אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

קורס מבוא לעיבוד ספרתי של תמונות

פרויקט סופי

שם הפרויקט:

Hole In The Wall

מגישים:

עדי מימון	206924979
ליאור עבדייב	314077033
עדן בלדב	315360479
איילה ראובן	314077033

מרץ 2023

תוכן עניינים

1. תקציר הפרויקט

תמונה 1

Hole in the Wall (בעברית: "ראש בקיר") הוא שעשועון טלוויזיוני יפני בו מטרת המשחק היא לעבור דרך קיר בו קיים חור בצורה כלשהי מבלי לגעת בקיר. במהלך המשחק השחקן עומד על שפת בריכה עם הפנים לכיוון קיר שמתקרב אליו במהירות, בקיר יש פתח בצורה מסוימת ועל השחקן להתאים את גופו לפי הצורה שמגיעה כך שיוכל לעבור את הקיר ובכך להימנע מנפילה לבריכה.

בפרויקט שלנו בחרנו להביא את המשחק מהטלוויזיה לסלון. כעת השחקן יעמוד מול מצלמה ומול מסך בו הוא רואה את עצמו וקיר עם חור בצורה כלשהי ועליו להתאים את עצמו לצורה שהוא רואה על המסך. מטרת המשחק היא לעבור כמה שיותר קירות שונים בלי לפגוע בקיר עצמו ולצבור כמה שיותר נקודות.

תמונה 2

את המשחק ניתן לשחק בשתי דרכים - שחקן יחיד או שני שחקנים. בכל דרך קיימות שלוש דרגות קושי. קלה , בינונית או קשה. במהלך כל שלב במשחק יופיע קיר עם חור בצורה כלשהי. הקיר ילך ויגדל וכאשר הקיר יהיה בגודל המסך השחקן יצטרך לעמוד בתוך הצורה. אם השחקן יעמוד בתוך הצורה הוא יצבור נקודות ויופיע הקיר הבא. ככל שהשחקן יתפוס יותר שטח מהצורה כך הוא יקבל יותר נקודות. בנוסף אם השחקן יעבור בהצלחה מספר קירות ברצף הקיר יתקדם מהר יותר ורמת הקושי של המשחק תעלה. אם השחקן יחרוג מהצורה או לא יעמוד בתוך הצורה הוא יפסל ולאחר שלוש פסילות יסתיים המשחק.

המשחק מבוסס על אלגוריתמי עיבוד תמונה ספרתיים לזיהוי השחקן/השחקנים בהינתן רקע נתון. האלגוריתמים כוללים חיסור רקע, Thresholds, מניפולציות במרחבי צבע שונים, Histogram Equalization, מציאת קונטורים ומעברים דרך פילטרים שונים. לשם התאמת המשחק לכל שחקן ולרוב הסביבות ייצרנו אלגוריתם רובוסטי עם מינימום אילוצים.

2. הקדמה

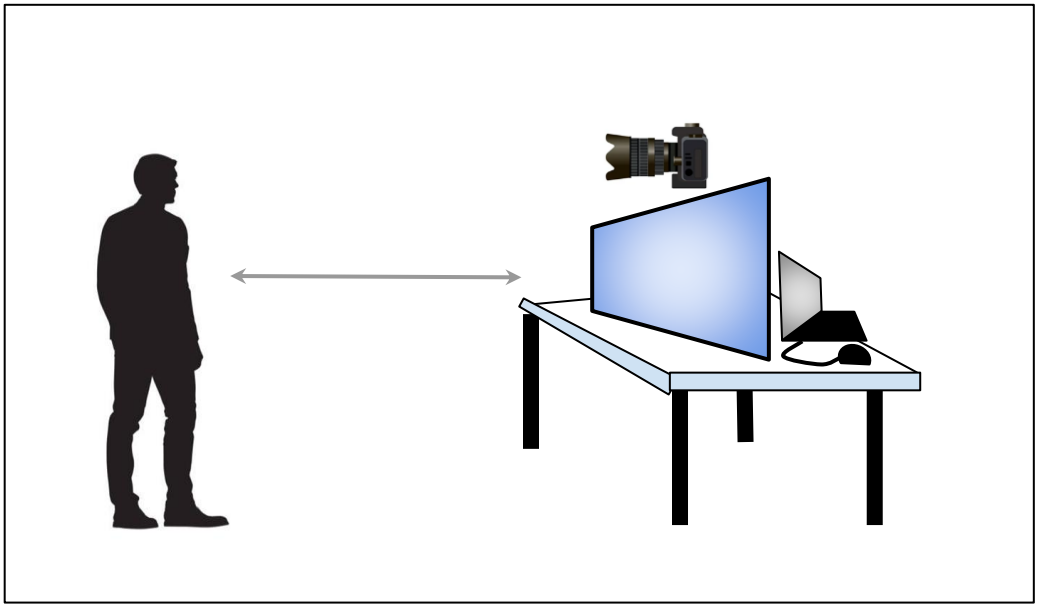
2.1. תיאור

מטרת פרויקט זה היא לדמות את המשחק "Hole in The Wall" בעזרת מצלמה בלבד בשימוש אלגוריתמים לעיבוד ספרתי של תמונה, כך שיהיה ניתן לשחק במשחק בכל מקום, לבד או עם חברים. נרצה שהמשחק ידמה באופן הטוב ביותר והטבעי ביותר את המחשק המקורי, לשם כך הוספנו אנימציה ואודיו ליצירת חוויה תלת-ממדית. המשחק יספק חווית משתמש מלאה החל מהצגת תפריט בחירה ראשי, שמירת שם השחקן/הקבוצה והתוצאה שקיבלו בלוח תוצאות, הצגת הניקוד במקרה של הצלחה או כשלון וסימון האזורים העיקריים בהם השחקן פגע בקיר במקרה של הפסד. בנוסף תפעול המשחק עצמו כמו בחירת מספר השחקנית ודרגת הקושי ועוד, גם כן מבוסס על עיבוד תמונה קלאסי לזיהוי בחירת כפתור מסוים, כך שהשחקן ממוקם מול המצלמה כבר משלב הפעלת המשחק ולא יידרש ממנו לגשת למחשב לשם ביצוע פעולות.

המשחק עצמו מאפשר אופציות שונות לביצוע המשחק: מספר שחקנים יחיד או זוג ודרגת קושי – קל, בינוי וקשה. דרגת הקושי משלבת צורות קשות יותר ומהירות ההגעה של הקיר. בנוסף המשחק לומד את השחקן במהלך המשחק ובמידה של הצלחה רצופה יעלה את רמת הקושי.

2.2. מהלך המשחק

מערך המשחק כפי שמתואר באיור 1, מבוסס על מצלמה ומסך המופנים כלפי השחקן/שחקנים. בהפעלת המשחק מהמחשב המחובר למסך ולמצלמה המסך הראשי (ראה תמונה 3) של המשחק יופיע על המסך שמול השחקן/שחקנים וברקע של המסך הראשי (ראה תמונה 4) מופיע הוידאו מהמצלמה כך שהשחקן/שחקנים יודעים איך למקם את עצמם אל מול המצלמה.



איור 1

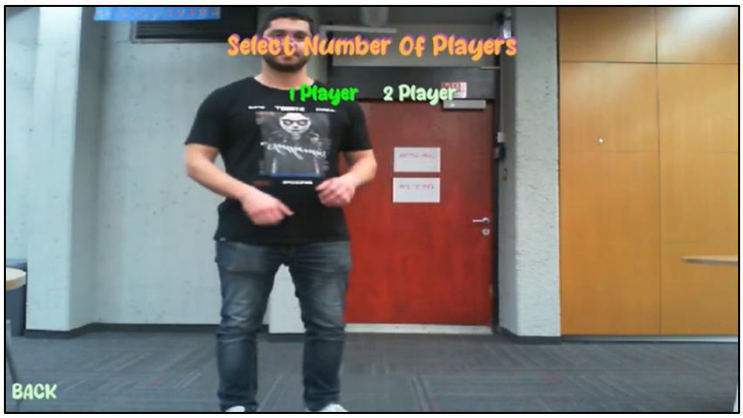
במסך הראשי יש 4 אופציות בחירה: התחלת משחק "play", הצגת לוח התוצאות "score board", הצגת הסרטון "trailer" ויציאה מהמשחק "quit". לחצן "take photo" המופיע למטה הינו לשימוש המפעיל של המשחק ומטרתו לקחת תמונת כיול (מבוצע באופן חד-פעמי) של הרקע ללא השחקן/שחקנים כך שבעצם לא נדרשות הנחות מיוחדות על הרקע במשחק (פירוט יינתן המשך).

תמונה 3

כפי שציינו קודם המשחק מספק יכולת שליטה במשחק באמצעות זיהוי האדם וסימון כפתור מסוים על ידי הרמת יד לאזור הכפתור הרצוי (ראה תמונה 4). תינתן אינדיקציה של בחירת כפתור מסוים על ידי שינוי צבע הכפתור המסומן ובחירה בכפתור תזוהה ע"י צליל לחיצה.

תמונה 4

בבחירת התחלת משחק "play", יופיעו האופציות לבחירת מספר שחקנים (תמונה 5), הכנסת שם הקבוצה (תמונה 6 - השלב היחידי במשחק שנדרש הזנה ידנית במחשב), בחירת דרגת קושי של המשחק (תמונה 8) וכללי המשחק (תמונה 7).



תמונה 7

תמונה 8

תמונה 6

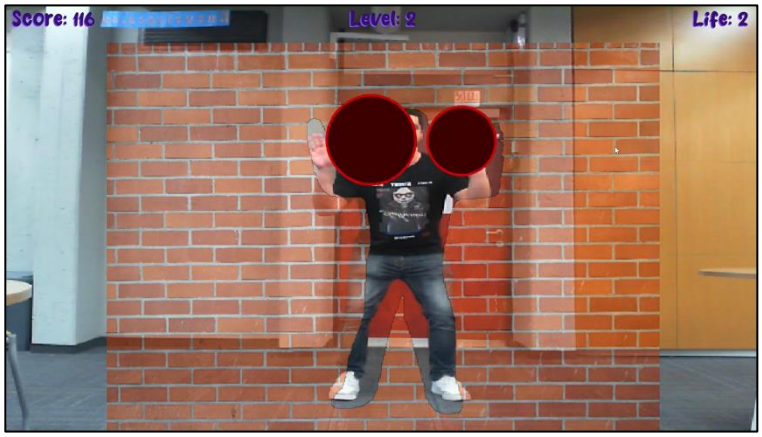
תמונה 5

תמונה 9

לאחר מכן יופיעו מסגרות המסמנות את מיקום השחקן על מנת שיתאים בגודל לצורות שיופיעו במהלך המשחק. ברגע שהמשחק מזהה את השחקן/שחקנים במיקום הנכון המשחק יציג ספירה לאחור (תמונה 9) ויתחיל באופן אוטומטי.

במהלך המשחק יופיעו צורות שונות על קיר הלבנים המתקדם לעבר השחקן (ראה תמונה 12), ברגע שהקיר מגיע לסוף (לקצה המסך) נלקחת תמונה מהמצלמה שעוברת למתודה המזהה את השחקן/שחקנים ומפרידה אותם מהרקע, עם המידע הזה בשילוב הצורה הנוכחית של הקיר מתודה אחרת מחשבת האם השחקן עבר את הצורה ללא פגיעה בקיר ומוציאה את אחוז ההצלחה ואחוז הכשלון (כמה מהשחקן פגע בקיר וכמה עבר את הצורה בהצלחה) ומהנתונים האלו מחושב הציון שיינתן לשחקן/שחקנים עבור הצורה הנוכחית ואלה יוצגו על המסך. בנוסף לכך, במידה והשחקן לא עבר באופן מלא את הקיר יסומנו במעגלים אדומים המקומות בהם פגע בקיר (ראה תמונה 11 – לפני הסימון, ותמונה 10 – לאחר הסימון).

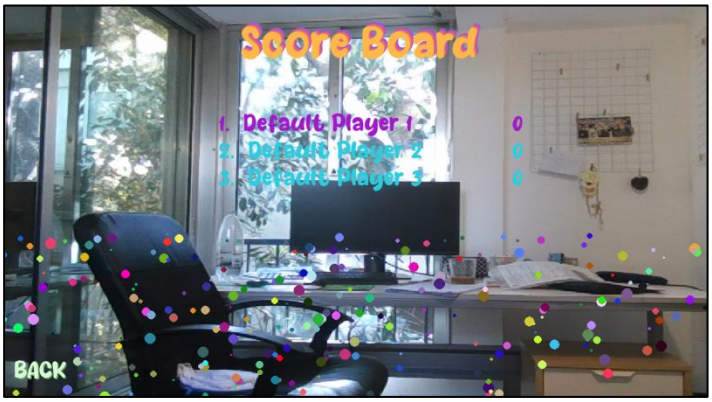
תמונה 10



תמונה 11

תמונה 12

לאחר שהמשחק נגמר (השחקן עבר את כל הקירות עם לכל היותר 2 פסילות או שהשחקן נפסל 3 פעמים) יופיע הניקוד של השחקן (תמונה 13) ולאחר מכן לוח התוצאות (תמונה 14) המציג את עשרת השחקנים עם הניקוד הגבוה ביותר.



תמונה 13

תמונה 14

2.3. אתגרים

קיימים אתגרים רבים לפרויקט שבין היתר נובעים מהרצון לצמצם את ההנחות והאילוצים שהאלגוריתם לוקח בחשבון. האתגרים העיקריים הינם :

- זיהוי השחקן/השחקנים בגבהים וגדלים שונים בהינתן סביבת רקע נתונה
- זיהוי בתנאי תאורה שונים
- התמודדות עם שינויי בהירות של מצלמה כתוצאה מחשיפה של שמש או תדירות שונה של מנורות
- צבעים דומים של בגדי השחקן וסביבת הרקע
- התמודדות עם זקן/שיער של השחקן/שחקנים
- התמודדות עם צל
- התמודדות עם הפרעות זמניות ברקע של עוברי אורח/שינויים ברקע
- תזמון נכון של התמונה מהשחקן/שחקנים והתמונה מהצורה המופיעה על הקיר
- ביצוע האלגוריתמים לזיהוי בזמן Real-Time

בנוסף לכך היו אתגרים טכניים הקשורים לביצוע האלגוריתם לזיהוי בזמן אמת (Real-Time). כמו למשל ביצוע זיהוי לבחירת כפתור בזמן אמת – מצד אחד לא נרצה שהאלגוריתם יזהה באופן מיידי בחירה של כפתור כיוון שייתכן והשחקן בטעות סימן כפתור מסוים ומצד שני נרצה השהייה נמוכה ברגע שהשחקן בחר כפתור. בנוסף האלגוריתם לזיהוי השחקן כבד חישובית בהשוואה לקצב ה-FPS של הצגת הפריימים מהוידאו במסך המשחק כך שאם היינו מבצעים אותו עבור כל frame היינו מקבלים השהייה של הוידאו ולא היתה מתקבלת תמונה רציפה. על מנת לפתור זאת השתמשנו ב-Threads הדוגמים את הפריימים מהמצלמה כל 100msec ובכך צמצמנו את ההשהייה כתוצאה מפעולת החישוב.

2.4. הנחות ואילוצי הפרויקט

האלגוריתם שנציג בהמשך מתמודד בצורה טובה עם מרבית מהאתגרים שהוצגו אך עדיין מניח את ההנחות הבאות:

- סביבת הרקע לא מכילה מראה/חומרים היכולים לשקף תאורה או תזוזה של אנשים מחוץ לרקע
- אין צל משמעותי הנובע מתזוזת השחקן/השחקנים ברקע
- בתחילת המשחק בסביבה חדשה/שינוי משמעותי בתאורה או ברקע יבוצע כיול במהלכו נלקחת תמונה של הרקע ללא השחקנים וללא מפריעים זמניים (מבוצע ע"י הכפתור "take photo" במסך הראשי, תמונה 3).

3. תיאור האלגוריתמים

בפרויקט השתמשנו בשיטות שונות של עיבוד תמונה על מנת לבצע סגמנטציה של האדם בצורה הטובה ביותר. האלגוריתם שלנו מצפה לקבל שני פריימים כקלט- תמונת רקע ותמונה של השחקן. הפלט שלו יהיה ה-MASK של האדם בעזרתו נוכל לבחון האם השחקן נמצא בתוך הצורה ועובר את הקיר או מחוץ לצורה ונפסל. כעת נסביר ונפרט על כל האלגוריתמים העיקריים בהם השתמשנו במהלך הפרויקט:

חיסור תמונות:

על מנת להוציא תמונת MASK של השחקן השתמשנו בשיטה של חיסור תמונות. חיסרנו בין תמונת ייחוס (תמונה של הרקע ללא השחקן) לתמונת המצב בה רצינו לבחון האם האדם עמד בקריטריונים ואכן נמצא בתוך הצורה. חיסור בין שתי התמונות נועד לזהות את השינויים שנוצרו בין התמונות ועל ידי כך לזהות את האדם מתוך הרקע. בחרנו להשתמש בשיטה הזאת מכיוון שכך אנחנו לא מוגבלים ע"י רקע ספציפי ולא דורשים אתחולים נוספים למשחק מלבד לקיחת תמונה לפני תחילת המשחק. אחד האתגרים שהתמודדנו איתם בשיטה זו הוא שלא יכולנו להבטיח שאף אחד מהשחקנים לא ילבש חולצה בצבע של הרקע או שגוון העור שלו יהיה מאוד דומה לשל הרקע. לכן כדי להיות רגישים יותר לשינויים של הרקע בחרנו לבצע החסרה עבור כל אחד מערוצי הRGB בנפרד. כך שאם השחקן לובש חולצה אדומה על רקע של דלת אדומה, נוכל להבדיל בין גוון האדום של החולצה לגוון האדום של הדלת. (בהמשך הפעלנו threshold על כל ערוץ בנפרד). אתגר נוסף בשיטה הזאת הוא שאם יש שינוי בצל או בתאורה אנחנו נבחין בשינוי הזה לאחר החיסור והוא יקשה עלינו לזהות את השחקן על גבי הרקע כי כעת תוצאת החיסור היא לא רק תמונה של האדם. על מנת להתמודד עם בעיה זאת השתמשנו באלגוריתמים נוספים בעיבוד תמונה עליהם נפרט בהמשך.

תוספת לערוץ ה Value:

על מנת לפתור בעיות בתמונה הנוצרות מצל הוספנו ערך שנבחר על ידי מדידות שונות לערוץ ה value. במרחב הצבעים ניתן לחלק את התמונות לסוגים שונים של ערוצי צבעים על מנת לקבל יותר מידע על צבעי התמונה. מרחב צבע מסוים הינו מרחב HSV אשר מורכב מהערוצים hue,saturation,value. ערוץ hue מייצג על צבע הפיקסל, ערוץ saturation מייצג את חוזקת הצבע ונע מלבן עד הצבע עצמו וערוץ value ממייצג את בהירות הפיקסל ונע בין שחור ללבן. לכן במידה וקיים צל/אור שיוצר הפרעות בעיבוד התמונה אנו יכולים להוסיף/להחסיר ערך מסוים לערוץ ה value על מנת לפתור בעיה זו.

Threshold:

ניתן להשתמש בפקודה זו על ערוץ בודד או תמונה בגווני אפור. בוחרים ערך סף כך שעבור כל פיקסל בעל ערך גבוה מהערך שהגדרנו הפיקסל יקבל את הערך 1 וכל פיקסל בעל ערך נמוך מהערך שהגדרנו הפיקסל יקבל את הערך אפס (ניתן גם להגדיר תמונת 0,255). כתוצאה מפקודה זו נקבל תמונת mask עם ערכים של אפס ואחד בלבד ובכך אנו מבצעים את הסגמנטציה של האובייקט שאותו אנו רוצים למצוא. בפרויקט שלנו ביצענו threshold על כל ערוץ RGB בנפרד.

Median Filter :

מסנן זה מקבל תמונה ועבור כל פיקסל מחליף את ערך הפיקסל בערך החציוני מבין שכניו של אותו פיקסל בעזרת מטריצת חלון עם גודל שנבחר על ידינו. מסנן זה מוריד רעשים כדוגמת salt & pepper על ידי שמירה של קצוות האובייקט ובכך משפר את איכות התמונה. בנוסף הפילטר עוזר לנו לצמצם חורים בתוך הMASK של השחקן. [2]

Histogram Equalization:

תמונה 15 – למעלה התמונה המקורית, ולמטה התמונה לאחר ביצוע histogram equalization

פקודה זו מחלקת מחדש את ההיסטוגרמה של התמונה באופן שיהיה יותר אחיד על פני כל הערכים האפשריים ובכך מגדילה את הניגודיות בתמונה. תמונה 15 מדגימה את ההשפעה של פונקציה זו עבור הפרדה מרקע. האלגוריתם שבו השתמשנו הוא מספריית open cv אשר ממפה את הפיקסלים באופן הבא:

- תחילה האלגוריתם מחשב את ההיסטוגרמה של התמונה. מתוך ההיסטוגרמה הלאגוריתם מחשב את פונקצית ההתפלגות המצטברת של הפיקסלים בתמונה ומנרמל אותה כך שהערך הגבוה ביותר בפונקציה יהיה 1. הפונקציה מתקבלת על ידי סכמת כל ערכי ההיסטוגרמה מערך הפיקסל הנמוך ביותר אל הגבוה ביותר.
- לאחר מכן האלגוריתם מבצע מיפוי מחדש כאשר פונקציית המיפוי היא פונקציית ההתפלגות המצטברת שהתקבלה (מוכפלת בערך המקסימלי האפשרי של פיקסל). אם ערך הפיקסל לפני הטרנספורמציה היה x אז בעזרת פונקציית CDF שהתקבלה ערך הפיקסל אליו ימופה הוא y=CDF(x).

התמונה שתתקבל לאחר הטרנספורמציה היא תמונה עם ניגודיות מוגברת עם דגש גם על הפרטים הקטנים בתמונה שלא היה ניתן להבחין בהם לפני כיוון שהיו מוסתרים בתוך אזורים עם צל או אור מוגברים. [2]

Dilate:

מסנן זה מבצע פעולה דומה כמו מסנן median. ההבדל במסנן זה הוא שכעת עבור כל פיקסל אנו מחפשים את השכן בעל הערך המקסימלי ומשנים את ערך הפיקסל להיות הערך המקסימלי שמצאנו. בעזרת מסנן זה נוכל לחבר בין קווים שנחתכו בתמונה ולמלא חורים בMASK של השחקן. המסנן עוזר להבליט צורות מסוימות בתמונה עלי ידי כך שניתן לבחור את צורת וגודל מטריצת הקרנל שבוחרת את הערך המקסימלי. לדוגמא ניתן לבחור מטריצת קרנל עגולה על מנת להבליט צורות עגולות בתמונה או ניתן לחבר בין חלקים רחוקים של הקונטור של השחקן על ידי הגדלת המטריצה. המטרה שלנו בשימוש במסנן זה הייתה שבהמשך נוכל להוציא קונטור שלם של השחקן ולא כמה קונטורים שונים (למשל קונטור של ראש וקונטור של נעל). [2]

Find Contours:

על מנת למצוא את הקונטור של השחקן השתמשנו בפונקציה find contours הנמצאת בספריית OpenCV המשתמשת ב modified Suzuki algorithm. האלגוריתם מקבל כקלט תמונה בינארית

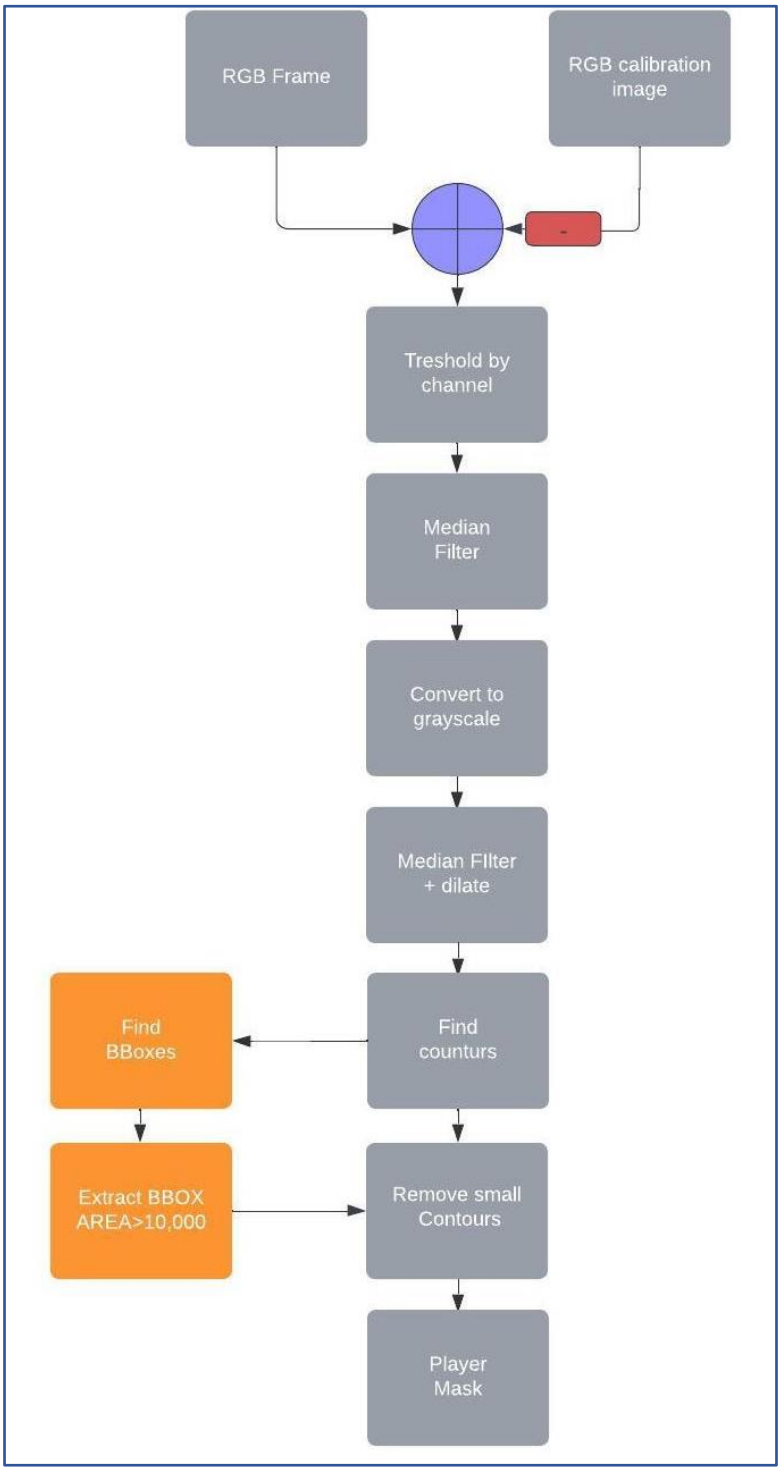
כאשר הרקע הינו אפס והאובייקט עם ערך אחד ומוציא כרשימה את הקווים המאפיינים צורות שונות בתמונה – contour. האלגוריתם פועל כך :

- תחילה האלגוריתם עובר שורה אחר שורה על מנת להגיע אל הפיקסל הראשון שאינו אפס.
- לאחר שמצאנו את הפיקסל הראשון שאינו 0 האלגוריתם מבצע מעקב אחר כל הפיקסלים שאינם אפס מסביב לפיקסל שמצאנו ועובר על הקו המאפיין של אותו אובייקט שמצאנו תוך מציאת כיוון ההתקדמות.
- סיום המעקב אחר הקצוות של האובייקט מתרחש כאשר האלגוריתם מגיע לנקודת ההתחלה.

בסיום האלגוריתם אנו מקבלים רשימה של contours אשר מכילה מידע נוסף גם על שטח האובייקט וכך ניתן למחוק רעשים בתמונה שמאופיינים בשטח קטן יחסית לאובייקט שאנו מחפשים. בנוסף האלגוריתם גם מתייג את הקונטורים שנמצאו ברשימה היררכית על מנת לסווג בצורה יעילה יותר את האובייקטים שנמצאו בתמונה. [1]

בפרויקט שלנו מצאנו את כל הקונטורים של התמונה. על מנת לנקות רעשים ברקע מצאנו את הBounding Box סביב כל הקונטורים ומחקנו את כל הקונטורים ששטח הBBOX שלהם קטן משטח הBBOX שיכול להתאים לקונטור של שחקן. האתגר בשיטה היה שאם לא הצלחנו להוציא קונטור אחד שמקיף את כל השחקן (למשל קיבלנו את נעליים וראש בנפרד מהגוף בגלל צבע גוף שקרוב לרקע או זקן שחור) אז מחקנו חלק משמעותי מהשחקן. על מנת להתמודד עם בעיה זו השתמשנו בפילטר dilate כמו שהסברנו קודם. (נראה דוגמאות בהמשך).

דיאגרמת בלוקים של האלגוריתם:



איור 2

4. תהליך העיבוד ותוצאות ביניים

תהליך עיבוד התמונה כלל שני עקרונות מרכזיים, הראשון הוא זיהוי מיטבי של השחקן והשני והלא פחות חשוב הוא הימנעות מהפרעה של אובייקטים אחרים בתמונה כמו תנועות שונות מאחורי הקלעים, כניסה של חפצים וצל. לעתים שני קריטריונים אלה עלולים לבוא אחד על חשבון השני, לדוגמא אנו יכולים לפתור את בעיית הצל אך בטעות גם נזהה פחות טוב את השחקן, ומצד שני נזהה בצורה מצוינת את השחקן אך גם נאפשר כניסה של יותר אלמנטים של צל.

השחקן מקבל ניקוד לפי שני קריטריונים:

1. כמה מתוך הצורה הוא מצליח למלא – כאן חשוב הזיהוי המיטבי של השחקן.
2. בכמה הוא חורג ממנה עד לאחוז חריגה שבה הוא נפסל – כל הפרעה שאינה חלק מהגוף של השחקן תגדיל את החריגה ובכך יכולה להביא לפסילה לא מוצדקת, לכן מזעור ההפרעות חשוב.

בכדי להגיע לתוצאה שממטבת את שני העקרונות נשתמש בכלים שונים של עיבוד תמונה כגון שימוש במרחבי צבעים, החסרת רקע (Background subtraction), מסננים, threshold, היסטוגרמה, מציאת קונטורים ועוד.

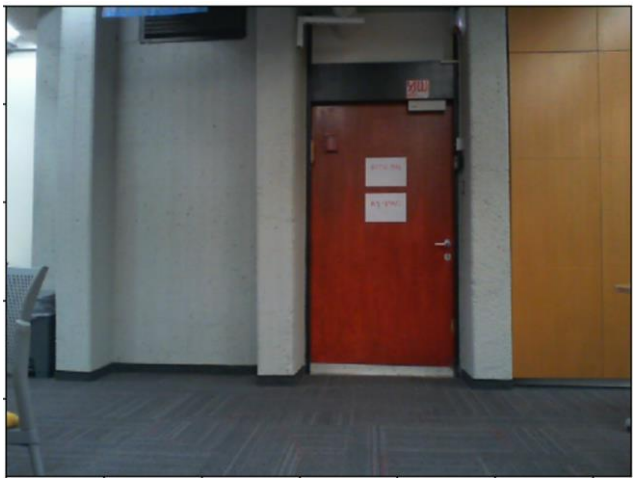
נראה אתגרים ובעיות בעיבוד התמונה וכיצד התמודדנו איתן.

זיהוי השחקן:

המשימה הראשית בתהליך העיבוד היא הפרדה מיטבית בין השחקן לבין הרקע והוצאת פלט של תמונה בינארית שבאמצעותה נוכל לדרג את הביצועים של השחקן. מכיוון שאנו לא יודעים מה השחקן ילבש ובאיזו סביבה יתקיים המשחק הזדקקנו לפתרון יצירתי שיעבוד בכמה שיותר מקרים ולכן נבסס את העיבוד על החסרת רקע – נדגום את הרקע ללא נוכחות של השחקן ובכל שלב נבצע החסרה בין הframe הנוכחי לבין תמונת הרקע.

כעת נדגים שלבים שונים בתהליך ונציג בעיות שונות ופתרונן:

1. דגימת הרקע:



תמונה 16

2. דגימה של השחקן והחסרה בין התמונה הנוכחית לתמונת הרקע, ונפעיל median Filter לניקוי רעשים :

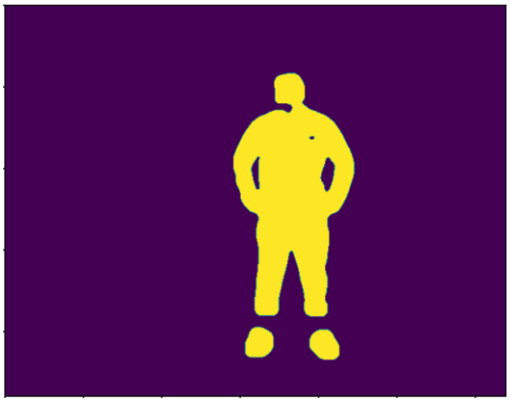
תמונה 17

תמונה 18

3. נבצע threshold על כל ערוץ בנפרד ונעביר את התמונה לייצוג בערוץ יחיד (grayscale) :

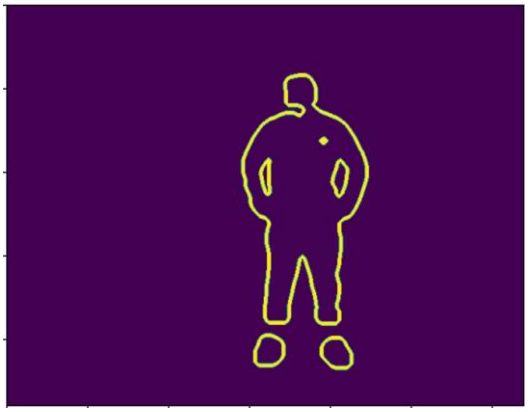
תמונה 19

4. נבצע threshold נוסף על התמונה ונציג תמונת mask:



תמונה 20

5. נציג קונטורים באמצעות הפקודה Findcontours:

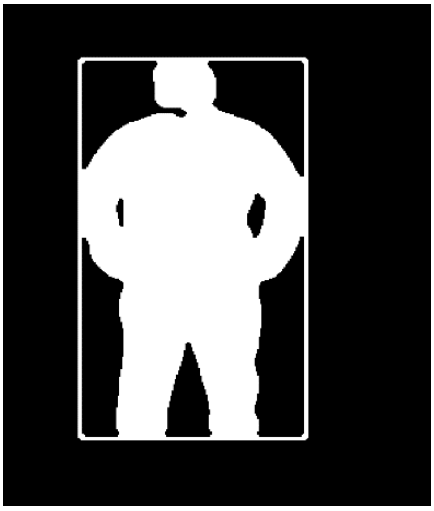


תמונה 21

התמודדות עם הפרעות:

הזיהוי שהדגמנו הוא זיהוי פשוט, לא נוצר שינוי ברקע מה שהקל על הזיהוי. כמובן שלא נוכל תמיד להסתמך על כך שלא ייווצרו שינויים כלל ולכן נרצה להשתמש בשיטות שונות כדי להתמודד עם הפרעות.

אחת הדרכים היא להתבסס על הקונטורים הגדולים ולהתעלם מקונטורים קטנים יותר (שיכולים להוות הפרעות בלתי רצויות) לכן שילבנו בעיבוד פקודה שמוציאה ערכי BBOX לכל קונטור בשם boundingRect, וכך ע"י חישוב שטח הBBOX (אורך * רוחב) נוציא את הקונטורים הגדולים, נגזור את הקונטור של השחקן ונציג אותו:

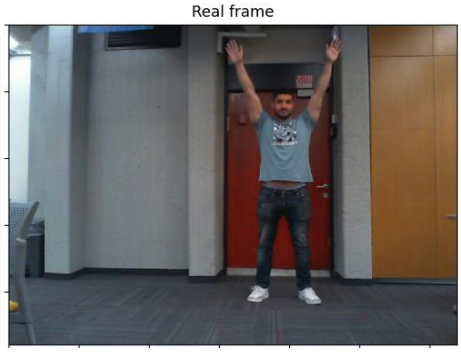


תמונה 22

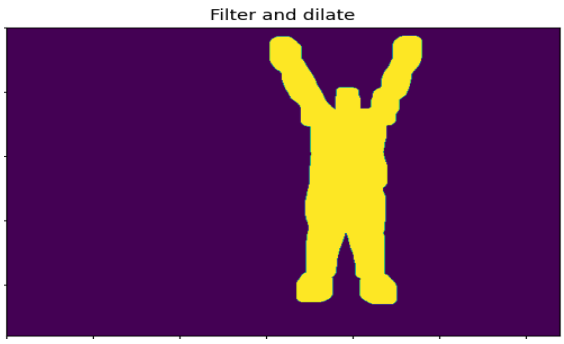
אמנם הצלחנו לתחום את האזור בו נמצא השחקן אך ניתן לראות שהאזור בין הנעליים לרגליים מחוץ לקונטור ולכן נקבל חיתוך של כפות הרגליים.

כדי לפתור את הבעיה, נשתמש בפקודה של ספריית open-cv בשם dilate, שמשתמשת בחלון בגודל מוגדר ומשנה את הערך בכל פיקסל לערך המקסימלי בתחום החלון, כך ניצור חיבור בין כפות הרגליים לרגליים במטרה לאחד את הגוף לתוך קונטור עם BBOX יחיד.

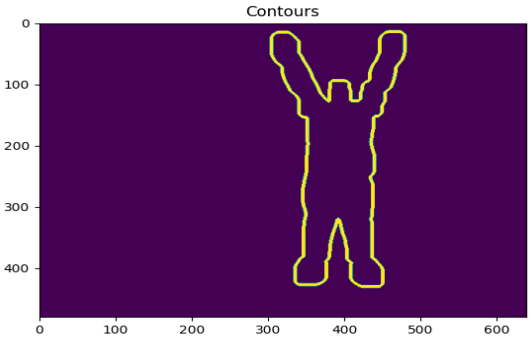
לצורך השוואה ניקח תמונה דומה ונשתמש בפונקציה dilate כדי לחבר את הרגליים ונעביר אותה בשלבים השונים :



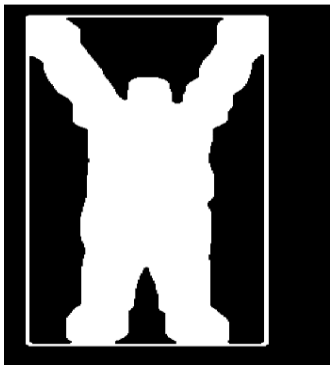
תמונה 23



תמונה 24



תמונה 25

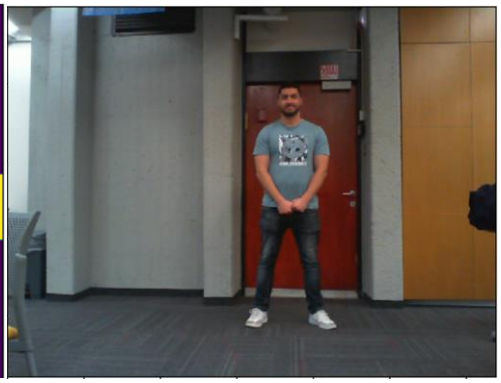


תמונה 26

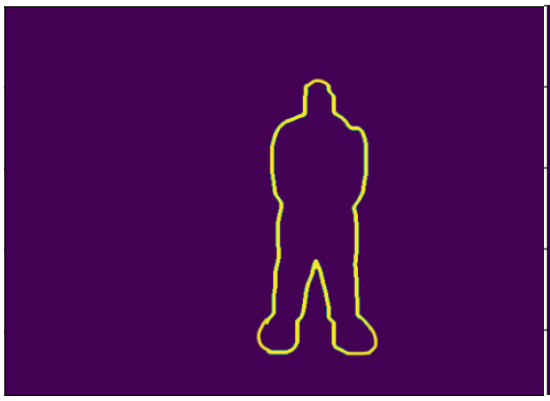
כעת ניתן לראות שאיחדנו הרגליים לגוף אחד ולכן נוכל לקבל contour שמכיל את כל הגוף.

נדגים את יעילות השיטה באמצעות הנחת הפרעה יזומה:

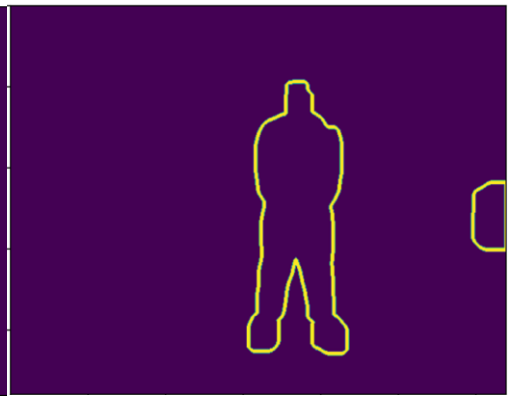
תמונה 28



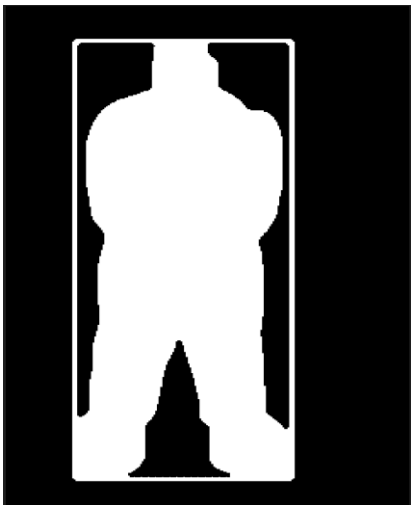
תמונה 27



תמונה 30



תמונה 29

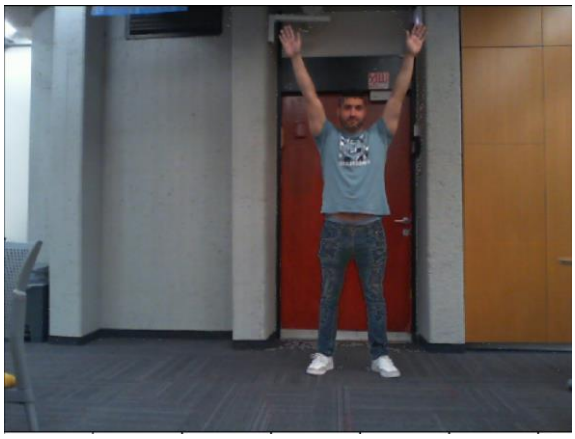


תמונה 31

העובדה שהצלחנו לאחד את השחקן לקונטור בעל שטח גדול, יחד עם הפעולה של מחיקת קונטורים קטנים עזרה לנו להוציא הפרעה משמעותית שיכלה לפגוע בניקוד של השחקן ולהביא לפסילתו.

התמודדות עם צל באמצעות מרחב צבעים HSV:

תופעה של צל היוותה בעיה משמעותית במהלך תהליך העיבוד. מכיוון שמטרתנו העיקרית בזיהוי היא להוציא תמונת MASK בינארית של השחקן, לעתים קיבלנו תמונת MASK בתוספת צל מה שעלול לגרום לפסילות כתוצאה של צל שחורג מהצורה הרצויה. הבעיה העיקרית עם צל היא שצריך להפריד בין צל שקיים גם בתמונת האתחול וגם בframe שמכיל את השחקן, לבין צל שנובע מהנוכחות של השחקן. לצורך כך השתמשנו בערוץ הvalue במרחב הצבעים HSV שמשפיע על החשכה והארה של תמונה.



תמונה 32

במהלך בניית המשחק שמנו לב שככל שהשחקן מתקרב לדלת הוא נכנס לתוך כוך וגורם לצל, לכן בחרנו להעלות ערכי value באזורים שמקבלים צל מהשחקן, מה שעזר להאיר את אותם אזורים ולעזור במיגור הצל.

התאמה בין השחקן לצורה:

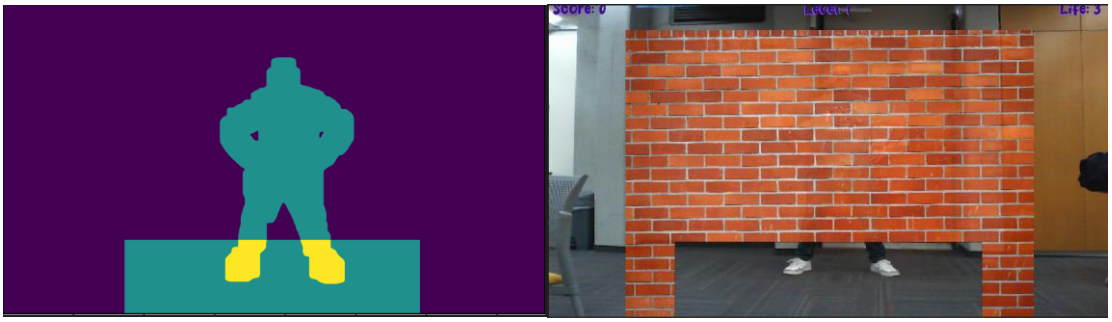
כפי שכבר ציינו, הצלחה או כישלון של השחקן במשחק נמדדת עפ״י ההכלה שלו בתוך הצורה ומזעור החריגה, את שני הקריטריונים הללו חישבנו באמצעות הנוסחאות הבאות:

1. $in = \frac{100}{\sum wall_i} * \sum corr_i$ - כמה מתוך הצורה השחקן ממלא.
2. $Out = 100 * \frac{\sum body_i - corr_i}{\sum body_i + wall_i}$ - חריגה של השחקן מהצורה.

כאשר:

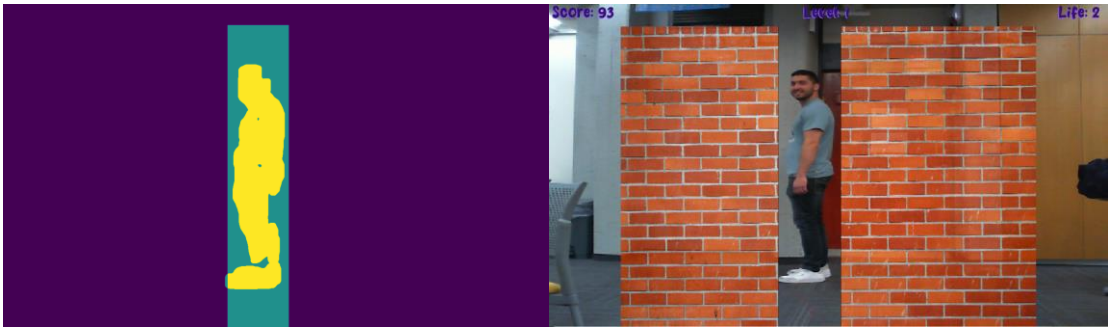
- body היא תמונת הMASK של השחקן וWALL היא תמונת MASK של הקיר.
- $corr_i = wall_i * body_i$ כמכפלה של איבר איבר.

על מנת לקבל ניקוד מירבי השחקן צריך למקסם את הביטוי in ולמזער את הביטוי Out, ולחלופין יפסל כאשר יתקל בחסם תחתון של in=20 או חסם עליון של Out=12.



תמונה 33

תמונה 34



תמונה 35

5. אנליזה

החלק המשמעותי במשחק הוא ההתאמה בין השחקן לצורה ומתן ניקוד בהתאם ואחד האתגרים שלנו היה ההתמודדות עם הצל שלעיתים מגדילה לנו את הmask של האדם ובכך אנחנו מקבלים תוצאות שגויות. כעת נבדוק את הפתרונות השונים בהם בחרנו להשתמש. נבדוק את הרגישות של ערכי ה-Thresholds עבור ערכי Value וארגמונטים של הפילטרים שונים. ננתח עבור המקרים השונים את הביצועים של המשחק. המטרה שלנו היא למצוא את הפרמטרים שעבורם נקבל MASK של השחקן בצורה האופטימלית ביותר.

הקריטריונים אותם נבחן הם:

- רגישות של ערכי ה-Thresholds השונים.
- גודל מטריצת קרנל של הפילטר Dilate.
- הערכים שנוסיף לערוץ ה-Value במרחב HSV.

את הניתוח נבצע בשיטת F-score כך ש:

- **True positive** - השחקן בתוך הצורה וקיבל ניקוד
- **False positive** - השחקן פספס את הצורה אך קיבל ניקוד
- **True negative** - השחקן פספס את הצורה ונפסל
- **False negative** - השחקן בתוך הצורה אך נפסל

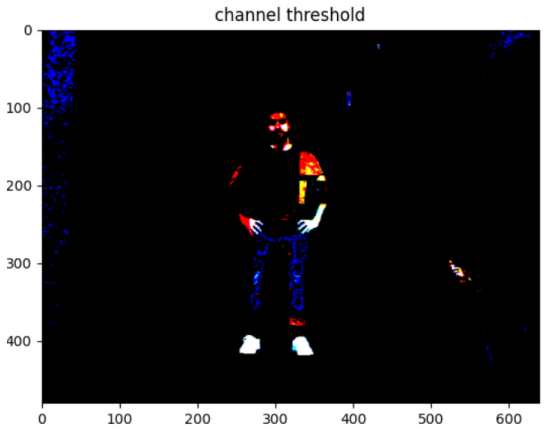
<u>עבור ערך Threshold גבוה:</u>

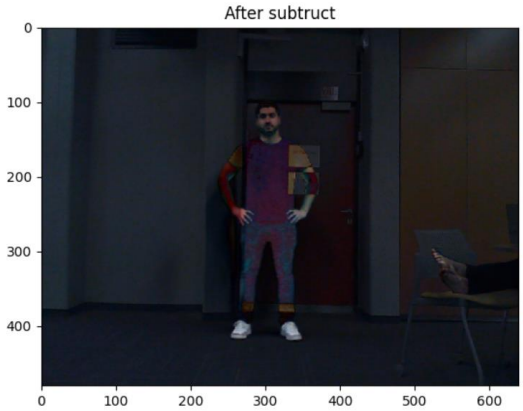
עבור ערכים גבוהים ב Threshold נצפה לקבל תמונה של השחקן עם רקע שחור בעלת חלקים שחורים רבים יותר מכיוון שאנחנו מאפסים הפרשים גדולים יותר. כתוצאה מכך אנחנו מצפים להפסיד חלקים גדולים יותר בMASK של השחקן וכתוצאה מכך יש שתי אפשרויות של תוצאות:

Fals positive- השחקן יצא מהצורה, אך מכיוון שחלקים רבים ממנו נמחקו, הMASK שלו יוצא בתוך הצורה וכתוצאה מכך הוא מקבל נקודות למרות ש"נפסל".

False negative- השחקן בתוך הצורה, אך חלקים רבים ממנו נמחקו והאלגוריתם לא מוצא אותו 'מספיק' בתוך הצורה ולכן פוסל אותו.

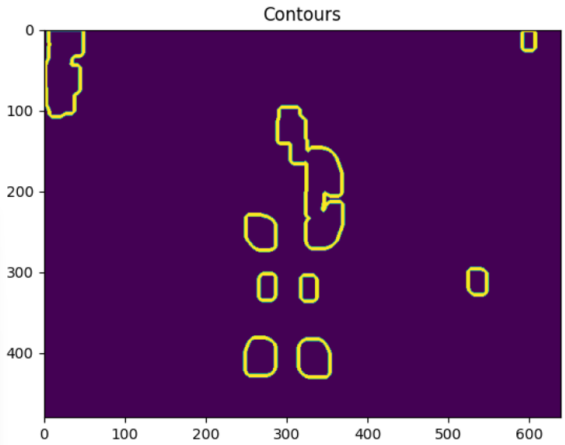
עבור TH=0.25 התוצאות שקיבלנו הן:





תמונה 37

תמונה 36

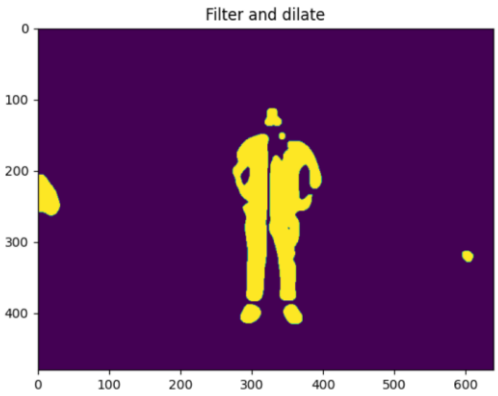


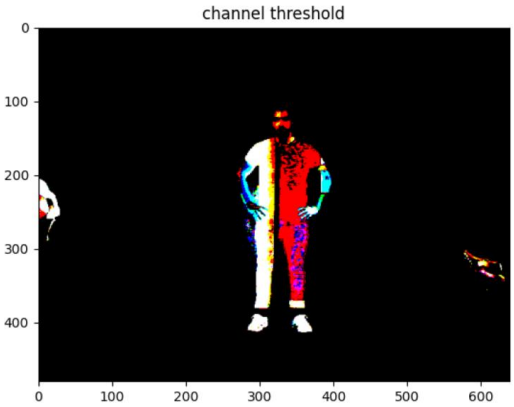
תמונה 39

תמונה 38

כלומר במקרה הזה אכן איבדנו חלקים גדולים של השחקן כפי שציפינו.

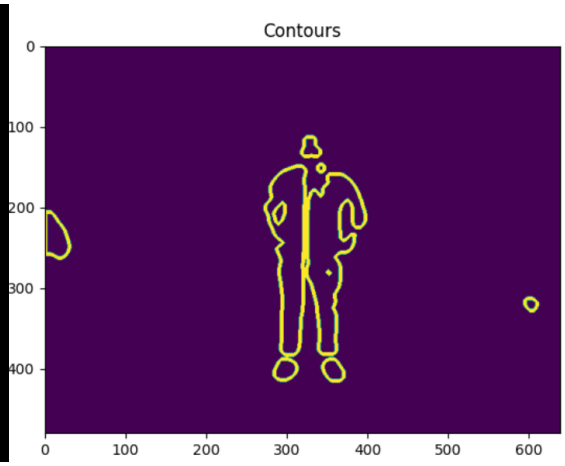
עבור TH=0.2 וגודל של הפילטר dilate הוא 9*9 התוצאות הן:





תמונה 41

תמונה 40

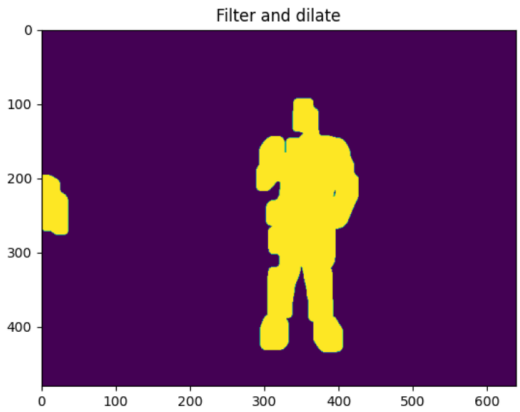


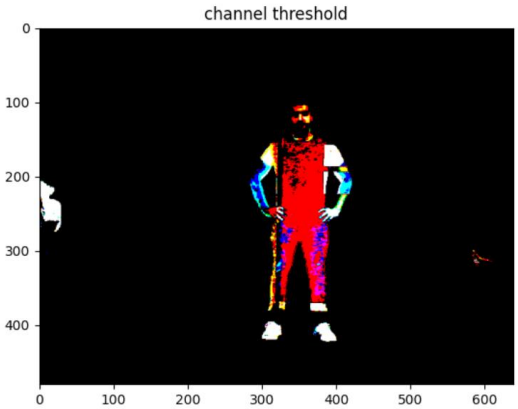
תמונה 43

תמונה 42

כעת הקטנו את הTH אך הוא עדיין גבוה. לכן ניתן ראות את החיתוך באמצע של הגוף שנמצא בחלק כהה של הרקע. התוצאות קצת יותר טובות אבל ניתן לראות שפילטר הdilate לא משפר לנו את התוצאות כמו שרצינו ואנחנו מקבלים חיתוך של הראש והנעליים. לכן בשלב הבא הגדלנו את הגרעין של הפילטר.

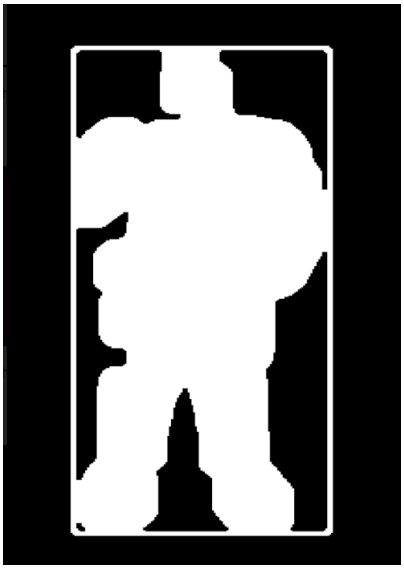
עבור TH=0.2 וגודל של מטריצת הפילטר dilate הוא 22*18 התוצאות הן:



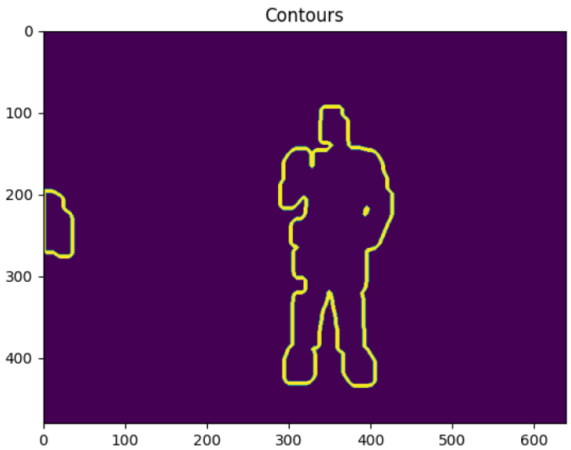


תמונה 45

תמונה 44



תמונה 47



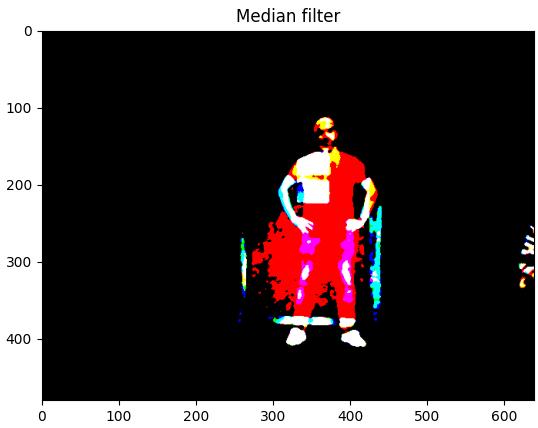
תמונה 46

כעת למרות הגדלת מטריצת הdilate קיבלנו תמונת MASK שיותר נראית כמו בן אדם אך עדיין לא מושלמת. זה מכיוון שהTH עדיין גבוה ואנחנו מאבדים חלקים מהשחקן בתמונה.

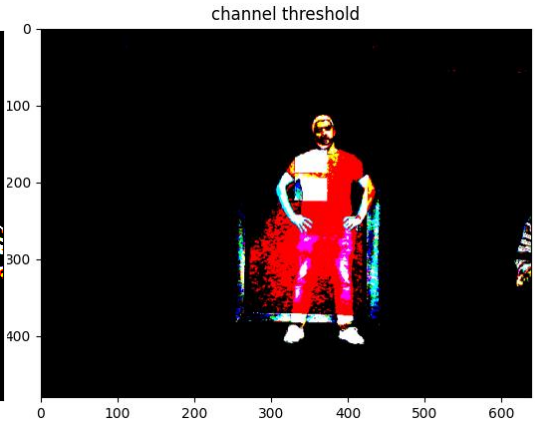
עבור ערך Threshold נמוך:

עבור ערכים נמוכים ב Threshold נצפה לקבל תמונה של השחקן עם רקע שחור בעלת רעשים רבים יותר מכיוון שכעת אנחנו מאפסים רק הפרשים קטנים ממש ולכן האלגוריתם במצב הזה רגיש לכל שינוי קטן בין תמונת הרקע לתמונה הנוכחית של השחקן (למשל תאורה שמשתנה או צל שנוסף). כתוצאה מכך אנחנו מצפים לקבל MASK של השחקן עם הרבה רעש מיותר מסביב. במצב הזה אנחנו מצפים לקבל הרבה תוצאות של Fals negative- השחקן נמצא בתוך הצורה, אך כתוצאה מהרעש סביבו האלגוריתם מוצא אותו 'נוגע בקיר' בהרבה מקומות ולכן פוסל אותו.

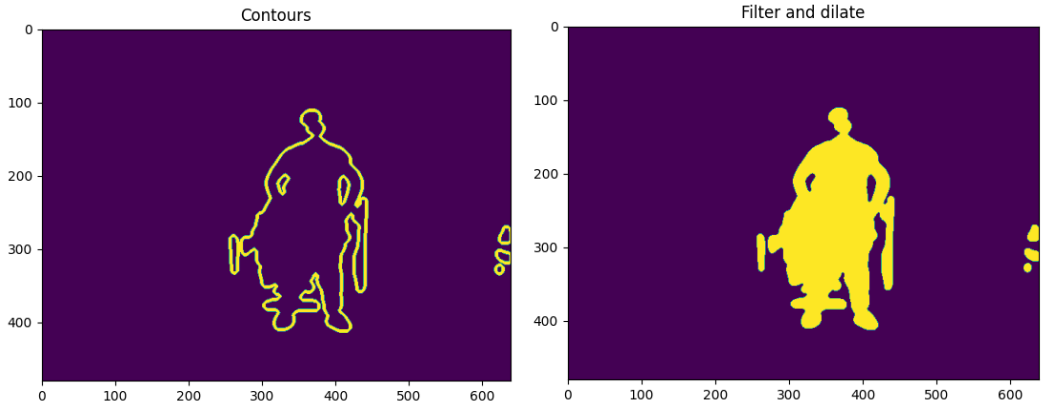
עבור TH=0.1 ו- value=0 התוצאות הן:



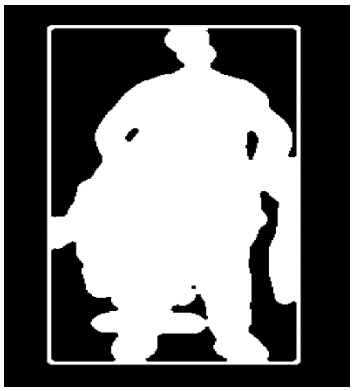
תמונה 49



תמונה 48



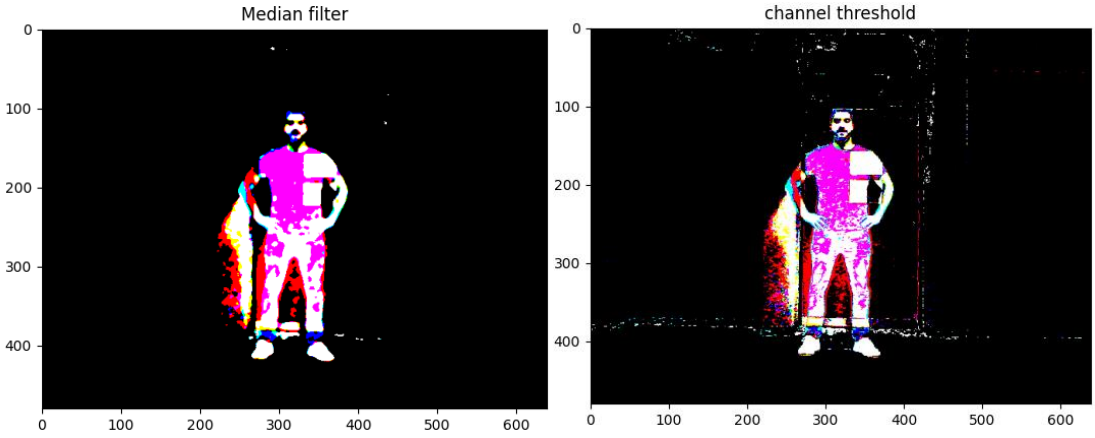
תמונה 50 תמונה 51



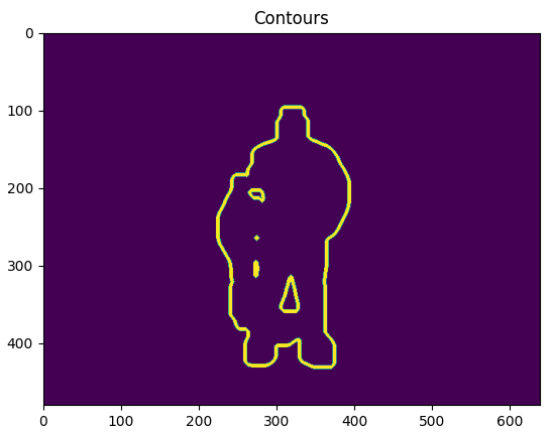
תמונה 52

כמו שציפינו לאחר הthreshold קיבלנו תמונה מורעשת. בשלב הזה ניתן לראות שפילטר median אכן מנקה לנו את הרעשים שברקע. תמונת הMASK שהתקבלה במצב הזה מורעשת מאוד עם הרבה רקע מיותר שנוצר כתוצאה מהצל ומכך שthreshold רגיש אליו במיוחד. בנוסף במקרה הזה לא התעסקנו בכלל עם ערך הvalue של התמונה. בשלב הבא ננסה להתמודד עם הצל על ידי הגדלת הערך של הvalue בתמונה.

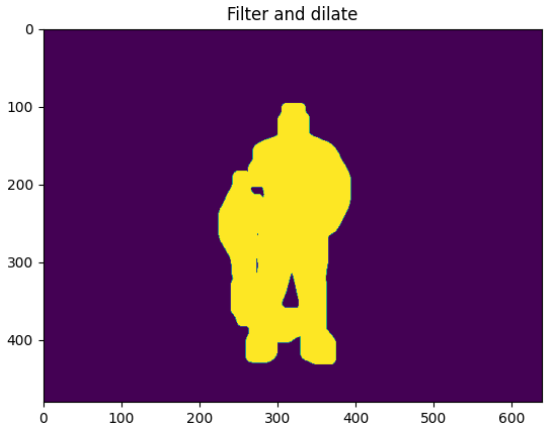
עבור TH=0.1 ו- value=0.2 התוצאות הן:



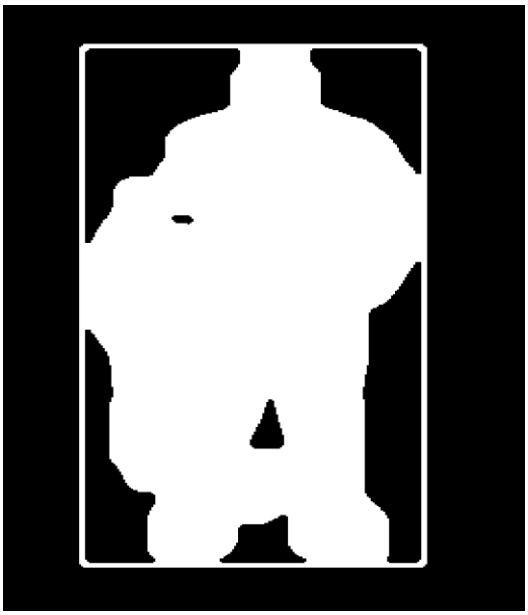
תמונה 53 תמונה 54



תמונה 56



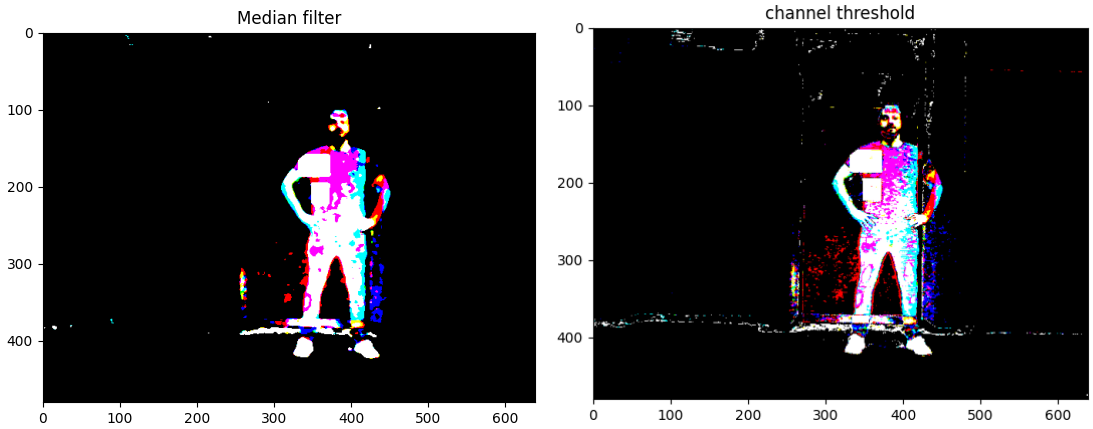
תמונה 55



תמונה 57

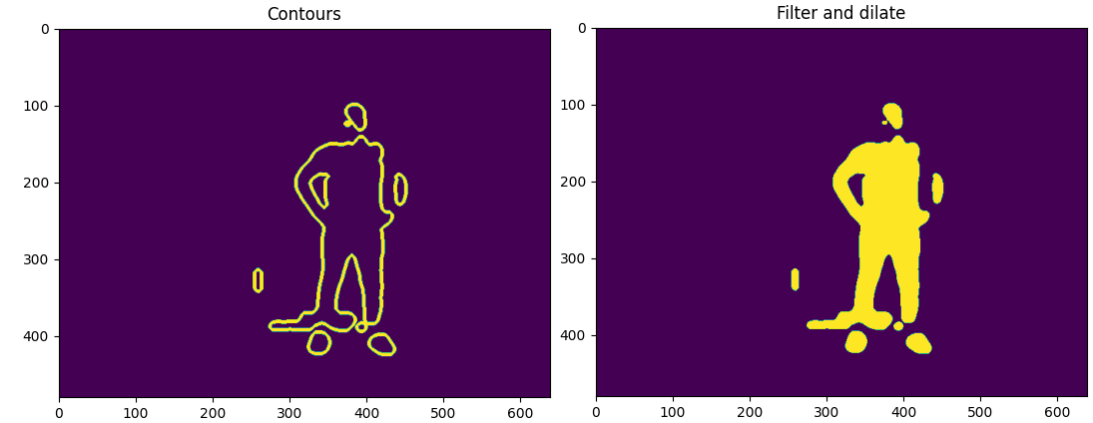
כעת בעזרת הגדלה של valueה הצלחנו לצמצם מעט את האפקט של הצל ברקע. אך ניתן לראות שעדיין קיבלנו מריחה גדולה בתמונה של הMASK. זאת מכיוון שמטריצת הdilate שלנו עדיין גדולה ומצמצמת חורים גדולים. על מנת לטפל כעת בבעיה נקטין את מטריצת הdilate. כעת נצפה לצמצם רק חורים קטנים וע״י לנסות לסנן את המריחה של הצל מהשחקן.

עבור TH=0.1, value=0.2 ו-dilate=6*6 התוצאות הן:



תמונה 58

תמונה 59



תמונה 60

תמונה 61

תמונה 62

כעת ניתן לראות שהצלחנו לצמצם משמעותית את המריחה של הצל מהאדם ולקבל MASK מאוד קרוב לצורה של השחקן, אך מכיוון שמטריצת dilate קטנה היא לא חיברה בין הראש לגוף ולכן לאחר החלוקה לקונטורים ומחיקה של BBOX קטנים קיבלנו חיתוך של הראש.

לסיכום לאחר כמה ניסיונות שונים של ניסוי וטעיה מצאנו שהתוצאות המקסימליות מתקבלות עבור TH=0.15, value=0.1 ומטריצת הdilate בגודל של 18*22. (התוצאות שהצגנו במהלך הדו"ח הם לפי הערכים הנ"ל).

6. סיכום

בפרויקט זה הצלחנו להביא את השעשועון הטלוויזיוני "Hole in The Wall" למחשב של כל אחד ואחת ובשימוש במצלמה יחידה. המשחק מתחילתו ועד סופו מופעל על-ידי אלגוריתמי זיהוי של השחקנים כך שניתן לתפעל את המשחק מרחוק וללא גישה למחשב. המשחק מאפשר וריאציות שונות של משחק כמו מספר משתתפים ורמת קושי, הוא משלב אנימציות וסאונדים שונים להשלמת חווית המשחק ועל מנת ליצור תחושה של משחק תלת-מימד אמיתי. בנוסף יש לציין כי שילוב האלגוריתמים הצליח לעמוד ברוב האתגרים שציינו בפרק ההקדמה כמו הפרעות זמניות ברקע, שינוי בהירות בתמונה ועוד, כך שמלבד ההנחות, כל אחד יכול לשחק במשחק מכל מקום.

פרט לאלגוריתם שאנו השתמשנו בו לזיהוי השחקנים, קיימים אלגוריתמים שונים לביצוע המשימה שלא מבוססים על תמונת הכיול הראשונים הנדרשת אצלנו. למשל יכולנו להשתמש באלגוריתמים מבוססי machine-learning לזיהוי השחקן/שחקנים כך שהמשחק יהיה חסין לגמרי להפרעות ברקע. בנוסף זה היה מאפשר הוספת פיצ'ר לזיהוי השחקן במידה וכבר שיחק בעבר. אולם כחלק ממסגרת הקורס בחרנו ללכת על אלגוריתמי עיבוד תמונה קלאסיים ללא שימוש במודלי AI. דרך נוספת היא להוסיף מעקב אחר השחקן/שחקנים ע"י זיהוי פיצ'רים ייחודיים לכל שחקן ומעקב תנועה בזמן. ואכן שקלנו להוסיף את האלגוריתם לשיפור הביצועים אך מהר מאוד התברר כי אלגוריתמים כאלו דורשים חישובים רבים מידי שלא היו ישימים כחלק ממערכת שאמורה לרוץ ב-Real-Time ולכן העדפנו את האלגוריתם הפשוט והאלגנטי ביותר שמגיע לתוצאות טובות.

בסופו של דבר הצלחנו באמצעות שילוב שיטות השונות להגיע לביצועים טובים בזיהוי שהתבטאו במהלך משחק רציף ונכון. שילוב אלגוריתמי עיבוד התמונה ביחד עם שיטת הניקוד יצר זרימה טובה של המשחק ושמר על איזון בהינתן מפריעים או זיהוי חלקי, כאשר גם אם התרחש שינוי ברקע וחוץ מהשחקן זוהה דבר נוסף, שיטת הניקוד המבוססת על אחוזי החפיפה של השחקן עם הצורה בקיר מושפעת ברובה מדמות השחקן שזוהה ומעט מההפרעות ברקע. במבחן האמת ביום הצגת הפרויקטים שיחקו במשחק משתתפים רבים אשר דיווחו על חווית משתמש טובה, הזיהוי התבצע כראוי, וכולם יצאו שמחים ומזיעים.

7. ביבליוגרפיה

[1] Suzuki, S., and Be, K. (1985). Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing 30, 32–46. doi: 10.1016/0734-189X(85)90016-7.

[2] https://docs.opencv.org/3.4/d4/d1b/tutorial_histogram_equalization.html