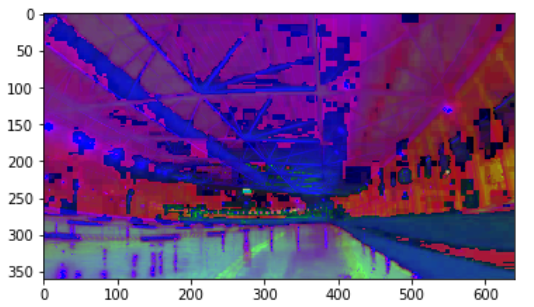
עיבוד התמונה התבצע רק במצבים בהם כמות הפיקסלים הכחולים היתה מרובה - המצב בו השחיין מתחת למים.

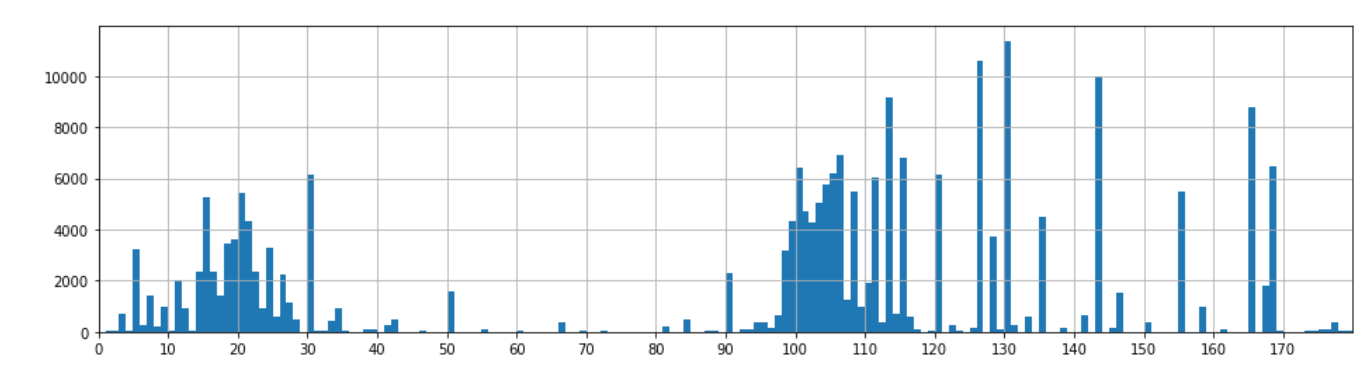
בעיה שניה היתה בעיית אי יעילות של בדיקת כל הפיקסלים בכל תמונה,

על מנת לחסוך זמן ריצה השתמשנו בהקטנת תמונה בעזרת resize לרבע מהגודל המקורי על מנת להקטין את כמות הפיסקלים המיועדים לבדיקה.

לאחר מכן העברת התמונה לפורמט HSV ובדיקת צבע של כל פיקסל שני.

Image in HSV format





בעיה נוספת היתה הימצאות של קווים אופקיים דקים לבנים (בין הלבנים שבקרקעית הברכה) שהמערכת זיהתה עוד בהתחלה והתריע על כך, התמודדנו עם הבעיה הזאת בכך שהגדרנו את האורך המינימלי בזיהוי הקוים, שיהיה ארוך יותר מקווים אלו.

עוד בעיה שהיתה הימצאות הפס האדום האופקי לאורך כל הבריכה (הנמצא קרוב לסוף), שהמערכת זיהתה בפריים עוד מאמצע הדרך והתריע "stop", כיוון שצבע הפיקסלים שלו מרחוק לא היו בצבע אדום,

התמודדנו ע"י כך שהגדרנו בתוך הפריים שטח רלוונטי שבתוכו מחפשים קווים. (חתכנו אותו משטח החיפוש(

**הדגמה:**

מצורף וידאו המציג את הפרויקט, ראשית את ההתרעה על הגעת השחיין לסוף הבריכה,

ובסוף הוידאו הוספנו קטע סרטון שעליו תחילה הפעלנו עיבוד נוסף – הזזת הפריים ימינה ואח"כ שמאלה כדי לדמות מצבים בהם השחיין סוטה מכיוון השחייה הנכון (האנכי) ולהציג את תגובת המערכת בהתאם לכל סטיה.