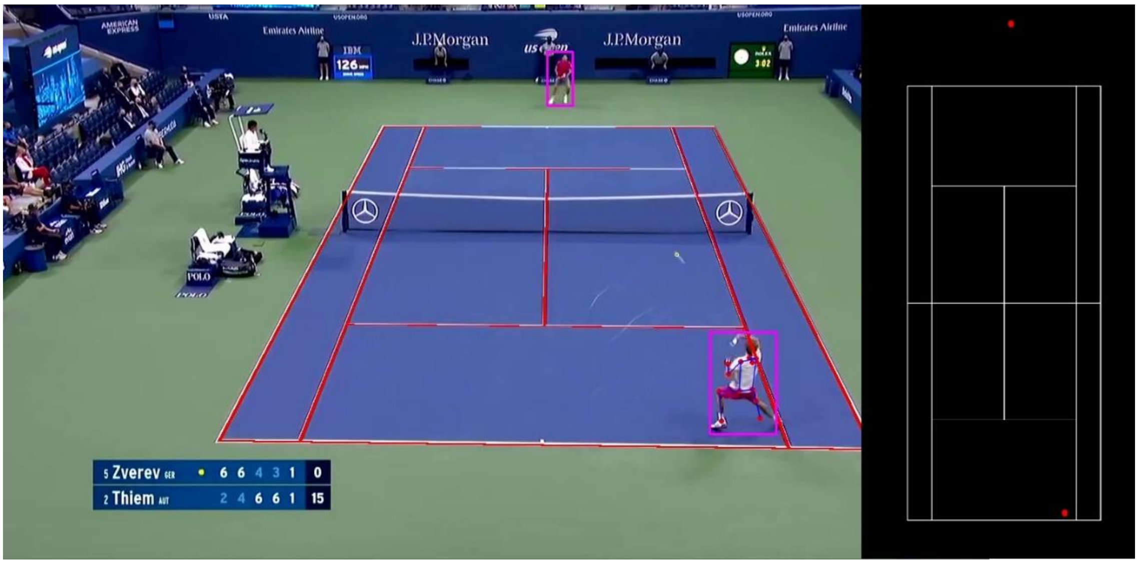
Geometric Image Processing Lab

עיבוד וניתוח משחקי טניס באמצעות ראייה ממוחשבת



מגיש: אביב כספי
מנחה אקדמי: אלון זבירין
מנהל מעבדה: ירון חונן
סמסטר: חורף 2020

תוכן עניינים

תוכן עניינים **2**
מבוא **4**
סקירת ספרות **7**
Hough transform 7
Probabilistic Hough Line Transform 9
מציאת טרנספורמציה הומוגרפית בין שתי תמונות 9
הפעלת טרנספורמציה הומוגרפית 10
Least Squares for line fitting 10
Resnet 11
מודל לסיווג חבטות טניס 12
TrackNet 13
SORT - SIMPLE ONLINE AND REALTIME TRACKING 13
הפתרון שלי ושיקולי תכנון **15**
זיהוי ומעקב אחר המגרש 15
זיהוי ומעקב אחר השחקנים 20
זיהוי שלד שחקן תחתון 23
זיהוי הכדור 24
זיהוי רגע החבטה 25
תיוג החבטה 26
סטטיסטיקות וסרטון מבט על 27
דאטה-סטים **31**
THETIS 31
Strokes 31
TrackNet 32
ניסויים **33**
זיהוי המגרש 33
מעקב מגרש 48
זיהוי השחקנים 52
זיהוי "שלד" השחקנים 57
סיווג סוגי חבטות 60
זיהוי הכדור 63
זיהוי השחקן העליון 67
זיהוי פריים החבטה 69
סטטיסטיקות 71
תוצאות **74**

פרטי מימוש 81

סיכום 81

סיכום 81

הצעות להמשך פיתוח 81

ביבליוגרפיה 83

מבוא

ניתוח משחקי ספורט מאז ומתמיד היווה עניין גדול, ומאז ומתמיד מאמני ספורט מחפשים את תוכניות האימונים והאסטרטגיות משחק הטובות ביותר על מנת לגרום לשיפור השחקנים והשגת הניצחון.

לרוב, על מנת לשפר אסטרטגיות משחק שונות, הגישה המקובלת הייתה לצפות במשחקי עבר ולנתח אותם, ובכך לקבל תובנות כיצד ניתן להשתפר.

קל להבין כי גישה זו אינה יעילה ומאפשרת למידה מדגימות קטנות מאד בגלל הצורך הפיזי לשבת ולצפות במשחק שלם על מנת להגיע לניתוח מספק.

ניתוח משחקים שונים על ידי ניתוח סטטיסטי הינה גישה נוספת שנפוצה בקרב מאמנים שונים, אך לרוב הגישה לנתונים אלה הסתכם במספר קטן של מאפיינים גסים כגון: מספר הגשות ראשונות (טניס), מספר ניצחונות עבור קבוצה מסוימת או מספר גולים לשחקן ספציפי (כדורגל).

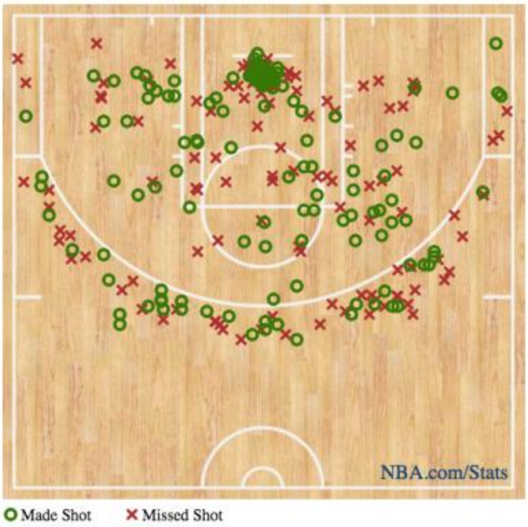
לכן, הניתוח התבסס על מאפיינים גסים שנוצרו בצורה ידנית על ידי אנשים שצפו במשחקים.

למרות תכונות אלה, לא ניתן להתכחש לעובדה כי ניתוח סטטיסטי זה אכן היה מוצלח לפעמים, אך עדיין מכיל כמה חסרונות גדולים:

- רמת הפירוט האפשרית להשגה בצורה ידנית הינה מוגבלת, כלומר לא ניתן לקבל פירוט על צורת תנועת הכדור רק על ידי צפייה במשחק. זה מוביל למידע גס.
- התיוג הידני יכול להוביל למידע לא מדויק או סובייקטיבי. המידע שתויג מושפע ממצב רוחו של המתייג, בנוסף המתייג יכול לפספס מכות מסוימות או להעלים עין מפרטים חשובים אחרים.
- איסוף המידע הינה עבודה ארוכה ומעייפת, ודורשת מומחים אשר יצפו במשחקים בריכוז גבוה.

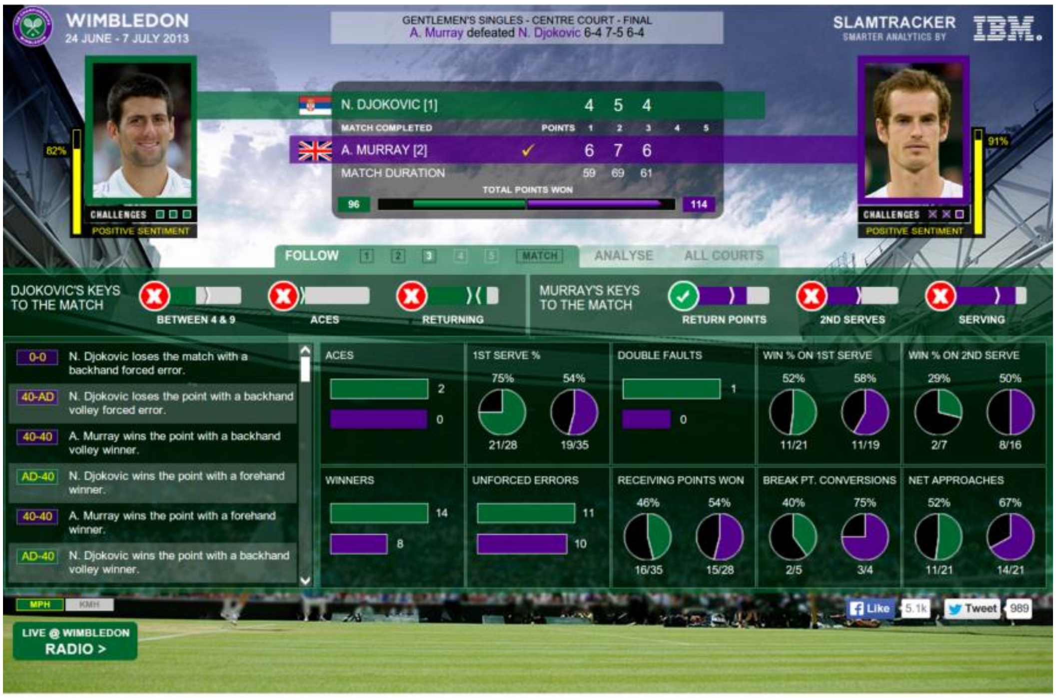
חסרונות אלה הובילו את החברה לפיתוח מודלים אוטומטיים המאפשרים תיוג בצורה אוטומטית ללא מגע אדם.

להלן דוגמא לשימוש במערכת אוטומטית במשחקי כדורסל המזהה את ניסיונות הקליעה לסל של השחקן קווין דורנט במשחק משנת 2016:



תיוג זה מאפשר לזהות אזורים בהם השחקן חלש יותר בקליעה מאשר אזורים אחרים וכך מאפשר למאמן להתמקד באזורים אלה בזמן האימון.

דוגמא נוספת לפיתוח בתחום זה הינו פיתוח משנת 2012 של חברת IBM אשר הציגה אפליקציה המציגה בזמן אמת ניקוד סטטיסטי מתוך משחקי טניס, על מנת להציג לצופים בבית פירוט של המשחק:



למרבה הצער, רוב המידע שנאסף על ידי מודלים כאלה ואחרים אינו פומבי בצורה מלאה. בנוסף, לרוב מערכות המבצעות עיבוד זה הינן יקרות מאד ודורשות יכולות מחשוב חזקות.

בפרויקט זה, אתמקד בבניית מודל לניתוח משחקי טניס, כאשר ארצה לבדוק האם ניתן לבצע ניתוח מספק על ידי שימוש בסרטונים של משחקים רשמיים בלבד בצורה אוטומטית לחלוטין.
הניתוח יתמקד בזיהוי המגרש, השחקנים והכדור, ויאפשר הוצאת סטטיסטיקות משחק שונות אשר יוכלו לשמש מאמנים לשיפור אסטרטגיות משחק וכדומה.
בנוסף לכך, כלל הניתוח יהיה ניתן לביצוע על ידי מחשב סטנדרטי ולא חזק במיוחד, ובזמן סביר.

היקף הפרויקט

על מנת למקד את היקף הפרויקט בחרתי להתמקד בכמה דברים ספציפיים בזמן העבודה:

- זווית הצילום שבחרתי לעבוד איתה, הינה זווית הצילום במשחקי טניס רשמיים, כאשר המצלמה נמצאת מאחורי השחקן בזווית גבוהה (בהמשך יוצגו דוגמאות לזווית צילום זו)
- עבודה רק עם סרטונים של מהלך משחק, ולא סרטונים המכילים קטעי מעבר.
- סוג המגרשים איתם עבדתי, הינם מגרשים "קשים", כלומר אספלט ולא מגרשים של חימר או דשא (במגרשים אלה הזיהוי יותר קשה כי הקווים מוסתרים לרוב או אינם בולטים).
- סיווג החבטות מתבצע לשחקן התחתון בפריים בלבד, בגלל הקרבה שלו למצלמה, ואיכות צילום נמוכה יותר לשחקן העליון.

סקירת ספרות

Hough transform

הגדרת קו

בקואורדינאטות קרטזיות, קו יכול להיות מוגדר בצורה הבאה: $y = ax + b$

כאשר a הוא השיפוע של הקו ו-b הינו החיתוך עם ציר y.

הבעיה בייצוג זה היא שלא ניתן להציג קו המקביל לציר y מפני ששיפועו יהיה אינסוף.

לכן, ניתן להסתכל על קווים גם בקואורדינטות אחרות – קואורדינטות פולריות.

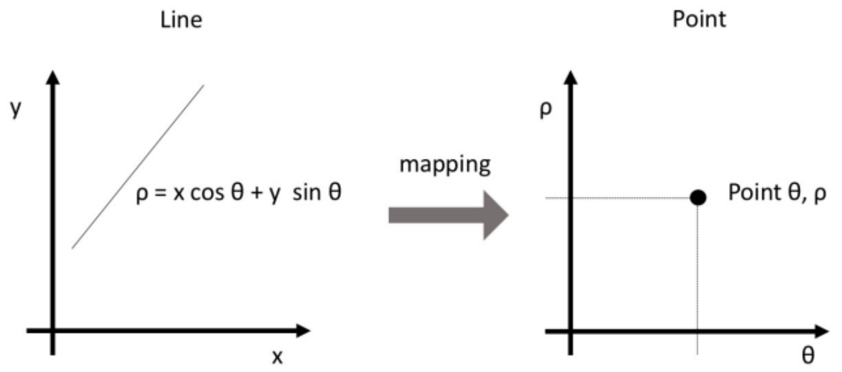
בקואורדינטות אלה, ניתן להגדיר קו בצורה הבאה: $\rho = xcos\theta + ysin\theta$

כאשר ρ ו-θ הם הקואורדינאטות הפולריות של החיתוך בין הקו לבין האנך לקו שעובר דרך ראשית הצירים.

Hough space

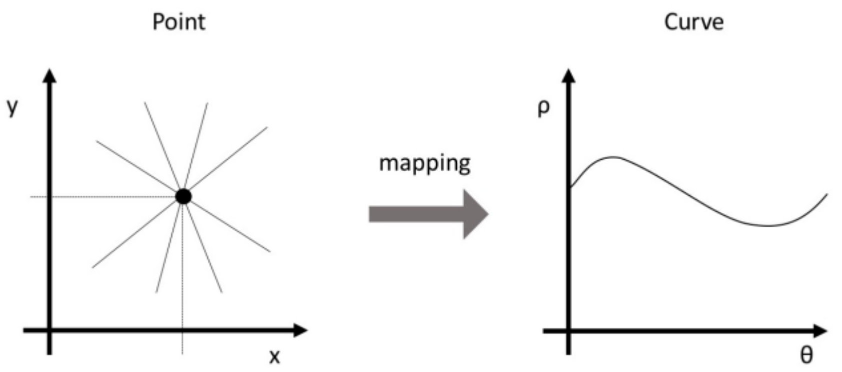
קווים ניתנים להצגה במישור Hough על ידי שני הפרמטרים שלהם ρ, θ.

לכן קו ממופה לנק' במישור Hough לפי פרמטרים אלה.

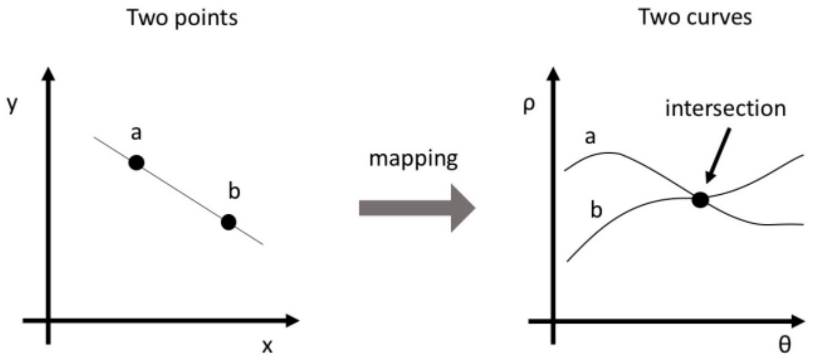


אלגוריתם לזיהוי קווים עם Hough

לאחר שראינו כי קו ממופה לנק' במישור Hough, מיפוי של כל הקווים העוברים דרך נק' כלשהי ממופה לעקום סינוסי כמו שניתן לראות בתמונה הבאה:



עבור שתי נק' במרחב, כל קו העובר דרך כל אחת מהן יוצר עקום נפרד במישור Hough, והחיתוך בין העקומים הנ"ל מייצג את הקו העובר בין שתי הנק'.

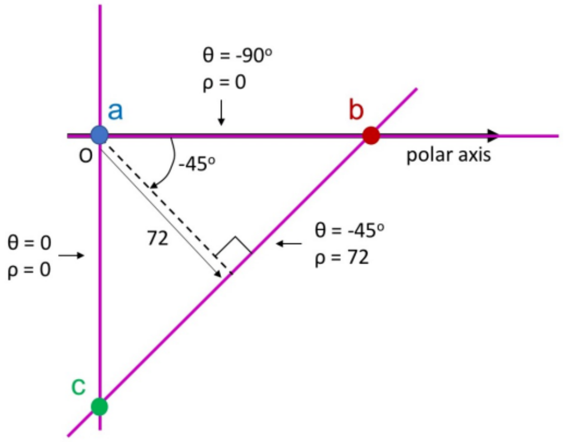


האלגוריתם למציאת קווים, מקבל תמונה בינארית של שפות, ועבור כל פיקסל לבן מחשב את התמרת Hough של נק' זו. לאחר חישוב העקומים לכל נק' בתמונה, האלגוריתם מחפש את נק' החיתוך במישור Hough בה מספר העקומים שנחתכים הינו הגדול ביותר או עוברת סף מסוים.

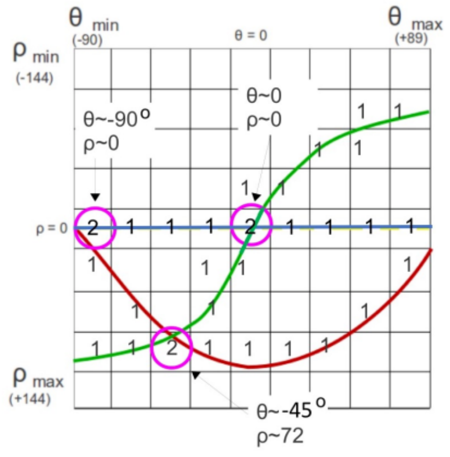
לדוגמא, בתמונה הבאה ישנן 3 נק' a, b, c כאשר ניתן לזהות 3 קווים בתמונה העוברים דרך כל זוג נק'.

בתמונה שאחריה ניתן לראות את מישור Hough אשר מציג את מספר הקווים העוברים בכל נק'.

ניתן לראות כי עבור הפרמטרים המתאימים לשלושת הקווים בתמונה הראשונה יש הצבעה השווה ל2 לעומת שאר המקומות בהן ההצבעה היא 0 או 1.



(a) 2D image with three points a, b and c and the lines that connect them (magenta).



(b) Hough transform accumulator array of (a).

Probabilistic Hough Line Transform

אלגוריתם זה דומה מאד לאלגוריתם הקודם שהוצג, למעט כמה הבדלים קטנים.

באלגוריתם זה מספר הנק' לו מבצעים חישוב קטן בהרבה מהאלגוריתם הקודם וכך נחסך הרבה זמן חישוב ללא איבוד מדיוק האלגוריתם.

באלגוריתם זה, רק סט רנדומלי קטן מתוך כלל הנק' בתמונה משמש לחישוב ההצבעות לקווים, בנוסף הסף שיש לעבור על מנת להיחשב כקו, קטן בהתאם.

מציאת טרנספורמציה הומוגרפית בין שתי תמונות

טרנספורמציה הומוגרפית הינה טרנספורמציה היכולה לבצע עיוות לתמונה בצורה הבאה:

כלומר, יכולה לסובב, להזיז, לעוות ולמתוח את התמונה.

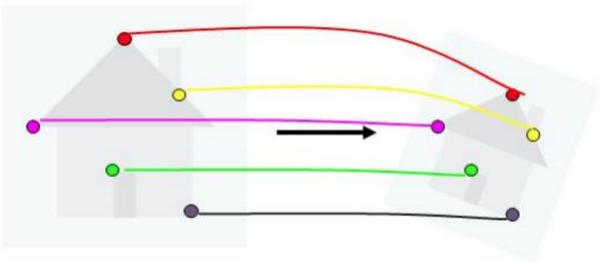
כאשר יש בידינו שתי תמונות של אותו גוף אך עם עיוות כלשהו, ניתן לייצג עיוות זה על ידי מטריצה 3x3 מהצורה הבאה:

$$\begin{bmatrix} u_i' \\ v_i' \\ w_i' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \qquad x_i' \rightarrow \frac{u_i'}{w_i'} \qquad y_i' \rightarrow \frac{v_i'}{w_i'}$$

כאשר x, y הם האינדקסים בתמונה אחת ו x', y' הם האינדקסים בתמונה השנייה.

כעת, על מנת למצוא את הקשר בין התמונות, יש צורך למצוא את המטריצה הנ"ל, כלומר למצוא את שמונת הפרמטרים שלה.

על מנת לעשות זאת, תחילה יש למצוא נק' התאמה בין שתי התמונות. כלומר, נק' אשר שייכות לאותה נק' בגוף האמיתי.



לאחר שנמצא את סט ההתאמות הנ"ל, נוכל למצוא את המטריצה המקשרת בין האינדקסים שמצאנו בצורה הבאה, כאשר נזכור כי הקשר בין כל זוג נק' התאמה הוא לפי המטריצה של הטרנספורמציה:

- The equations we get are

$$x_i' = \frac{ax_i + by_i + c}{gx_i + hy_i + 1} \qquad y_i' = \frac{dx_i + ey_i + f}{gx_i + hy_i + 1}$$

$$\Rightarrow \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i' x_i & -x_i' y_i & -x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -y_i' x_i & -y_i' y_i & -y_i' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ 1 \end{bmatrix} = \begin{bmatrix} \vdots \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

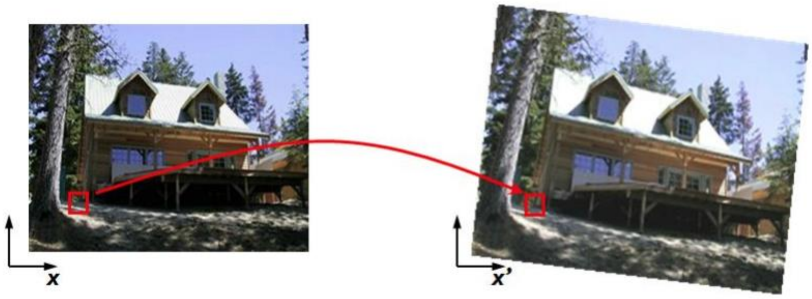
נוכל לפתוח את הכפל במטריצה לפי המתואר בתמונה, ואז להציג את המשוואה הנ"ל בצורה הומוגנית.
כאשר כל זוג נק' נותן לנו 2 שורות במטריצה השמאלית, ועל מנת לפתור את מערכת המשוואות ההומוגנית יש צורך בלפחות 8 שורות במטריצה השמאלית (כמספר המשתנים בוקטור), לכן יש צורך בלפחות 4 זוגות של נק' התאמה בין שתי תמונות.
על מנת לפתור את המשוואות הנ"ל ניתן להשתמש באלגוריתם *SVD* אשר לא אפרט עליו.
אלגוריתם זה מאפשר לקבל פתרון למשוואות ולקבל את הערכים המתאימים למשנים *a-h*.
לאחר מציאת המשתנים הנ"ל, יש בידינו את המטריצה ההומוגרפית המקשרת בין שתי תמונות של אותו גוף.

הפעלת טרנספורמציה הומוגרפית

כאשר יש בידינו טרנספורמציה הומוגרפית המקשרת בין שתי תמונות של אותו גוף, ניתן להפעיל אותה על אינדקסים בתמונה אחת על מנת לקבל את האינדקסים המתאימים בתמונה המתאימה לה.
על פי המשוואה הבאה:

$$\begin{bmatrix} u_i' \\ v_i' \\ w_i' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \qquad x_i' \to \frac{u_i'}{w_i'} \qquad y_i' \to \frac{v_i'}{w_i'}$$

כעת, על מנת למצוא את הקשר ההפוך בין התמונות, ולמצוא איזה אינדקסים בתמונה הראשונה מתאימים לאינדקסים נתונים בתמונה השנייה, יש צורך להפוך את המטריצה הנ"ל ולהפעילה על האינדקסים שנתונים.



Least Squares for line fitting

Least Squares זאת שיטה להערכת פתרון למערכת משוואות, כאשר מספר המשוואות גדול ממספר המשתנים, על ידי מזעור השגיאה הריבועית של כל משוואה.

כאשר יש בידינו מספר נקודות נתונות ונרצה להתאים קו ישר המתאים לכל הנק' הנ"ל נוכל להשתמש בשיטה זו בצורה הבאה:

קו מוגדר בצורה הבאה – $y = mx + b$

כעת נרצה למצוא קו כך שהשגיאה הריבועית שלו בכל נק' היא מינימלית, לכן תחילה נגדיר מהי השגיאה עבור נק' כלשהי (x_i, y_i) אותה נרצה למזער:

$$e_i = |y_i - (mx_i + b)|^2$$

וסך השגיאה שנרצה למזער:

$$e = \sum_i |y_i - (mx_i + b)|^2$$

כעת נוכל לרשום את השגיאה בצורה מטריצית:

$$e = \left\| \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} - \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} \right\|^2 = ||y - Xc||^2$$

כאשר c הוא וקטור הפרמטרים של הקו, y ו- X הינם הערכים המתאימים לנק' לפי המתואר במשוואה.

כעת נרצה למזער ערך זה, ונוכל לבצע זאת על ידי מטריצת הפסאודו אינברס של c.

$$\hat{c} = \begin{pmatrix} \hat{m} \\ \hat{b} \end{pmatrix} = argmin_c(e) = argmin_c\left(||y - Xc||^2\right) = (X^T X)^{-1} X^T y$$

התוצאה שקיבלנו הינה הפרמטרים m ו- b מתאימים לקו שממזער את השגיאה הריבועית המתוארת למעלה.

Resnet

התחום של ראיה ממוחשבת ביצע קפיצה משמעותית ביכולות העיבוד ברגע שהוצגו לראשונה מודלים המכילים בתוכם בלוקים של קונבולוציה. המודל הראשון שהוצג הינו מודל בשם AlexNet.

מאז, התפתחו הרבה מודלים שונים ורשתות עמוקות הנבנות על עיקרון זה.

אחת הבעיות בה נתקלו כאשר ניסו לפתח רשתות קונבולוציה עמוקות, היא התאפסות של הגרדיאנטים בזמן האימון בגלל עומק הרשת. ככל שעומק הרשת גדל, כך הגרדינאטים שהצליחו לחזור על מנת לאמן בלוקים בתחילת המודל קטנו משמעותית, עד הגעה למצב בו לא ניתן לאמן מודלים בצורה זאת.

אחד הפיתוחים המשמעותיים בפתרון הבעיה הינו בלוק בשם Residual block. בלוק זה מאפשר מעבר נקי של הגרדיאנטים בזמן האימון ובכך מונע מהם להתאפס.

בתמונה הבאה ניתן לראות דוגמא למבנה הבלוק:

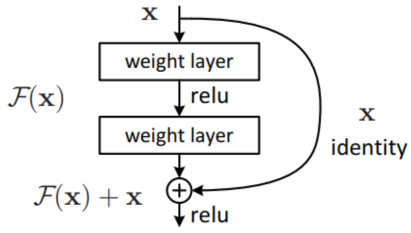


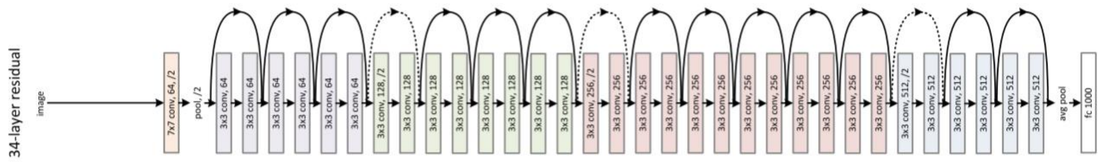
Figure 2. Residual learning: a building block.

ניתן לזהות כי בנוסף למסלול של x העובר בשני דרגות של חישוב, ישנו מסלול זהות אשר מאפשר לx לעקוף את החישוב הנ"ל, ובכך בזמן פעפוע הגרדיאנטים בשלב ה backward, הגרדיאנטים יוכלו לעבור בחופשיות ללא התאפסות.

פיתוח בלוק זה איפשר פיתוח רשתות עמוקות בהרבה מהרשתות שהיו קיימות עד אז.

אחת מהרשתות היא Resnet אשר פותחה למספר עומקים שונים (בפרויקט אתמקד בResnet50 – 50 שכבות).

דוגמא למבנה הרשת עבור עומק של 34:



את מודל זה אפשר לאמן לכמה משימות שונות, ביניהם סיווג של עצמים, סגמנטציה של עצמים, זיהוי ומיקום עצמים שונים ומציאת keypoints של חלקי גוף של אנשים.

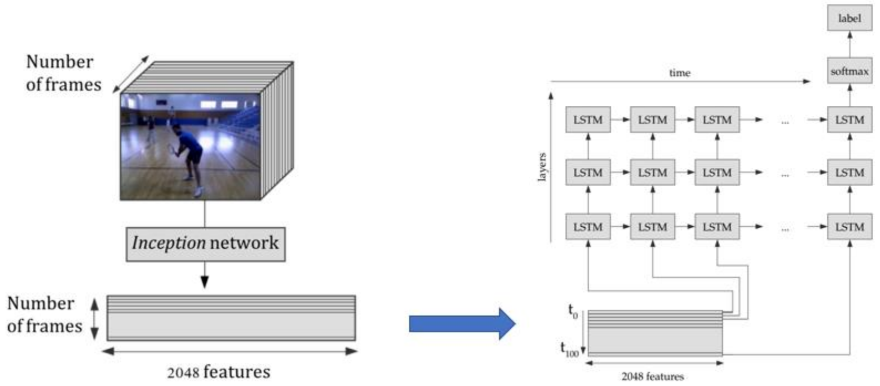
מודל לסיווג חבטות טניס

על פי התזה [1] , על מנת לסווג סוגי חבטות בטניס, ניתן להשתמש במודל למידה עמוקה, המחולק לשני חלקים.

חלק ראשון – רשת קונבולוציה עמוקה אשר אחראית על הוצאת הפיצ'רים מתוך כל הפריימים.

חלק שני – Recurrent neural network אשר אחראית על קבלת רצף של פיצ'רים של פריימים בזמן, וביצוע חישוב המאפשר ביצוע סיווג לסוג המכה שהופיעה בפריימים.

מבנה המודל:



כאשר החלק הראשון של המודל הינו הFeatures extractor, אשר נלקח מתוך רשת בשם Inception שאחראית על סיווג תמונות, ואומנה על דאטה סט בשם ImageNet.

חלק זה מקבל סט של פריימים מתוך סרטון המכיל חבטת טניס, ומוציא סט של פיצ'רים לכל פריים.

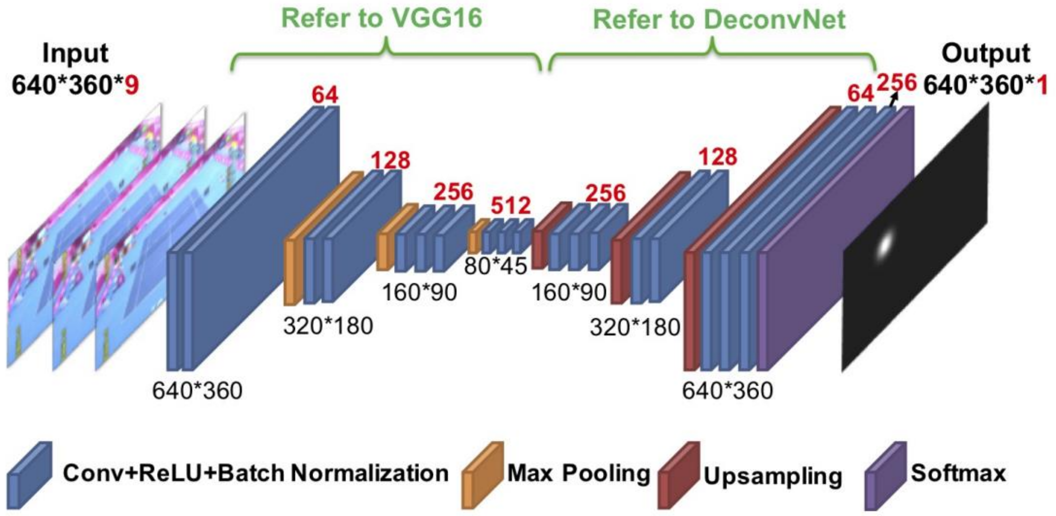
החלק השני של המודל מקבל את סט הפיצ'רים הנ"ל, ומעביר אותו דרך מודל הנקרא LSTM (Long short term memory), כאשר כל פיצ'רים השייכים לפריים מסוים, הינם קלט של המודל בזמן אחר.

כמו שניתן לראות בתמונה, מודל LSTM מבצע את החישוב בזמן, כלומר הפריים הראשון נכנס ראשון למודל, עובר עיבוד בשלוש שכבות של LSTM, כך שכל שכבה מעבירה את הפלט שלה לשכבה הבאה, ואותה שכבה בזמן העיבוד של הפריים העוקב.

לבסוף, בשכבה האחרונה מקבלים ניקוד עבור כל סוג חבטה אפשרי, ולאחר הפעלת פונקציית softmax מתקבלת ההסתברות לכל חבטה.

TrackNet

TrackNet הינו מודל המופיע במאמר [2], אשר משמש לזיהוי כדור טניס במשחקים טניס רשמיים.
הרשת העמוקה במאמר זה היא מסוג של Encoder-Decoder, כלומר חלק ראשון שאחראי על הקטנת הקלטת והוצאת פיצ'רים, וחלק שני שאחראי על הגדלת התוצאה לממדי הקלט ויצירת מפת הסגמנטציה.
להלן תיאור מפורט של הרשת:



אחד היתרונות הגדולים של רשת זו, הוא האפשרות להכניס כקלט מספר פריימים עוקבים במקום פריים בודד. במקרה של המאמר משתמשים בשלושה פריימים עוקבים כקלט. אופציה זו מאפשרת לרשת לחזות את מיקום הכדור גם בפריים בהם הוא מוסתר על ידי השחקן או קשה לזיהוי, על ידי שימוש בתנועת הכדור מהפריימים הקודמים.
פלט הרשת הינה מפת סגמנטציה המציגה את מיקום הכדור, ועל מנת למצוא את המיקום המדויק, משתמשים ב Hough Transform עבור מעגלים (אלגוריתם דומה למה שהוצג בדו"ח, אך עם חיפוש מעגל במקום קו ישר).
בנוסף לרשת זו, מצורף למאמר דאטה סט מתאים, המכיל פריימים ממשחקי טניס רשמיים, בהם מיקום הכדור מתויג בדיוק, וגם מצב הכדור מתויג (מצב מתאר האם הוא מופיע בפריים, מוסתר על ידי עצם כלשהו או מחוץ לפריים).

SORT - SIMPLE ONLINE AND REALTIME TRACKING

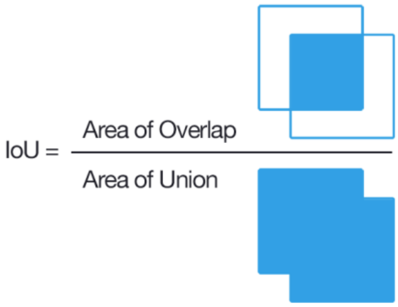
אלגוריתם זה, הינו אלגוריתם המאפשר לבצע מעקב אחר אנשים לאחר זיהוי מיקומם.
כלומר, בנוסף לרשת היודעת לזהות מיקום אנשים כמו שהצגתי קודם לכן (Resnet), ניתן להשתמש באלגוריתם זה עבור סרטונים, כאשר עבור כל פריים נשתמש ברשת לזיהוי האנשים, ולאחר מכן באלגוריתם SORT על מנת לזהות מי האנשים שזיהיתי בפריים והאם הם זוהו בעבר או שהם אנשים חדשים שלא הופיעו קודם לכן.

אלגוריתם זה משתמש אך ורק בbounding box שנמצא על ידי הרשת העמוקה לזיהוי אנשים ובגודל הקופסא על מנת לקבוע מי האנשים בסרטון.

המודל שומר עבור כל דמות שהוא זיהה, את המיקום האחרון של הדמות, את הסקלה, את השטח ואת המהירות (כיוון וגודל) של הדמות.

כאשר מגיע פריים חדש, תחילה מתבצע חיזוי למיקום החדש עבור כל הדמויות שזוהו עד כה.

לאחר מכן מתבצע חישוב IOU (Intersection over union) בין החיזויים של המיקומים החדשים לבין כל הזיהויים החדשים בפריים החדש, חישוב זה כולל בדיקת היחס בין חיתוך האזורים לבין האיחוד ביניהם (כאשר האזורים זהים - היחס מקסימלי).



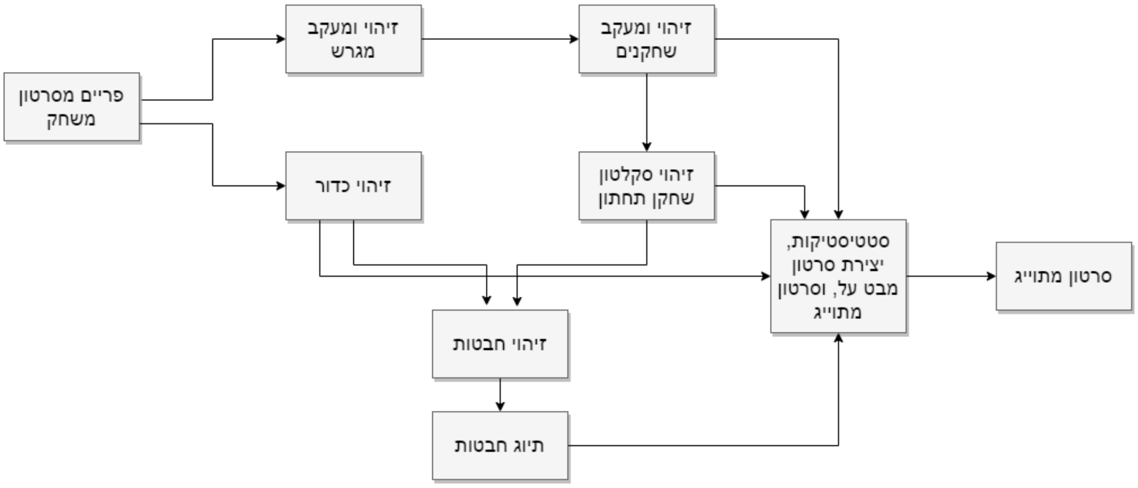
לפי ערך IOU שהתקבל לכל חיזוי וזיהוי חדש, נקבע מי מהדמויות החדשות היא דמות שזוהתה בעבר, או האם הגיעה דמות חדשה שיש להתחיל לעקוב אחריה.

בגלל מבנה האלגוריתם, וחוסר השימוש ברשתות עמוקות או חישובים מורכבים (בדומה להרבה אלגוריתמי זיהוי מחדש אחרים), מהירות החישוב גדולה מאד.

הפתרון שלי ושיקולי תכנון

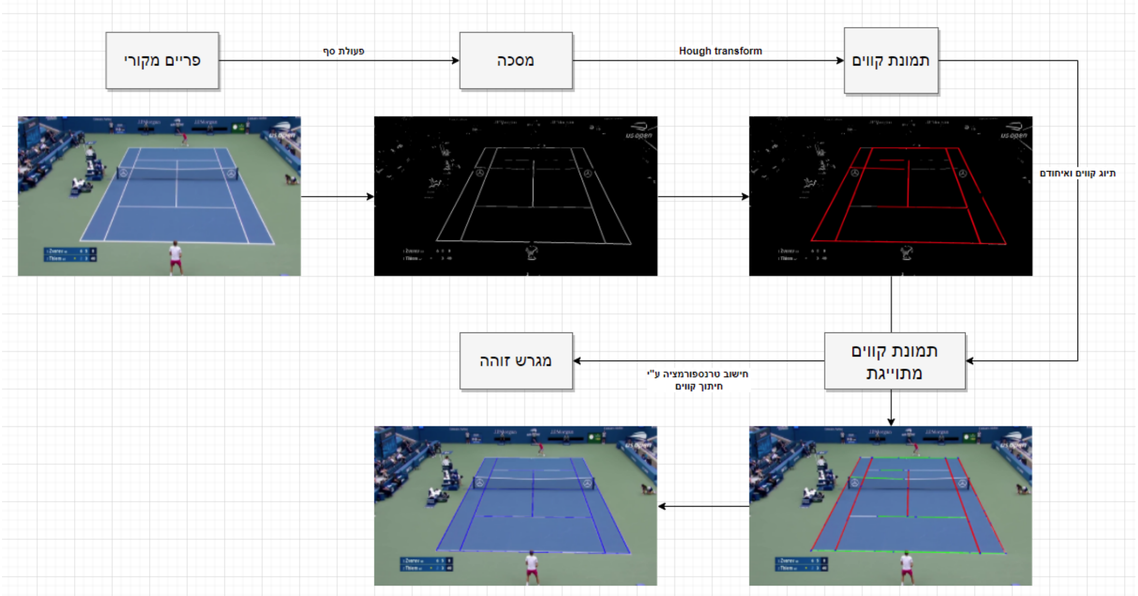
בחלק זה אציג את הפתרון שלי למשימת הפרויקט, וכיצד בחרתי למשש כל חלק מהמשימות.
את העבודה על הפרויקט חילקתי למספר משימות כאשר התמודדתי עם כל משימה בנפרד, בנוסף לכך ניסיתי לעבוד על המשימות בסדר הגיוני כך שאוכל להשתמש בפתרונות קודמים לפתרון המשימה הבאה.
להלן דיאגרמת בלוקים של שלבי פעולת המודל שלי לניתוח משחקים:

תרשים זרימה לפעולת המודל



זיהוי ומעקב אחר המגרש

המשימה הראשונה שבחרתי להתמודד איתה הינה זיהוי המגרש בפריים הראשון, והמשך מעקב אחר מיקומו בשאר הפריימים.
בתמונה הבאה ניתן לראות דיאגרמת בלוקים של תהליך זיהוי המגרש:

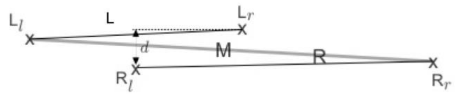


שלב 1: הפעלת פעולת סף על התמונה על מנת לקבל מסכה שמציגה רק את הפיקסלים הלבנים בפריים (ידוע כי המגרש הינו לבן, והבדלי תאורה יכולים להשפיע על עוצמת הערך הלבן).
שלב 2: הפעלת Probabilistic Hough line transform (ניתן לראות בסקירת הספרות כיצד פעולה זו עובדת). בסיום שלב זה נקבל סט של קווים שזוהו בפריים והם חשודים כקווים של המגרש.

שלב 3: תיוג הקווים מחלק את הקווים שזוהו לשני קטגוריות: קווים אופקיים וקווים אנכיים, כאשר התיוג נעשה על ידי חישוב שיפוע הקווים בפריים וזיהוי האם הקו אופקי או אנכי בצורה זו.

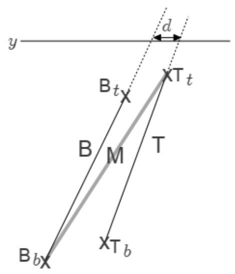
נשים לב, כי לאחר שלב 2 נותרנו עם המון קווים אשר מייצגים את אותו קו במגרש, לכן על מנת להקטין את מספר הקווים שזוהו, ביצעתי איחוד לקווים אשר מייצגים את אותו הקו בצורה הבאה (מוסבר גם בהמשך בשלב הניסויים):

עבור קווים אופקיים:



נאחד את הקווים L ו R לקו הסופי M אם המרחק d קטן מסף מסוים.

ועבור קווים אנכיים:



נאחד את B ו T לקו M אם d קטן מסף מסוים כאשר הישר y מוגדר להיות קו המקביל האופקי של הפריים.

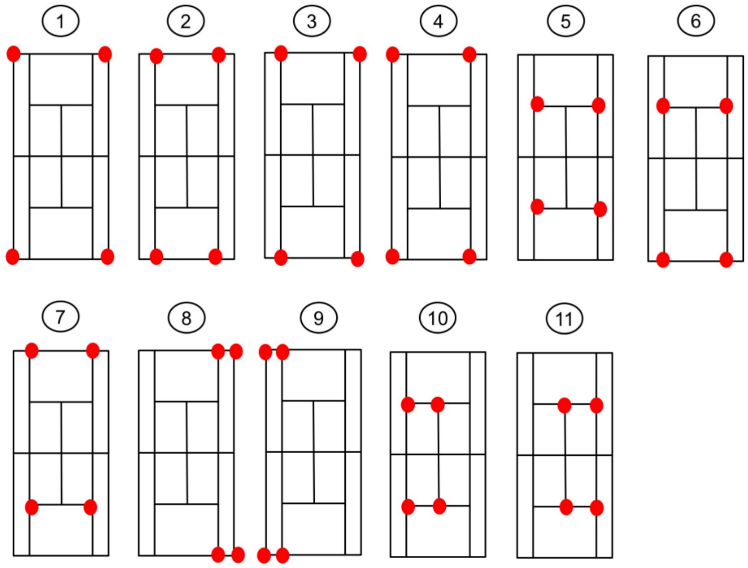
שלב 4: חישוב 4 נק' חיתוך בין כל זוג קווים אופקיים לכל זוג קווים אנכיים, חישוב טרנספורמציה בין קונפיגורציה של תמונת רפרנס של המגרש לבין 4 הנק' שמצאתי (חישוב הטרנספורמציה מוסבר בסקירת הספרות), כאשר כל קונפיגורציה בודקת 4 נק' חיתוך שונות על מגרש הרפרנס. לאחר חישוב הטרנספורמציה לכל קונפיגורציה של נק' חיתוך, מחושבת פונק' ניקוד אשר בודקת כמה הטרנספורמציה שנמצאה מתאימה למגרש בפריים המקורי (החישוב מתבצע על ידי ביצוע עיוות לתמונת הרפרנס של המגרש בעזרת הטרנספורמציה שנמצאה, ואז בדיקה כמה פיקסלים לבנים בפריים נמצאים על הפיקסלים הלבנים של תמונת המגרש המעוותת).

לאחר חישוב הניקוד לכל קונפיגורציה ולכל סט של 4 נק' חיתוך, נבחרת הטרנספורמציה שהובילה לניקוד הגבוה ביותר.

להלן דוגמא לחישוב פונקציית הניקוד עבור סט נק' חיתוך וקונפיגורציות שונות (בירוק ניתן לראות את נק' החיתוך שנמצאו בפריים ובאדום בצד שמאל את הנק' של הקונפיגורציה הנוכחית):

ניתן לזהות כיצד 4 נק' החיתוך שנמצאו בפריים מתאימות לקונפיגורציה האחרונה, ואכן קונפיגורציה זו קיבלה את הניקוד הגבוה ביותר.

הקונפיגורציות השונות שנבחנו הינן:



חישוב הניקוד מתבצע בצורה הבאה:

לאחר עיוות תמונת הרפרנס של המגרש, בודקים עבור כל הפיקסלים של תמונת העיוות האם הם נמצאים על פיקסל לבן או לא בתמונת המסכה.

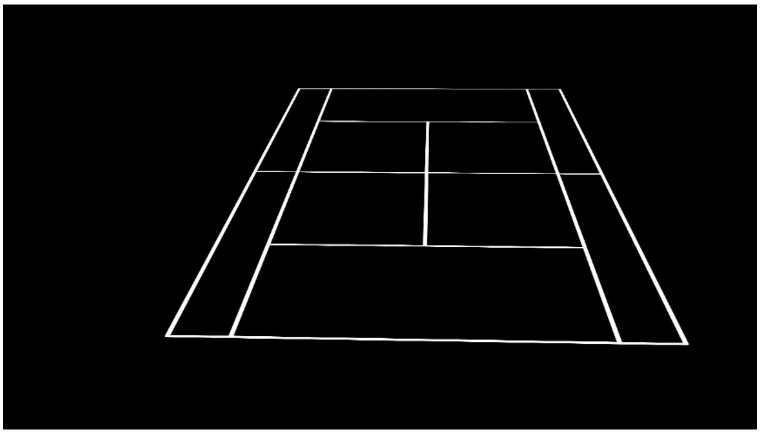
מוסיפים לניקוד:

+1 אם הפיקסל במגרש המעוות אכן נמצא על פיקסל לבן בפריים

-0.5 אם הפיקסל במגרש המעוות נופל על פיקסל שחור בפריים

0 אם הפיקסל במגרש המעוות נופל מחוץ לפריים

המגרש שזוהה בסוף התהליך הינו:

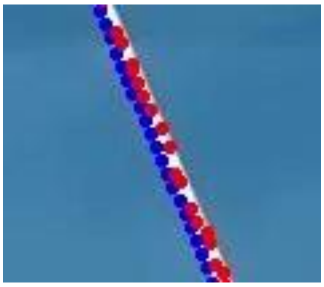


לאחר זיהוי המגרש בפריים הראשון, אני משתמש במידע שמצאתי על מנת למצוא את מיקום המגרש בפריימים הבאים.

תחילה, בזמן זיהוי המגרש אני שומר את הקונפיגורציה המתאימה של המגרש שבעזרתה מצאתי את מיקום המגרש בתהליך הזיהוי, כלומר שמרתי את 4 נק' החיתוך מתמונת הרפרנס של המגרש שאיתם זיהיתי את המגרש (נק' אלה הן חיתוך בין הקווים הבולטים ביותר בפריים הראשון).

לאחר מכן, עברתי על 4 הקווים שהחיתוך שלהם נותן את 4 נק' החיתוך המתאימות בפריים. עבור כל קו בנפרד, בחרתי 100 נק' הנמצאות על הקו במרחקים שווים, ועבור כל נקודה, חיפשתי בפריים החדש בסביבת אותה

נקודה, את הפיקסל בעל הבהירות הגבוהה ביותר אשר ערכו גדול מסף מסוים.



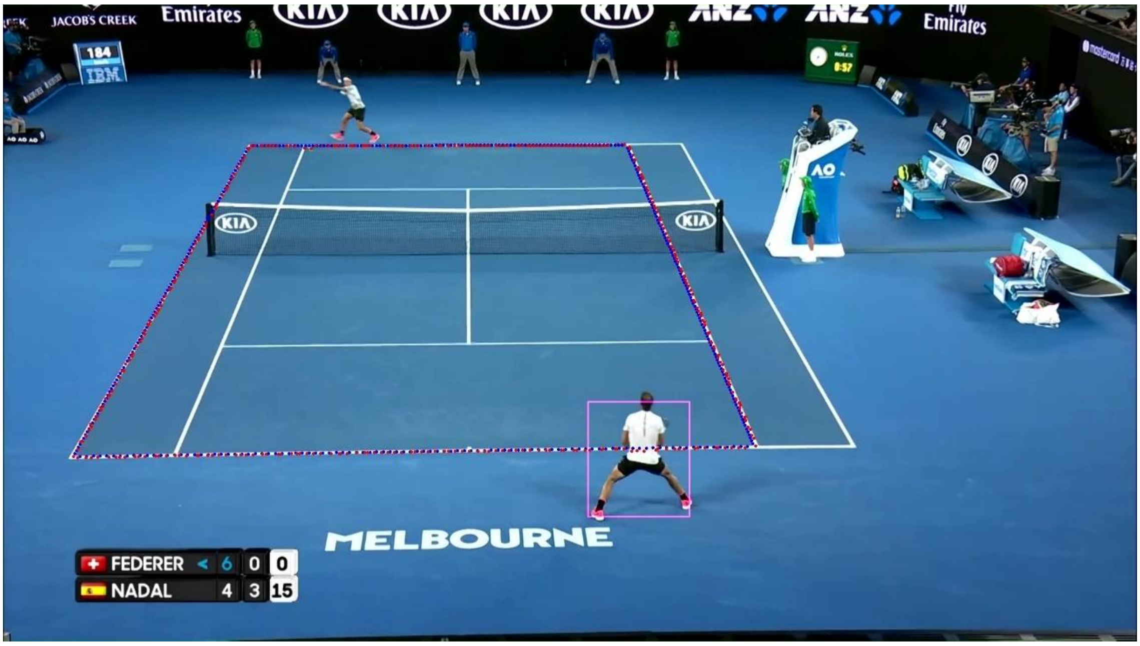
בתמונה ניתן לראות בכחול את הנק' אשר שייכות לקו המגרש מהפריים הקודם, ובאדום את הנק' שנבחרו בפריים החדש לפי הלוגיקה שתיארתי.

לאחר מעבר על כל הנק', אם נמצאו יותר מ-50 נק' חדשות לכל קו בפריים הקודם, אני מוצא קו המתאים לנק' החדשות על ידי שימוש באלגוריתם Least Square שמוסבר בסקירת ספרות. אם לא נמצאו 50 נק', אבצע חיפוש מחדש למגרש לפי אלגוריתם הזיהוי שתואר קודם לכן.

לאחר ביצוע הפעולה הזאת לארבעת הקווים של המגרש, אקבל 4 קווים חדשים לפריים הנוכחי, ואמצא את הטרנספורמציה בין נק' החיתוך של קווים אלה לקונפיגורציה של מגרש הרפרנס ששמרתי בזיהוי המגרש הראשוני.

כעת יש בידי את הטרנספורמציה שמתאימה לי בין רפרנס המגרש, לבין המגרש בפריים, כלומר המגרש זוהה בפריים החדש.

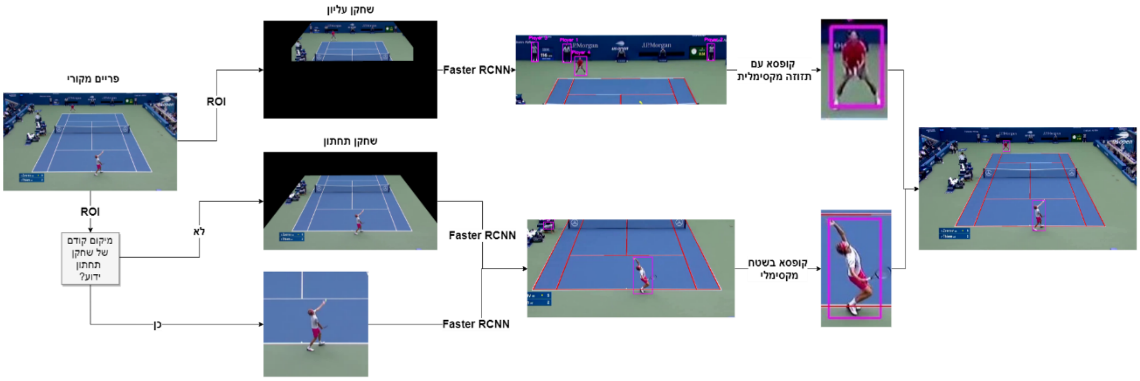
ניתן לראות בתמונה הבאה את זיהוי הנק' החדשות בפריים מלא:



זיהוי ומעקב אחר השחקנים

המשימה השנייה שבחרתי להתמודד איתה היא זיהוי השחקנים ומעקב אחריהם במהלך המשחק.

להלן דיאגרמת בלוקים של תהליך הזיהוי:



תחילה התמקדתי בשחקן התחתון מפני שזיהוי שלו הינו קל יותר בגלל הקרבה למצלמה ובגלל שמספר האנשים סביבו קטן מאד עד אפסי ברוב המקרים.

זיהוי השחקן התחתון נעשה על ידי שימוש ברשת עמוקה מסוג Faster RCNN שאומנה על הדאטה סט ImageNet אשר מכיל סט של תמונות לביצוע detection בעזרת bounding box.

זיהוי השחקן נעשה בפריים הראשון לאחר זיהוי המגרש, לכן קלט הרשת הינו האזור בפריים בו השחקן התחתון יכול להימצא, כלומר הקלט הינו החצי התחתון של המגרש.

בתמונה הבאה ניתן לראות דוגמא לקלט הרשת בפריים הראשון:

חשוב לציין כי "אזור העניין" הינו טיפה גדול יותר מחצי מגרש, על מנת לאפשר זיהוי גם במקרים בהם השחקן נמצא קרוב לרשת כך שחלק גופו העליון נמצא באזור המגרש העליון.

לאחר מעבר ברשת העמוקה, אקבל סט של bounding boxes אשר מתאים לכל האנשים שזוהו בקלט, מתוך סט זה אני בוחר את ה"קופסא" בעלת השטח הגדול ביותר להיות השחקן בפריים (הנחה זו סבירה מפני שלרוב אם יזוהו דמויות נוספות בפריים הראשון, הן יהיו של שופט\ נער כדורים\ קהל אשר יושבים או מכופפים כך שהשטח שלהם קטן משל השחקן אשר עומד).

בתמונה הבאה ניתן לראות כיצד זוהה השחקן התחתון בפריים:

לאחר זיהוי השחקן, מתבצע מעקב בפריים העוקבים לפריים הראשון.

בכל פריים אשתמש שוב ברשת העמוקה Faster RCNN , אך כעת הקלט יהיה סביבה של מיקום השחקן בפריים הקודם (בהתאם להנחה כי השחקן לא יכול לזוז הרבה ממיקומו בפריים הקודם), בנוסף לכך אקטין את סף הביטחון של הרשת בדמות כך שתוכל לזהות את השחקן גם במקרים בהם הרשת לא בטוחה בזיהוי (בגלל הביטחון כי השחקן חייב להיות בסביבה שנבחרה).

דוגמא לסביבה בה נחפש את המיקום הבא:

לאחר זיהוי השחקן התחתון, ניתן להתקדם לזיהוי השחקן העליון.

זיהוי זה הינו יותר קשה מכמה סיבות: מרחק השחקן מהמצלמה פוגע באיכות הצילום והזיהוי, לרוב מופיעים בסביבת השחקן שופטי קו או אנשים אחרים אשר יכולים לפגוע בזיהוי השחקן.

לכן, יש צורך להשתמש בשיטה אחרת לזיהוי השחקן העליון.

על מנת לזהות את השחקן העליון תחילה בדומה לזיהוי השחקן התחתון בחרתי את "אזור העניין" להיות חצי

המגרש העליון:

כעת השתמשתי ברשת העמוקה Faster RCNN על מנת לזהות את כל האנשים בפריים, ושמרתי את מיקומם לכל אורך הסרטון.

בנוסף לזיהוי האנשים בפריים השתמשתי באלגוריתם SORT שמתואר בסקירת הספרות, על מנת לבצע מעקב אחר מיקום כל דמות בפריים.

כלומר לדוגמא אם בפריים הראשון זוהו 4 אנשים, ובפריים הבא זוהו 5 אנשים, אלגוריתם SORT ידע להגיד לי איזה bounding box בפריים החדש שייך לאיש שזוהה בפריים הקודם ואיזה bounding box שייך לדמות חדשה. בצורה זאת אבצע מעקב אחר הדמויות המופיעות לכל אורך הסרטון.

חשוב לציין כי ישנם מקרים בהם קיימים מספר פריימים בהם הרשת העמוקה אינה מצליחה לזהות אדם מסוים ולאחר מכן מזהה אותו שוב כאדם חדש לגמרי, לכן בסוף התהליך אקבל סט של אנשים שזוהו בכל פריים וייתכן כי חלק מהאנשים שזוהו שייכים לאותה דמות בסרטון אך בפריימים שונים.

לאחר זיהוי כל האנשים בסרטון, אני מבצע חישוב למרחק אותה עברה כל דמות שזוהתה, כלומר אעבור על כל אדם שזוהה, אחשב את מרכז המסה שלו בכל פריים, ואחשב כמה מרכז זה זז לאורך הסרטון.

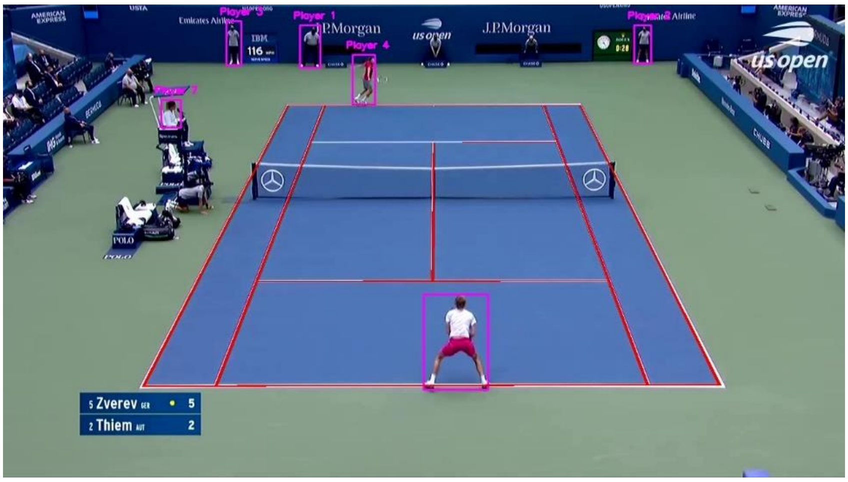
מתוך ההנחה כי השחקן יהיה הדמות שזזה הכי הרבה בסרטון וכי שופטי הקו אינם זזים כלל, אוכל לסנן מתוך כל האנשים שזוהו את אלה שזזו הכי הרבה ובכך לזהות את השחקן.

לאחר זיהוי הדמות שזזה הכי הרבה, אוכל לסנן מתוך סט האנשים שזוהו את האנשים שמופיעים בפריימים זהים לדמות זו (כלומר כל אדם שזוהה במקביל לדמות שזזה הכי הרבה, הוא איננו השחקן). כך אשאר עם סט אנשים אשר זוהו בקטעים זרים מהקטע בו זיהיתי את הדמות שזזה הכי הרבה.

כעת אפעיל שוב את אותה לוגיקה ואבחר שוב את הדמות שזזה הכי הרבה מאלה שנותרו ואגדיר אותה כשחקן.

לאחר מספיק הפעלות חוזרות אשאר רק עם דמויות שזזו הכי הרבה בכל מקטע וכך אזהה את השחקן העליון.

בתמונה הבאה ניתן לראות את זיהוי האנשים בחלק העליון של המגרש בזמן העיבוד:



ובתמונה הבאה ניתן לראות את הזיהוי הסופי של השחקן העליון:

זיהוי שלד שחקן תחתון

המשימה הבאה היא זיהוי השלד של השחקן התחתון.

עבור הפרויקט בחרתי להתמקד בעיקר בשלד של השחקן התחתון ולא של העליון, מהסיבה שאיכות התמונה של השחקן העליון אינה מספיק טובה על מנת להוציא את השלד שלו בצורה מספיק טובה.

על מנת לבצע זאת, השתמשתי ברשת עמוקה מסוג Keypoint RCNN (מידע נוסף נמצא בסקירת הספרות).

מודל זה מקבל תמונה המכילה אנשים, ומוציא חיזוי למיקום 17 נק' בגוף האדם (ביניהם ידיים, רגליים וכו').

לאחר מציאת מיקום השחקן התחתון במשימה הקודמת, השתמשתי במיקום זה על מנת להכניס לרשת את השחקן התחתון בלבד (ללא שאר הפריים).

פלט הרשת נתן לי בדיוק את המידע הנדרש לבניית השלד שלו.

בגלל אופי העבודה, בו מבצעים חישוב “frame by frame” , כלומר כל פריים נכנס בנפרד לרשת, מתרחשת בעיה של ריצוד הנק'. כלומר, חיזוי נק' הגוף של השחקן זזות כמה פיקסלים כל פריים ואינן קבועות במקום. בעיה זו גורמת לסוג רק ריצוד או רטט של שלד השחקן והחיזוי אינו חלק.

על מנת להתמודד עם הבעיה הזאת בחרתי להוסיף למודל הנ"ל שלב עיבוד נוסף בסוף העבודה.

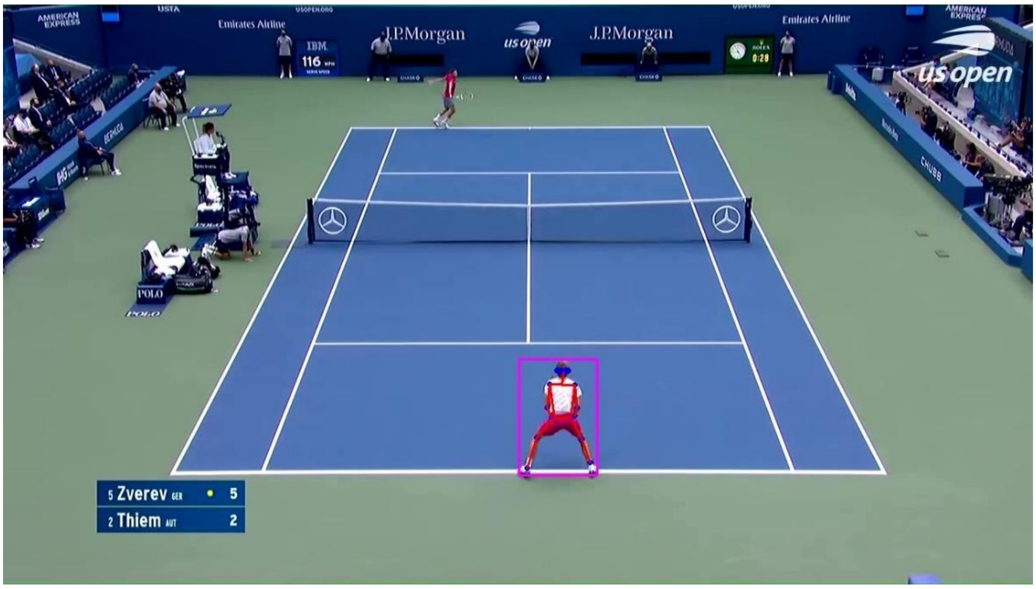
שלב זה אחראי על החלקת התוצאות המתקבלות מהרשת.

ההחלקה מתבצעת עבור כל חלק גוף בנפרד, כאשר עבור כל חלק גוף ההחלקה מחושבת עם חלון של 7 פריימים ברצף, ומנסה להתאים פונקציה ריבועית למיקומי האיבר שזוהה.

לדוגמא, אם נבצע החלקה למיקום יד ימין, נעבור על כל המיקומים שמצאנו ליד ימין בזמן העיבוד, ועבור כל רצף של 7 פריימים נתאים פונקציה ריבועית למיקומי היד.

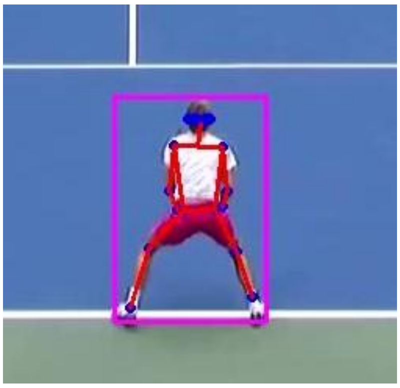
כך בסוף התהליך נקבל את המיקומים החדשים לאחר ההחלקה, והשלד של השחקן ינוע בצורה חלקה ללא ריצודים וקפיצות.

בתמונה הבאה ניתן לראות את התוצאה הסופית של הוצאת השלד מהשחקן:



בתמונה ניתן לראות בכחול את הנק' שזוהו על ידי הרשת ובאדום את הקווים המחברים בין הנק'.

בתמונה הבאה ניתן לראות את הקלט של הרשת – סביבה מסביב למיקום השחקן:



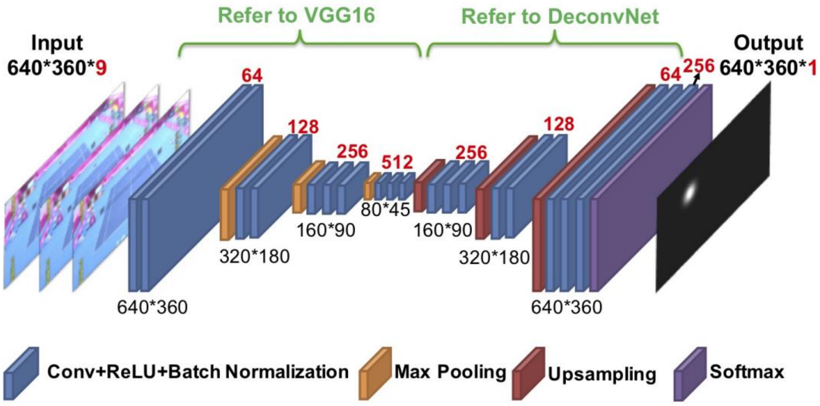
זיהוי הכדור

המשימה הבאה היא זיהוי הכדור בפריים.

על מנת לבצע משימה זאת עבדתי לפי המאמר TrackNet [2], ובניתי בעצמי את הרשת העמוקה המתוארת במאמר (ניתן לראות פירוט על המאמר והרשת בסקירת הספרות).

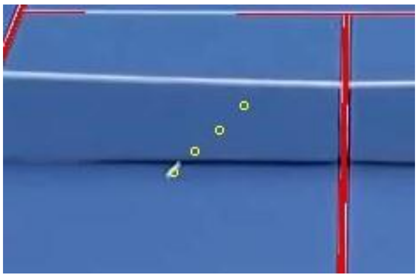
אזכיר בקצרה כי הרשת מקבלת כקלט שלושה פריימים עוקבים מתוך משחק רשמי של טניס, והפלט שלה זאת מפת סגמנטציה המציגה היכן נמצא הכדור בפריים האחרון. את הרשת אימנתי על סט הפריימים שמתואר במאמר אשר נבנה על סרטונים ממשחקים רשמיים.

תזכורת למבנה הרשת:



על מנת למצוא את מיקום הכדור בפלט הרשת, אני משתמש בHough Transforms עבור מעגל, כאשר אני מחפש מעגל בגודל המתאים לכדור טניס סטנדרטי בתמונת הסגמנטציה שהתקבלה, ובוחר את המעגל עם מספר ההצבעות הגבוה ביותר בטרנספורמציה.

בתמונה הבאה ניתן לראות את זיהוי הכדור בפריים, בנוסף סימנתי גם את 4 המיקומים הקודמים של הכדור כדי להציג את תנועתו:



זיהוי רגע החבטה

לאחר שיש בידי את מיקום המגרש, השחקנים והכדור, מצאתי את רגעי החבטה בסרטון על ידי הצלבה בין מיקום הכדור למיקום השחקנים.

על מנת לבצע זאת, תחילה הסתכלתי על אינדקס y של הכדור וחיפשתי נק' קיצון, כאשר הנחתי כי נקודות קיצון מעידות על שינוי בכיוון תנועת הכדור.

נק' מקסימום מסמלת נק' החשודה כי בה הכדור שינה את כיוון תנועתו בפריים מלמטה למעלה (החלק התחתון בפריים בעל ערכי y גדולים יותר), ובצורה דומה נק' המינימום הינן נקודות החשודות כשינוי כיוון התנועה מלמעלה למטה.

כלומר, נק' המקסימום חשודות כרגעי חבטה של השחקן התחתון, ונק' מינימום חשודות כרגעי חבטה של השחקן העליון.

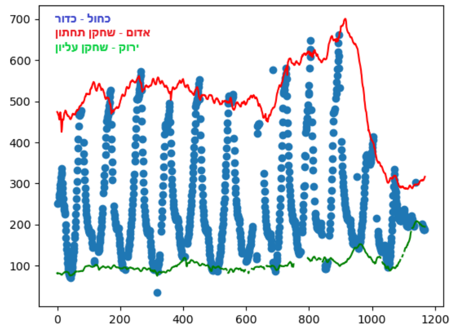
כעת, לאחר שהסקתי זאת ומצאתי את נק' הקיצון, עבור כל נקודה שנמצאה חישבתי את מרחק הכדור מהשחקן המתאים.

כאשר הנק' של השחקן מהן חישבתי את המרחק הן: מרכז המסה שלו, מיקום יד ימין ומיקום יד שמאל (אם קיים).

עבור שלושת הנק' הנ"ל בחרתי את המרחק המינימלי, ובדקתי האם הוא קטן מסף מסוים.

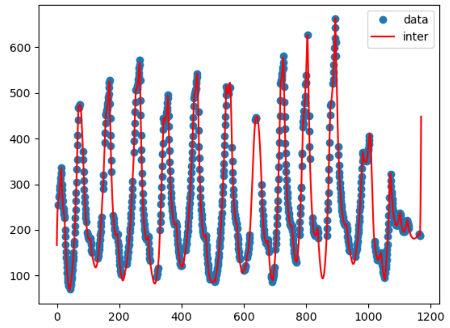
במקרה ואכן המרחק קטן מהסף, הסקתי כי הסיבה שהכדור שינה את תנועתו היא מחבטה של אותו שחקן.

בגרפים הבאים ניתן לזהות את אינדקס y של הכדור והשחקנים ולזהות מתי הייתה חבטה לפי החיתוך ביניהם:

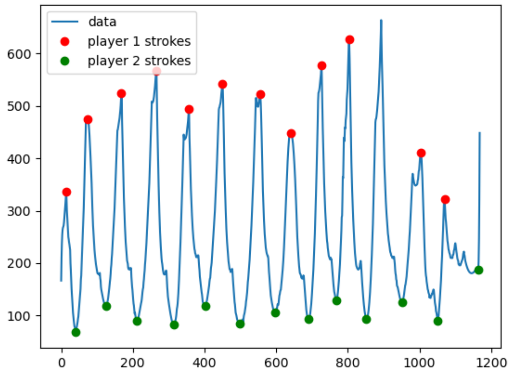


ניתן לראות בגרף כי הנק' בהן זוהה הכדור אינן רציפות (לרוב בגלל מהירות תנועתו, או בגלל זיהוי שגוי של מיקומו), לכן על מנת למצוא את נק' הקיצון היה צורך לבצע אינטרפולציה למיקום הכדור.

בגרף הבא ניתן לראות את האינטרפולציה שביצעתי כאשר ביצעתי אינטרפולציה ריבועית:



ובגרף הבא ניתן לזהות את רגעי החבטה שזוהו לפי הלוגיקה שהסברתי קודם לכן:

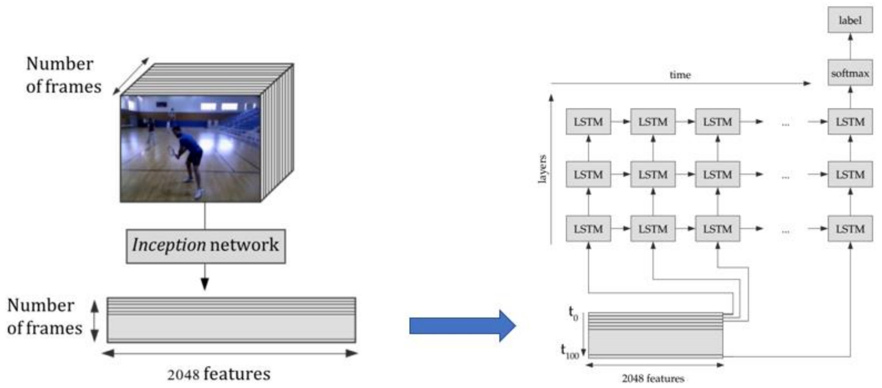


ניתן לזהות בגרף כי ישנו מקרה בודד בו לא זוהתה חבטה, אך מקרה זה אינו נפוץ ולרוב מתרחש בגלל זיהוי לא מספק של השחקן\הכדור או יציאת חלק מדמות השחקן מהפריים.

תיוג החבטה

על מנת לתייג את סוג החבטה בניתי מודל לפי המתואר בתזה [1], תחילה בניתי בעצמי את המודל המתואר בתזה, המכיל חלק שאחראי על הוצאת הפיצ'רים מתוך הפריימים, וחלק נוסף האחראי על חיזוי המכה על ידי סט של פיצ'רים של רצף פריימים (המודל מוסבר בסקירת הספרות).

להלן תזכורת למבנה המודל:

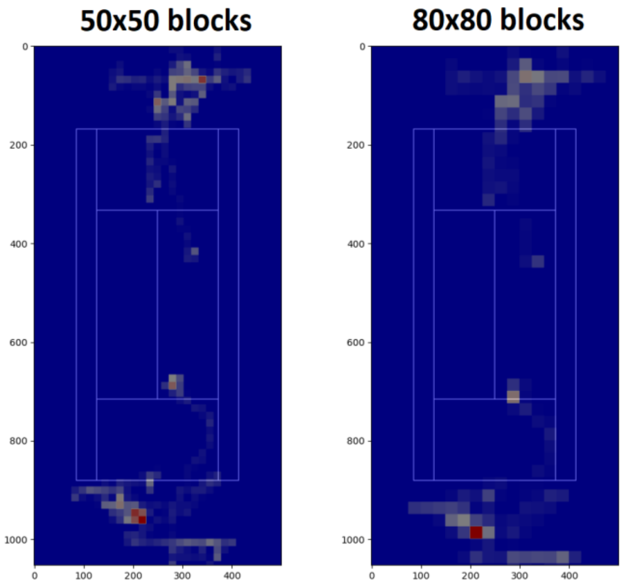


את הרשת אימנתי על ידי הסט שבניתי בעצמי Strokes Dataset (בהמשך אסביר בהרחבה על סט זה).
הרשת מקבלת כקלט רצף של פריימים, ומוציאה כפלט את ההסתברות שחבטה מסוימת הופיעה בפריימים.
עבור הפרויקט בחרתי לעבוד עם שלושה סוגים של חבטות, Forehand, Backhand, Smash/Service.
לאחר זיהוי רגע החבטה של השחקן התחתון, נלקחים מספר פריימים המתאימים לבערך 2 שניות, מסביב לרגע החבטה עצמה.
פריימים אלה מוכנסים אל המודל הנ"ל על מנת לקבל חיזוי איזו חבטה התרחשה.
בתמונה הבאה ניתן לזהות את הרגע בו התבצע חיזוי לסוג החבטה:

סטטיסטיקות וסרטון מבט על

בשלב זה כל הנותר לעשות הוא להשתמש במידע שצברנו מהשלבים הקודמים ולהסיק ממנו מסקנות.
תחילה לאחר שידוע לנו מיקום השחקנים בכל פריים, ניתן הוציא heatmap המציג את מיקומם בסרטון שלם.
על מנת לחשב את הheatmap, כל שנותר לעשות הוא לחלק את המגרש לריבועים קטנים יותר (בחרתי ריבועים בגודל 80x80 אך כל גודל אפשרי לפי הרזולוציה שרוצים), לאחר מכן לעבור על כל הפריימים בסרטון, ועבור כל פריים להוסיף ניקוד בריבוע בו נמצא כל שחקן.
בסיום חישוב זה, נותר רק להציג את הheatmap שהתקבל על גבי תמונה של המגרש וכך נוכל להציג היכן שהה השחקן הרבה זמן.

להלן דוגמא לסטטיסטיקת מיקום השחקן עבור אחד הסרטונים עבור ריבועים בגודל 80x80 ו- 50x50:



בנוסף לכך, עבור השחקן התחתון (אשר עבורו מתבצע סיווג לחבטות) ניתן לחשב סטטיסטיקות כגון מספר החבטות בהן השתמש במהלך המשחק, או המיקום במגרש בו בחר להשתמש בכל חבטה.

סטטיסטיקה נוספת שניתן לחשב היא המרחק שעבר כל אחד מהשחקנים.

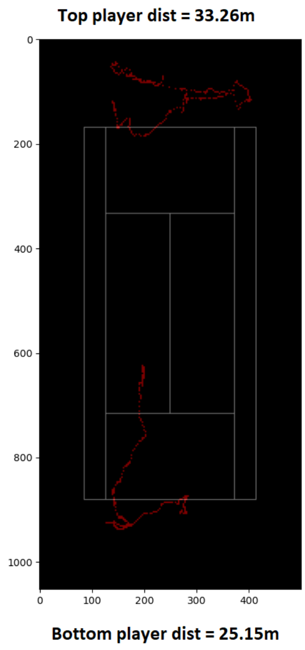
על מנת לחשב זאת, כל שנותר הוא לחשב את המרחקים שזז הbounding box של כל שחקן בפיקסלים ולהעביר מידע זה למטרים.

חישוב המרחק מתבצע על מגרש הרפרנס לאחר העברת המיקומים של השחקנים למגרש, ולאחר מכן מעבר על כל המיקומים וחישוב המרחק בין כל שני מיקומים רצופים.

את מגרש הרפרנס בניתי בצורה כזאת שכל סנטימטר בעולם האמיתי, יעבור לפיקסל יחיד בתמונת הרפרנס.

כלומר, על מנת לחשב את מספר המטרים שעבר כל שחקן, כל שעלי לעשות הוא לחלק ב100 את מספר הפיקסלים שעבר בזמן המשחק.

בתמונה הבאה ניתן לראות דוגמא למסלולים שעבר כל שחקן והמרחק שעבר:



לאחר חישוב סטטיסטיקות שונות, יצרתי סרטון חדש למהלך המשחק, כאשר הסרטון מציג את המשחק ממבט על.

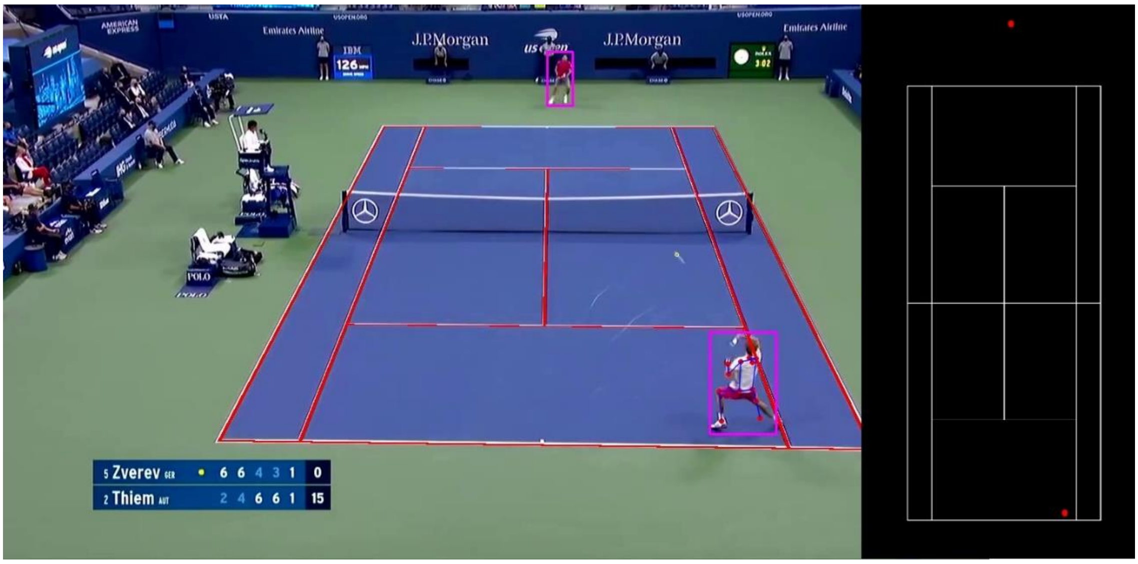
על מנת ליצור את הסרטון הנ"ל, השתמשתי במיקום השחקנים שמצאתי, בנוסף לטרנספורמציה שחושבה על מנת לקשר בין המגרש בפריים לבין המגרש בתמונת הרפרנס.

תחילה, נזכר כי הטרנספורמציה שמצאתי בחלק של זיהוי ומעקב המגרש, מבצעת עיוות של מגרש הרפרנס למגרש בפריים. כלומר, אם ארצה למצוא את מיקום השחקן במגרש הרפרנס (ממבט על), כאשר אני יודע את מיקומו בפריים, יש צורך בטרנספורמציה ההופכית לטרנספורמציה שמצאתי, כלומר יש להפוך את המטריצה של הטרנספורמציה.

כעת כל שנותר הוא להפעיל את הטרנספורמציה ההפוכה על מיקומי השחקנים בכל פריים (כאשר חשוב לשים לב כי יש להשתמש בטרנספורמציה המתאימה לפריים המתאים, מפני שייתכן כי היא שונה מפריים לפריים בגלל תזוזת המצלמה).

מיקום השחקן מחושב כמיקום הרגליים שלו (החלק התחתון של הbounding box המתאים לשחקן), מפני שהרגליים שלו נמצאות כמעט על אותו מישור כמו המגרש ולכן הטרנספורמציה תמצא את מיקום השחקן במדויק בתמונה ממבט על (כיצד להפעיל את הטרנספורמציה מוסבר בנוסף בסקירת הספרות).

בתמונה הבאה ניתן לראות פריים מתוך סרטון בו בוצע המעבר מהפריים למבט על:



דאטה-סטים

בחלק זה אציג את הדאטה-סטים השונים בהם השתמשתי במהלך הפרויקט.

THETIS

סט זה מכיל 1980 סרטונים של אנשים מבצעים חבטות טניס מסוגים שונים.

סט הסרטונים מורכב מ-12 סוגי חבטות שונות אשר מבוצעות על ידי 31 חובבנים ו-24 מקצוענים.

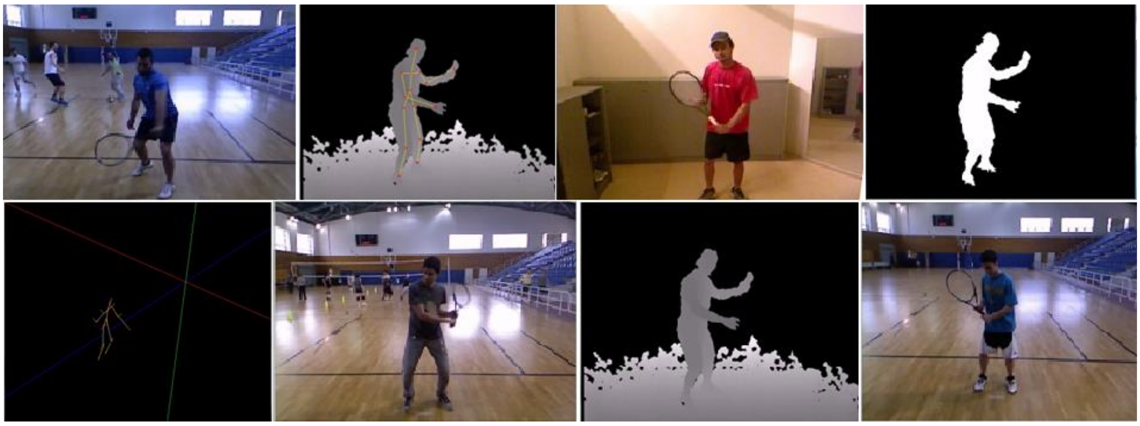
בנוסף לסט הסרטונים, לכל סרטון מצורפים: תיוג המכה בסרטון, סרטון ממצלמת עומק, סרטון המכיל מסכה המציגה את האדם בלבד, וסרטונים המכילים את ה-skeleton של האדם.

הסרטונים מצולמים כאשר האדם החובט עומד מול המצלמה, ומבצע את החבטה ללא כדור.

בסט זה השתמשתי על מנת לבדוק את המודל שמנסה לזהות את סוגי המכות של השחקנים במשחק.

בגלל אופי סט זה, הגעתי למסקנה כי למעט בדיקת איכות המודל, לא אשתמש בו לשאר המשימות מפני שאינו מתאים לזווית הצילום ולסוג הסרטונים עליהם הפרויקט מתמקד.

להלן דוגמא לכמה פריימים מתוך סט זה:



Strokes

סט זה הינו סט שיצרתי בעצמי, הנועד לבצע את זיהוי החבטות בסוגי הסרטונים בהם הפרויקט מתמקד.

על מנת ליצור את סט זה עבדתי בכמה שלבים:

1. הורדת סרטונים של משחקים רשמיים מYouTube
2. סימון ידני של רגעי החבטות של השחקן התחתון בפריים (על ידי מערכת תיוג שמריצה את הסרטונים ומסמנת פריימים בלחיצת כפתור במקלדת)
3. הרצת מודל לזיהוי השחקן התחתון וחיתוך חלון מסביב למיקום השחקן
4. תיוג ידני של סוגי החבטות (על ידי מערכת תיוג שמריצה את הסרטונים ושומרת תיוג לפי לחיצה ידנית במקלדת)

סט זה מכיל 1401 סרטונים מתוך עשרות משחקים שונים, של שחקנים שונים ובמגרשים שונים.

סוגי החבטות שתויגו בסט זה הינם Forehand, Backhand, Smash, Service,

כאשר Smash ו- Service קיבלו את אותו תיוג בגלל מאפיינים דומים מאד.

הסבר נוסף על אופן בניית הסט ניתן למצוא בחלק של הניסויים תחת סיווג סוגי חבטות.

להלן דוגמא לכמה פריימים מתוך סרטונים בסט זה:

TrackNet

סט זה הינו סט של פריימים מתוך משחקים רשמיים, כאשר בכל פריים מתויג מיקום הכדור המדויק.

השתמשתי בסט זה על מנת לאמן מודל שמסוגל לבצע סגמנטציה לכדור טניס במשחקים רשמיים.

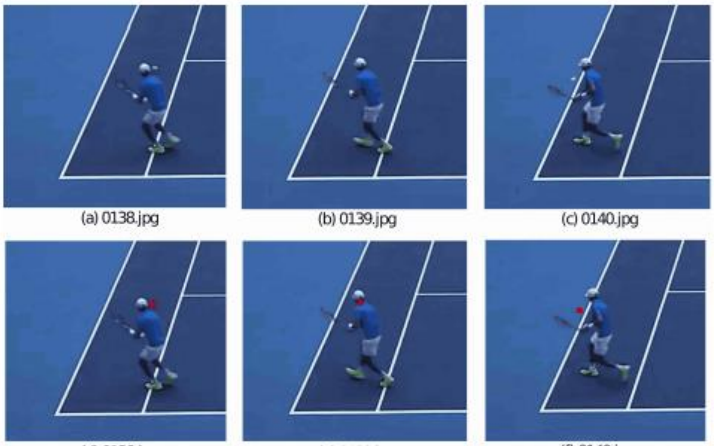
סט זה מכיל 20,844 פריימים, כאשר לכל פריים מצורף תיוג המכיל את: אינדקסים x, y של מיקום הכדור בפריים, האם הכדור מופיע בפריים או לא, וסיווג של תנועת הכדור (תנועה רגילה, חבטה, פגיעה במגרש).

השימוש בסט זה מתבצע כך שבכל שלב לוקחים 3 פריימים רצופים, ומחפשים את מיקום הכדור בפריים האחרון.

דוגמא לרצף של שלושה פריימים מתוך סט זה:

(a) 0021.jpg (b) 0022.jpg (c) 0023.jpg

דוגמא לרצף של שלושה פריימים כאשר הכדור מוסתר על ידי השחקן בחלק מהם:



(a) 0138.jpg (b) 0139.jpg (c) 0140.jpg

(d) 0138.jpg (e) 0139.jpg (f) 0140.jpg

<u>ניסויים</u>

בחלק זה של הדוח אציג חלק גדול מהניסויים שהרצתי בזמן העבודה על הפרויקט, ייתכן כי הרבה מהמידע שרשום לפני כל ניסוי חוזר על עצמו מהחלק שמציג את הפתרון, אך השתדלתי להוסיף כל מידע שנראה לי רלוונטי לכל ניסוי.

זיהוי המגרש

בחלק זה עבדתי על זיהוי המגרש ומציאתו בתמונה.

<u>מציאת קווים בתמונה</u>

בחלק זה אעבוד תחילה עם התמונה הבאה של מגרש, ובחלקים בהם אשתמש בתמונות נוספות, אציין זאת לפני.

תמונה מקורית:

משימה ראשונה הייתה מציאת הקווים בתמונה, על מנת לבצע זאת, תחילה השתמשתי במידע הקיים על מבנה וצבע המגרש.

ידוע לי כי צבע הקווים במגרש תמיד לבן, אך בגלל שינויי תאורה, ייתכן כי הצבע של הקווים ינוע בין אפור ללבן.

לכן תחילה, אבצע פעולת סף עם ערך 200 על תמונה שחור-לבן של המגרש שהתקבלה, כך שאקבל תמונה בינארית בה רק פיקסלים עם ערך אפור גדול מ200 יופיעו.

כעת המשכתי למציאת הקווים מתוך התמונה שהתקבלה.

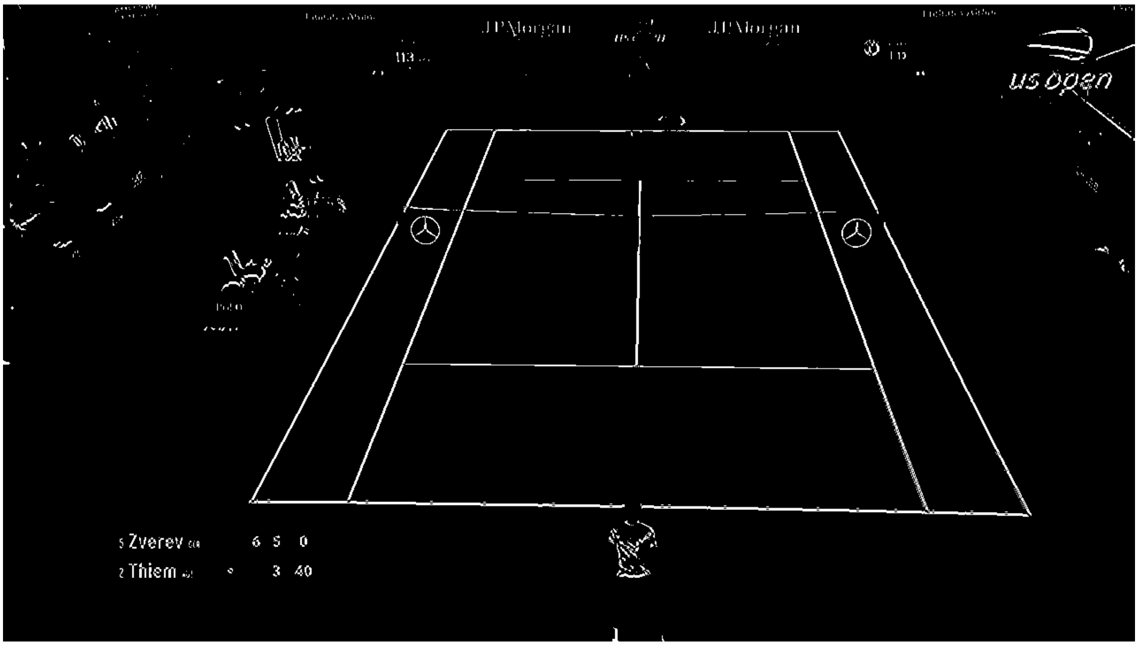
ניסוי 1 – **Hough Transform**

הניסיון הראשון היה שימוש בהתמרת Hough, למציאת הקווים בתמונה הבינארית שקיבלתי.

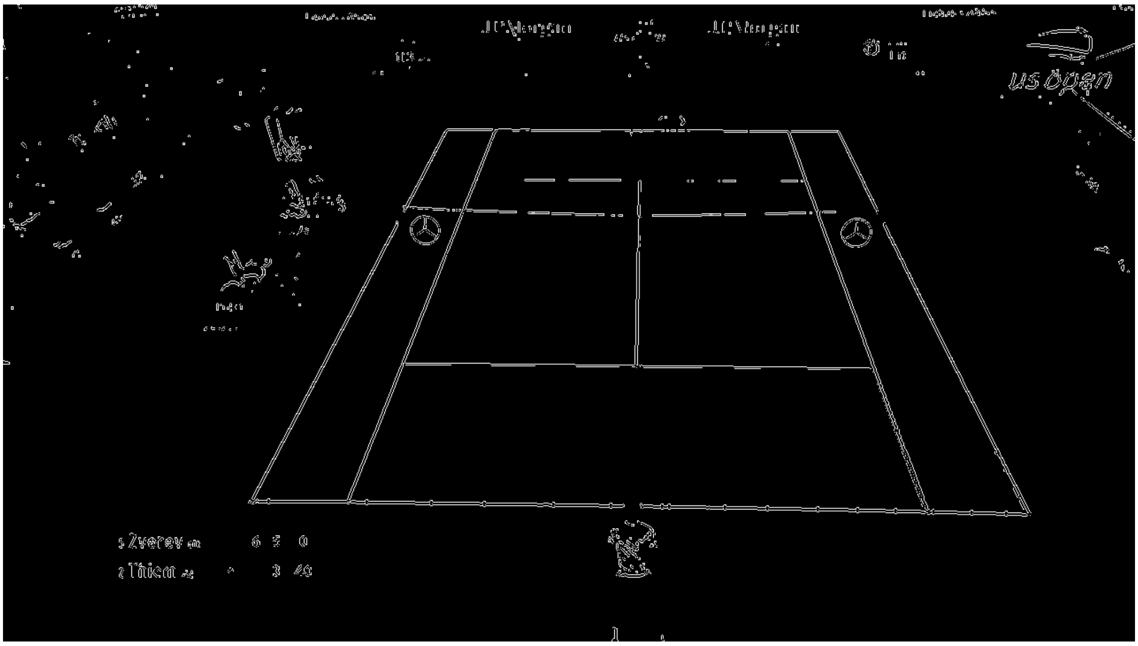
לחישוב ההתמרה השתמשתי בשיטה המתוארת בסקירת הספרות.

<u>תוצאות הניסוי עם שימוש ב Canny edge detector כצעד מקדים:</u>

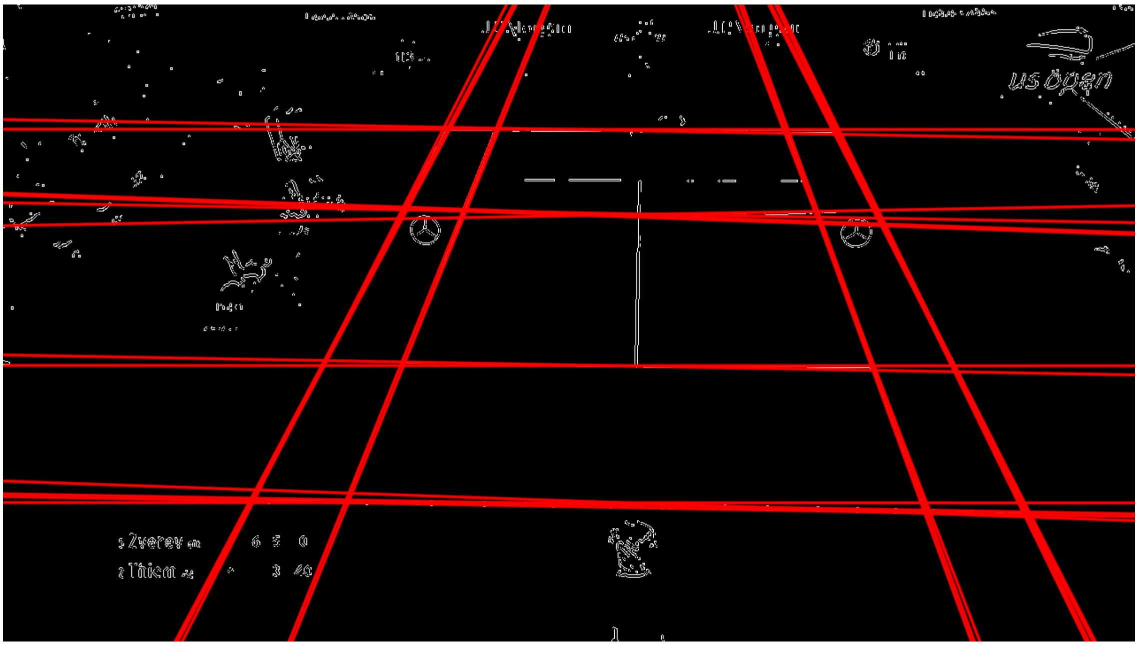
תמונה בינארית לאחר ביצוע פעולת סף ופילטר:



תמונה לאחר שימוש בCanny edge detector :



תוצאה סופית של Hough transform :



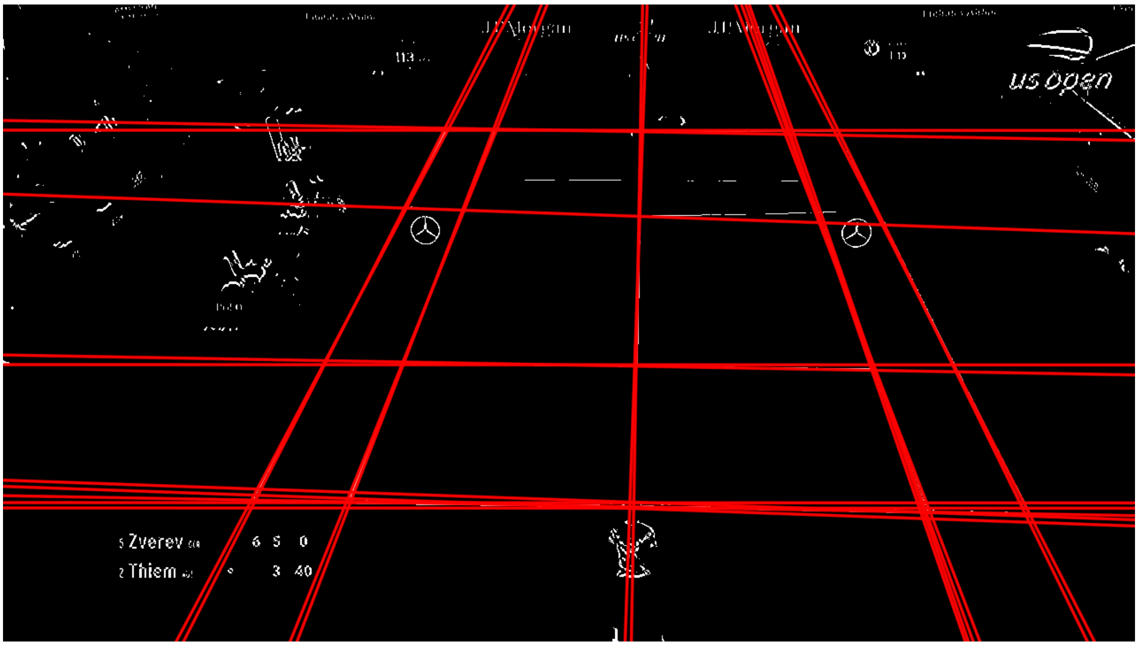
ניתן לראות כי הפעולה אכן הצליחה לזהות את רוב הקווים של המגרש, וכי כמעט ואין זיהויים שגויים, אך ניתן גם לראות כי הפעולה אינה נותנת לנו קו יחיד לכל קו במגרש אלא סט של המון קווים לכל קו במגרש.

בנוסף, לא ניתן לזהות בעזרת התמרה זו היכן מתחיל והיכן מסתיים כל קו.

כעת, ניסיתי לראות האם יש צורך בCanny , או האם אפשר לוותר עליו בגלל השימוש בפעולת הסף המקדימה.

<u>תוצאות הניסוי ללא שימוש ב Canny edge detector</u>:

תמונה סופית של Hough Transform:



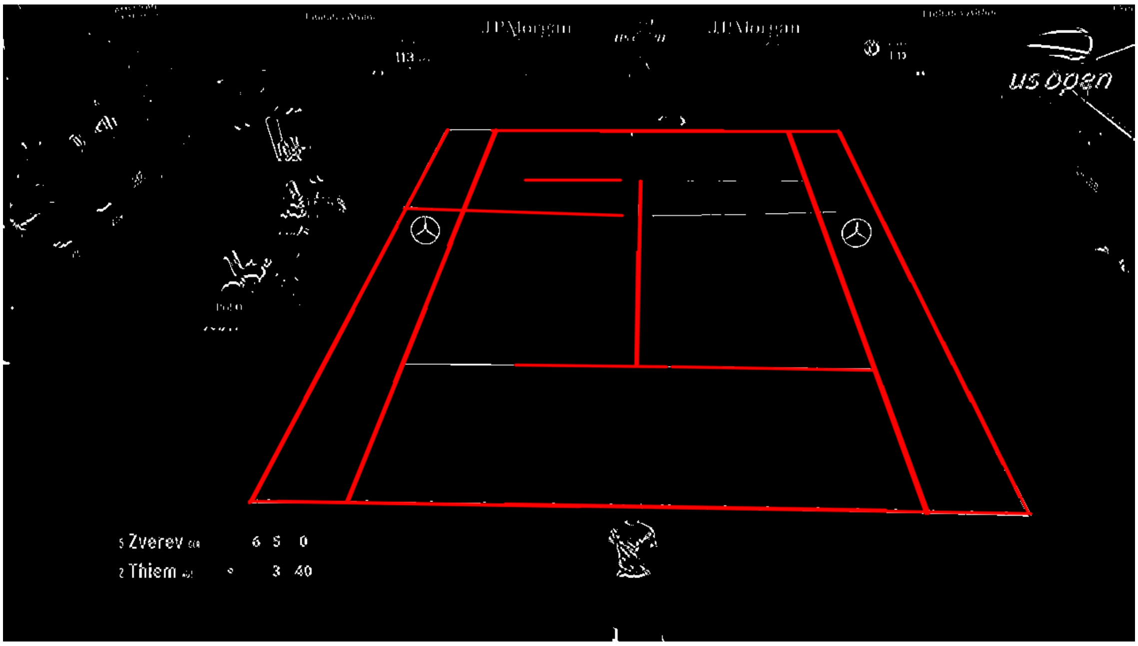
ניתן לראות כי הפעלת Canny לא הייתה נחוצה, אך עדיין נותרנו עם המון זיהויים לקו בודד, ועדיין אין לנו מידע על מיקום הקו המקורי בתמונה.

ניסוי 2 - Probabilistic Hough Line Transform

כעת ניסיתי שימוש בהתמרת Hough ההסתברותית למציאת קווים, בתקווה כי התמרה זו תוכל לתת לי הערכה יותר מקומית של הקווים, ולשפר לי את התוצאות ומהירות החישוב.

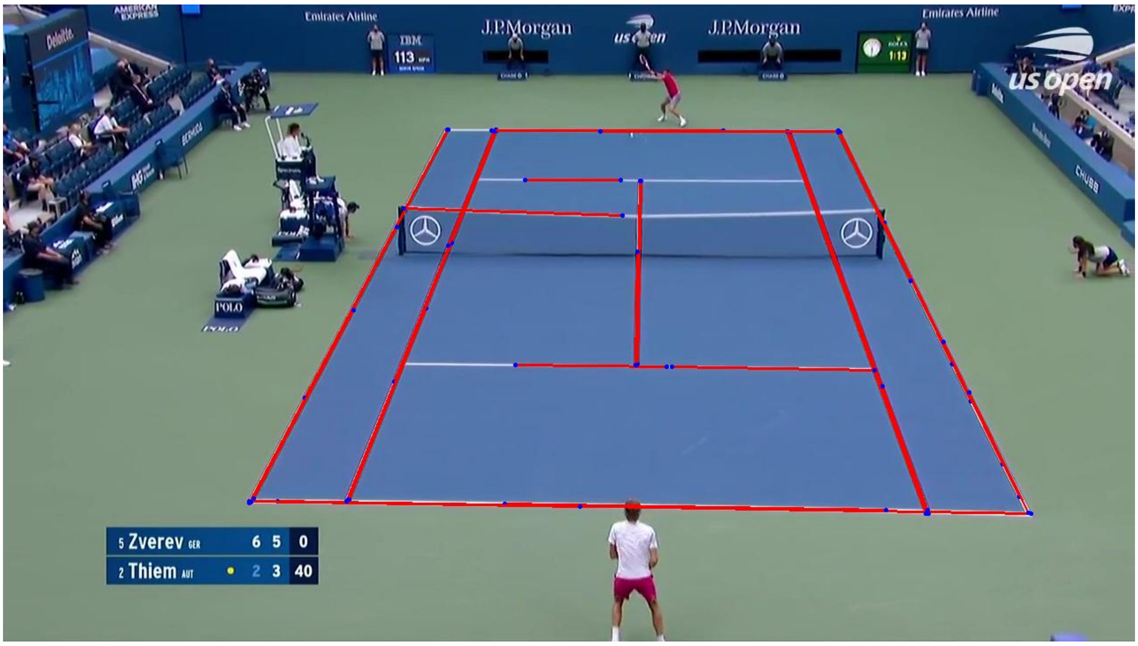
הפעלת ההתמרה נעשתה כמו שתואר בסקירת הספרות.

<u>תוצאות הניסוי</u>:



ניתן לראות כי כעת, זיהוי המגרש יותר מדויק מבחינת מיקום בתמונה, בנוסף ניתן לזהות היכן קווים מתחילים ומסתיימים.

חשוב לציין, כי פעולה זו עדיין מוצאת המון קווים השייכים לאותו קו מקורי בתמונה, ניתן לראות זאת בתמונה הבאה המציגה את כל הקווים שנמצאו עם נק' בהתחלה ובסוף כל קו.



בגלל פיצול הקו המקורי לכמה זיהויים שונים, יהיה צורך בהפעלת פעולת איחוד לקווים האלה, על מנת לקבל קו יחיד לכל זיהוי.

בסוף ביצוע התמרה זו, קיבלנו מועמדים מוצלחים מאד לקווים של המגרש.

סיווג הקווים

לאחר מציאת מועמדים לקווים של המגרש, יש צורך לסווג אותם לשתי קטגוריות (כרגע שתיים, מאוחר יותר ייתכן שאסווג ליותר), קווים אנכיים וקווים אופקיים.

על מנת לבצע את הסיווג, השתמשתי בהנחה כי עבור קו אנכי, השינוי בציר y צריך להיות גדול יותר מהשינוי בציר x ועבור קו אופקי ההפך נכון.

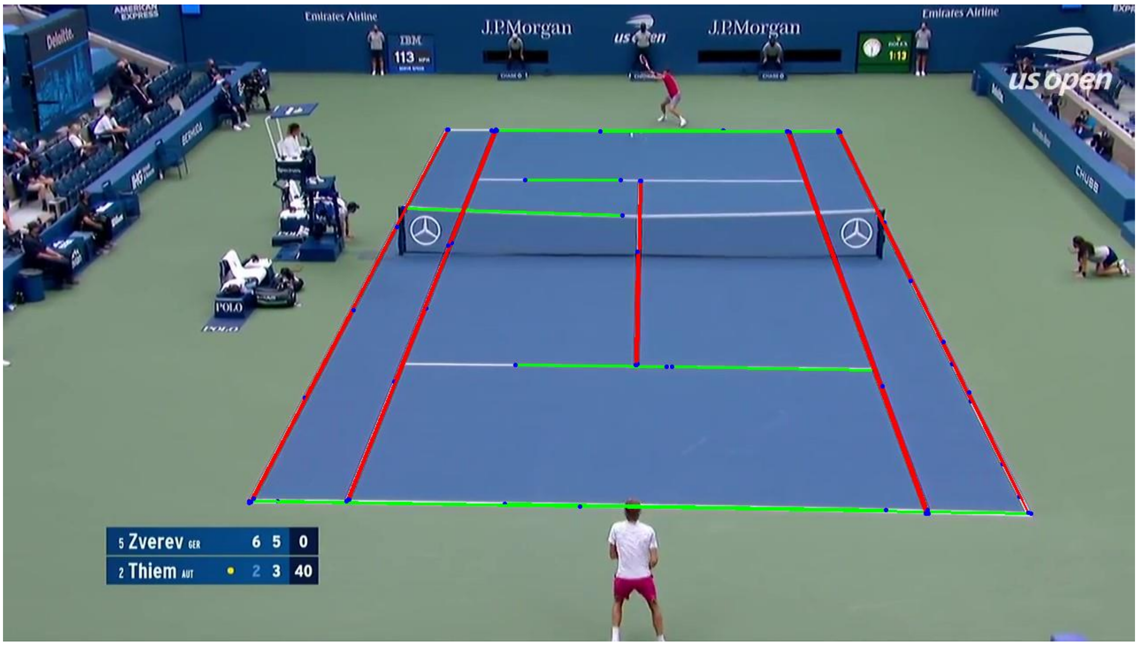
כעת על ידי שימוש בהנחה הזאת, ביצעתי סיווג לקווים שנמצאו בניסוי הקודם.

ניסוי 1 – שימוש ב $dx < dy$ לסיווג

<u>תוצאות הניסוי:</u>

קווים אופקיים מסומנים בצבע ירוק וקווים אנכים בצבע אדום.

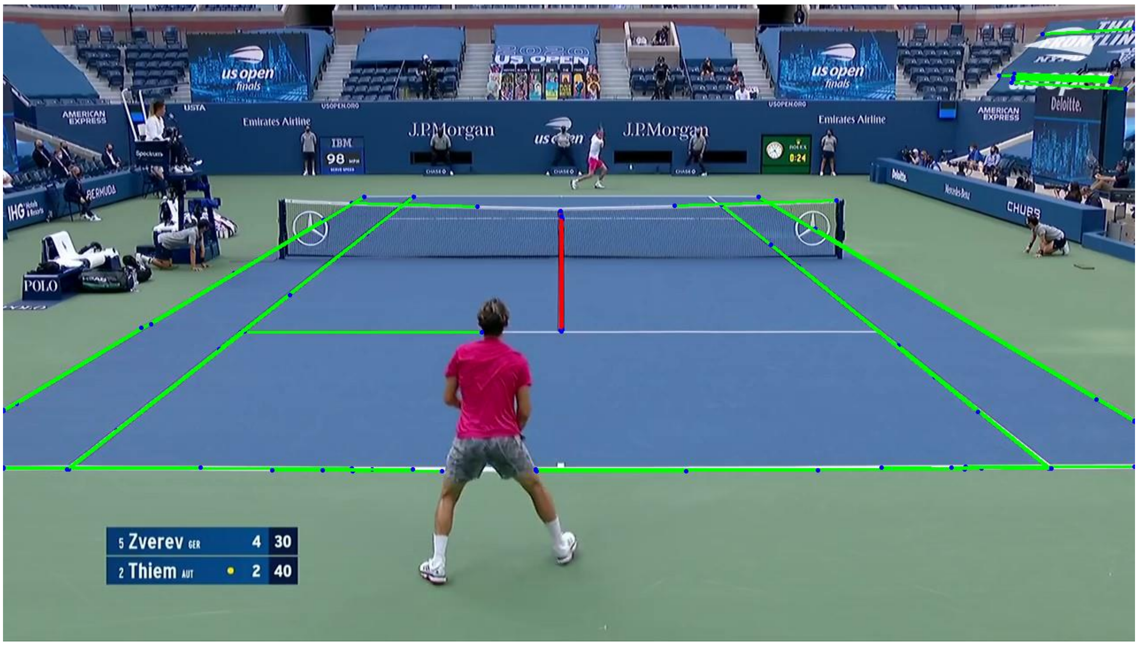
עבור התמונה המקורית:



ניתן לראות בוצע סיווג מוצלח מאד לקווים, וכי אין אף טעות.

כעת, ניסיתי להריץ את אותה פעולה על תמונה נוספת, מזווית טיפה אחרת.

תוצאה עבור התמונה החדשה:



ניתן לראות כי בזווית הזאת, קווים שהם אנכיים מסווגים כאופקיים, בגלל שהשינוי שלהם בציר x גדול מהשינוי בציר y בגלל זווית הצילום.

כדי להתגבר על כך, תחילה ניסיתי לשחק עם היחס בין dx לdy כך שבתמונות מהזוויות שהצגתי, הסיווג יהיה נכון.

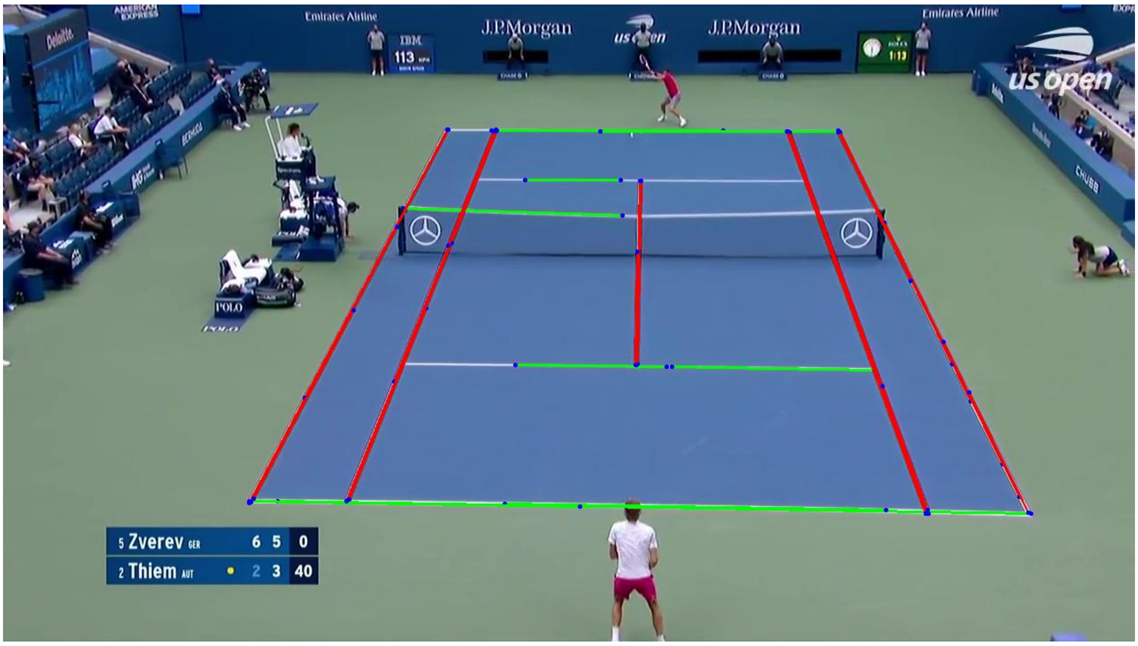
ניסוי 2 – שימוש ב $dx < 2dy$ לסיווג

כעת בגלל הנחת העבודה כי זווית הצילום איתה אעבוד בפרויקט תהיה מאחורי השחקן, הגעתי למסקנה כי אם אגדיר שהשינוי בציר x צריך להיות פי 2 גדול יותר מהשינוי בציר y על מנת לסווג קו כאופקי, אקבל סיווג נכון לקווים האנכיים. (ההנחה היא שזווית הצילום משפיעה בעיקר על הקווים האנכיים, וכי הקווים האופקיים יהיו מספיק אופקיים גם עבור התנאי החדש)

אם ארצה להוסיף אופציה לזוויות צילום נוספות, יהיה צורך לחשוב מחדש על דרך סיווג הקווים.

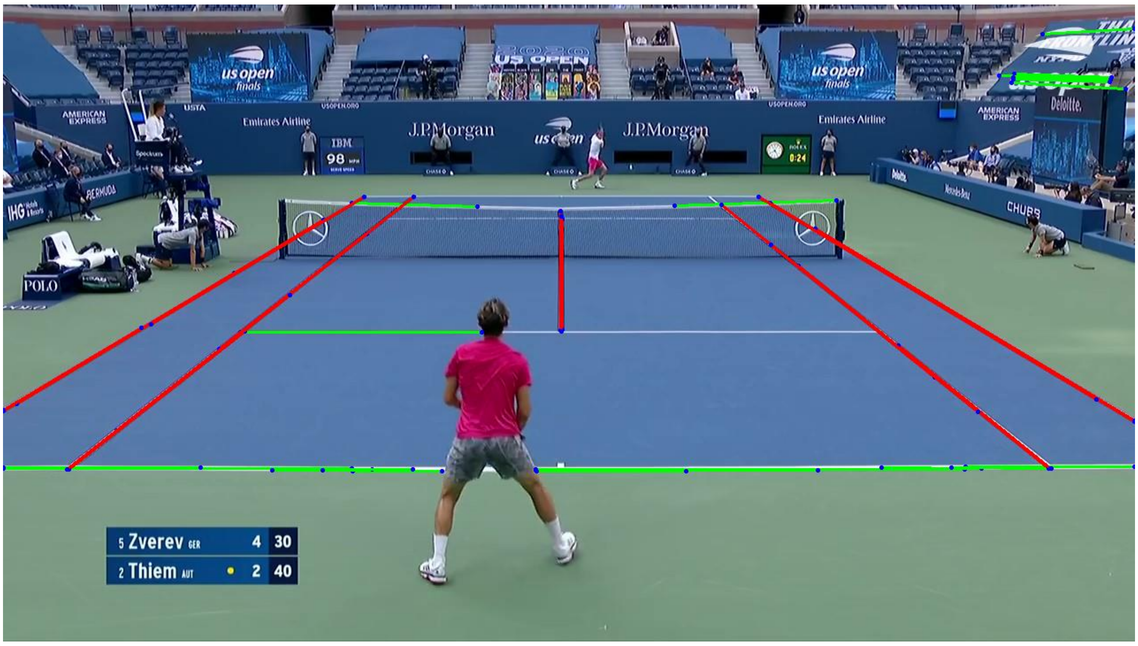
<u>תוצאות הניסוי עבור שתי זוויות הצילום:</u>

עבור התמונה המקורית:



ניתן לזהות, כי הסיווג בזווית המקורית לא נפגע.

עבור התמונה השנייה:



כעת ניתן לראות כי הצלחנו לסווג נכון את כל הקווים של המגרש.

בהרצת הניסוי על התמונה השנייה, שמתי לב כי יש זיהוי שגוי של קווים מחוץ לתחומי המגרש, אשר לרוב מתרחש באזורים שהם מעל הקו העליון של המגרש או מתחת לקו התחתון של המגרש, וכי הסיווג של אותם קווים הוא רק לקווים אנכיים.

לכן, החלטתי לעבוד עם פעולת סיווג נוספת עבור קווים אופקיים כך שאוכל לסנן קווים שנמצאים מחוץ למגרש.

ניסוי 3 – סינון קווים אופקיים

על מנת לסנן קווים אופקיים מחוץ למגרש, תחילה זיהיתי את הקווים האנכיים הגבוהים ביותר והנמוכים ביותר שזוהו בתמונה.

עבור קווים אלה, בחרתי את הנקודה הגבוה ביותר בתמונה והנמוכה ביותר בתמונה, והגדרתי מרחק זה כh.

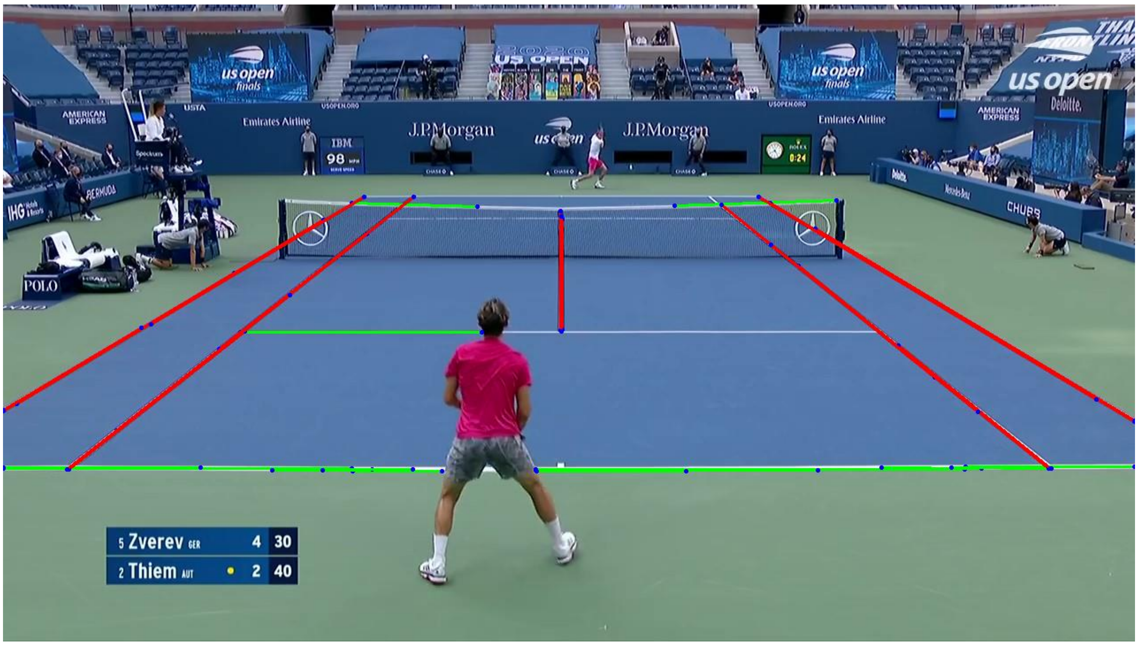
כעת הנחתי כי כל הקווים האופקיים שאזהה בתמונה צריכים להיות בין שתי הנקודות הנ"ל (עם תוספת שגיאה לכל כיוון).

כלומר, הגדרתי כי כל קו אופקי היושב מחוץ לטווח $[\min y - \frac{2}{15}h, \max y + \frac{1}{15}h]$ ימחק מהקווים האופקיים.

(בגלל שy נמוך יותר שייך לאזורים בחלק העליון של התמונה, אשר שייכים לאזורים רחוקים יותר מהמצלמה, הזיהוי שם יותר קשה, לכן נתתי תוספת שגיאה יותר גדולה)

<u>תוצאות הניסוי :</u>

עבור התמונה השנייה:



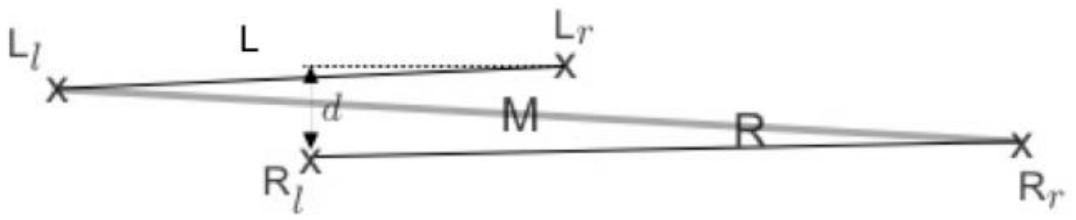
ניתן לזהות, כי הקווים שזוהו בחלק הימני העליון של התמונה, כעת סוננו.

איחוד קווים

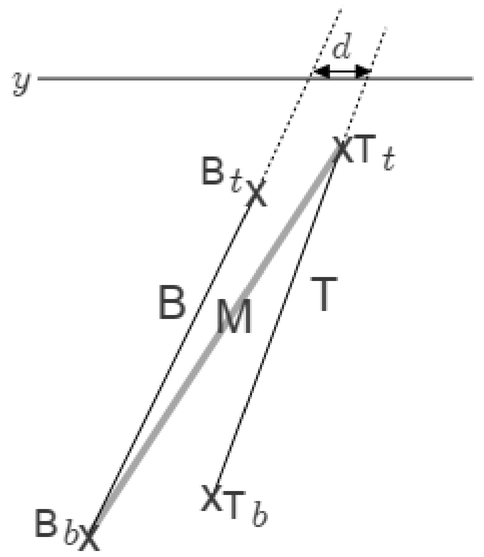
כמו שראינו בחלק הקודם, זיהוי הקווים מחזיר לנו קבוצה של קווים אשר חלקם שייכים לאותו קו מקורי.
על מנת לקבל קו בודד לכל קו במגרש, יש צורך לאחד את כל הקווים המתאימים לכדי קו בודד.

ניסוי 1 – איחוד שני קווים לפי נק' מכל קו

תחילה, ניסיתי לבצע איחוד קווים בצורה הבאה:
זיהוי כי שני קווים שייכים לאותו קו במגרש על ידי חישוב מרחק בציר y עבור קווים אופקיים והשוואה לסף כלשהו, ועבור קווים אנכיים, חישוב מרחק בציר x עבור נק' חיתוך של הקווים בקו אופקי כלשהו והשוואת המרחק לסף כלשהו.
לאחר מכן, איחוד הקווים על ידי לקיחת נק' הקצה השמאלית מהקו הראשון ונק' הקצה הימנית מהקו השני (עבור קווים אופקיים).
לדוגמא עבור קווים אופקיים:

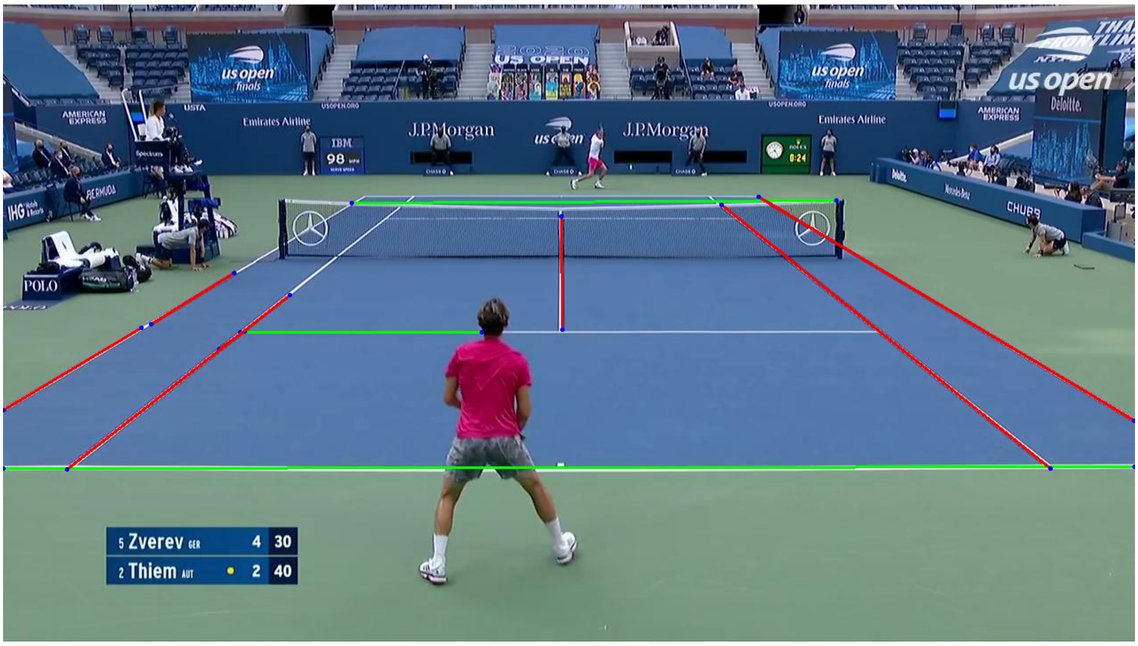


כאשר נאחד את הקווים L ו R לקו הסופי M.
ועבור קווים אנכיים:



כאשר נאחד את B ו T לקו M.

<u>תוצאות הניסוי:</u>



ניתן לראות כי אכן קיבלנו פחות קווים השייכים לאותו קו מקורי, אך בנוסף שמתי לב כי חלקים שכוסו בקווים המקוריים שמצאתי, כעת אינם מכוסים, כמו הקווים השמאליים בחלקם העליון.

לאחר בדיקה, מצאתי כי ישנם מקרים בהם קו אחד מוכל לגמרי בקו השני, כלומר הקצוות של הקו הראשון נמצאות בתוך הקו השני.

לכן ביצעתי ניסוי נוסף על מנת לפתור את הבעיה שנוצרה.

ניסוי 2 – איחוד שני קווים על ידי נק' מקסימום ומינימום ביניהם

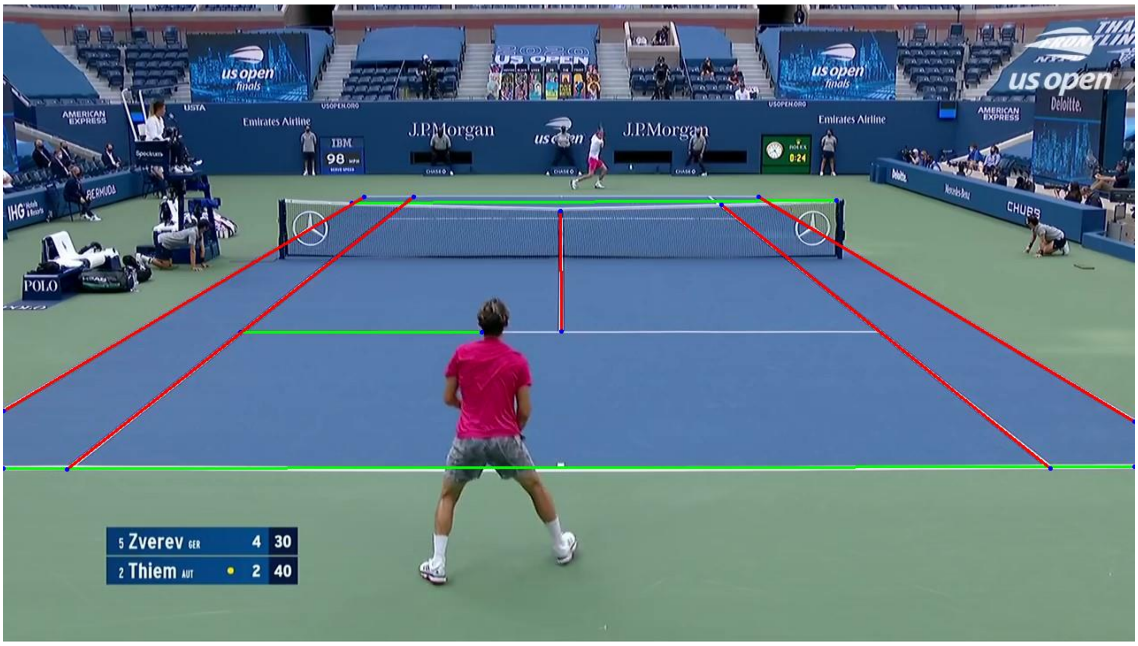
כעת על מנת לפתור את הבעיה שנוצרה בניסוי הקודם, במקום לאחד שתי נק' על ידי נק' קצה מקו אחד ונק' קצה מקו שני.

החלטתי למצוא את נק' המקסימום והמינימום בין 4 נק' הקצה של הקווים, ובעזרתן להגדיר את הקו המאוחד.

כך בעצם מקרים בהם קו אחד מוכל בשני לא ישפיעו על האיחוד.

חשוב לציין כי נק' מקסימום ומינימום עבור קו אנכי שונות מהנק' עבור קו אופקי, כאשר עבור קו אנכי נק' מינימום הינה הנק' הנמוכה ביותר ומקסימום היא הנק' הגבוהה ביותר. עבור קו אופקי נק' מינימום היא הנקודה השמאלית ביותר ונק' מקסימום היא הנקודה הימנית ביותר.

<u>תוצאות הניסוי:</u>



ניתן לראות כי כעת הקווים שנמצאו התאחדו בצורה טובה וכי לא איבדנו קווים בפעולת האיחוד, כמו שקרה בניסוי הקודם.

מציאת טרנספורמציה

בחלק זה, השתמשתי בקווים שנמצאו בניסויים הקודמים, על מנת למצוא טרנספורמציה בין תמונת רפרנס בינארית של מגרש אמיתי (שנבנתה לפי מידות המגרש), לבין המגרש בתמונה של המשחק.

כך יהיה אפשר לזהות בדיוק היכן כל הקווים של המגרש ולבצע Overlay של מיקום המגרש לתמונת המשחק.

על מנת לבצע זאת, תחילה מצאתי נק' חיתוך בין כל הישרים האופקיים שמצאתי לבין הישרים האנכיים, ואז מצאתי התאמה בין הנק' האלה לנק' הנמצאות בתמונת הרפרנס של המגרש.

והשתמשתי בשיטה שמתוארת בסקירת הספרות למציאת הטרנספורמציה המתאימה לנק' הנ"ל.

כדי למצוא את ההתאמה הטובה ביותר, השתמשתי בקונפיגורציות שונות של נק' על מגרש הרפרנס, עד שמצאתי את הקונפיגורציה המתאימה ביותר לתמונת המשחק.

הקונפיגורציות והשיטה המלאה מתוארות בחלקים הקודמים של הדו"ח.

ניסוי 1 – מציאת קונפיגורציה מתאימה

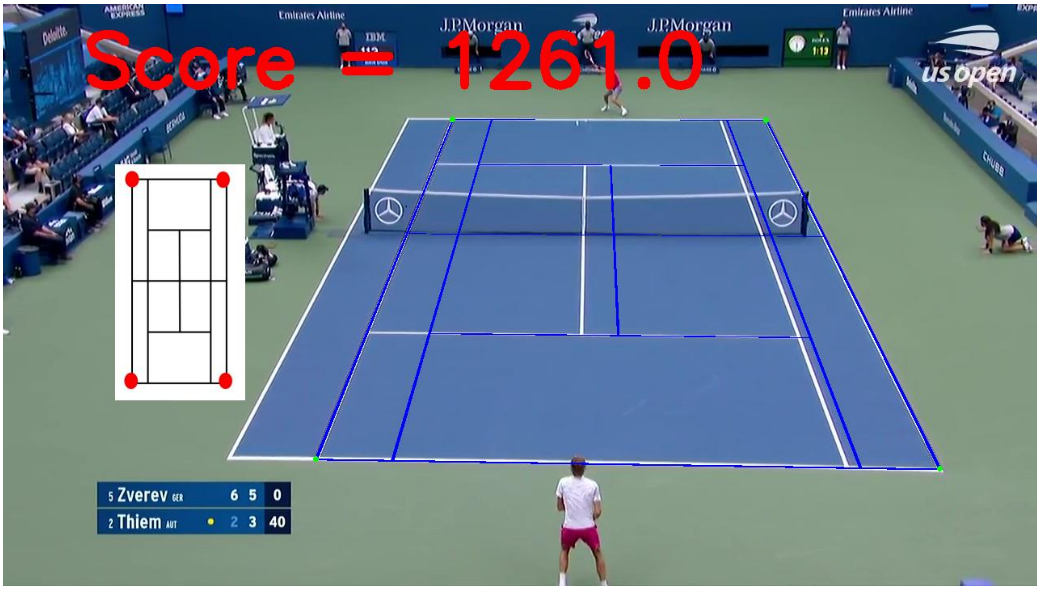
בניסוי זה, עבור בחירה של שני קווים אופקיים ושני קווים אנכיים, ניסיתי למצוא את הקונפיגורציה המתאימה ביותר של נק' החיתוך בין הקווים.

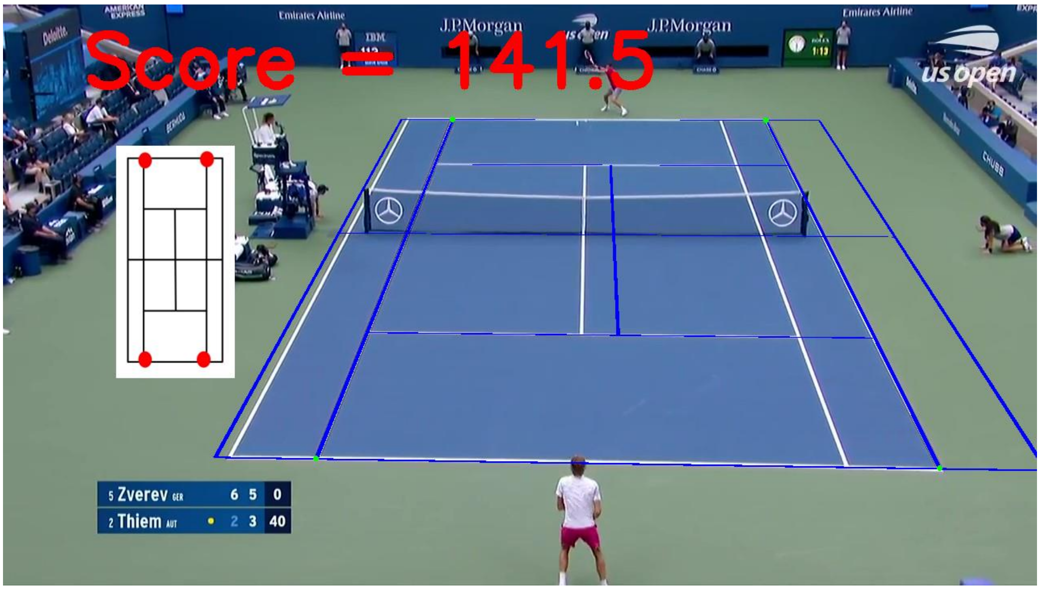
על מנת לעשות זאת, חישבתי ניקוד לכל קונפיגורציה כמו שמתואר בסקירת הספרות, ובחרתי את הקונפיגורציה עם הניקוד המקסימלי.

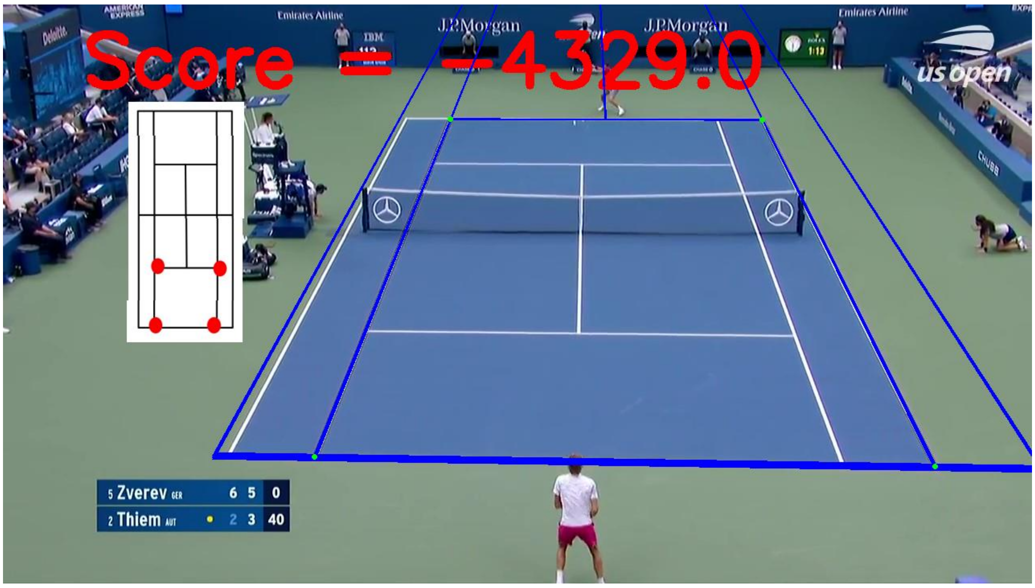
<u>תוצאות הניסוי עבור קונפיגורציות שונות:</u>

ניתן לראות את נק' החיתוך שנבחרו בירוק, ואת ההטלה של תמונת הרפרנס של המגרש על תמונת המשחק לפי הטרנספורמציה שחושבה. בנוסף, הציון שניתן לקונפיגורציה רשום בתמונה והקונפיגורציה שנוסתה מופיעה גם כן.

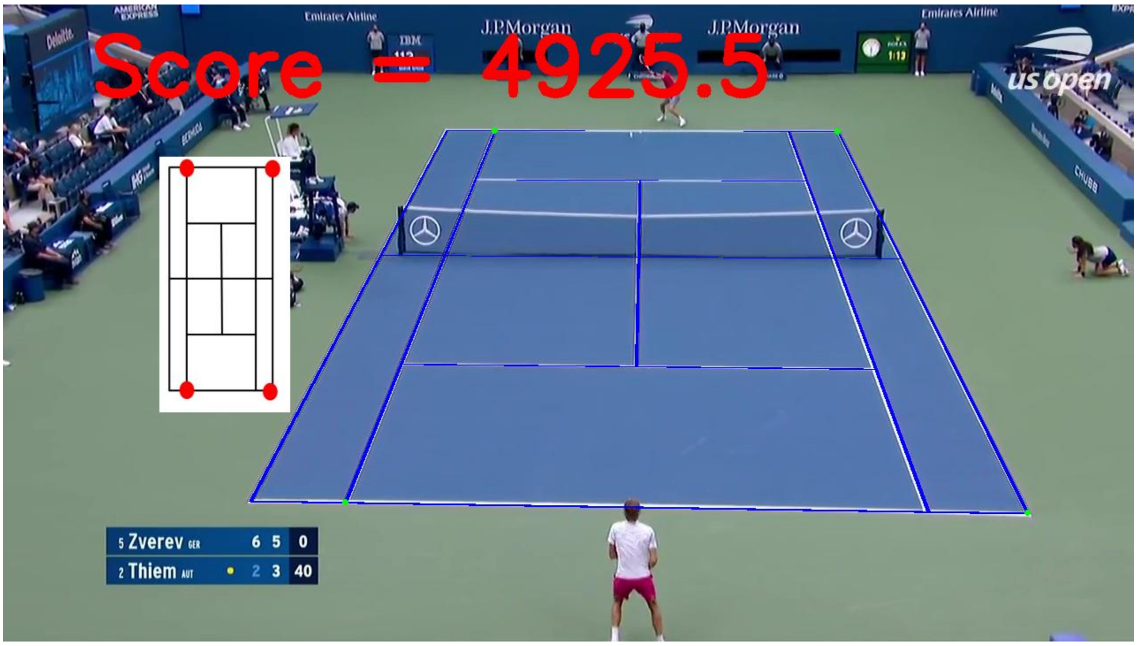
עבור קונפיגורציות לא מוצלחות:

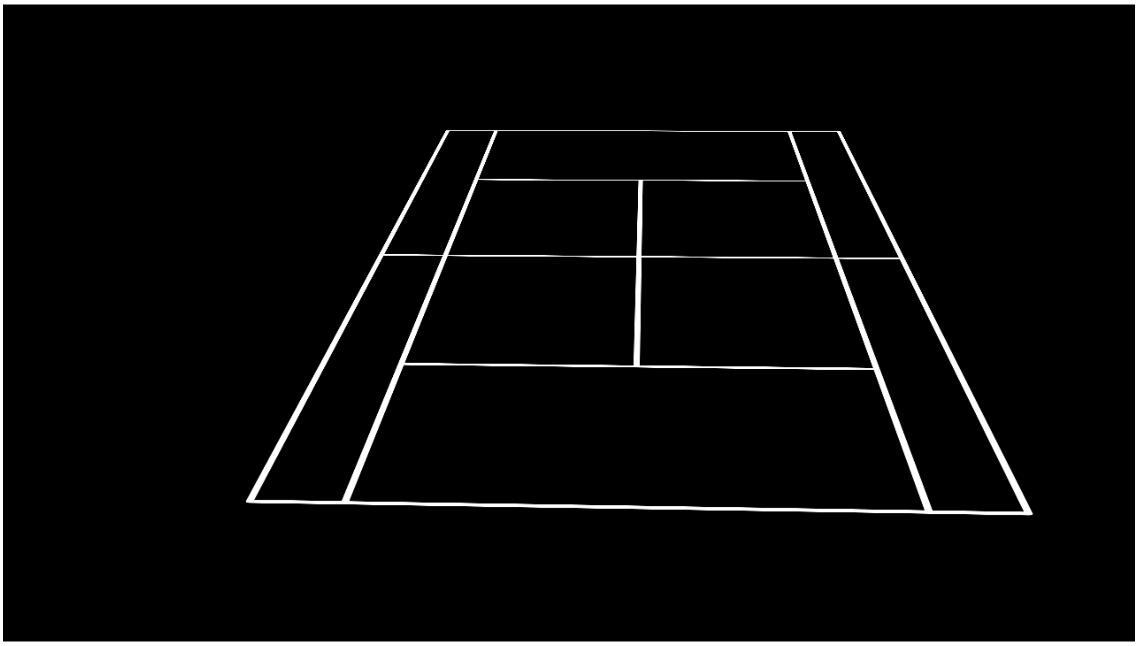






עבור קונפיגורציה מוצלחת:





ניתן לראות כי הניקוד שקיבלו קונפיגורציות לא מוצלחות, הרבה יותר נמוך מהניקוד שקיבלה קונפיגורציה מוצלחת.

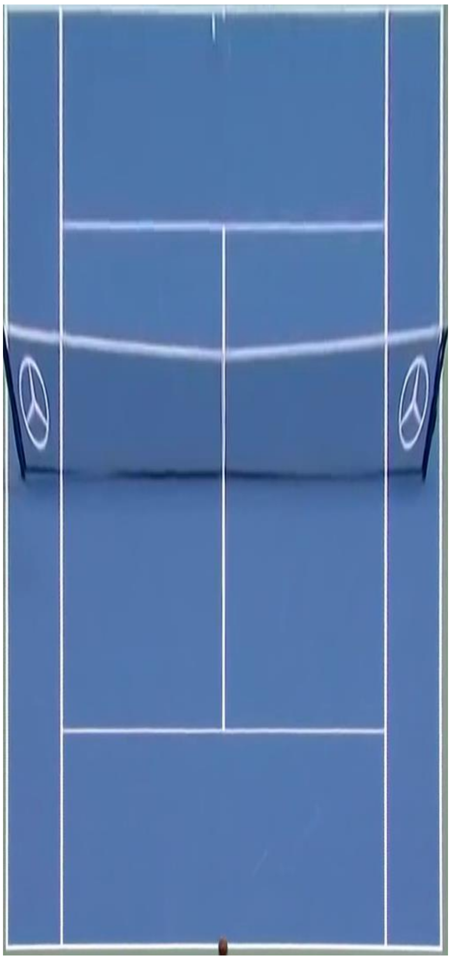
בנוסף, ניתן לראות כי ההטלה שקיבלנו של המגרש לתמונת המשחק, היא בדיוק גבוה מאד.

ניסוי 2 – מציאת טרנס' הפוכה והפעלתה

כעת לאחר שמצאנו את הטרנספורמציה שמטילה את תמונת הרפרנס של המגרש לתמונת המשחק, נרצה לבצע היפוך למטריצה שהתקבלה כדי לקבל את ההטלה ההפוכה.

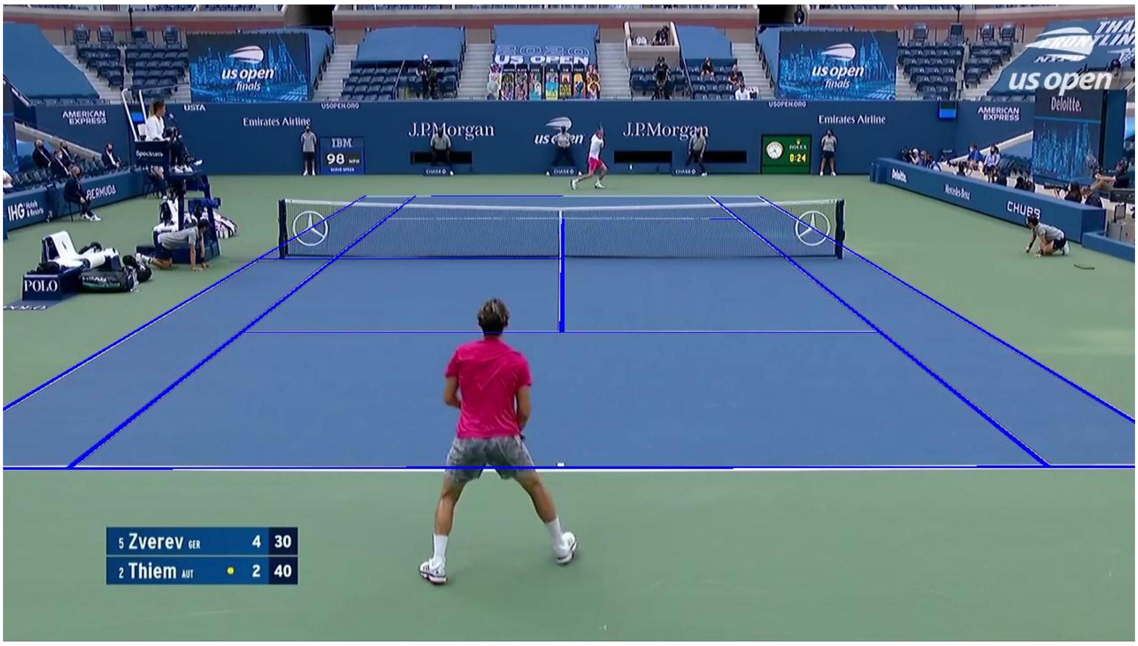
<u>תוצאות הניסוי עבור זוויות צילום שונות</u>:

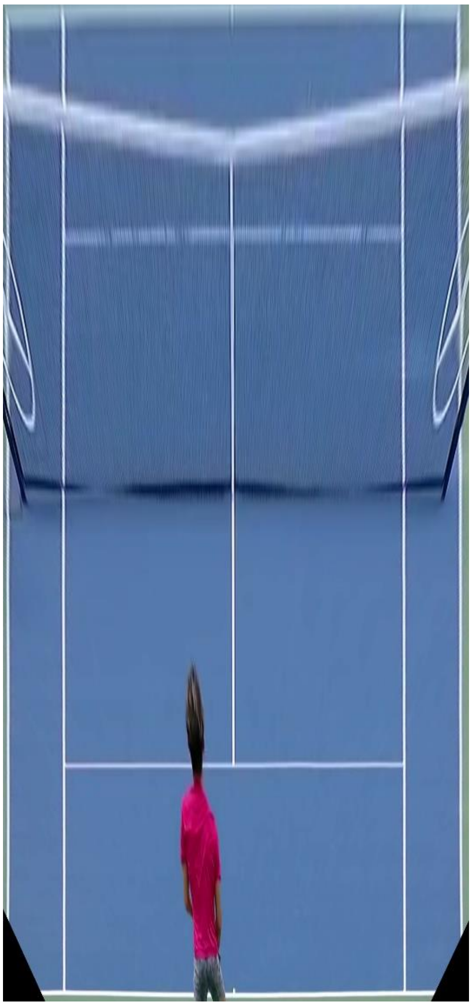
עבור תמונה מזוויות גבוהה:



ניתן לראות תוצאה טובה מאד בזיהוי המגרש והטלת המשחק למבט על.

עבור תמונה מזווית נמוכה:





גם כעת ניתן לראות תוצאות טובות מאד, כאשר בהטלה ממבט על, התמונה מרוחה יותר, וזה הגיוני כי בסופו של דבר מוצג בדיוק מה שמופיע במשחק, ובגלל זווית הצילום יש צורך לעוות את התמונה יותר.

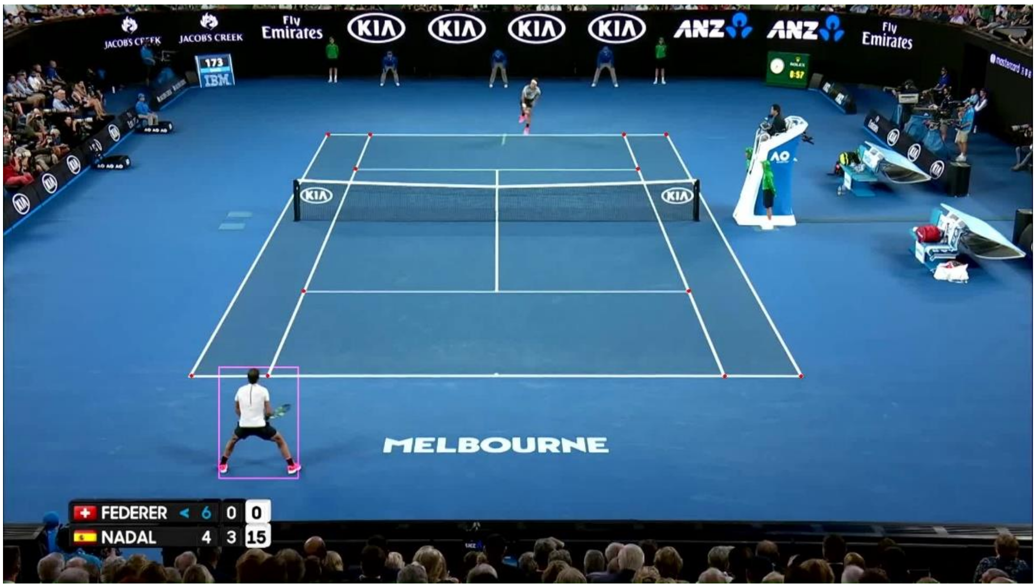
מעקב מגרש

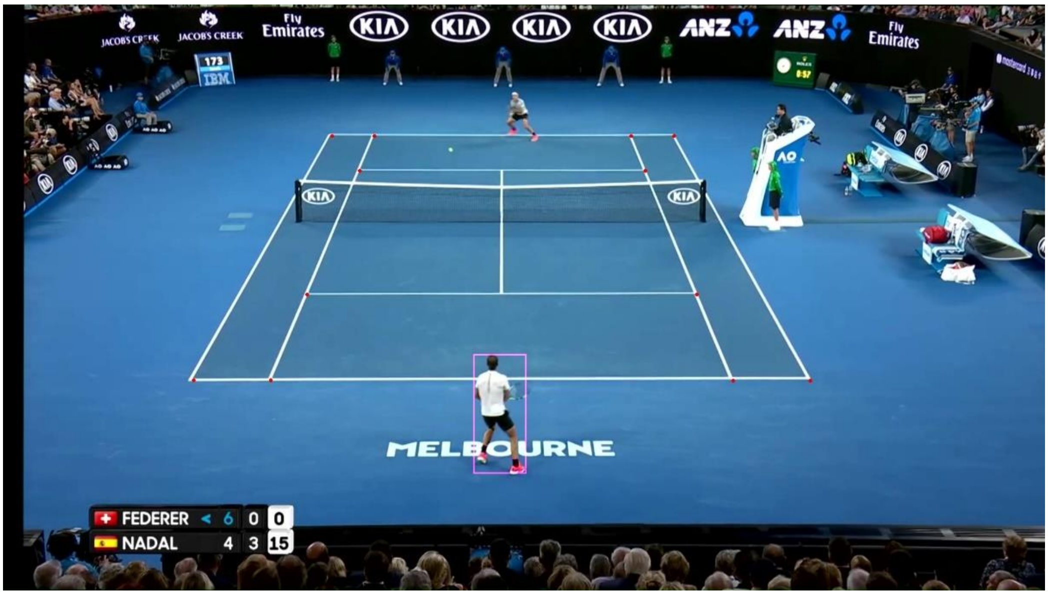
לאחר זיהוי המגרש בפריים הראשון, כעת יש צורך לבצע מעקב אחרי מיקום המגרש לאורך מהלך המשחק כולו.
ברוב קטעי המשחק, ישנים חלקים בהם המצלמה זזה, תזוזה זו גורמת לשינוי במיקום המגרש.
פתרון ראשון שחשבתי עליו הינו זיהוי המגרש מחדש עבור כל פריים, אך בגלל שמציאת המגרש בפעם הראשונה יכולה לקחת כמה שניות, ביצוע חיפוש זה לכל פריים הינו ארוך ואינו משתמש במידע שיש לנו ממיקום המגרש בפריים הקודם.
לכן, תחילה ניסיתי לבצע מעקב אחרי המגרש שמצאתי בפריים הראשון על ידי שימוש במיקום הקיים.

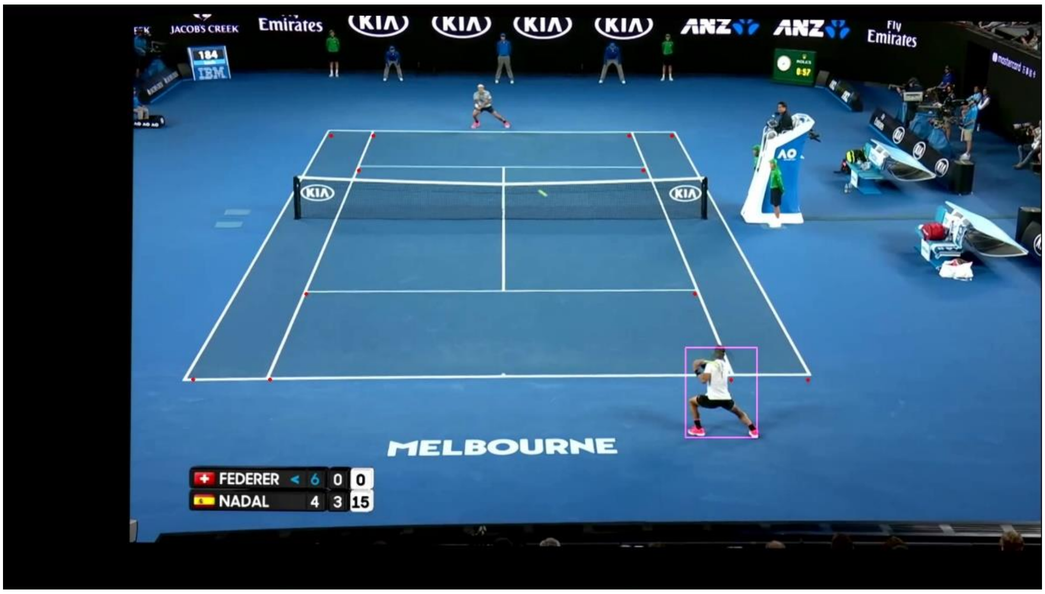
ניסוי 1 – מעקב מגרש על ידי מציאת סיבוב, הזזה וסקאלה בין פריימים

בניסוי הראשון ביצעתי הנחה, כי במקום להניח כי המצלמה זזה, אני מניח כי התמונה הוזזה, סובבה ובוצע שינוי סקאלה.
כעת במקום למצוא את נק' המגרש מחדש, אשתמש בנק' שמצאתי בזיהוי המגרש הראשוני, ואבצע חיפוש לוקאלי בסביבת כל נק', על מנת למצוא את הנק' בפריים החדש (בגלל שמדובר בפריימים עוקבים, הנחתי כי הנק' בפריים החדש תהיה בסביבה קרובה למיקום הנק' המקורי), את זיהוי הנק' בפריים החדש ביצעתי על ידי ביצוע Cross Correlation בעזרת סביבת הנק' בתמונה המקורית, בסביבה יותר גדולה בפריים החדש.
לאחר זיהוי הנק', חישבתי טרנספורמציה אפינית בין מיקום הנק', על מנת למצוא את המטריצה שתעביר אותי מהפריים החדש לפריים המקומי.
לאחר מציאת הטרנספורמציה, הפעלתי אותה על הפריים החדש על מנת לקבע את המגרש במיקום אחד לאורך הסרטון.

<u>תוצאות הניסוי:</u>







ניתן לזהות בתמונות את הנק' האדומות, אשר מסמלות את מיקום המגרש בפריים הראשון.

בנוסף, ניתן לראות כי אכן המגרש נשאר ממורכז לאורך הסרטון

https://www.youtube.com/watch?v=9iziytuwXnM

https://www.youtube.com/watch?v=gQGPANF_uuA

הבעיה העיקרית בשיטה זו, היא כי ההנחה שהתמונה זזה במקום המצלמה, אינה חזקה מספיק. השינוי במיקום המצלמה ובזווית שלה אינו שקול להזזה בתמונה, מפני שבזמן לקיחת התמונה, מתבצע מעבר מתלת מימד, לדו מימד, וייתכן כי יכנס מידע שלא היה קיים ובצורה זהה, ייתכן כי מידע שהיה קיים לא יופיע בפריים החדש.

לכן ניתן לראות כי למרות שהמגרש נשאר ממורכז בסרטון, מיקום הנק' שנמצאו בפריים הראשון, לא נשאר באותו מקום בדיוק ויש טיפה תזוזה.

בנוסף חיפוש הנק' בכל פריים, לפעמים לקח שנייה אחת לפריים, וזה הוריד את קצב החישוב בצורה משמעותית, לכן יש צורך למצוא שיטה אחרת המבצעת את החיפוש בצורה טובה יותר.

ניסוי 2 – מעקב אחר קווי המגרש שנמצאו בתהליך הזיהוי

שיטה שנייה שניסיתי, היא מעקב אחר הקווים המקוריים שמצאתי בתהליך זיהוי המגרש.

תחילה, שמרתי את הקונפיגורציה המתאימה של המגרש שבעזרתה מצאתי את מיקום המגרש בתהליך הזיהוי.

לאחר מכן, עבור כל קו בנפרד, בחרתי 100 נק' הנמצאות על הקו במרחקים שווים, ועבור כל נקודה, חיפשתי בפריים החדש בסביבת אותה נק', את הפיקסל בעל הבהירות הגבוהה ביותר אשר ערכו גדול מסף מסוים.

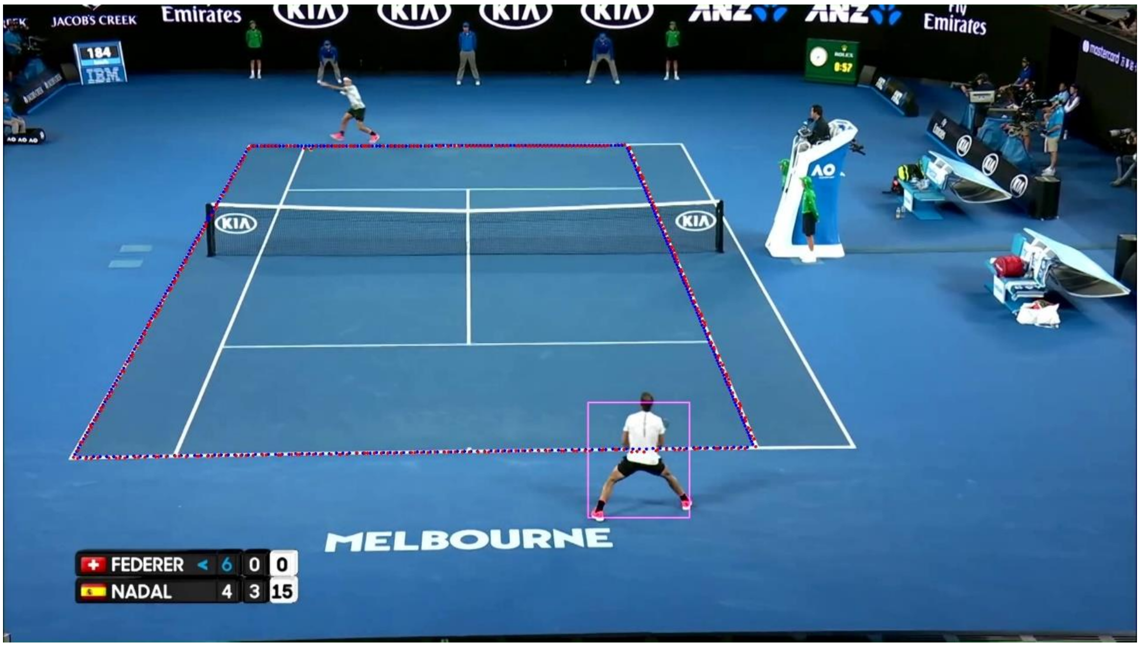
לאחר מעבר על כל הנק', אם נמצאו יותר מ-50 נק' חדשות, אני מוצא קו המתאים לנק' החדשות על ידי שימוש באלגוריתם Least Square שמוסבר בסקירת ספרות. אם לא נמצאו 50 נק', אבצע חיפוש מחדש למגרש.

לאחר ביצוע הפעולה לארבעת הקווים של המגרש, אקבל 4 קווים חדשים ואבצע פעולות דומות לפעולות שביצעתי בחיפוש המגרש.

פעולות אלה כוללות, מציאת חיתוך בין הישרים, ומציאת הטרנספורמציה ההומוגנית שמקשרת בין נק' החיתוך לנקודות של הקונפיגורציה של המגרש.

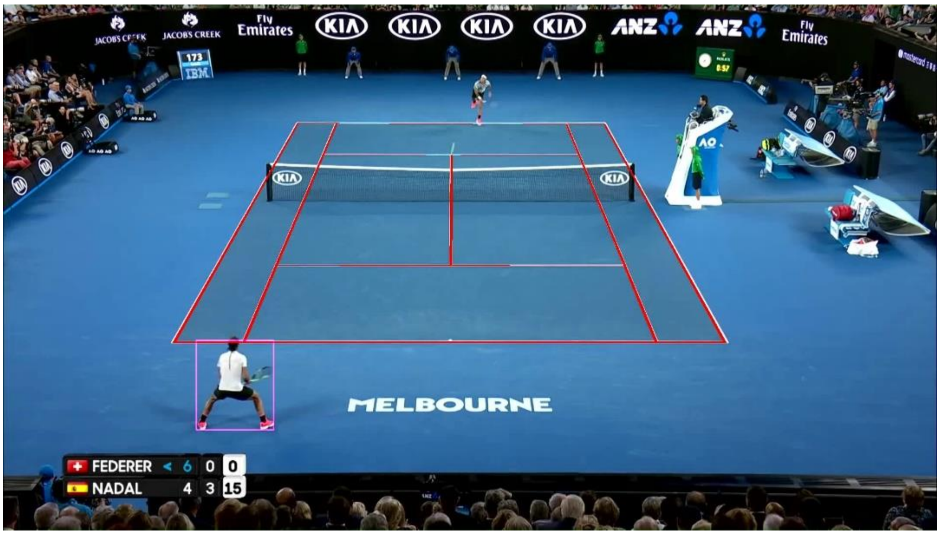
כעת יש בידי את הטרנספורמציה שמתאימה לי בין רפרנס המגרש, לבין המגרש בפריים.

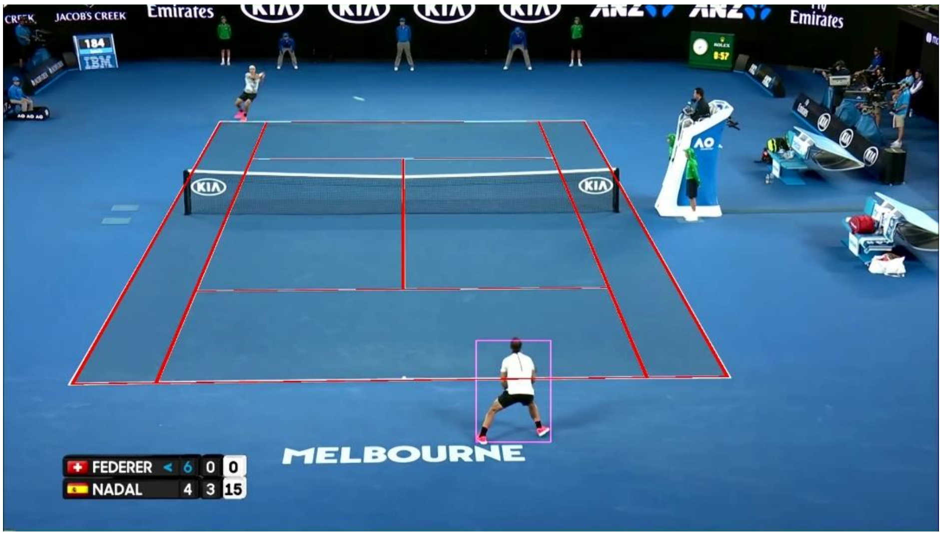
<u>תוצאות הניסוי:</u>



ניתן לראות בתמונה את הנק' על הקו בפריים הקודם אשר מסומנות בצבע כחול, ואת הנק' שנמצאו בפריים הנוכחי לפי האלגוריתם בצבע אדום.

ניתן לראות בעיקר בקו הימני, כי הנק' של הקו בפריים הקודם נמצאות טיפה שמאלה ממיקום הקו בפריים הנוכחי, וכי הנק' החדשות שמצאנו (בצבע אדום) ממוקמות על הקו הנכון בפריים.





ניתן לראות כי גם בשינוי זווית הצילום, מיקום המגרש עדיין מאד מדויק.

בסרטון ניתן לראות התאמה זו בצורה יפה מאד,

https://www.youtube.com/watch?v=N0C0caSjOnw

זיהוי השחקנים

בחלק זה אתמקד בזיהוי השחקנים במגרש.

הקשיים בזיהוי השחקנים הם: מספר אנשים גדול במגרש (שחקנים, קהל, שופטים), גודל השחקן המרוחק הינו קטן יותר ולכן קשה יותר לזיהוי, השחקן הקרוב נמצא עם הגב למצלמה.

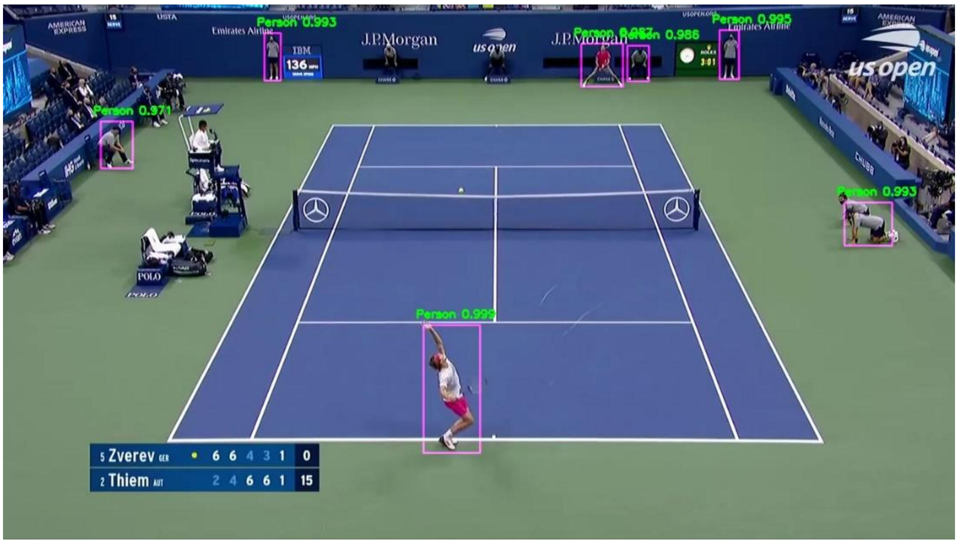
<u>שימוש ברשת עמוקה בצורה נאיבית</u>

הגישה הראשונה שניסיתי, היא שימוש ברשת עמוקה מאומנת מראש, אשר מסוגלת לזהות אובייקטים שונים.

ניסוי 1 – זיהוי כל האנשים במגרש

בניסוי זה השתמשתי ברשת Faster RCNN שאומנה על הדאטה סט COCO אשר מכיל סט של תמונות לביצוע detection בעזרת bounding box.

<u>תוצאות הניסוי:</u>



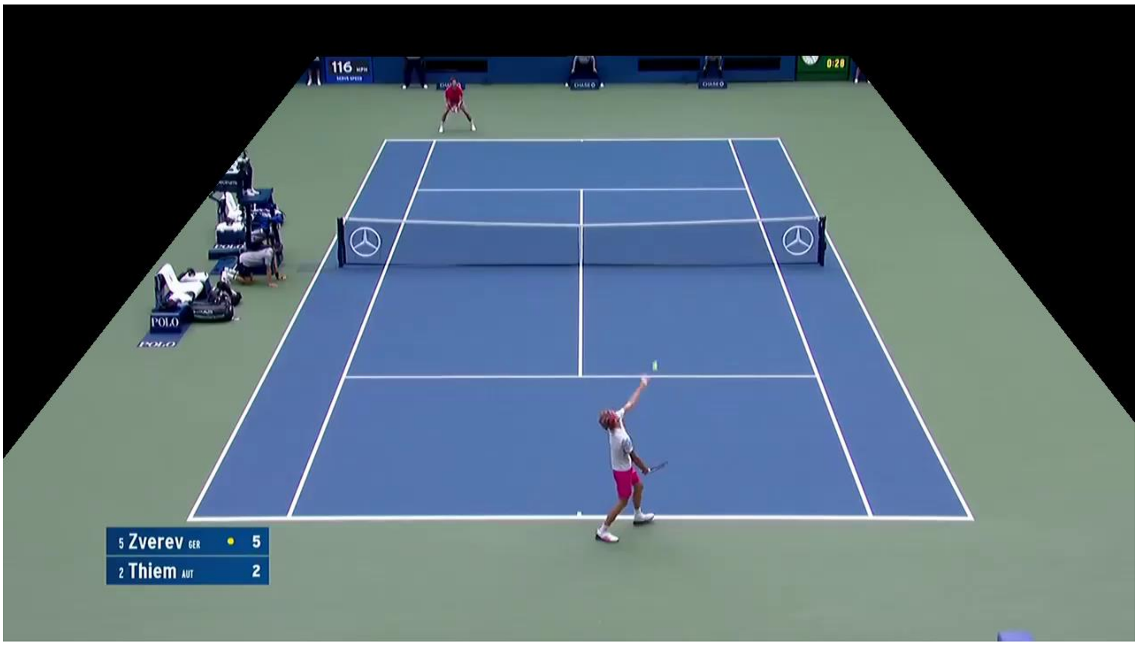
ניתן לזהות בתמונות, כי הזיהוי של האנשים די מוצלח, אך בגלל תזוזת השחקן העליון, לפעמים הוא אינו מזוהה.

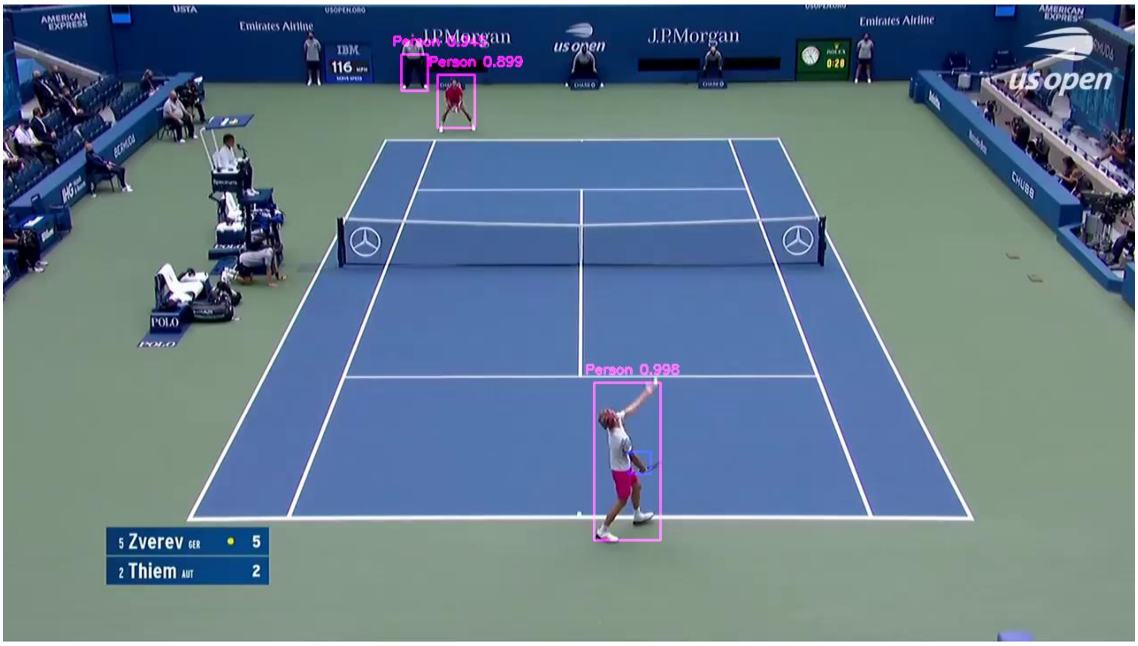
הבעיה שנוצרת היא כיצד להפריד בין השחקנים לבין שאר האנשים בתמונה, וכיצד להצליח לזהות את השחקן המרוחק ברוב הפריימים.

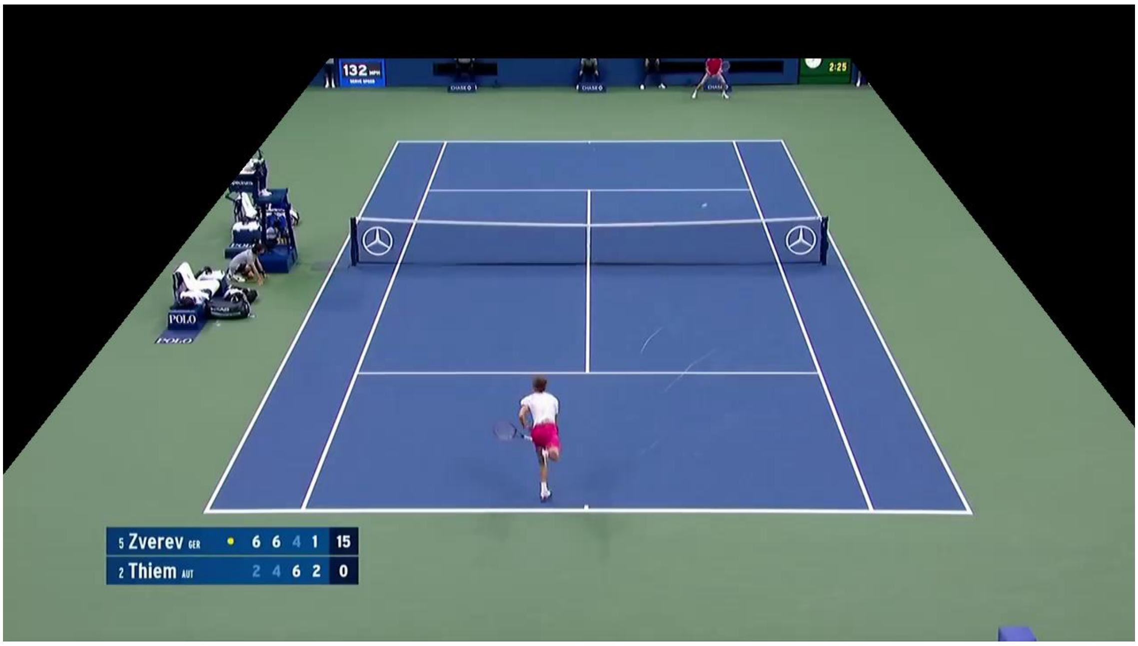
ניסוי 2 – שימוש במידע על המגרש לזיהוי השחקנים

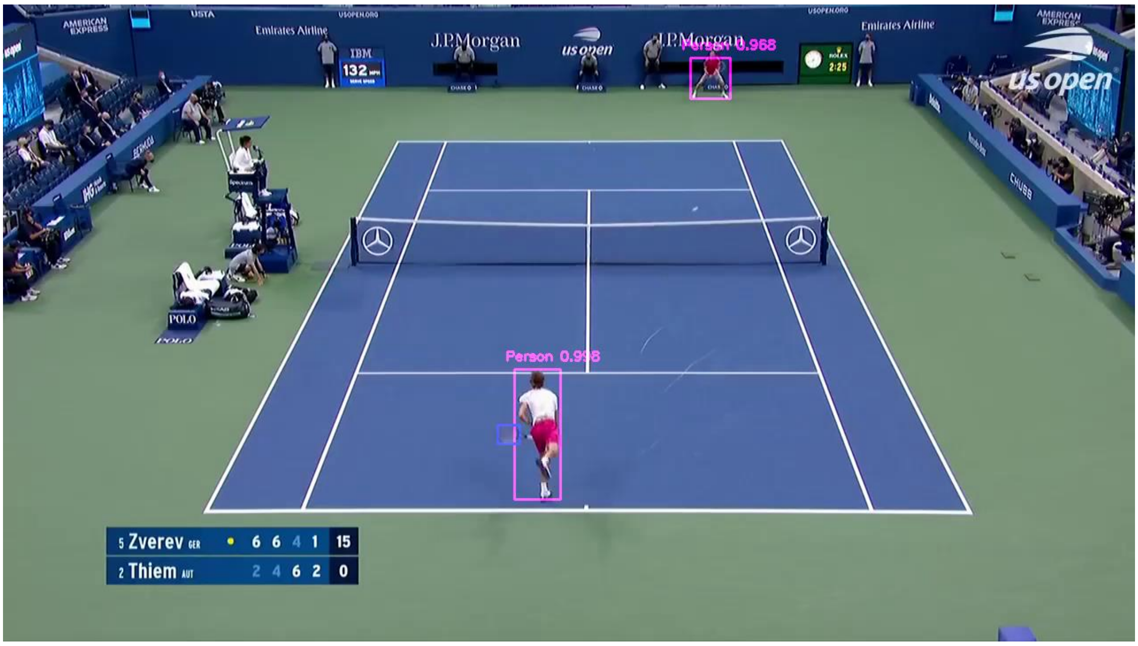
לאחר ששמתי לב, כי ישנם הרבה זיהויים של אנשים שאינם נמצאים על המגרש ואינם השחקנים.
השתמשתי במידע שמצאתי בנוגע למיקום המגרש, על מנת למחוק את האזורים בפריים שאינם קשורים למגרש,
ובכך להקטין את מספר הזיהויים הלא רלוונטיים.

<u>תוצאות הניסוי:</u>









ניתן לראות כי כעת הזיהוי הרבה יותר טוב, אך אחת הבעיות שנתקלתי בהן, הוא שבחלק מזיוויות הצילום, השחקן המרוחק נמצא מחוץ לאזור המגרש ואז רק חלק מגופו מופיע בתוצאה, וזה גורר לכך שבחלק מהמקרים הוא אינו מזוהה.

ניסוי 3 – זיהוי שחקן תחתון

לאחר שזיהיתי כי השחקן העליון בפריים הינו קשה יותר לזיהוי, החלטתי תחילה להתמקד בזיהוי השחקן התחתון ויותר מאוחר לחזור לזיהוי השחקן העליון.

על מנת לעשות זאת, ובהמשך לניסוי הקודם, תחילה ביצעתי זיהוי למגרש, ובחרתי רק את חציו התחתון של המגרש כאזור בו יש לחפש את השחקן התחתון.

דוגמא לחלק המגרש בו מתבצע החיפוש הראשוני:

ניתן לשים לב כי האזור שנבחר מכיל טיפה יותר מחצי מגרש, זאת תוספת שבחרתי בגלל המקרים בהם השחקן קרוב לרשת כך שפלג גופו העליון מופיע מעל חצי מהגרש.

לאחר החיפוש בפריים הראשון, נקבל סט של אנשים שזוהו כפוטנציאלים להיות שחקנים (לרוב רק השחקן מזוהה, אך לפעמים גם שופט או נער הנמצאים בפריים יזוהו), על מנת להפריד בין השחקן לשאר הדמויות, בחרתי לסנן את הדמויות לפי גודלם בפריים, כלומר השחקן יהיה האדם בעל הbounding box הגדול ביותר.

הנחה זו הינה הגיונית מפני שלרוב שאר הדמויות שיזוהו יהיו מכופפות (כמו נער הכדורים שנמצא בפריים בדוגמא למעלה), או רחוקות יותר ולכן קטנות יותר.

לאחר מציאת השחקן בפריים הראשון, אין צורך לחפש אותו בכל חצי המגרש בשאר הפריימים, אלא רק בסביבת מיקומו הקודם.

בהנחה כי השחקן לא יכול להתרחק יותר מידי ממיקומו האחרון, החיפוש נעשה רק בסביבת המיקום האחרון כך שגם תהליך החיפוש מואץ לאחר הפריים הראשון.

<u>תוצאות הניסוי:</u>

את תוצאות הניסוי ניתן לראות בסרטון הבא:

https://www.youtube.com/watch?v=N0C0caSjOnw

דוגמא לפריים בו זוהה השחקן התחתון:

זיהוי "שלד" השחקנים

לאחר זיהוי מיקום השחקנים, המשימה הבאה היא לזהות את מבנה הגוף של השחקנים והוצאת "השלד" שלהם.

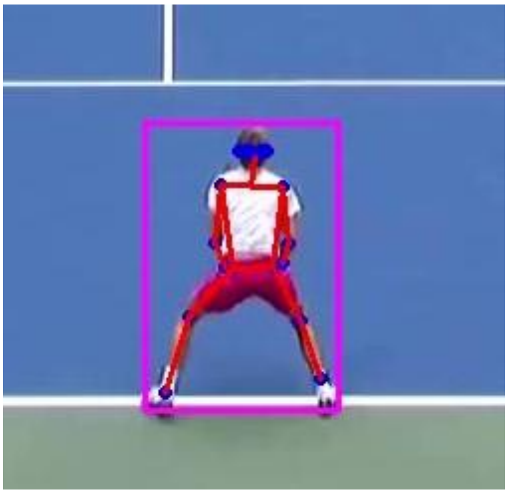
על מנת לבצע זאת, השתמשתי במודל מסוג keypoint RCNN, שבעצם זה מודל הזהה למודל שהשתמשתי בו לזיהוי השחקנים, אך כעת הוא אומן להוצאת מיקומי האיברים בגוף, ויודע לזהות 17 נק', הכוללות את הכתפיים, ידיים, אגן, רגליים וראש.

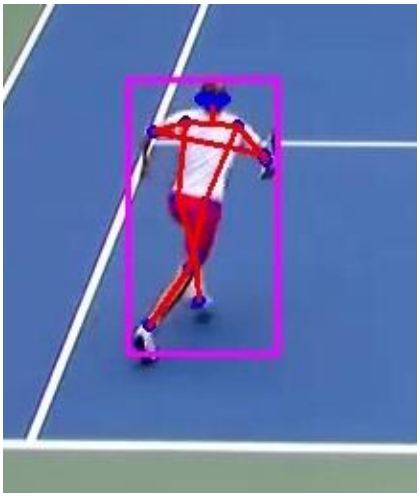
על ידי שימוש במידע שמתקבל מהמודל אני בונה "שלד" של האדם על ידי חיבור הנק' באמצעות קווים.

ניסוי 1 – זיהוי השלד בשימוש ישיר במודל

בניסוי זה, תחילה השתמשתי במידע על מיקום השחקן שמצאתי בניסויים הקודמים, והכנסתי רק את הסביבה של הפיקסלים שמכילה את השחקן לתוך מודל זיהוי הkeypoints.

<u>תוצאות הניסוי:</u>





ניתן לראות בתמונה העליונה, כי המבנה גוף של השחקן זוהה בצורה די טובה.

אך, בתמונה התחתונה ניתן לראות כי ישנם פריימים בהם יש טעות בזיהוי של אחת הנקודות ולכן המבנה גוף יוצא מעוות.

בעיה נוספת שנתקלתי בה היא תזוזות של השלד, הנובעות מכך שזיהוי השלד נעשה בכל פריים בנפרד וללא קשר לזיהוי שבוצע בפריים הקודם, ניתן לזהות את הבעיה בסרטון (https://www.youtube.com/watch?v=2FYbg3iQpbA)

פתרון אפשרי לבעיה הזאת, הוא שימוש בהחלקה על מיקומי האיברים, כך שנוכל להכניס לחישוב את מיקומי הנקודות בפריימים הקודמים והבאים.

ניסוי 2 – הוספת החלקה לזיהוי השלד

כעת, על מנת להימנע ברעידות במיקום הסלקטון, הרצתי החלקה מסוג savgol_filter, אשר מעבירה חלון על הדאטה של מיקומי הנקודות, ומנסה להתאים פולינום מדרגה כלשהי לנקודות בחלון על ידי שימוש ב Least Square, בניסוי זה בחרתי בחלון בגודל 7, ופולינום מדרגה 2.

<u>תוצאות הניסוי:</u>

את תוצאות הניסוי ניתן לראות בסרטון הבא :

https://www.youtube.com/watch?v=rlrYXenCOgE

ניתן לזהות בסרטון, כי כעת הצלחנו להיפטר מהרעידות שהופיעו בניסוי הקודם.

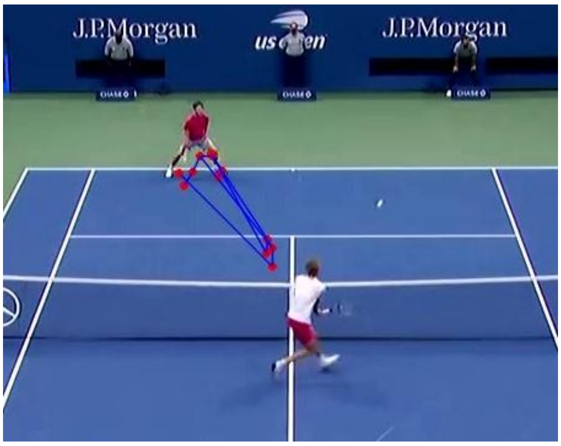
ניסוי 3 – זיהוי נק' שגיאה

כעת, ניסיתי להתמודד עם הבעיה בה יש פריימים בודדים בהם יש טעות בזיהוי הנקודות בגוף השחקן, וזה גורם לשינויים גדולים במבנה הגוף שנוצר.

על מנת להתמודד עם הבעיה הזאת, העברתי חלון על הדאטה של מיקומי הנקודות, וחיפשתי outliers בכל מקטע.

כאשר מצאתי נקודות שהן outliers, החלפתי אותן בממוצע החלון.

דוגמא למקרה בו הזיהוי שגוי (לפני תיקון הבעיה):



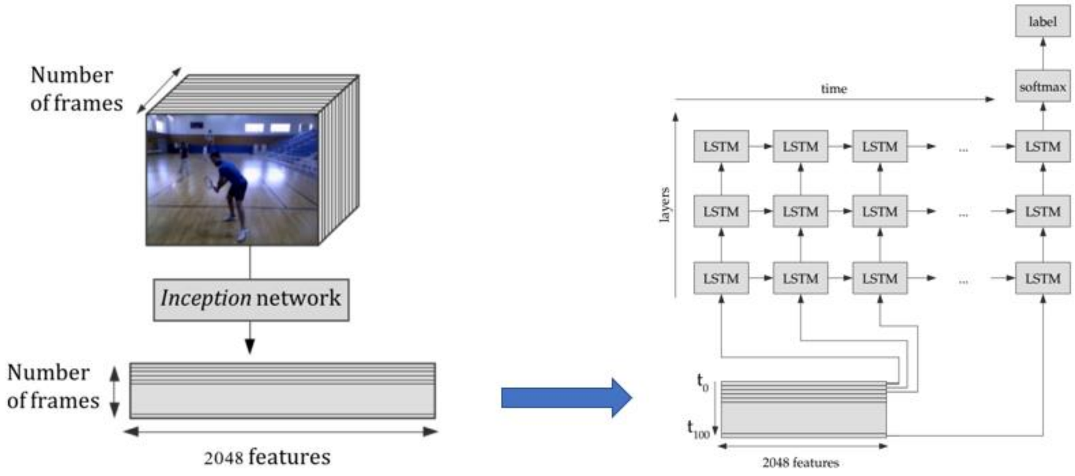
<u>תוצאות הניסוי:</u>

ניתן בהחלט לזהות כי כעת הזיהוי של הנק' השתפר, אך עדיין אינו מושלם בפריים זה. בסרטון הבא ניתן לראות תוצאות של הרצה לאחר התיקון: https://youtu.be/KhFQqFOcokk

סיווג סוגי חבטות

בניסויים הבאים, בניתי מודל לזיהוי סוגי החבטות שיש במשחק טניס.

המודל שבניתי מוסבר בחלק של סקירת הספרות, להלן תמונה של מבנה המודל לתזכורת:



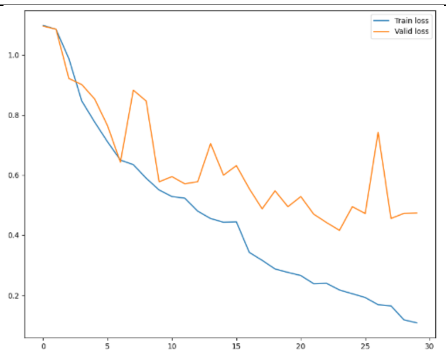
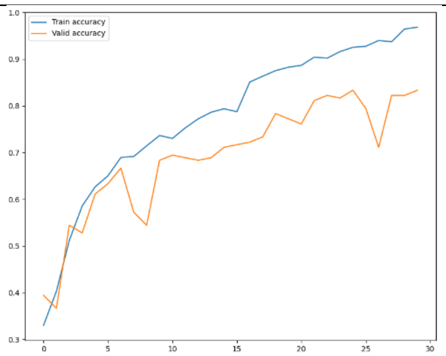
ניסוי 1 – אימון המודל בעזרת סט הסרטונים THETIS

שלב ראשון בניסויים, היה לאמן את המודל בעזרת סט THETIS, כאשר פיצלתי את סט הסרטונים כמו שמומלץ במאמר של הסט. סט הולידציה מורכב בכל הרצה מכל הסרטונים המתאימים ל3 שחקנים שנבחרו בצורה אקראית, סט המבחן מורכב מכל הסרטונים המתאימים לשחקן קבוע (בחרתי אקראית בשחקן מס' 40 לכל הניסויים), וסט האימון מכיל את כל הסרטונים המתאימים לשאר השחקנים בסט.

במהלך ניסוי זה, ניסיתי כמה היפר פרמטרים שונים על ידי שימוש ב Cross validation, עד אשר מצאתי את הפרמטרים הטובים ביותר.

אימון המודל נעשה כך, שבכל איטרציה, המודל מקבל את סרטון מלא של מכה, ומוציא חיזוי למכה שמתרחשת בסרטון, כלומר רק לאחר מעבר על כל הפריימים מתייחסים לחיזוי המודל, ולפי כך מאמנים אותו.

<u>תוצאות הניסוי:</u>

Loss	Accuracy
	

ניתן לראות כי המודל הצליח להתאמן בהצלחה על סט THETIS, ולהגיע לדיוק של בערך 82% על סט הולידציה. לאחר בחינה חוזרת של סט זה, החלטתי כי הוא אינו מתאים לצרכים של הפרויקט, לכן מעבר לבדיקה על יכולות החיזוי של המודל, בחרתי לא להשתמש בו יותר לסיווג המכות העתידי.

לאחר שזהיתי שסט זה אינו מתאים, בחרתי לבנות בעצמי סט של סרטונים ולתייג אותם בצורה ידנית לפי סוגי המכות בסרטון.

את הסט אבנה מסרטונים של משחקים אמיתיים, כאשר אוציא מתוך הסרטונים סביבה מסביב לשחקן התחתון, ואשתמש בחלק זה כסרטון מהסט החדש.

ניסוי 2 – בניית הסט החדש Strokes Dataset

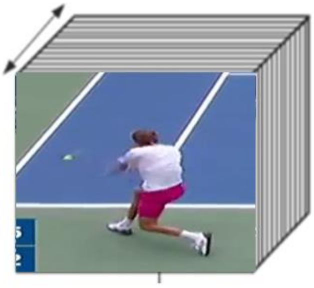
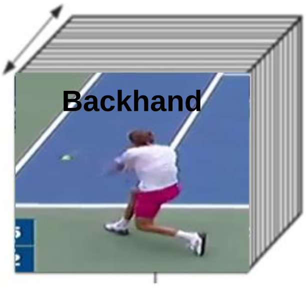
את בניית הסט ביצעתי בשלבים הבאים:

- בחירת סרטונים מ-YouTube של משחקים אמיתיים.
- בניית פלטפורמת לסימון מהיר של אירועי חבטה.
- סימון ידני של כל הרגעים בהם יש מכה של השחקן התחתון בסרטון.
- חיתוך אוטומטי של סרטון באורך 2 שניות מסביב לרגע המכה.
- שימוש במודל לזיהוי השחקן התחתון, על מנת למצוא את מיקומו וחיתוך הסרטון כך שיכיל את השחקן בלבד.
- מעבר על כל הסרטונים שנוצרו (1400 סרטונים), תיוג המכה בכל סרטון, זיהוי סרטונים שחתוכים בצורה לא טובה, זיהוי סרטונים בהם זיהוי השחקן אינו מדויק.
- לאחר שזיהיתי את כל השגיאות (באזור 200 סרטונים), הרצה חוזרת של עיבוד הסרטונים הנ"ל על ידי שימוש בפרמטרים שונים לקבלת סרטון מוצלח, במקרה שזה לא עבד ויתרתי על הסרטונים שנותרו.

בסוף תהליך זה נותרתי עם 1355 סרטונים מתויגים.

<u>דוגמא למהלך הבנייה:</u>

סרטון מקורי	זיהוי רגע המכה	חיתוך סרטון מסביב למכה

לקיחת אזור מסביב לשחקן	סיווג המכה
	

ניתן לראות דוגמאות לתוצאות בניית הסט בסרטון הבא:

https://www.youtube.com/watch?v=P3YSSecYbel

ניסוי 3 – אימון המודל בעזרת הסט החדש Strokes Dataset

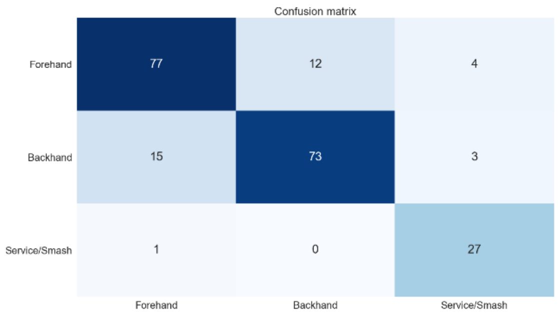
כעת, יש בידי סט חדש המתאים לצרכי הפרויקט, לכן השתמשתי בו לאימון המודל לחיזוי המכות.

את הסט פיצלתי לסט אימון וסט ולידציה, כאשר על מנת להימנע מקשר כלשהו בין הסרטונים השייכים לסטים שונים, פיצלתי את הסט שיצרתי כך שסט הולידציה מורכב מ212 סרטונים השייכים ל4 סרטונים מקוריים, וסט האימון מורכב מ1143 סרטונים השייכים לסרטונים מקוריים אחרים.

את האימון ביצעתי בצורה דומה לאימון שבוצע עם הסט הTHETIS.

<u>תוצאות הניסוי:</u>

Loss	Accuracy

Confusion Matrix


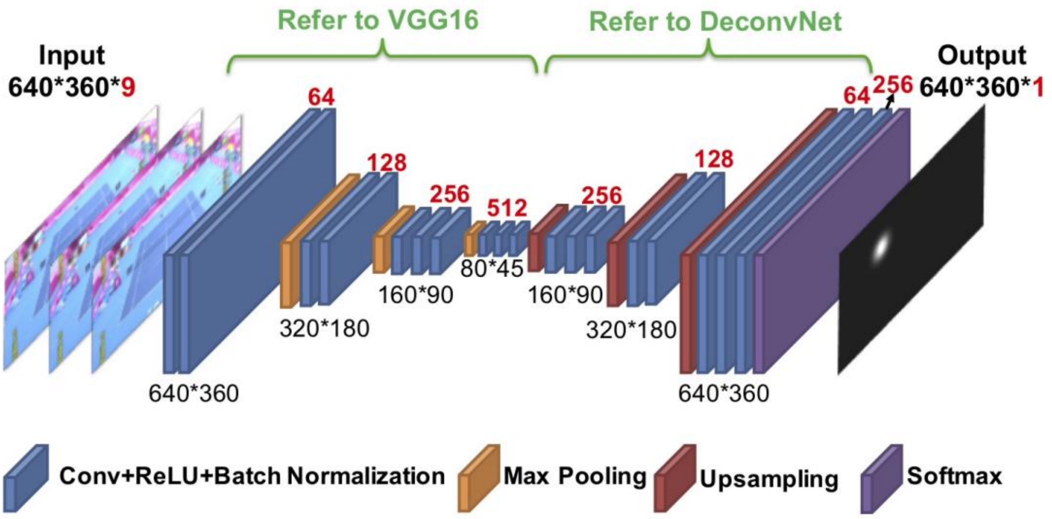
ניתן לראות כי המודל מצליח ללמוד ולסווג בצורה נכונה את המכות בסרטונים.

ניתן לראות כי המודל הגיע לדיוק של 83.5% על סט הולידציה.

בנוסף, ניתן לזהות את הייחודיות של מכת ה Service/Smash , מפני שכמעט ולא נעשתה שגיאה בזיהוי מכה זו.

זיהוי הכדור

על מנת לזהות את הכדור במהלך המשחק, השתמשתי ברשת עמוקה בשם TrackNet ובדאטה סט המתאים לרשת זו, מבנה הרשת וסט הסרטונים מוצג בסקירת הספרות. להלן תזכורת למבנה הרשת:



כאשר קלט הרשת הוא 3 פריימים עוקבים, ופלא הרשת הוא מפת הסתברויות של מיקום הכדור.
בחרתי לממש בעצמי את הרשת ואת מודל האימון של הרשת על מנת שתתאים לצרכי הפרויקט.
את המודל אימנתי לפי ההמלצה של כותב המאמר הפרמטרים הבאים:

Parameters	Setting
Learning rate	1.0
Batch size	2
Steps per epoch	200
epochs	500
Initial weights	random uniform
Range of initial weights	$[-0.05, 0.05]$

ניסוי 1 – אימון עם פלט בעומק 256

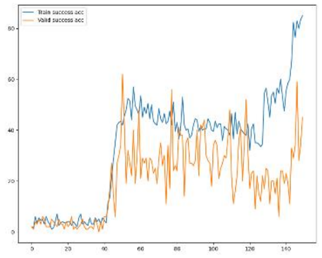
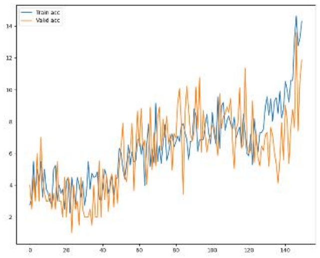
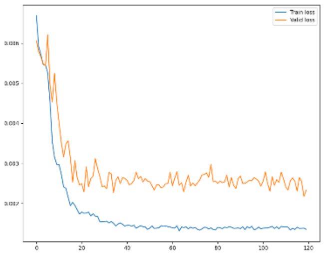
במאמר הקשור למודל, אימנו את הרשת כך שפלט הרשת יהיה מפת הסתברויות בעומק 256 לכל פיקסל בתמונה, כך שבכל ערוץ נקבל את ההסתברות לערך אפור עבור אותו פיקסל (זהה לפעולת קלסיפיקציה לערך אפור).

לאחר קבלת תמונת הפלט והמחלקה של כל פיקסל (ערך אפור), על מנת למצוא את מיקום הכדור בחיזוי, אני משתמש בHough Transforms עבור מעגל, כאשר אני מחפש מעגל בגודל המתאים לכדור טניס סטנדרטי בפריים, ובוחר את המעגל עם מספר ההצבעות הגבוה ביותר בטרנספורמציה.

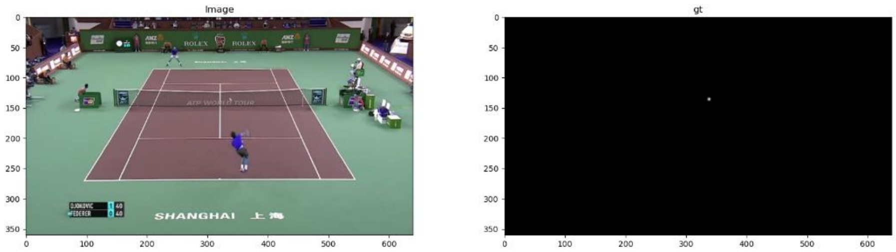
בזמן האימון מחושבים שני מדדים עיקריים (בנוסף לפונקציית ההפסד), מדד ראשון הינו דיוק הפלט של הרשת בפיקסלים השייכים לכדור, חישוב מדד זה נעשה על ידי השוואת ערכי הפיקסלים של מיקום הכדור במפת האמת לערכי הפיקסלים של מיקום הכדור במפת החיזוי, כלומר ישנה "התעלמות" מהפיקסלים השחורים בגלל מספרם הגדול. מדד שני הינו דיוק הצלחות הזיהוי, כאשר הצלחת זיהוי הכדור מוגדרת כך שהכדור שהרשת זיהתה, נמצא במרחק של עד 5 פיקסלים ממיקום הכדור האמיתי, ובמקרים בהם במפת האמת אין כדור והרשת הוציאה גם כן כי אינה מזהה כדור. שאר המקרים האפשריים נחשבים ככשלון בזיהוי הכדור (כאשר זוהה יותר מכדור אחד, לא זוהה כדור אך קיים כדור במפת האמת וכו').

בניסוי זה, פעלתי בצורה דומה לפירוט שהיה במאמר ביחס לפרמטרי האימון.

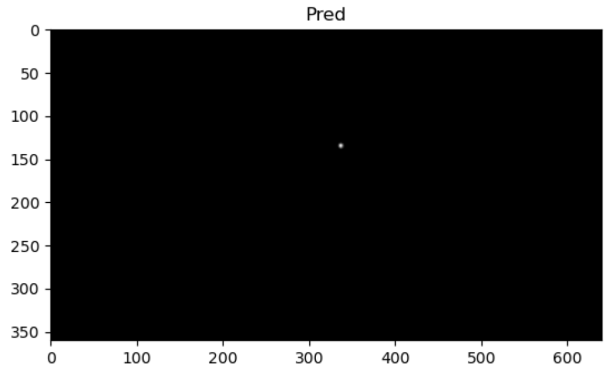
תוצאות האימון:

Success Accuracy	Accuracy	Loss
		

דוגמא לתמונה מסט הולידציה:



ניתן לראות מצד שמאל את הפריים המקורי ומצד ימים את המיקום האמיתי של הכדור.



ניתן לראות כי במקרה הנתון המודל הצליח לזהות את מיקום הכדור.

אך בניגוד למה שניתן לראות בתמונה, ניתן לזהות בגרף כי המודל לא הצליח להגיע לדיוק משמעותי וכי האימון נתקע על בערך 13% דיוק על ערכי פיקסלי הפלט בסט הולידציה ועל 60% דיוק הצלחות זיהוי הכדור בסט הולידציה.

ניתן לראות דוגמא לסרטון בו בוצע זיהוי לכדור בעזרת המודל הנ"ל בקישור הבא:

https://youtu.be/lwQJspX1Tf4

ניסוי 2 – אימון עם פלט בעומק 2

לאחר התוצאות של הניסוי הקודם, ולאחר המון ניסויים והרצות עם פרמטרים טובים על מנת לשפר את זיהוי הכדור, הגעתי למסקנה כי אחת הסיבות לתוצאות שלא מוצלחות שקיבלתי, היא הסיבה שמבצעים קלסיפיקציה לכל פיקסל ל256 ערכים שונים, כאשר עבור תמונה אחת, רוב הערכים הינם 0 וישנם ערכים בודדים השונים מאפס.

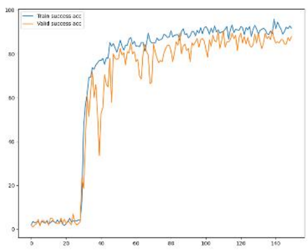
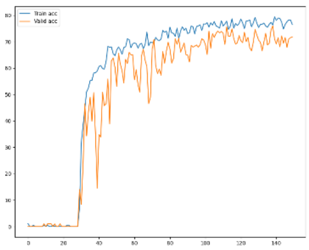
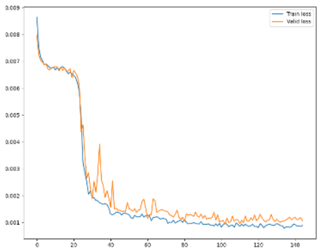
המודל שאומן עד כה, "העדיף" להתכנס למודל אשר מוציא 0 לכל הפיקסלים.

על מנת להתמודד עם כך, שיניתי את פלט הרשת שלי לגודל 2, כאשר מתבצעת פעולת סגמנטציה פשוטה, בה כל פיקסל השייך למיקום הכדור יסומן ב1 (במקום בערכים בטווח 0-255), ושאר הפיקסלים ב-0.

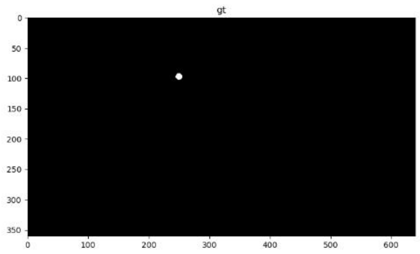
כעת המודל יצטרך להכריע בין 2 ערכים ולא בין 256.

שאר הניסוי נותר כמו שהו.

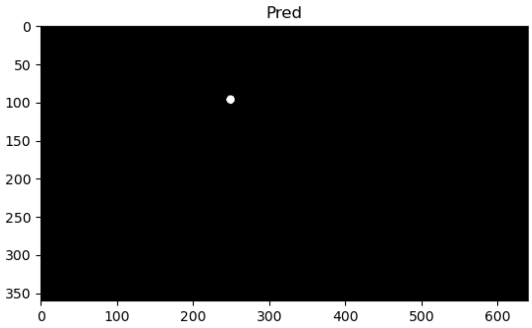
<u>תוצאות האימון:</u>

Success Accuracy	Accuracy	Loss
		

דוגמא לתמונה מסט הולידציה:



ניתן לראות מצד שמאל את הפריים המקורי ומצד ימים את המיקום האמיתי של הכדור.



כעת, ניתן תחילה לזהות כי הפלט הינו בעל 2 ערכים בלבד, וכי מיקום הכדור שזוהה הינו במקום האמיתי.

בנוסף עבור ניסוי זה התקבל דיוק גבוה בהרבה מהדיוק שהתקבל בניסוי הקודם והוא עומד על 81.88% על ערכי פיקסלי הפלט בסט הולידציה ול91.9% דיוק הצלחה בזיהוי הכדור בסט הולידציה.

ניתן לראות דוגמא לסרטון בו בוצע זיהוי לכדור בעזרת המודל הנ"ל בקישור הבא:

https://youtu.be/EyeQWq19voo

זיהוי השחקן העליון

זיהוי השחקן העליון הינה משימה יותר קשה מאשר זיהוי השחקן התחתון מכמה סיבות:

- השחקן העליון יותר רחוק מהמצלמה, לכן הגודל והאיכות שלו יורדים בסרטון.
- לרוב מאחורי השחקן העליון נמצאים שופטי הקו, אשר מפריעים לזיהוי השחקן וגורמים לזיהוי שגוי.

ניסוי 1 – זיהוי נאיבי של השחקן העליון

תחילה בדומה לזיהוי השחקן התחתון, התמקדתי רק בחצי המגרש העליון (על ידי שימוש במיקום המגרש שמצאתי).

דוגמא:

לאחר מכן ניסיתי בצורה נאיבית לזהות את השחקן העליון בעזרת המודל שהשתמשתי לזיהוי השחקן התחתון.

ניתן לראות בסרטון הבא את תוצאות הניסוי:

https://www.youtube.com/watch?v=3CC6I5cvPV4

הבעיה העיקרית הייתה זיהוי שגוי של השופטים מאחורי השחקן, וחוסר יכולת להבדיל איזה מהדמויות היא השחקן ואיזה שופט.

ניסוי 2 – ביצוע מעקב לכל אחת מהדמויות שזווה

על מנת להתמודד עם הבעיה שנתקלתי בה בניסוי הקודם, החלטתי לנסות לבצע מעקב אחר כל אחת מהדמויות שמצאתי בפריים, על ידי שימוש באלגוריתם SORT אשר מתואר בסקירת הספרות.

לאחר ביצוע המעקב אקבל סט של bounding box לכל דמות שזוהתה בכל פריים, ואוכל למצוא מתוך סט זה את הדמות המתאימה לשחקן לפי מציאת הדמות שמרחק התזוזה שלה הוא הגדול ביותר לאורך הסרטון.

בנוסף בגלל שהזיהוי של הדמויות יכול להיפגע בחלק מהפריימים בגלל איכות לא מספיק טובה של הסרטון, ישנם פריימים בהם השחקן לא יזוהה, ולאחר כמה פריימים יזוהה מחדש כדמות חדשה.

על מנת למצוא לאורך כל הסרטון את הדמות המתאימה לשחקן, ביצעתי חיפוש איטרטיבי של הדמות שזזה הכי הרבה, והסרתי את שאר הדמויות שהמקטע בסרטון בו הן מופיעות נחתך עם המקטע בו הדמות עם המרחק תזוזה המקסימלי. כך בסופו של דבר נותרתי עם מספר דמויות אשר נמצאות במקטעים זרים של הסרטון והן

בעלות מרחק תזוזה מקסימלי והשתמשתי בדמויות אלה להגדרת מיקום השחקן (הנחתי כי הדמויות שזזו הכי הרבה מתאימות לשחקן מפני ששאר הדמויות הן לרוב שופטים שאינם זזים כלל).

בסרטון הבא ניתן לראות כיצד בוצע המעקב אחר כל דמות:

https://youtu.be/MwnOWPlq_KY

ובסרטון הבא ניתן לצפות בתוצאה הסופית של זיהוי ומעקב השחקן העליון:

https://youtu.be/MwyRMRbPMjU

זיהוי פריים החבטה

על מנת לבצע חיזוי של חבטות השחקנים, תחילה עלי למצוא את רגע החבטה.

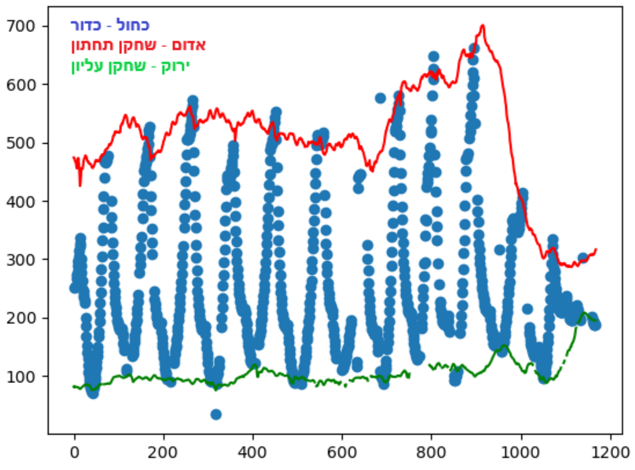
על מנת לבצע זאת השתמשתי במיקום הכדור על מנת לזהות שינויים חדים בכיוונו, ולזהות קרבה לכל אחד מהשחקנים.

ניסוי 1 – זיהוי קרבה בין הכדור לשחקן

תחילה, זיהיתי כי קוארדינטת y של הכדור, היא הקואורדינטה הדומיננטית בזווית הצילום בה בחרתי לעבוד בפרויקט.

על ידי מיקום y של הכדור אוכל לזהות מתי התבצע מכה.

ניתן להסתכל על הגרף הבא אשר מציג את קואורדינטת y של השחקנים ושל הכדור בכל פריים במשחק:

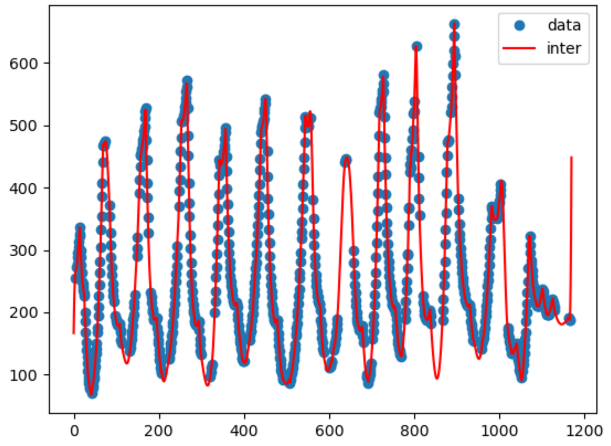


ניתן לראות בבירור מתי הכדור משנה את כיוון תנועתו, ומתי מיקום השחקן ומיקום הכדור נחתכים.

בנוסף ניתן לראות כי בגלל מהירות הכדור וזיהוי לא מושלם, ישנם אזורים בגרף בהם יש רווחים במיקומי הכדור.

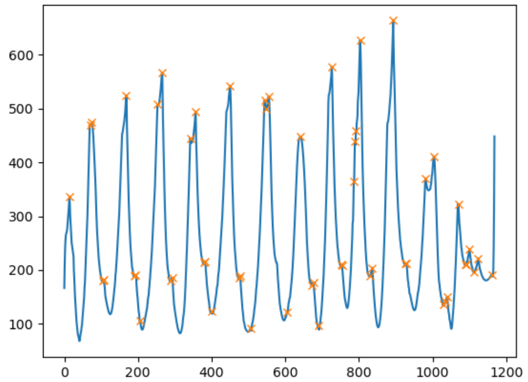
לכן, תחילה אבצע אינטרפולציה למידע הכדור כך שאוכל למצוא בצורה מדויקת יותר את נק' הקיצון של מיקום הכדור (נק' קיצון במיקום הכדור בציר y לרוב תעיד על שינוי בכיוונו ולכן על נקודה בה הייתה חבטה).

בגרף הבא ניתן לראות את האינטרפולציה הריבועית שביצעתי למיקומי הכדור:

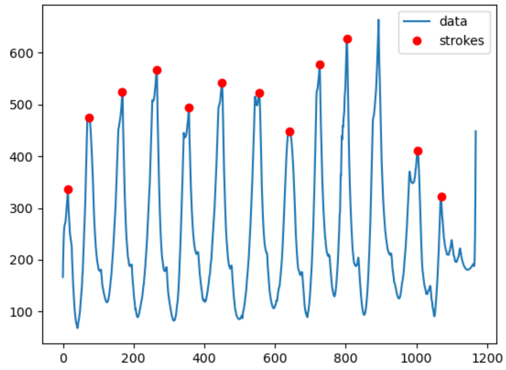


כעת לאחר ביצוע האינטרפולציה, אחפש נקודות מקסימום מקומיות ואבחר נקודות אלה להיות מועמדות לרגע

החבטה:

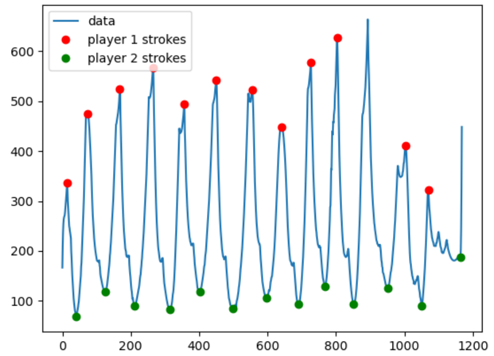


בשלב זה יש בידי מספר נקודות החשודות כרגעים בהם הייתה חבטה של השחקן התחתון (בעל ערכי y גדולים). על מנת לסנן את הנקודות, ולהשאיר רק את הנק' המתאימות לחבטה, מתוך כל הנק' הנ"ל בחרתי רק נק' בהן מרחק הכדור ממיקום השחקן (את מיקום השחקן הגדרתי בשלוש נק': מרכז המסה של השחקן, מיקום כף יד ימין ומיקום כף יד שמאל) קטן מסף מסוים.
בנוסף, קבעתי כי לא ייתכן כי יהיו שתי חבטות במרחק של פחות מ30 פריימים (שנייה אחת), לכן במקרים בהם זוהתה יותר מחבטה אחת, השארתי את הנק' בה המרחק בין הכדור לשחקן מינימלי.
בגרף הבא ניתן לראות את הנק' שנבחרו מתוך כל הנק' הקודמות כמתאימות לחבטה:



בגרף ניתן לזהות כי ישנה נק' מקסימום אחת שלא סומנה, זה נובע מזיהוי לא מספק של השחקן כמו שניתן לראות בסרטון (רוב גופו נמצא מחוץ לפריים).
לאחר בדיקה כי שיטה זו אכן מוצאת את הפריימים בהם יש חבטה, פעלתי בצורה דומה על מנת למצוא את רגעי החבטה של השחקן העליון בפריים, כאשר ההבדל היחיד הוא חיפוש נק' מינימום במקום מקסימום בערך y של הכדור וחישוב המרחק ביחס לשחקן העליון.

תוצאות הזיהוי:



ניתן לראות בסרטון הבא את תוצאות ניסוי זה, כאשר בזמן שזוהה כחבטה תופיעה הודעה מעל השחקן החובט:
https://www.youtube.com/watch?v=ivjCtC0M7pE

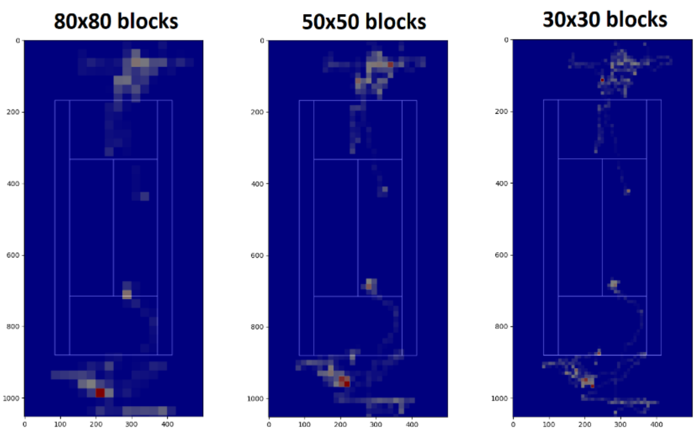
סטטיסטיקות

לאחר עיבוד של הסרטון, יש בידי מספיק מידע על מנת ליצור כמה סוגי סטטיסטיקות מעניינות לגבי המשחק בסרטון.

ניסוי 1 – מיקום השחקנים במגרש לאורך המשחק

סטטיסטיקה ראשונה שבחרתי לעבוד עליה היא מיקום השחקנים במגרש לאורך המשחק.
ברגע שיש בידי את מיקום השחקנים בפריים, ואת הטרנספורמציה ההומוגרפית שמקשרת בין המגרש בפריים למגרש הרפרנס, ניתן להשתמש במידע זה על מנת למצוא את מיקום השחקנים במגרש הרפרנס בכל רגע.
על מנת למצוא את מיקום השחקנים כל שעלי לעשות הוא להשתמש בטרנספורמציה ההופכית לטרנספורמציה של המגרש שמצאתי, ולהפעיל אותה על מיקום רגלי השחקנים (מוסבר בסקירת הספרות כיצד זה נעשה).
כעת כשיש בידי את מיקום השחקנים במגרש הרפרנס, כל שיש לעשות הוא לחלק את המגרש לבלוקים בגדלים קבועים ולספור כמה פריימים בילה השחקן בכל בלוק.
בניסוי זה ניסיתי כמה גדלים שונים לבלוקים.

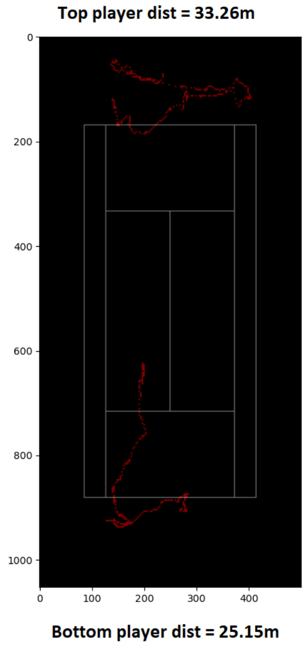
<u>תוצאות הניסוי:</u>



ניתן לזהות כי הבלוקים הקטנים מציגים ממש את המסלולים שעשו השחקנים, אך על מנת לקבל תמונה יותר רחבה, ניתן להסתכל על בלוקים גדולים יותר, ולהבין היכן היה כל שחקן רוב הזמן.

ניסוי 2 – חישוב מרחק ריצה של כל שחקן

סטטיסטיקה נוספת שניתן לחשב עבור הסרטון, היא המרחק שעבר כל אחד מהשחקנים במהלך המשחק.
על מנת לבצע זאת, השתמשתי במיקום השחקנים במגרש הרפרנס בכל פריים, ותחילה חישבתי את המרחק שנע כל שחקן במגרש הרפרנס, על ידי חישוב המרחק בין המיקומים בכל פריים.
לאחר מציאת מרחק זה, השתמשתי בכך שבניתי את מגרש הרפרנס ביחס של פיקסל אחד עבור כל סנטימטר של המגרש.
כלומר, גודל מגרש הרפרנס בפיקסלים הוא בדיוק גודל המגרש בסנטימטרים. כלומר, כל שעלי לעשות על מנת למצוא את המרחק שנע כל שחקן, הוא לחלק את המרחק שעבר בפיקסלים ב-100 על מנת לקבל את המרחק במטרים.
נקודה חשובה שצריך לשים לב אליה, היא הרזולוציה של מיקום השחקנים, בגלל שכל שינוי בפיקסל בודד במיקום גורר שינוי של סנטימטר בודד. כלומר, אם המיקום אינו מדויק לכדי סנטימטר, התזוזות הקטנות של המיקום יגררו סכימה של תזוזה שלא באמת הייתה.
על מנת להתמודד עם בעיה זאת, בחרתי להנמיך את הרזולוציה של הדיוק במיקום, ולפני חישוב המרחק לכל שחקן, עיגלתי את המיקומים של השחקנים 30 פיקסלים כלפי מטה (אשר מתאימים ל30 סנטימטר בעולם האמיתי), כך שאם המיקום של השחקן משתנה בפחות מ30 פיקסלים, זה לא ישפיע על המרחק שהוא נע.
<u>תוצאות הניסוי:</u>

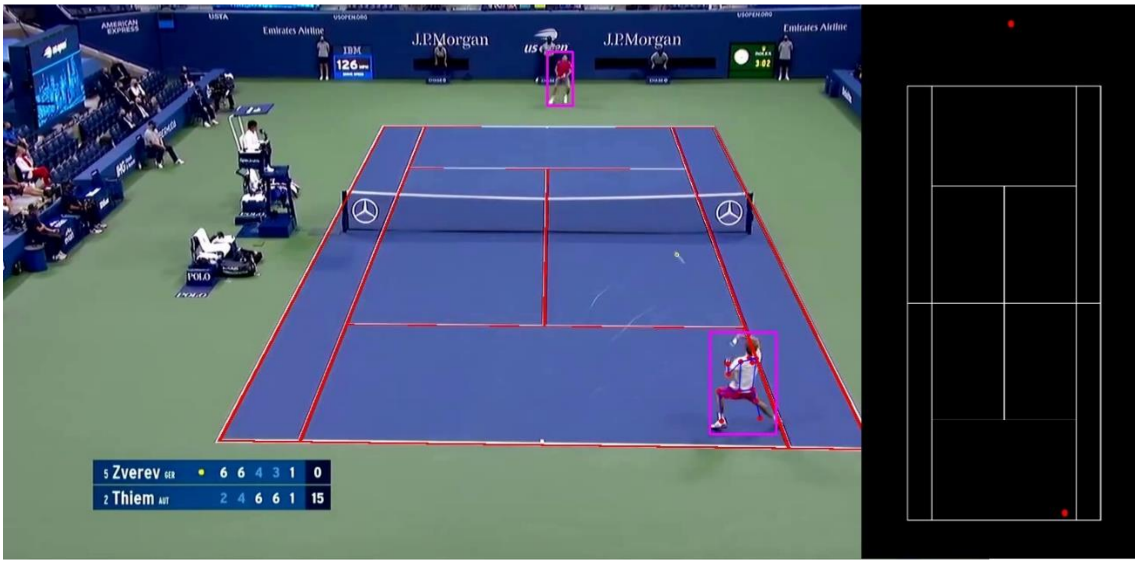


ניתן לראות בצורה ברורה את המסלול שביצע כל שחקן במהלך המשחק, ובנוסף לקבל הערכה מוצלחת למרחק שזזו במהלכו. בנוסף לכך, כעת ניתן להוסיף לסרטון הסופי את המרחק שנע כל שחקן בכל רגע.

ניסוי 3 – בניית מודל מבט על למשחק

בנוסף לסטטיסטיקות שחישבתי עד כה, ניתן להשתמש במידע שמצאתי לבניית מודל משחק ממבט על. כלומר להציג את מהלך המשחק על מגרש הרפרנס, כך שנוכל להציג בכל רגע את מיקום השחקנים "ממבט על". על מנת לבצע זאת אשתמש במיקום השחקנים שמצאתי בחישובי הסטטיסטיקות הקודמות, ואבנה סרטון כך שכל פריים מציג את המיקום של השחקנים בפריים המתאים בסרטון המקורי.

<u>תוצאות הניסוי:</u>



ניתן לזהות כי מיקום השחקנים במודל ממבט על מדויק, ומציג בצורה יפה את מהלך המשחק.

דוגמא לתוצאות הניסוי ניתן לראות בסרטון הבא:

https://www.youtube.com/watch?v=5CL4IAxqeYE

תוצאות

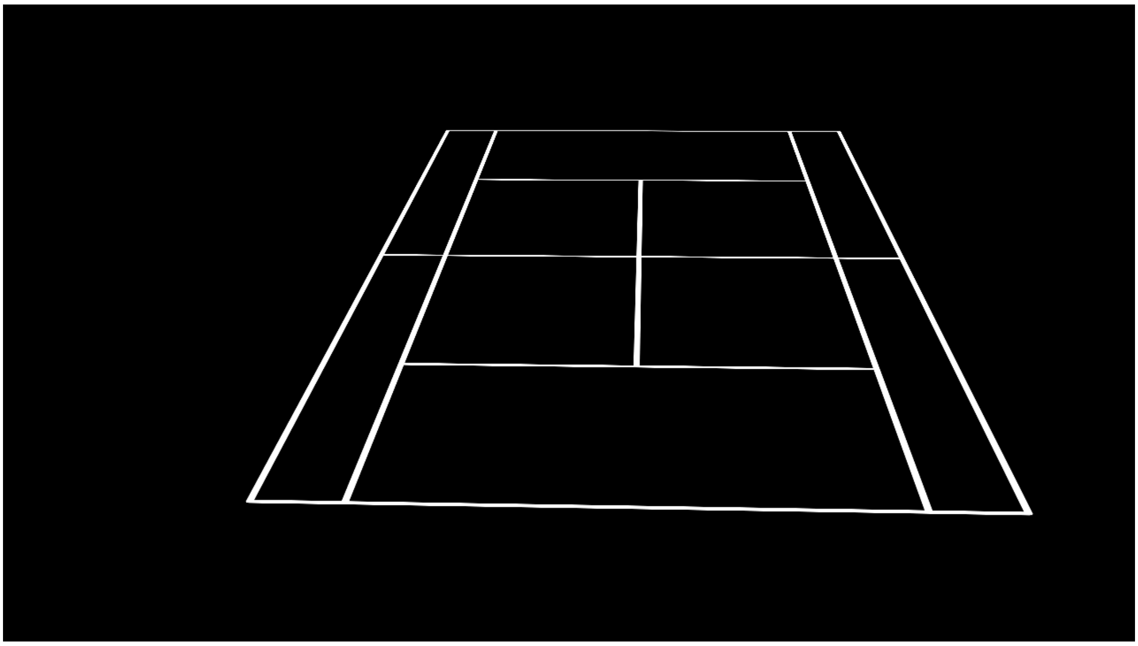
בפרויקט זה, הראיתי כי ניתן ליצור מודל המבצע עיבוד וניתוח למשחקי טניס רשמיים, רק באמצעות מודלים של ראייה ממוחשבת.

מטרות שהושגו בפרויקט:

- זיהוי מגרש ומעקב אחר מיקומו לאורך המשחק
- זיהוי ומעקב שני השחקנים
- זיהוי ומעקב של הכדור
- הוצאת "שלד" לשחקן הקרוב למצלמה
- זיהוי רגעי החבטות
- סיווג סוג החבטה של השחקן הקרוב למצלמה
- יצירת מודל משחק ממבט על המכיל את שני השחקנים
- חישוב סטטיסטיקות שונות

כעת אציג כמה דוגמאות לתוצאות מתוך כל מטרה שהושגה, כאשר אוסיף סרטונים להדגמה לחלק מהמטרות שהושגו (חלק מהסרטונים הופיעו בחלק של הניסויים).

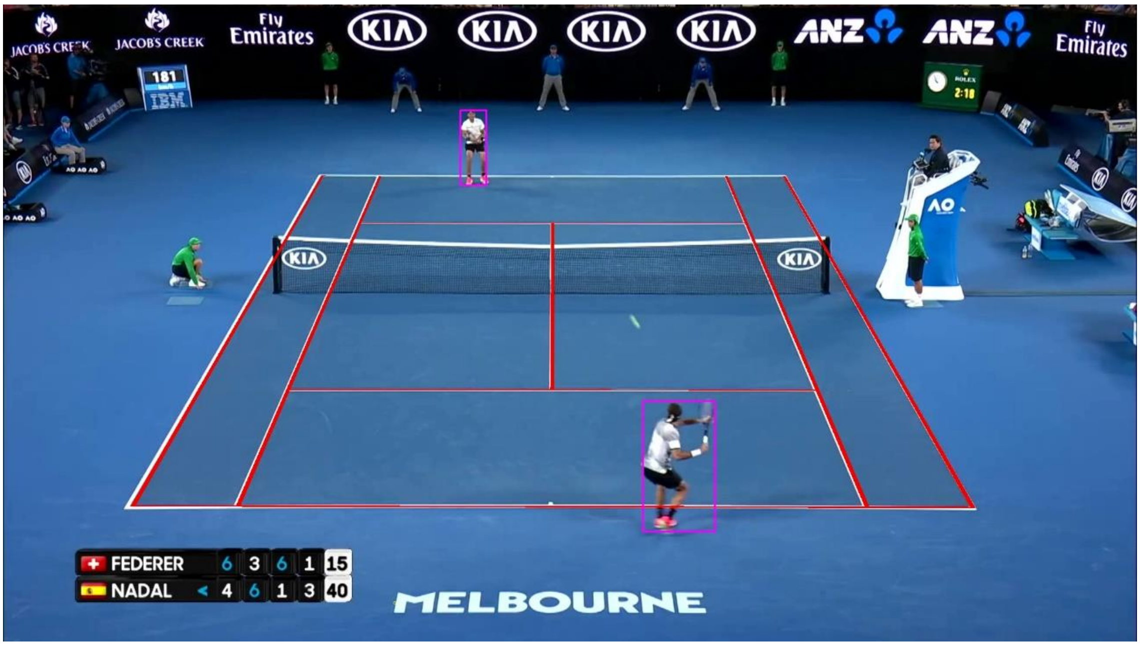
זיהוי מגרש:



בנוסף ניתן לראות הדגמה בסרטון הבא, אשר מציג בצורה יפה כיצד גם כאשר המצלמה זזה מקצה לקצה, המגרש עדיין מזוהה בצורה מדויקת:

https://www.youtube.com/watch?v=N0C0caSjOnw

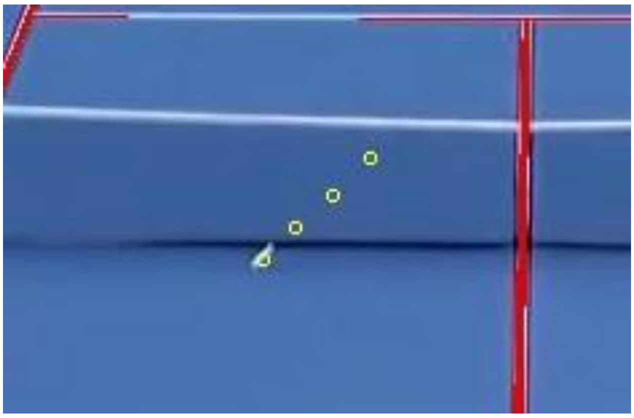
זיהוי שחקנים:



בסרטון הבא ניתן לראות הדגמה לזיהוי ומעקב אחר שני השחקנים, ניתן לזהות כי בכל שלב במהלך הסרטונים שני השחקנים מסומנים ומקומם ידוע:

https://www.youtube.com/watch?v=MwyRMRbPMjU

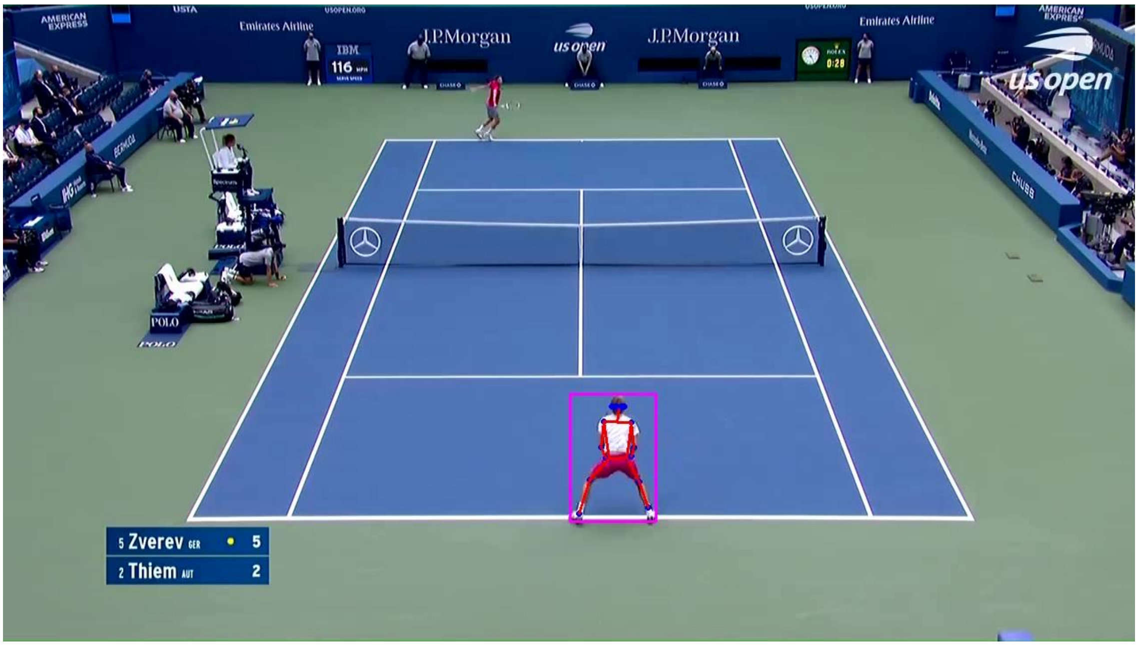
זיהוי הכדור:

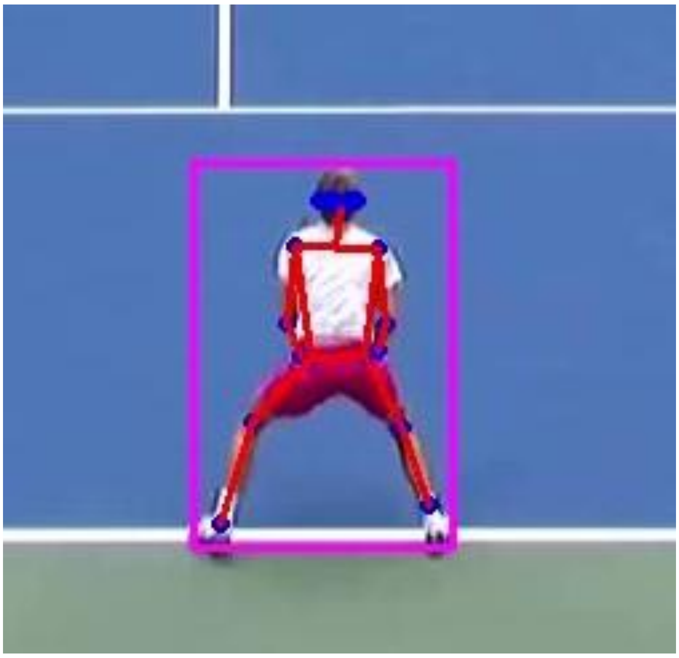


התמונה הנ"ל מציגה את 4 הזיהויים האחרונים של הכדור עבור סרטון מסוים.

בנוסף לכך, ניתן לראות הדגמה של זיהוי הכדור בסרטון הקודם שהצגתי https://www.youtube.com/watch?v=MwyRMRbPMjU.

זיהוי שלד השחקן הקרוב למצלמה:

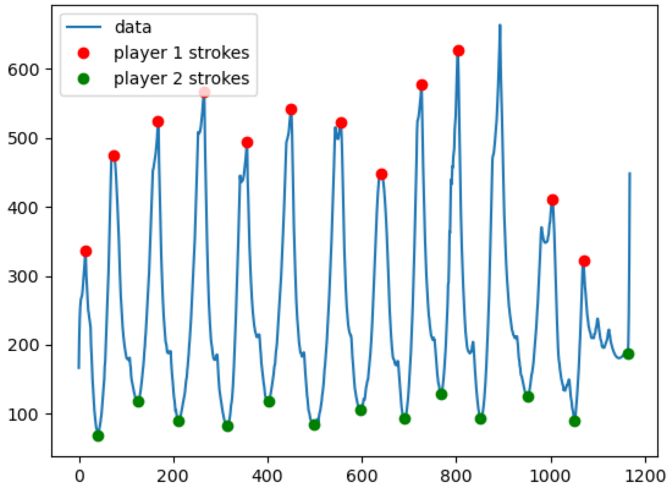




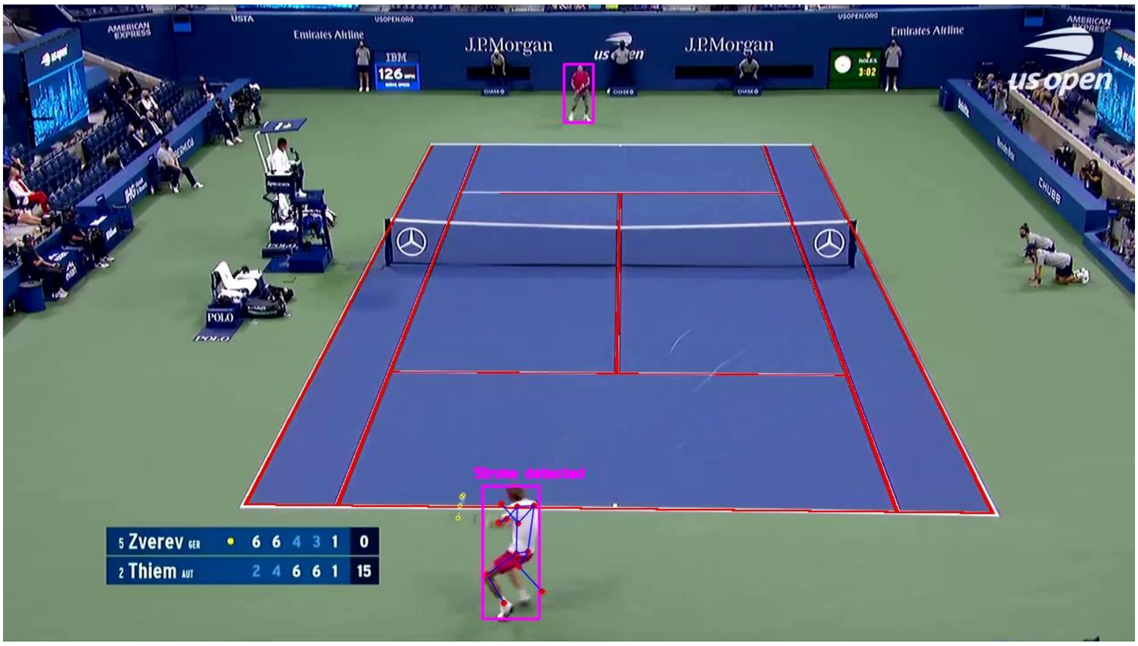
בסרטון הבא ניתן לראות הדגמה של זיהוי השלד:

https://www.youtube.com/watch?v=rlrYXenCOgE

זיהוי רגעי החבטות:



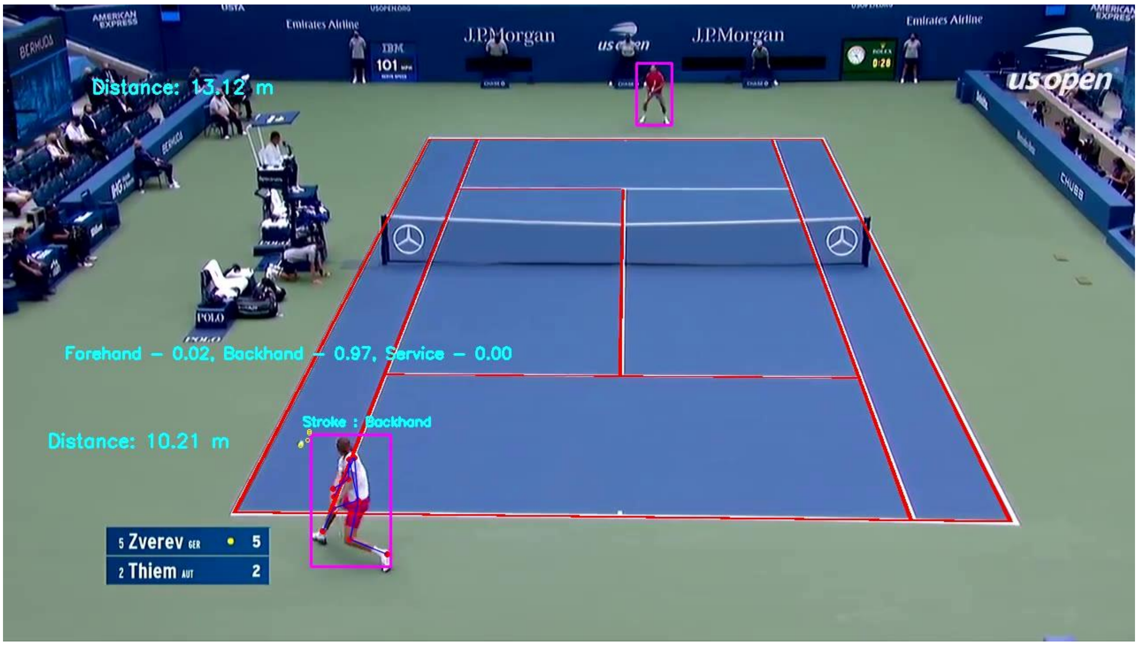
בגרף ניתן לראות את אינדקס y של הכדור בכל פריים, ואת הנק' שנבחרו לרגעי החבטות.



בסרטון הבא ניתן לראות הדגמה אשר מציגה מתי זוהתה חבטה ומתי לא עבור שני השחקנים:

https://www.youtube.com/watch?v=ivjCtC0M7pE

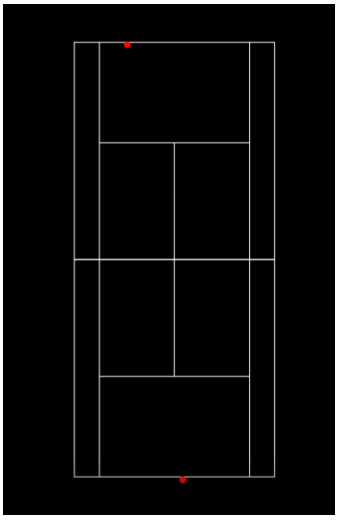
סיווג סוגי חבטות לשחקן הקרוב למצלמה:



בסרטון הבא ניתן לראות הדגמה לביצוע זיהוי וסיווג החבטות:

https://youtu.be/uvk9m17j25g

יצירת מודל משחק ממבט על עבור שני השחקנים:

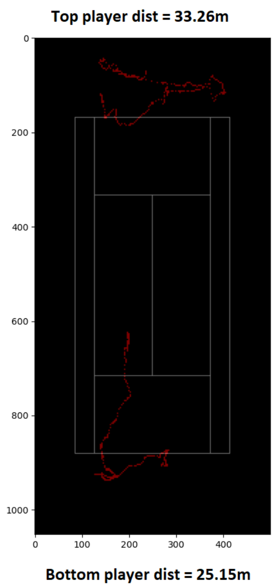


בסרטון הבא ניתן לראות הדגמה של משחק ושל המודל משחק ממבט על שנוצר עבור הסרטון:

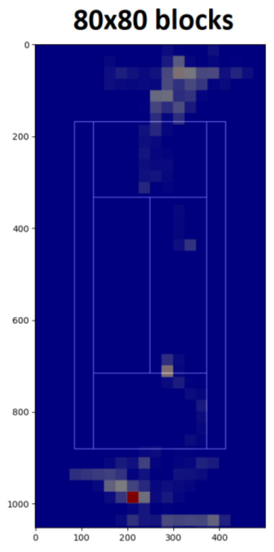
https://www.youtube.com/watch?v=5CL4IAxqeYE

סטטיסטיקות שונות:

מרחק ריצה עבור כל שחקן -



מיקום כל שחקן במהלך המשחק -



בנוסף לכך, ניתן לקבל מידע על מספר החבטות השונות שביצע השחקן והמיקום בו בחר להשתמש בכל אחת מהן.

הדגמה מלאה:

בסרטון הבא אציג בצורה מלאה את כל החלקים של הפרויקט, ניתן לזהות בסרטון את זיהוי המגרש, השחקנים, הכדור, סוגי החבטות של השחקן הקרוב למצלמה, את המרחק שעבר כל שחקן בכל רגע ואת מודל מבט העל של המשחק:

https://youtu.be/-Hf69zMGh9w

פרטי מימוש

הפרויקט נכתב בכולו בPython.

כחלק מהמימוש בחרתי להשתמש בכמה ספריות הממשות חלק מהאלגוריתמים שהצגתי וחלק מהרשתות העמוקות, ביניהם:

- OpenCV – עבודה עם הסרטונים, מימוש אלגוריתמי זיהוי והתאמת קווים ומעגלים כגון Least squares, Hough transform.
- PyTorch, Torchvision - מימוש של חלק מהרשתות העמוקות כגון TrackNet, ושימוש ברשתות שאומנו מראש כגון Faster RCNN.
- Scipy – ביצוע החלקות למידע ואינטרפולציה.

בפרויקט בחרתי לממש את רוב הקוד בעצמי ולהשתמש בספריות הנ"ל למימוש חלק מהאלגוריתמים השונים, למעט שימוש בקוד מוכן של אלגוריתם SORT שהצגתי קודם לכן.

סיכום

סיכום

בפרויקט זה הצגתי כיצד ניתן לנצל מודלים ואלגוריתמים של ראיה ממוחשבת על מנת לבצע עיבוד וניתוח איכותי למשחקי טניס רשמיים, על מנת לקבל תמונה ברורה יותר של המשחק הכוללת סטטיסטיקות שונות.

במהלך הפרויקט הצלחתי להשיג כמה מטרות שונות שהוזכרו בחלק הקודם:

- זיהוי מגרש ומעקב אחר מיקומו לאורך המשחק
- זיהוי ומעקב שני השחקנים
- זיהוי ומעקב של הכדור
- הוצאת "שלד" לשחקן הקרוב למצלמה
- זיהוי רגעי החבטות
- סיווג סוג החבטה של השחקן הקרוב למצלמה
- יצירת מודל משחק ממבט על המכיל את שני השחקנים
- חישוב סטטיסטיקות שונות

לדעתי זה תחום מאד מעניין אשר ניתן לפיתוח מעמיק יותר עד כדי הגעה לשימושים אמיתיים בשטח.

בנוסף, לדעתי המודלים הנ"ל מאפשרים למאמני טניס שונים לנצל את התוצאות על מנת לשפר את יכולת האימון שהם מעבירים והתכוננות טובה יותר למשחקים עתידיים.

הצעות להמשך פיתוח

לדעתי מספר הכיוונים האפשריים להמשך פיתוח הינו גדול מאד, וניתן להתמקד בחלקים שונים בפרויקט ולשפר אותם, או ללכת לכיוון של שדרוג המודל מעבר ליכולותיו כעת. מספר אפשרויות לדעתי להמשך פיתוח:

- הגדלת סט הסרטונים Strokes ושיפור המודל לסיווג חבטה (אפשרות לסיווג מספר גדול יותר של סוגי חבטות)
- ביצוע סיווג לחבטות השחקן העליון בפריים (הוספת סט סרטונים המתאים לשחקן זה)

- ייעול החישובים על מנת להאיץ את פעולת המודל
- הוספת כדור למודל ממבט על
- ביצוע העיבוד בזמן אמת וחיזוי חבטה עתידית
- זיהוי האם הכדור נחת בתוך המגרש או לא (שימוש כשופט בזמן משחק)
- הוספת סטטיסטיקות מורכבות כמו ספין הכדור ומהירות החבטה.

ביבליוגרפיה

1. Silvia Vinyes Mora, “Computer Vision and Machine Learning for In-Play Tennis Analysis: Framework, Algorithms and Implementation”, October, 2017.
2. Yu-Chuan Huang, "TrackNet: Tennis Ball Tracking from Broadcast Video by Deep Learning Networks," Master Thesis, advised by Tsì-Uí İk and Guan-Hua Huang, National Chiao Tung University, Taiwan, April 2018.
3. Yu-Chuan Huang, I-No Liao, Ching-Hsuan Chen, Tsì-Uí İk, and Wen-Chih Peng, "TrackNet: A Deep Learning Network for Tracking High-speed and Tiny Objects in Sports Applications," in the IEEE International Workshop of Content-Aware Video Analysis (CAVA 2019) in conjunction with the 16th IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS 2019), 18-21 September 2019, Taipei, Taiwan.
4. Bewley, Alex and Ge, Zongyuan and Ott, Lionel and Ramos, Fabio and Upcroft, Ben, “Simple online and realtime tracking” in 2016 IEEE International Conference on Image Processing (ICIP)
5. Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun, “Deep Residual Learning for Image Recognition” in 2015
6. Sofia Gourgari and Georgios Goudelis and Konstantinos Karpouzis and Stefanos Kollias, “THETIS: Three Dimensional Tennis Shots a Human Action Dataset” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2013
7. Bin Xiao and Haiping Wu and Yichen Wei, “Simple Baselines for Human Pose Estimation and Tracking”, 2018.
8. Hayden Faulkner, “Temporal Event Annotator”, https://github.com/HaydenFaulkner/TemporalEventAnnotator.