

פרויקט מסכם עיבוד תמונה – "מלך ההקפצות"

מרצה: פרופ' תמי ריקלין רביב
מתרגלות: שירה כרמי, יעל אלקין

מגישים:

דורון גוזנברג 318183589

מאור בראון 316119767

עומר הורוביץ 322587395

מתן גבאי 209268689



תוכן עניינים

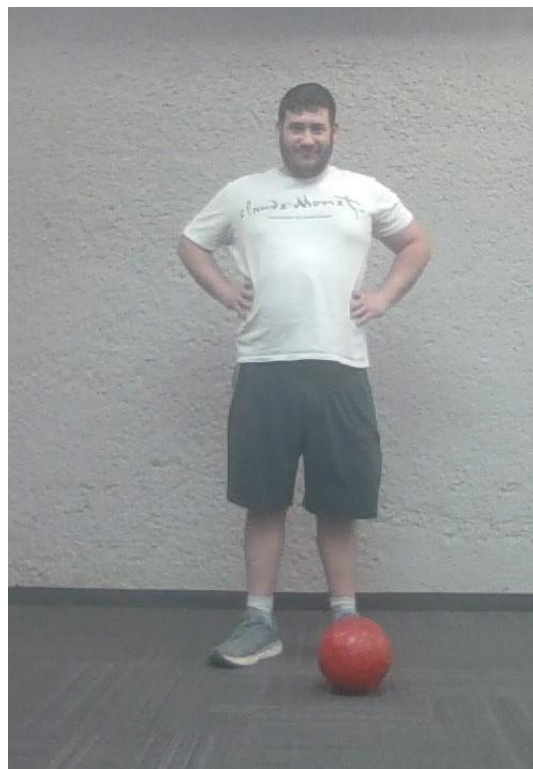
3	תקציר הפרויקט :
4	תיאור הפרויקט
6	תיאור האלגוריתמים
7	זיהוי הכדור:
12	אלגוריתם לזיהוי הקפצת כדור:
14	זיהוי פגיעה ברצפה:
20	אלגוריתם זיהוי נגיעה ביד.
26	אנליזת התוצאות- בחינת ביצועי אלגוריתם זיהוי הרצפה
32	מסקנות:
33	סיכום:
34	ביבליוגרפיה:

תקציר הפרויקט :

משחק הכדורגל הוא בין המשחקים הפופלארים בעולם. כאשר מילוני ילדים ברחבי העולם מקדישים את מרבית זמנם בבילוי במגרשי הכדורגל. אחת הדרכים להשתפר במשחק היא להשתפר בשליטה בכדור על ידי שליטה במיומנות ההקפצה. ילדים רבים מתקשים בספירת ההקפצות באופן סימולטני עם ביצוע ההקפצות. לכן יצרנו תחרות שתדרבן את השחקנים להשתפר פלאים במשחק ההקפצות תוך כדי הנגשת הספירה למשתמש בצורה הנוחה ביותר. מלך ההקפצות היא תחרות הקפצת כדור כדורגל. ראשית השחקן מכניס את שמו המלא למערכת ולאחר הזנקה מקפיץ את הכדור עד שנפסל. פסילה מוגדרת להיות אחת משלושת האפשרויות הבאות:

- א. הכדור נופל לרצפה.
- ב. הכדור יוצא מתחומי המשחק (טווח הראייה של המצלמה)
- ג. השחקן נוגע בכדור בעזרת היד.

מספר ההקפצות נספר באופן אוטומטי, כאשר השחקן נפסל באחת הדרכים המתוארות לעיל, האלגוריתם פוסל את השחקן ומציג את מספר ההקפצות שצבר, מתאר את סיבת הפסילה ואת מיקומו בטבלה ביחס לשאר המשתתפים.



תיאור הפרויקט

תיאור המערכת:

לאחר הכנסת שם השחקן, מנהל המשחק מבקש מהשחקן לעמוד מול המצלמה במרכז הפריים כאשר ידיו על מותניו ומנהל המשחק מסמן את ידיו של השחקן, ומפעיל את הספירה לאחר שבסופה מגיעה הזנקת המשחק. השחקן מקפיץ את הכדור שאותו נזהה באלגוריתם, נזהה הקפצה חוקית ונספור את מספר ההקפצות החוקיות עד אשר נפסל לאור אחת מן הסיבות הבאות: הכדור נופל לרצפה, הכדור יוצא מתחומי המשחק (טווח הראייה של המצלמה) או נגיעת היד של השחקן בכדור. את הסיבות הללו אנו נזהה בזמן אמת וכאשר השחקן נפסל האלגוריתם פוסל את השחקן ומציג את מספר ההקפצות שצבר, מתאר את סיבת הפסילה ואת מיקומו בטבלה ביחס לשאר המשתתפים.

מיקום המצלמות ויעודן:

בפרויקט אנו משתמשים במצלמה אחת. המצלמה תמוקם מול השחקן כאשר הוא עומד מולה בתוך שטח המשחק שהוגדר לו. המצלמה תמוקם כך שבתוך הפריים ייתפס כל גופו של השחקן ומעט שטח מעל ראשו, בנוסף הפריים יכלול חלק נכבד מן הרצפה מלפני רגלי השחקן לצורך זיהוי פגיעה ברצפה. יעוד המצלמה הוא לקלוט את הקפצות הכדור ופסילת השחקן.

מטרות:

מטרת העל של הפרויקט היא לספק פתרון יעיל ואמין לספירת הקפצות כדור חוקיות בזמן אמת ובאופן רציף.

1. זיהוי הכדור בכל פריים.
2. זיהוי הקפצה חוקית.
3. זיהוי מיקום הידיים בתחילת המשחק.
4. מעקב אחרי ידי השחקן לאורך המשחק בזמן אמת.
5. ספירת ההקפצות
6. זיהוי פסילת השחקן באחת מהאפשרויות הבאות:
 - a. זיהוי מגע בין ידי השחקן לכדור.
 - b. זיהוי נפילת הכדור על הרצפה.
 - c. יציאת הכדור מהפריים.
7. עיצוב המשחק, חווית המשתמש והצגת הניקוד וסיבת הפסילה בזמן אמת.

אילוצי/הנחות הפרויקט:

1. הכדור אדום.
2. השחקן אינו לובש בגדים אדומים.
3. הרקע אינו מכיל חלקים אדומים.
4. למעט השחקן והכדור אין אובייקטים נוספים שזזים בפריים.

אתגרי הפרויקט:

1. למצוא אלגוריתם שיצליח לעקוב אחרי תנועות ידיו של השחקן בדגש על תנועות חדות (נראה כמריחה בפריים).
2. למצוא אלגוריתם שיזהה הקפצה תקינה ויספור כל הקפצה כהקפצה יחידה ויקלוט את כל ההקפצות התקינות.
3. מציאת אלגוריתם לזיהוי נפילת הכדור לרצפה שיתמודד בצורה מיטבית עם מקרה קצה שבו הכדור נופל סמוך לרגלי השחקן (עלול להראות כהקפצה תקינה).
4. זיהוי הכדור בכל פריים ללא קשר למרחקו מן המצלמה ולמהירותו.
5. התמודדות עם מצבי קצה הקשורים למיקומים של השחקן ושל הכדור ביחס למצלמה וביחס בין מיקום השחקן לכדור עצמו.
6. מציאת אלגוריתמים יעילים חישובית שיאפשרו הרצה של המשחק בזמן אמת ללא פגיעה באיכות המשחק וחוויית המשתמש.
7. אינטגרציה של כל חלקי המשחק לכדי קוד אחד מתפקד.



בתמונה זו רואים תמונת הפרשים שבה מתקיים מצב שבו הכדור נופל על הרצפה אך קרוב מדי לרגלי השחקן ולכן זוהה כהקפצה תקינה.

תיאור האלגוריתמים

הקוד שלנו מכיל ארבעה אלגוריתמים עיקריים:

אלגוריתם ראשון הינו אלגוריתם שמטרתו לזהות את הכדור, כאשר אין זיהוי של הכדור נסיק כי הכדור יצא מתחום המשחק ולכן נפסול את השחקן.

אלגוריתם שני שמטרתו לזהות מתי התרחשה הקפצה ובנוסף מניית הקפצות השחקן.

אלגוריתם שלישי שמטרתו לזהות את ידיו של השחקן ולעקוב אחר תנועתן, ולפסול את השחקן במקרה של פגיעה ביד.

אלגוריתם רביעי שמטרתו לזהות האם התרחשה פגיעה של הכדור ברצפה אשר תוביל לפסילת השחקן.

על מנת לשפר את חווית המשתמש של השחקן השתמשנו בתיקיית "Pygame". יצרנו מסך המציג ספירה לאחור של השחקן וכאשר הוא מגיע ל10 הקפצות מופיעה תמונה של זיקוקים עם קריאת שמח "wow" בנוסף, כאשר השחקן נפסל הוצאנו כפלט על המסך הודעה של "game over" לצד כמות ההקפצות שהשחקן הצליח להקפיץ וסיבת פסילתו

בתוספת סאונד של הפסד. יחד עם זאת מימשנו באותה תיקייה טבלה שבה השחקן מכניס את שמו המלא על מנת להיכנס לתחרות ובסוף המשחק הטבלה מתעדכנת עם מיקום השחקן בטבלה.

Game
Over

Your score is: 56

ball out of frame

זיהוי הכדור:

רקע ומוטיבציה:

חלק עיקרי במשחק ההקפצות הוא לזהות היכן נמצא הכדור בכל פריים, על מנת שנוכל לעקוב אחר תנועתו ולזהות כל הקפצה וכל פסילה.

קלט האלגוריתם – פריים ממצלמת הוידאו .

פלט האלגוריתם – שיעור נקודת מרכז המסה של הכדור ורדיוסו, בנוסף תחימת הכדור בכל פריים על ידי המעגל שהתקבל מנקודת מרכז המעגל והרדיוס.

מהלך האלגוריתם:

באלגוריתם זה השתמשנו במספר טכניקות של עיבוד תמונה:

- HSV – אלגוריתם לזיהוי צבע.
- פעולות מורפולוגיות.
- מציאת הקונטור של הכדור.

הקלט של האלגוריתם הוא פריים של הסרטון שמיוצג ב RGB.

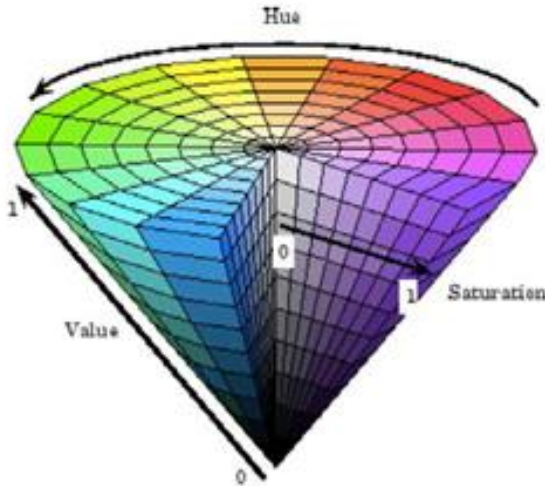


תחילה אנו ממירים את התמונה מייצוג RGB לייצוג HSV

HSV:

מרחב הצבעים HSV (נקרא לפעמים גם HSL) במודל זה כל צבע מיוצג על ידי גוון (Hue), רוויה (Saturation), ערך (Value).

זו בעצם השיטה הכי אינטואיטיבית לייצוג צבע.



- **גוון**- מייצג את הגוון של הצבע עצמו, הגוון מיוצג לעתים קרובות כזוויות בגלגל צבעים (הגוון האדום הוא בעצם ערך 0, צהוב זה בערך 60, ירוק זה 120, כחול זה 240, ואדום זה בחזרה 360).
- **רוויה**- מספר אשר מיוצג בין 0 ל 1 (בעצם דומה לערבוב צבע היסוד עם גועש לבן) כאשר 0 מייצג צבע לבן ו 1 מייצג צבע שלא עורבב בכלל עם לבן.

- **ערך**- מספר בין 0 ל 1 אומר בעצם עד כמה הצבע מואר או חשוך.

לאחר מכן אנו מבצעים 2 מסיכות שיסננו את הצבע האדום מהתמונה.

השתמשנו ב 2 מסיכות מהסיבות הבאות:

אובייקטים אדומים עשויים להופיע לעיתים בגוונים שונים עקב תנאי תאורה שונים. שימוש בשתי מסכות מבטיח זיהוי מיטבי בסיטואציות שונות, ובכך האלגוריתם הופך גמיש ויעיל יותר ומגדיל את האמינות של משימת זיהוי הצבע האדום.

לאחר פעולה זו התקבלה תמונה בינארית שמסמנת את האזורים האדומים בתמונה.



לאחר מכן אנו משתמשים במסכות מורפולוגיות על מנת לנקות רעשים ולהחליק את הכדור.

פירוט מלא על הפעולות המורפולוגיות יינתן בהמשך בחלק של אלגוריתם זיהוי הרצפה

והתקבלה התמונה הבאה אשר אמורה לייצג רק את הכדור:



מציאת קונטורים בתמונה:

קונטורים (קווי מתאר) - הם קווים רציפים או עקומות אשר מתארים את היקפם של האובייקטים בתמונה.

הפונקציה `cv2.findContours` משמשת לזיהוי קווים אלה בתמונות בינאריות, כאשר הפיקסלים הם 0 או 255. פונקציה זו שימושית במיוחד בזיהוי אובייקטים, ניתוח צורות וחילוץ תכונות.

תיאור דרך פעולה של cv2.findContours:

- האלגוריתם מקבל תמונה בינארית.
 - התמונה נסרקת מהצד השמאלי העליון לצד ימין תחתון של התמונה, פיקסל אחרי פיקסל עד שמגיעים לפיקסל לבן אשר מסמן הימצאותו של אובייקט.
 - האלגוריתם עוקב אחר גבול האובייקט על ידי הסתכלות על 8 השכנים המחוברים של הפיקסל (פיקסלים צמודים ישירות אופקית, אנכית או אלכסונית) וקביעה איזה שכן הוא חלק מקו המתאר.
 - מאחסן כל קו מתאר שזוהה בתמונה כרשימה של נקודות.
- ניתן לאחסן את כל קווי המתאר שזוהו בזיכרון, אך אחסון כל פיקסל בקו המתאר עשוי להיות עתיר זיכרון דבר אשר יכול להאט את האלגוריתם.
1. מכיוון שהמשחק שלנו הוא אינטראקטיבי, ריצת האלגוריתם בזמן אמת היא נושא קריטי, לכן אנו בחרנו בשיטת `cv2.CHAIN_APPROX_SIMPLE` אשר חוסכת בזיכרון על ידי הסרת כל הנקודות המיותרות ודחיסת קווי המתאר, ובכך מאחסנת רק נקודות קריטיות לזיהוי קו המתאר.
 2. כיוון שאנו רוצים לזהות את היקף הכדור בלבד, בחרנו בשיטת `RETR_EXTERNAL` – שיטה זו מחזירה רק את קווי המתאר החיצוניים ביותר. זה שימושי כאשר מעוניינים רק בצורה החיצונית של האובייקטים, תוך התעלמות מכל קווי מתאר שיכולים להימצא בתוך אותם אובייקטים. מצב זה יכול להפחית משמעותית את מספר קווי המתאר שנמצאו והוא מהיר יותר ביחס לשאר.
- לאחר שקיבלנו את כל הקונטורים אשר מופיעים בתמונה, אנו בוחרים בקונטור הגדול ביותר, כיוון שזה הקונטור שמייצג את הכדור(בהנחה שאין אובייקטים אדומים גדולים נוספים בתמונה).
- השלב הבא הוא מציאת המעגל הקטן ביותר אשר מכיל את כל קו המתאר המקסימלי (שזוהה קודם לכן) מכאן אנו מחלצים באופן חד חד ערכי את נקודת מרכז המעגל ואת רדיוסו של הכדור.
- נשים לב שבחרנו לסמן רק את הקונטור המקסימלי שרדיוסו גדול מערך סף מסוים אשר מצאנו על ידי כך שהרחקנו את הכדור למרחק המקסימלי מהמצלמה שניתן בגבולות המשחק ומצאנו את רדיוסו במקרה זה.
- סיבה נוספת לשימוש בתנאי היא על מנת לסנן מקרים בהם גוון עורו של השחקן נתפס כאדום מספיק כך שעלול להיחשב ככדור.
- כיוון שביחס לכדור כל עיגול על גוף השחקן הוא קטן לכן האלגוריתם יתעלם מהעיגולים שיימצאו על גוף השחקן.

לבסוף סימנו בכל פריים את המעגל שמצאנו אשר מייצג את מיקום הכדור בכל רגע נתון. יש לציין שיצרנו מערך positions שכל איבר במערך מכיל את נקודת מרכז המעגל ובכל פריים דגום הוספנו איבר חדש למערך. מערך זה ישמש אותנו למימוש אלגוריתם לזיהוי הקפצה כקלט האלגוריתם.



אלגוריתם לזיהוי הקפצת כדור:

רקע ומוטיבציה:

חלק עיקרי במשחק ההקפצות הוא לזהות הקפצת כדור תקינה, על מנת שנוכל לספור את כמות ההקפצות ולהציגן לאורך המשחק ובסופו.

קלט האלגוריתם – שלוש איברים סמוכים של מערך positions שמתאר את מרכז הכדור בכל פריים.

פלט האלגוריתם – זיהוי הקפצה ועדכון כמות ההקפצות הנוכחית.

מהלך האלגוריתם:

תחילה חישבנו את המרחק שעבר מרכז הכדור בין כל שני פריימים סמוכים

$$dx_1 = x_1 - x_0, dx_2 = x_2 - x_1$$

$$dy_1 = y_1 - y_0, dy_2 = y_2 - y_1$$

נשים לב שהמרחק שמעניין אותנו כרגע הוא המרחק בציר y שכן אם $dy_i, i = 1, 2$ חיובי אז הכדור בירידה, ולהיפך אם $dy_i, i = 1, 2$ שלילי הכדור נמצא בעלייה. הקפצה תהיה מצב שבו dy_1 יהיה חיובי ו dy_2 יהיה שלילי, לכן התנאי של זיהוי גרדיאנט מספיק בכיוון הנכון יהיה:

$$\begin{aligned} & dy_2 < -3 \quad \bullet \\ & dy_1 > 2 \quad \bullet \\ & dy_1 \cdot dy_2 < 0 \quad \bullet \end{aligned}$$

נשים לב שאת הערכים 2 ו 3- בחרנו לאחר ניסוי וטעיה של גרדיאנט מספיק שלא יהיה זיהוי שווא של הקפצה עקב תנועה קטנה מדי.

על מנת לזהות הקפצה נרצה שלא רק יהיה שינוי בגרדיאנט אלא גם שינוי חד במהירות של הכדור כלומר שהתאוצה שלו תהיה גבוהה מספיק לכן יצרנו מערך speeds שכל איבר בו הוא מהירות הכדור בכיוון x ובכיוון y . נשים לב שהזמן שעובר בין כל זוג פריימים הוא 1.

ולכן $dx_i = v_{x_i}$ ו $dy_i = v_{y_i}$ לכל i .

על מנת לחשב את התאוצה בכיוון x ובכיוון y לכל זוג פריימים נחסיר את המהירויות בין כל זוג פריימים בהתאמה $a_{x_i} = v_{x_{i+1}} - v_{x_i}$ ו $a_{y_i} = v_{y_{i+1}} - v_{y_i}$.

על מנת לחשב את התאוצה השקולה נבצע:

$$\hat{a} = \sqrt{a_{y_i}^2 + a_{x_i}^2}$$

את התאוצה השקולה שחישבנו נשווה לערך סף ובמידה וזה גדול מערך הסף נסיק שהתקיים התנאי השני להקפצה שאומר שבזמן הקפצה הכדור מוקפץ על ידי רגלו של השחקן כלומר תאוצתו גדולה.

לכן התנאי של זיהוי שינוי מהירות מספיק יהיה:

$$\bullet \hat{a} > 10$$

נשים לב שאת הערך 10 בחרנו לאחר ניסוי וטעיה של שינוי המהירות.

כאשר שני התנאים מתקיימים נסיק שקרתה הקפצה.

.if gradient_bounce_detected and velocity_bounce_detected

זיהוי פגיעה ברצפה:

רקע ומוטיבציה:

אחת המטרות שצוינה למעלה הייתה לזהות את הרצפה ולהפריד אותה מן השחקן, האתגר היה שכאשר הכדור קופץ על הרצפה, הקפיצה נספרת כהקפצה עקב אלסטיות כדור הכדורגל.

קלט האלגוריתם - שני פריימים עוקבים ממצלמת הוידאו.

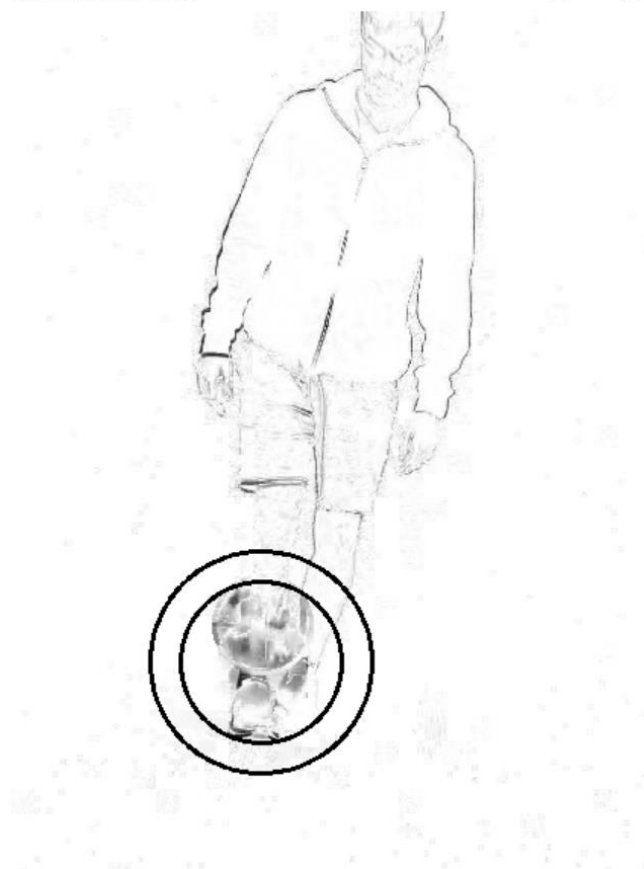
פלט האלגוריתם – תמונת הפרשים שעליה מסומנת טבעת סביב הכדור.

מהלך האלגוריתם:

ראשית, זיהינו את הרצפה ע"י שימוש בעובדה שהרצפה סטטית לאורך כל המשחק, ואילו השחקן והכדור בתנועה.

על מנת לבצע מטרה זו, בחרנו להשתמש בתמונות הפרשים כך שכאשר שני פיקסלים בעלי מיקום זהה בין שני פריימים עוקבים, יהיו בעלי ערך זהה והחיסור בין שני הפיקסלים יהיה אפס, ואילו מפני שהשחקן והכדור זזים ערכי הפיקסלים בין שני פריימים עוקבים יהיו שונים ולכן החיסור יהיה שונה מאפס.

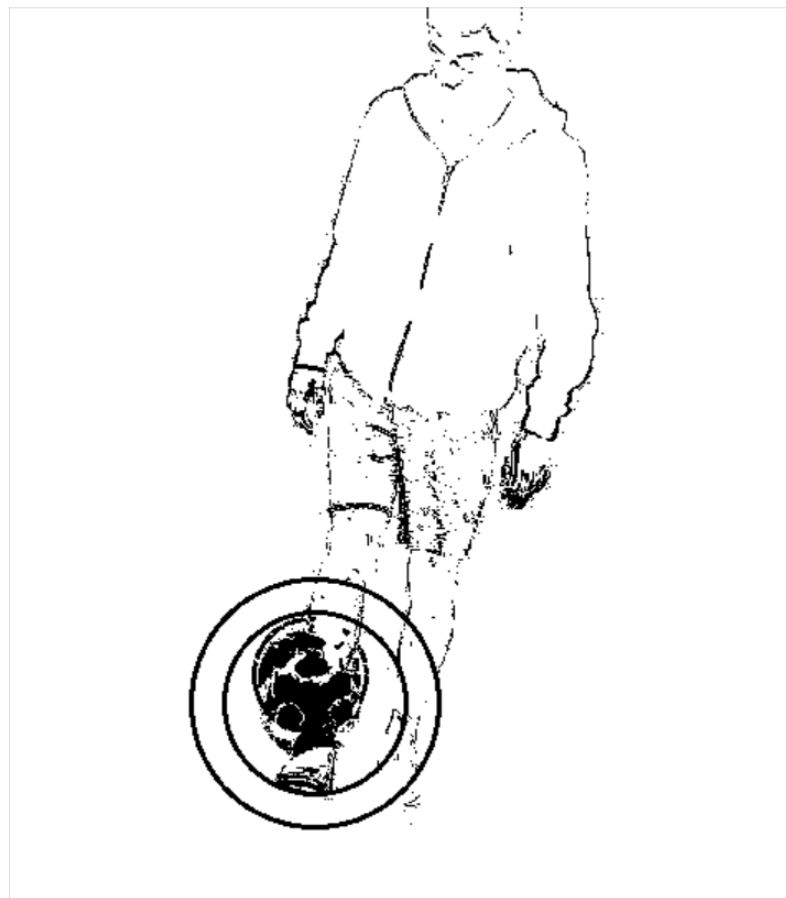
לאחר תמונת ההפרשים, נקבל אמנם הפרדה מסויימת של הרצפה מן השחקן- אולם נקבל תמונה רועשת מאוד שלא תאפשר לנו להחליט האם הכדור פגע ברצפה.



בתמונה ניתן לראות כי אכן החלקים הסטטיים בתמונה הם בגוונים לבנים יותר, ואילו החלקים שזזים בגוונים שחורים יותר, זאת כיוון שהפעלנו אופרטור **not** על תמונת ההפרשים לצורכי אסתטיקה ונוחות תצוגה.

יחד עם זאת, קיבלנו תמונה לא בינארית שמכילה ערכים אפורים, מה שמקשה על ביצוע האלגוריתם, ולכן דבר ראשון שעשינו היה להפוך את התמונה לבינארית באמצעות קביעת **threshold** מתאים כך שערכי פיקסלים שמתחת לסף יהפכו ללבנים באופן מוחלט וערכי פיקסלים שמעל הסף יהפכו לשחורים באופן מוחלט.

לאחר קביעת ערך הסף קיבלנו תמונה כזו:



כעת אכן קיבלנו תמונה בינארית, אך נשים לב שללא שימוש בשום פילטר נקבל רגישות גבוהה לשינויים בין פריימים, כך שאפילו שינויים קלים שלא דווקא נובעים מהתנועה עצמה אלא מגורמים כמו אור וצל- ישפיעו על תמונת ההפרשים.

בנוסף, קיבלנו תמונה עם הרבה חורים, מה שמקשה על הפרדה מיטבית בין השחקן והכדור לרצפה.

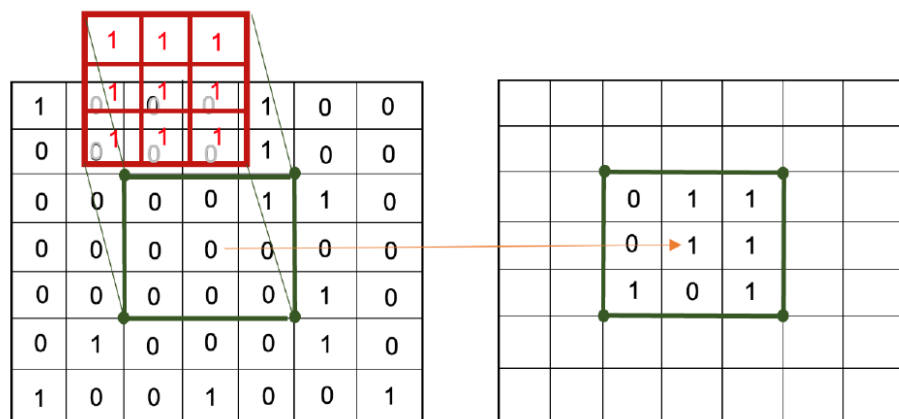
כדי להתמודד עם תוצאות אלו, הפעלנו על התמונה הבינארית פעולות מורפולוגיות של erosion, dilate ושילובם - close.

לצורך כך, תחילה הגדרנו חלון בגודל 9X9 של אחדות ובאמצעותו הפעלנו את הסינון.

Dilation - זוהי פעולת הרחבה.

ההרחבה מתבצעת על ידי קונבולוציה של התמונה עם גרעין, לרוב הגרעין הינו box-filter סטנדרטי.

אם לפחות פיקסל אחד מתחת לגרעין הינו שחור, אז הפיקסל בתמונה יהיה שחור גם כן, כמראה בחץ הכתום בתרשים.



את פעולת ההרחבה ניתן לעשות מספר פעמים וכך לקבל הרחבה משמעותית יותר כאשר מספר האיטרציות גדול יותר.

הפעלנו את פעולה זו למשך 3 איטרציות וקיבלנו את תוצאת הביניים הבאה:



נשים לב שבתמונה לעיל קשה לזהות את השחקן, בפרט את רגלי השחקן וגם את הכדור- שנראים כמקשה אחת.

יחד עם זאת, הצלחנו להעלים בצורה טובה (אך עדיין לא מושלמת) את רעשי הרקע שמסביב.

על מנת להתמודד עם בעיה זו, נוסיף גם את פעולת ה-erosion.

Erosion - זוהי פעולת צמצום. פעולה זו הפוכה לפעולת ה-dilation, כך שנסמן פיקסל כשחור רק אם כל הפיקסלים בגרעין שחורים גם כן.

Close - היא פעולה שמשלבת את פעולת ההרחבה (dilate) ופעולת הצמצום (erosion), כלומר, תחילה נרחיב את האזורים השחורים בתמונה ואחר כך נחזיר אותם לגודלם המקורי. פעולת closing מסייעת גם במילוי חורים.

לאחר פעולה זו נקבל את התוצאה הסופית והרצויה:



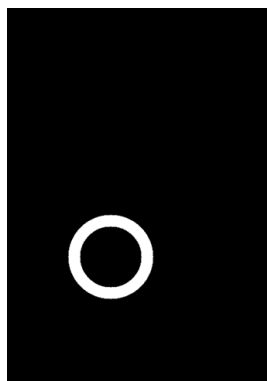
כפי שניתן לראות, לאחר כל הפעולות שהפעלנו, קיבלנו תמונת הפרשים אינפורמטיבית, שמפרידה לנו את שתי הרגליים. בתמונה זו אין הפרדה בין הכדור לרגל, אבל היא צולמה במהלך הקפצה ולכן זה צפוי ורצוי.

הגדרת הטבעת ושימושיה לצורך זיהוי הרצפה-

לאחר שקיבלנו תמונת הפרשים מספיק טובה שמפרידה בין האזורים הסטטיים בתמונה לבין האזורים הדינאמיים ללא השפעת אור וצל ותנועות חדות, נרצה ליישם באמצעותה את זיהוי נפילת הכדור על הרצפה.

לצורך כך, הגדרנו טבעת שרדיוסה הפנימי הוא רדיוס הכדור+מרחק ביטחון, ורדיוסה החיצוני גדול ב-20 יחידות מהרדיוס הפנימי. את המידע על מיקום הכדור אנו נקבל מהאלגוריתם הראשון שמזהה את הכדור ונותן מידע על מרכז הכדור ורדיוסו בכל פריים ומקיף אותו בעיגול. הסיבה שבחרנו לקחת מרחק ביטחון עבור הרדיוס הפנימי של הטבעת היא שהקפת הכדור זזה יחד עם הכדור בין הפריימים ובתמונת הפרשים התוצאה היא טבעת שחורה לא רצויה שפוגעת בדיוק האלגוריתם לזיהוי הפגיעה ברצפה.

לאחר שהגדרנו את הטבעת, נרצה לחשב את אחוז הפיקסלים השחורים בטבעת. הסיבה לבחירה בשיטה זו היא על מנת להפריד בין הקפצה אמיתית של השחקן להקפצה על הרצפה, כאשר אם השחקן שנע מקפיץ את הכדור אחוז הפיקסלים השחורים בטבעת יהיה יותר גדול בצורה משמעותית מהמצב שבו הכדור קפץ על הרצפה, מכיוון שבעת ההקפצה, הרגל הנעה של השחקן תיכנס לטבעת. (נזכיר- פיקסלים שחורים מייצגים את החלקים הנעים בתמונה ואילו פיקסלים לבנים מייצגים את החלקים הסטטיים).



פילטר הטבעת אשר מקיפה את הכדור
לאורך המשחק

חישוב אחוז הפיקסלים השחורים בטבעת התבצע באמצעות סכימת הפיקסלים השחורים בטבעת (פעולת bitwise and בין הפיקסלים השחורים של תמונת הפרשים לבין הפיקסלים הלבנים במסכת הטבעת- כלומר הטבעת עצמה), סכימת כל הפיקסלים בטבעת, וחלוקה בין הסכימות הללו. היינו צריכים להגדיר סף שמתחתיו הטבעת תיחשב "לבנה" ולאחר אנליזה ראינו שערך הסף המתאים הוא 25 אחוז.

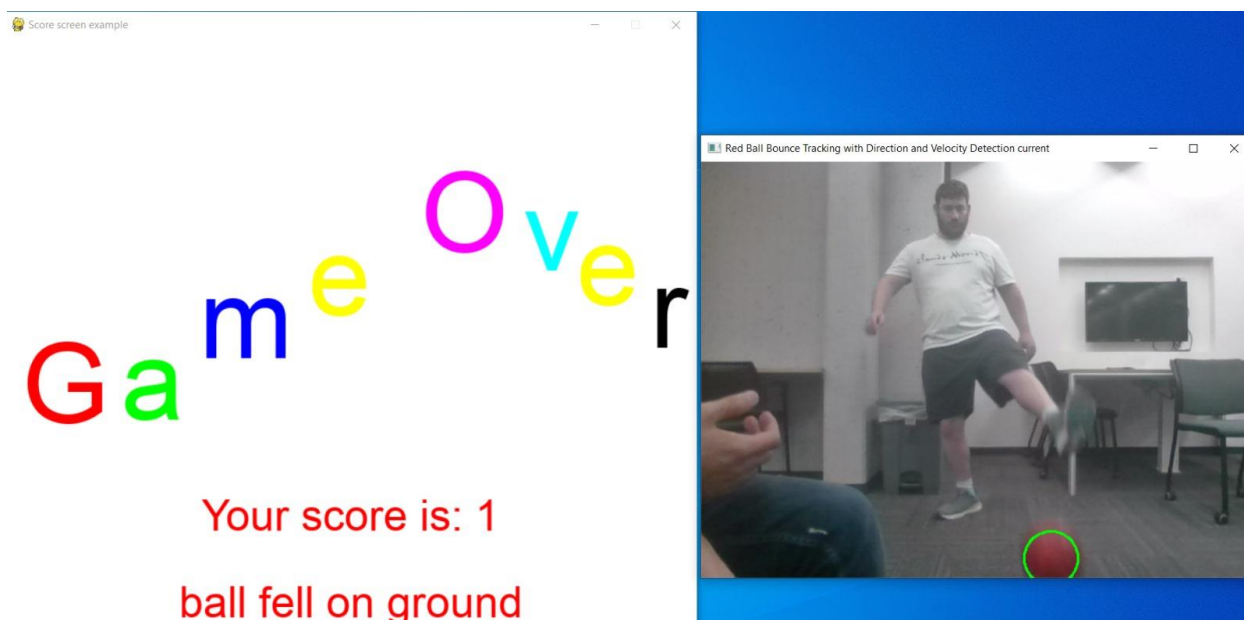
אולם, לצורך קביעת פסילה לא נוכל להשתמש רק במדד של הטבעת, מכיוון שפעמים רבות כאשר הכדור נמצא בתנועתו באוויר והרקע מאחוריו סטטי (לדוגמה כאשר השחקן משמאל או מימין לכדור).

לכן נדרוש שיתקיימו שלושה תנאים:

- התרחשה הקפצה, שכן הכדור תמיד קופץ לאחר פגיעתו ברצפה.
- הטבעת תיחשב כלבנה (אחוז הפיקסלים השחורים יהיה מתחת לסף).
- הכדור יימצא בחצי התחתון של התמונה.

רק אם כל שלושת התנאים הללו יתקיימו, נזהה את ההקפצה כפגיעה ברצפה ונפסול את השחקן.

לבסוף כשזוהתה נפילה על הרצפה עולה הודעת הפסילה וסיבתה, כמו כן מוצג גם מספר ההקפצות שנצברו עד כה. בנוסף לכך מוצג פריים הפסילה.



אלגוריתם זיהוי נגיעה ביד

רקע ומוטיבציה:

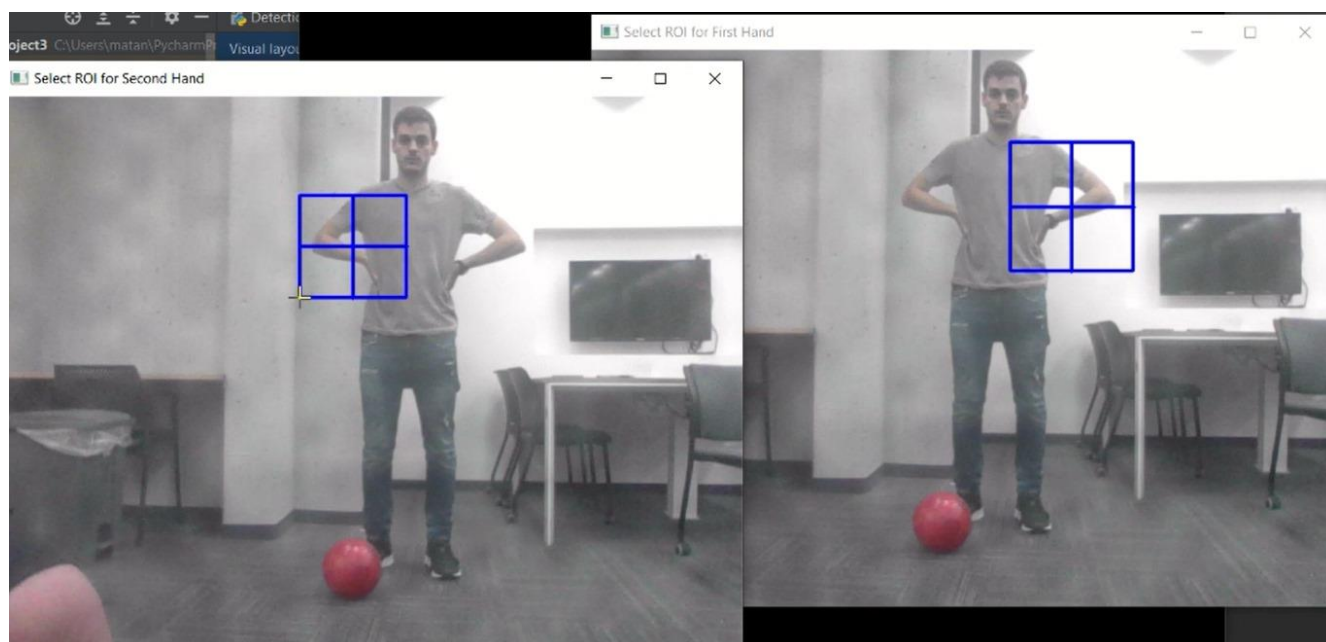
במשחק הכדורגל נגיעה של היד בכדור אינה חוקית לכן קבענו כי נגיעת היד בכדור במהלך המשחק תוביל לפסילה. שכן שימוש ביד במהלך משחק ההקפצות פוגם במטרת המשחק שהיא שיפור יכולות השחקן וספירה של ההקפצות.

קלט האלגוריתם – זיהוי הקפצה חיובית, שלוש איברים סמוכים של מערך positions שמתאר את מרכז הכדור בכל פריים, מיקומי ריבועי הידיים שסומנו בתחילת התהליך ומתעדכנים על ידי CSRT.

פלט האלגוריתם – הודעת פסילה לאור נגיעה ביד הספציפית (שמאל או ימין) וכמות ההקפצות החוקיות שנצברו עד כה.

מהלך האלגוריתם:

בתחילת המשחק מנהל המשחק מסמן את ידיו של השחקן בשתי מלבנים באופן ידני, כאשר יד שמאל מסומנת ראשונה ויד ימין מסומנת לאחריה. האלגוריתם מקבל את שמונת קודקודי המלבנים בהתאמה. לאחר מכן מתבצע מעקב אחרי ידי השחקן על ידי אלגוריתם CSRT.



: CSRT

אלגוריתם המעקב CSRT (Discriminative Correlation Filter with Channel and Spatial Reliability) הוא אלגוריתם מתקדם למעקב אחר אובייקטים המשתמש ביישומים של ראייה ממוחשבת כדי לעקוב אחרי תנועת אובייקטים ספציפיים בתוך הווידאו.

מושגים מרכזיים בCSRT

מסנני קורלציה: אלגוריתם CSRT משתמש במסנני קורלציה כדי לקבוע את מיקומו של אובייקט בתמונה. ה-Tracker משתמש במסננים אלו כדי ליצור מודל של האובייקט על סמך המראה שלו, אשר מתעדכן במהלך הזמן.

מבחינה מתמטית נגדיר $x(t)$ המייצג את הפיקסלים המשויכים לאובייקט שנמצא במלבן הראשוני המסומן בזמן t .

נסמן את פילטר הקורלציה כ- $h(t)$ והוא מחושב על ידי ביצוע התמרת פורייה של $x(t)$ והכפלה בצמוד שלו.

$$H(w) = \mathcal{F}[x(t)]\overline{\mathcal{F}[x(t)]}$$

ולאחר ביצוע התמרה הפוכה ניתן לקבל:

$$h(t) = \mathcal{F}^{-1}[H(w)]$$

על מנת למצוא את האזור בפריים החדש שתואם בצורה הטובה ביותר לאובייקט אנו מפעילים את פילטר הקורלציה $H(t)$ על הפריים החדש $y(t)$.

בצורה הבאה:

$$C(t) = \mathcal{F}^{-1}[H(w)\mathcal{F}[y(t)]]$$

מעקב בהתבסס על ערוצים: אלגוריתם CSRT משתמש במספר ערוצי תמונה (כגון צבע, גרדיאנטים) כדי לעקוב אחר האובייקט. כל ערוץ מספק פרספקטיבה ייחודית על תכונות האובייקט (פיצ'רים), והשימוש במספר ערוצים משפר את יכולת העקיבה אחר האובייקט הרצוי, ניתן לעקוב אחר אובייקטים שונים במצבים שונים ומגוונים לבחירתנו.

אמינות בערוץ: לא כל הערוצים תורמים באותה מידה למעקב יעיל. CSRT מעריך את האמינות של כל ערוץ ונותן לכל ערוץ ציון. הערוצים עם האמינות הגבוהה כלומר בעלי ציון גבוה יותר ישפיעו יותר על תהליך המעקב.

מבחינה מתמטית נגדיר את $c_i(t)$ כתגובת הקורולציה עבור הערוץ i בזמן t . ניקוד האמינות $s_i(t)$ לכל ערוץ מחושב על ידי תגובת הקורולציה

$$s_i(t) = \frac{1}{N} \sum_{j=1}^N c_i(t, j)$$

כאשר N זהו מספר הפיקסלים בתוך המלבן התוחם.

האמינות הערוץ $r_i(t)$ עבור ערוץ i נקבעת על בסיס הניקוד $s_i(t)$ ועל גבי ניקוד סך הערוצים:

$$r_i(t) = \frac{s_i(t)}{\sum_{j=1}^M s_i(t, j)}$$

כאשר M זהו מספר הערוצים הכולל.

אמינות מרחבית: אלגוריתם CSRT מתייחס גם לאזורים מרחביים של האובייקט. חלקים מסוימים יכולים להיות יותר אמינים למעקב מאחרים בגלל יציבות הרקע, ייחודיות האזור או גורמים אחרים. רכיב זה מסייע ל-CSRT לשמור על פוקוס במעקב אחר חלקי האובייקט המרכזיים, גם אם אזורים אחרים מוסתרים או מטושטשים.

מבחינה מתמטית נגדיר את $P_{ij}(t)$ אשר מגדיר את ניקוד האזור המרחבי j בערוץ i בזמן t . נגדיר את $Q(t)$ להיות מדד האמינות המרחבית הכולל והוא נקבע על ידי ניקוד האזורים המרחביים השונים.

$$Q(t) = \sum_{i=1}^M \sum_{j=1}^K P_{ij}(t)$$

כאשר K מתאר את מספר האזורים המרחביים על גבי כל הערוצים.

כיצד CSRT עובד:

אתחול: בתחילת המעקב, CSRT מקבל אזור אותו אנו מגדירים כמלבן מסביב לאובייקט למעקב. מלבן זה מסומן על גבי הפריים הראשון של הסרטון.

יצירת מסנן הקורלציה: CSRT יוצר מסנן קורלציה על סמך האובייקט בתוך המלבן המסומן. מסנן זה משמש לזיהוי ומעקב אחר האובייקט בתמונות הבאות.

מעקב בין פריימים:

CSRT מחליט את מיקום האובייקט הטוב ביותר בפריים החדש בעזרת מסנן הקורלציה. הוא משתמש במספר ערוצי תמונה כדי לשפר את הדיוק, ומתחשב בשינויים בצבע, בטקסטורה או בגרדיאנט.

Tracker משייך ציון אמינות לכל ערוץ ולכל אזור מרחבי בתוך האובייקט. זה עוזר ל-CSRT להתאים את המיקוד על החלקים הכי אמינים.

עדכון המודל: ככל שהאובייקט נע והרקע של האובייקט שבתוך הריבוע משתנה, CSRT מעדכן את מודל המסנן קורלציה שלו כך שישקף את השינויים הללו. זה מבטיח ש**Tracker** יתאים לשינויי מיקום האובייקט.

פלט של אזור המעקב המתעדכן: לאחר העיבוד של התמונה, CSRT מחזיר מלבן עדכני שמציין את המיקום המעודכן של האובייקט. מלבן זה ישמש להמשך המעקב בתמונות הבאות.

יתרונותיו של אלגוריתם ה-CSRT:

גמישות: השילוב של שימוש בערוצים אמינים שונים ובאמינות באזורים מרחביים שונים מספק יכולת התמודדות עם מצבי קיצון שונים כגון: שינויי תאורה, טשטוש וראות חלקית של האובייקט (קשיים שאיתם התמודדנו בפרוייקט).

הסתגלות: ה-CSRT מתאים את עצמו למראה המשתנה של האובייקט לאורך הזמן, מאפשר מעקב רציף גם כאשר התנאים משתנים.

לסיכום, CSRT הוא **Tracker** גמיש וחזק המתאים למגוון תרחישים של מעקב אחר אובייקט.

התלבטות לשימוש באלגוריתם המעקב

נציין כי התלבטנו בין שני אלגוריתמים למעקב אחר תנועות הידיים בין הפריימים בוידאו והם : optical flow (Lucas-Kanade) או CSRT tracker .

האלגוריתם שאיתו בחרנו לממש את המשימה הוא CSRT tracker מכמה סיבות:

- התמודדות יותר טובה עם שינויים חדים של תנועת האובייקט הנעקב עקב שימוש בערוצים שונים ובאמינות המרחבית.
- עמידות בפני רקעים משתנים ומורכבים.
- עקיבה איכותית גם כאשר ישנה חסימה חלקית של האובייקט הנעקב.

זיהוי פסילה עקב נגיעה ביד

באופן סימולטני למעקב הידיים , האלגוריתם משתלב יחד עם ההאלגוריתם לזיהוי הקפצה. כאשר מתקיים התנאי על קיום הקפצה נכנסים אל האלגוריתם לבדיקה של חוקיותה.

תחילה נבדוק האם קיים בהקפצה הנוכחית זיהוי של הכדור, לאחר מכן נבדק האם מיקומי מרכז הכדור הם בתוך מלבן היד השמאלית.

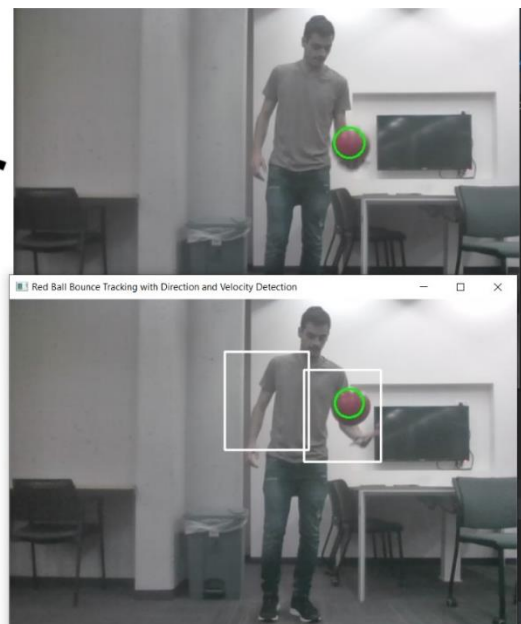
$$((p1[0] \leq x \leq p2[0]) \text{ and } (p1[1] \leq y \leq p2[1]))$$

ברגע שזוהתה הקפצה בתוך מלבן היד אנו מסיקים כי התבצעה נגיעה ביד השמאלית. פוסלים את השחקן ומציגים לו את מספר ההקפצות שלו.

Game Over

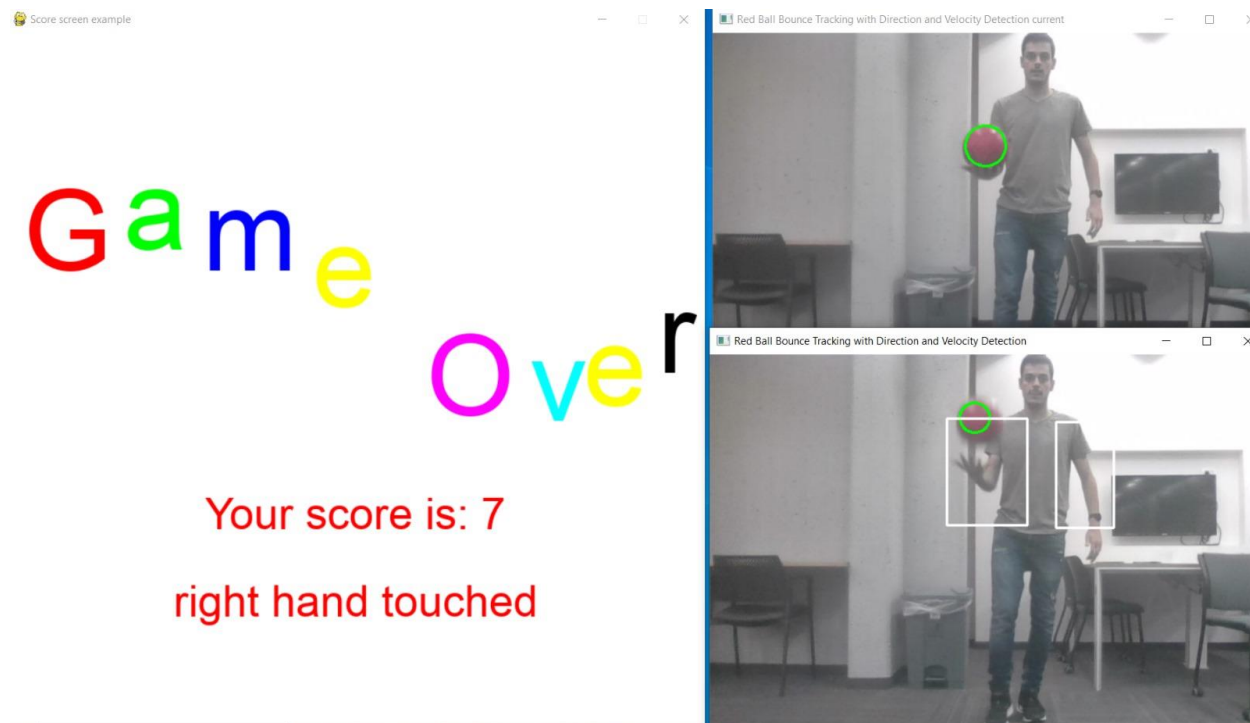
Your score is: 1

left hand touched



אותה בדיקה מתבצעת עבור היד הימנית לאחר מכן.

$$((p3[0] \leq x \leq p4[0]) \text{ and } (p3[1] \leq y \leq p4[1]))$$



תצוגת הטבלה הסופית:

לבסוף נציג את טבלת המקפיצים כך שנראה את שם ושם משפחתו של המקפיץ ואת מספר ההקפצות אותן השיג. בסיום המאורע (יום הפרויקטים) נכריז על המנצח הסופי.

DataFrame Viewer			
	First Name	Last Nam	Score
1	matan gabi	gabai	56

אנליזת התוצאות- בחינת ביצועי אלגוריתם זיהוי הרצפה

הקדמה לניסויים: כדי לאמוד את ביצועי אלגוריתם זיהוי הרצפה שלנו, בחנו מספר קטעי וידיאו של מהלך המשחק ב"תנאים קיצוניים", ובחנו את השפעת התנאים הקיצוניים על זיהוי הרצפה.

התנאים הקיצוניים כוללים:

- א. נפילת הכדור במרחק קטן מאוד מרגלי השחקן כך שבפריים נראה כאילו השחקן הקפיץ את הכדור אך בפועל הכדור נפל על הרצפה, ביחס למצב בו הכדור נפל במרחק מספיק גדול מרגלי השחקן כך שהאלגוריתם לא לוקח בחשבון את השחקן.
- ב. נפילת הכדור על הרצפה מלפני השחקן כך שבפריים השחקן משפיע על האלגוריתם (השחקן גורם לאלגוריתם הפרשים עקב תזוזתו בתוך הטבעת אותה הגדרנו קודם לכן בפירוט האלגוריתם) ביחס למצב בו הכדור נופל לצד השחקן כך שרק כף רגלו של השחקן עלולה להשפיע על האלגוריתם בכך שכף רגלו של השחקן נכנסת לטבעת המוגדרת.
- תנאים אלה עשויים להשפיע על ביצועי האלגוריתם ועל היכולת שלנו לזהות בצורה מהימנה את הנפילה של הכדור על הרצפה.
- על מנת לבחון את איכות ה-threshold שקבענו ערכנו ארבעה ניסויים, שכל ניסוי מייצג את אחד ממקרי הקצה הנ"ל.
- ניסוי ראשון:** בדקנו את נכונות סיווג האלגוריתם עבור המקרה בו הכדור נופל קרוב לרגלי השחקן ומלפניו.
- ניסוי שני:** בדקנו את נכונות סיווג האלגוריתם עבור המקרה בו הכדור נופל קרוב מרגלי השחקן ולצידו.
- ניסוי שלישי:** בדקנו את נכונות סיווג האלגוריתם עבור המקרה בו הכדור נופל רחוק מרגלי השחקן ומלפניו.
- ניסוי רביעי:** בדקנו את נכונות סיווג האלגוריתם עבור המקרה בו הכדור נופל רחוק מרגלי השחקן ולצידו.
- כל ניסוי ביצענו שלוש פעמים עבור thresholds פוטנציאליים שונים (15,25,35), על מנת לבחור את ה-threshold האידיאלי.

השערת הניסויים: לפי הצורה בה מימשנו את אלגוריתם זיהוי הרצפה, ההערכה שלנו היא שהביצועים עבור נפילת הכדור במרחק קטן מרגלי השחקן או בדיוק מלפנים לשחקן יהיו פחות טובים מאשר הביצועים עבור נפילת הכדור במרחק רב מרגלי השחקן או לצד השחקן, כלומר אחוז הפיקסלים השחורים בטבעת (המייצגים את האזורים הזזים בפריים) יהיו גדולים יותר. זאת כמובן מכיוון שזיהוי נפילה על הרצפה מתבסס על תמונת ההפרשים ועל ידי הפרדה בין חלקים נייחים בתמונה לחלקים זזים בתמונה. לכן ככל שיהיה יותר "מרווח ביטחון" של הכדור מחלקים זזים בתמונה, נזהה בצורה מדויקת יותר את נפילת הכדור על הרצפה.

השערה נוספת שלנו הייתה שה-threshold האידיאלי מבין שלושת האפשרויות יהיה האמצעי מבניהם(25).

דרך פעולה: עבור כל אחד מהניסויים נשתמש ב-confusion matrix, כפי שלמדנו בהרצאות הקורס.

עבור כל ניסוי, יצרנו מאגר של סרטונים הרלוונטי לאותו ניסוי, ובדקנו את אחוזי הדיוק בהתאם.

נשים לב שעבור נפילה אמיתית על הרצפה ישנן שתי אפשרויות:

- **TP** - זיהוי נכון של ההקפצה כפגיעה ברצפה
- **FN** - זיהוי כוזב של ההקפצה כהקפצה חוקית, למרות שבפועל הייתה נפילה על הרצפה.

באופן דומה, עבור הקפצה חוקית ישנן שתי אפשרויות:

- **TN** - זיהוי נכון של ההקפצה כהקפצה חוקית.
- **FP** - זיהוי כוזב של ההקפצה כפגיעה ברצפה, למרות שבפועל הייתה הקפצה חוקית.

עבור כל אפשרות כזאת מדדנו את המספר היחסי שהמאורעות התרחשו, ביחס למספר הניסיונות, או במילים אחרות- חישבנו את אחוז השגיאות עבור כל שגיאה (FP, FN) ואת אחוז ההצלחות עבור כל הצלחה (TP, TN).

את אחוזי הדיוק חישבנו לפי המשוואה: $ACC = \frac{TP+TN}{TP+TN+FP+FN}$

תוצאות הניסוי:

עבור כל ניסוי יצרנו 25 סרטונים בהם הכדור נפל על הרצפה ו-25 סרטונים בהם הכדור הוקפץ על ידי השחקן.

ניסוי ראשון – נפילת הכדור קרוב לרגלי השחקן ומלפניו

עבור threshold=15:

TP	TN	FP	FN
14	25	0	11

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{14 + 25}{50} = 78\%$$

עבור threshold=25:

TP	TN	FP	FN
17	24	1	8

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{17 + 24}{50} = 82\%$$

עבור threshold=35:

TP	TN	FP	FN
20	20	5	5

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{20 + 20}{50} = 80\%$$

ניסוי שני – נפילת הכדור קרוב לרגלי השחקן ולצידו

עבור threshold=15:

TP	TN	FP	FN
19	24	1	6

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{19 + 24}{50} = 86\%$$

עבור threshold=25:

TP	TN	FP	FN
21	23	2	4

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{21 + 23}{50} = 88\%$$

עבור threshold=35:

TP	TN	FP	FN
21	22	3	4

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{21 + 22}{50} = 86\%$$

ניסוי שלישי – נפילת הכדור רחוק מרגלי השחקן ומלפניו

עבור threshold=15:

FN	FP	TN	TP
5	1	24	20

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{20 + 24}{50} = 88\%$$

עבור threshold=25:

FN	FP	TN	TP
3	1	24	22

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{22 + 24}{50} = 92\%$$

עבור threshold=35:

FN	FP	TN	TP
3	2	23	22

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{22 + 23}{50} = 90\%$$

ניסוי רביעי – נפילת הכדור רחוק מרגלי השחקן ולצידו

עבור threshold=15:

FN	FP	TN	TP
2	1	24	23

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{23 + 24}{50} = 94\%$$

עבור threshold=25:

FN	FP	TN	TP
0	1	24	25

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{25 + 24}{50} = 98\%$$

עבור threshold=35:

FN	FP	TN	TP
0	2	23	25

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{25 + 23}{50} = 96\%$$

מסקנות:

כפי ששיערנו, אכן ה-threshold האידיאלי הוא 25, מכיוון שכפי שניתן לראות הוא משיג את ה-ACC הגבוה ביותר בכל אחד מהניסויים, ה-ACC שהתקבל היה מספיק גבוה בכל אחד מהניסויים, ולכן זה ה-threshold הנבחר.

כמו כן ניתן לראות שכפי ששיערנו, הניסוי שהשיג את ה-ACC הגבוה ביותר הוא הניסוי הרביעי בו ברגע קפיצת הכדור על הרצפה, הכדור היה רחוק מהשחקן ולצידו. הניסוי שהשיג את ה-ACC הנמוך ביותר הוא הניסוי הראשון בו ברגע קפיצת הכדור על הרצפה, הכדור היה קרוב לשחקן ומלפניו.

יחד עם זאת, בפועל, מקרה בו כדור נופל גם קרוב מדי לשחקן וגם מלפניו זה מקרה קצה נדיר, מכיוון שאם הכדור נופל ליד הרגל של השחקן, רוב הסיכויים שהוא גם יצליח להקפיץ אותו.

לכן נוכל להגיד שסך הכל ביצועי האלגוריתם שלנו הם טובים עבור הבחירה של $\text{threshold}=25$.

סיכום:

הפרויקט שבחרנו טמן בתוכו קשיים מגוונים איתם התמודדנו באמצעות הכלים השונים שלמדנו בקורס ובאופן פרטי. הפרוייקט דרש למידה עצמאית, מאמץ וזמן רב.

המערכת המאפשרת שימוש נוח ואמין של משתמשי משחק מלך ההקפצות, אשר מספקת בזמן אמת, מידע מיטבי על כמות ההקפצות של השחקן כך שלא יצטרך להתעסק במטלת הספירה המבלבלת תוך כדי המשחק שעלולה לפגום בהישגיו, תוך בדיקה של הקפצות חוקיות, ללא מעבר על חוקי המשחק עקב נגיעת יד של השחקן בכדור או נפילתו של הכדור על הרצפה.

האלגוריתם הינו שילוב של שלושה אלגוריתמים עיקריים:

1. זיהוי הכדור ובפרט זיהוי הקפצה של כדור

2. זיהוי נפילת הכדור על הרצפה

3. מעקב אחר תנועת הידיים של השחקן ופסילתו עקב נגיעה ביד

מצאנו שאלגוריתם זיהוי הכדור עובד מצויין והצלחנו לזהות את הכדור בכמעט כל הפריימים

בנוסף הצלחנו לזהות הקפצה של כדור בצורה מצוינת ללא חזרתיות של ספירה עבור הקפצה בודדת או ספירת הקפצה מיותרת

עבור אלגוריתם זיהוי הרצפה הצלחנו להפריד בצורה מאוד טובה את הרצפה מהשחקן והכדור על ידי תמונות הפרשים ויצרנו אלגוריתם שאותו גם ניתחנו למעלה במצבים שונים שמוגדרים כמצבי קיצון שאיתם האלגוריתם התמודד בזמן אמת עבור זיהוי נפילת הכדור כל הרצפה

אלגוריתם מעקב הידיים עבד בצורה טובה מאוד ובמרבית הפריימים הידיים של השחקן היו מסומנות במלבן המוגדר למעקב ובאופן ספציפי האלגוריתם שהשתמשנו בו מיועד למצבי קיצון של תנועה חדה של ידי שחקן ושינויים של מיקומי היד וזוויותיה

הפרויקט הביא איתו התמודדות עם אתגרים שונים

העבודה על הפרוייקט הייתה מאתגרת בלשון המעטה.בפן המקצועי בכך שהיינו צריכים ללמוד ולנסות אלגוריתמים שונים שנלמדו בכיתה וכאלו שלמדנו באופן פרטי על מנת להצליח לבצע את המשימה.בפן החברתי של התמודדות עם עבודה בצוות של 4 וקביעת לוחות זמנים. מאידך העבודה הייתה מספקת ומהנה ונתנה לנו תחושת מסוגלות והישגיות רבה כמהנדסים צעירים לעתיד שיכולים לספק מערכת שלמה אמינה,עובדת ובעלת ערך עבור משתמשים רבים.

ביבליוגרפיה:

אתר שמסביר על מציאת קונטורים-

<https://www.opencvhelp.org/tutorials/image-analysis/finding-drawing-contours/>

-Lecture-3 HSV

https://moodle.bgu.ac.il/moodle/pluginfile.php/4325488/mod_resource/content/1/Class3-Pyramids-Color-Transformation.pdf

פעולות מורפולוגיות-

<https://www.geeksforgeeks.org/python-opencv-morphological-operations/>

הסבר על CSRT-

<https://medium.com/@khwabkalra1/object-tracking-2fe4127e58bf>

תנאים פיזיקליים

[חישוב מהירות הכדור](#)

[חישוב תאוצת הכדור](#)

תמונת הפרשים

https://en.wikipedia.org/wiki/Image_subtraction#:~:text=Image%20subtraction%20or%20pixel%20subtraction%20or%20difference%20imaging,and%20a%20new%20image%20generated%20from%20the%20result

confusion matrix

https://moodle.bgu.ac.il/moodle/pluginfile.php/4351020/mod_resource/content/1/Class7-hough-descriptors.pdf