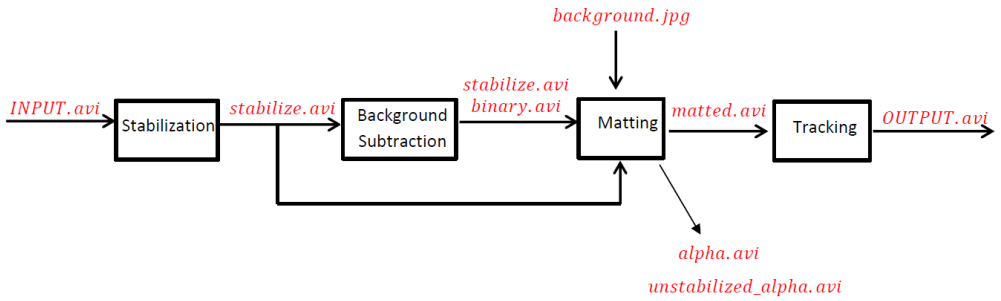
פרויקט מסכם לקורס

**עיבוד וניתוח וידאו 0512-4263**

בפרויקט זה ניישם את הצעדים הבאים:

- מקבלים כקלט וידאו לא מיוצב של אדם הולך, המטרה העיקרית היא לגזור את האדם מן התמונה ומלבישים אותו על רקע חדש (גם הוא קלט). כדי לבצע זאת אנו נדרשים בהתחלה לייצב את הוידאו, אז לייצר תמונה בינארית אשר האובייקט (האדם) יקבל 1 ו הרקע 0. אחר זאת אנו מבצעים alpha matting של האדם להרקע החדש בהתחשבות בגלישת הצבע מה הרקע הישן. ואז בסוף נבצע עקיבה על האדם בתמונה החדשה, כלומר נשרטט מלבן סביבו.

- את המערכת למעלה ניתן לתאר בעזרת הבלוקים בתרשים למטה:




איור 1- תרשים זרימה לפרויקט עיבוד וידאו

- Stabilization: מקבל כקלט וידאו 'INPUT.avi' לא יציב, ומייצב אותו ומוציא כפלט את 'stabilized.avi', כמו כן מקבל פרמטרים השייכים לפונקציה שאחראית על מציאת נקודות עניין לייצוב הוידאו.
- Background Subtraction: מקבל כקלט את הסרטון המיוצב מייצר את הסרטון הבינארי binary.avi וכן סרטון של האדם על האזור של המסיכה – extracted.avi
- Matting: מקבל כקלט את הסרטון הבינארי עם הסרטון המיוצב והרקע החדש ומייצר את הסרטון עם הרקע החדש matted.avi שבו רואים את האדם על פני הרקע החדש כמו כן הבלוק הזה גם מייצר עוד את סרטון של alpha.avi שבו ערכי alpha של כל פיקסל .
- Tracking: מקבל כקלט את 'matted.avi' וכן סרטון בינארי , מוצא על סמך הסרטון את הבינארי את המיקום של האובייקט ומבצע עקיבה במהלך הוידאו, ההעקיבה מתבצעת ע"י מלבן סביב האדם ההולך לאורך הוידאו. נשמור את הסרטון הנ"ל ב 'OUTPUT.avi'.

בנוסף למודלים אלו ישנו קובץ קונפוגרציה בשם config.py בו ניתן לשלוט על הפרמטרים הנשלחים למודולים השונים וכן קובץ main.py שבו מרציצים מודלים אלו.

# Stabilization:

קלט- וידאו רועד(לא יציב).
פלט- וידאו מיוצב, פרמטרי התנועה .

כמו שהוזכר בהצגת הפרוייקט המטרה של בלוק הזה היא לייצב את סרטון הקלט.
הגישה שבה נקטנו על מנת לבצע חלק זה הייתה למצוא טרנפורמציה פרוז'טיבית הממפה מישור של תמונה בזמן t לתמונה בזמן t-1 ובצורה זו לקזז בעצם את התנועה של המצלמה כיוון שאז בעצם אוביקטים סטטים שהיו בפריים הקודם יהיו מיושרים לפי הפריים הקודים ויישארו סטטיים בפריים הנוכחי .
מבחינה תאורטית אם לא היה שינוי של הרקע כלל פתרון של הטלה של הפריימים על פני הפריים הראשון היה צריך לתת מענה זהה אך מעשית לאחר מספר יחסית גדול של פריימים אנו מבחינים שהטרנפורמציה לא מתקבל כראוי והתמונה בפריים לא ממשיכה ברצף את התמונה מהפריים הראשון ולכן בחרנו בשיטה זו.

לצורך מציאת התמרה זו עשינו שימוש במספר פונקציות:
ראשית מצאנו נקודות עניין בפריים הנוכחי -זאת באמצעות הפונקציה של goodFeatureToTrack opencv ,הפונקציה עושה שימוש באלגוריתם שלharris corner detector על מהת למצוא קודקודים בתמונה (שהם כפי שנלמד בהרצאה מידע אינפורמטיבי ויחיודי כדי לתאר תמונה כפי שנלמד בהרצאה), הפרמטרים המועברים לפונקציה זו ניתנים לשליטה באמצעות קובץ הקונפגורציה.
בשלב הבא , אנו מחפשים התאמה של נקודות עניין אלו בפריים הקודם , כדי לבצע זאת אנו משתמשים מוצאים את האoptical flow של הנקודות האלה בין הפריימים ומחפשים למצוא התאמה על בסיס ה optical flowבפריים הקודם. חיפוש זה נעשה באמצעות הפונקציה א calcOpticalFlowPyrLK אשר מחשבת את ה optical flow של סט נקודות בין שתי תמונות על בסיס השיטה של lk בפרמידה ומחזירה איזה נקודות תואמות להם בפריים היעד(הפריים הקודם) , וסטטוס לאיזה נקודות נמצאה התאמה.

כעת לאחר מציאת סטים של קודקודים תואמים אנו מוצאים את ההומגרפיה בין התמונות .
יש לציין שבשלב ראשון ניסנו לחשב טרנספורמציה אפינית אך הבחנו שיש תזו זות של רעש גם בציר z (באזור הדלת בסרטון הנתון) ולכן בחרנו למצוא טרנספורמציה פרוזקטיבית(כאמור טרנפורמציה אפינית יודעת רק לתקן פועלות של scale הזזה וסיבוב ) . את הטרנפורמציה אנו מוצאים באמצעות הפונקציה findHomography פונקציה זו עושה גם בשימוש באלגורתים ransac על מנת למצוא את הטרנפורמציה הטובה ביותר על בסיס סט הנקודות (מחשבת באופן איטרטיבי על בסיס בחירה אקראית של 4 נקודות בכל פעם ואז בודקת מול מספר ה ouliers עפ"י אומדן סף מסוים והטרנפורמציה שמתארת בצורה הטובה ביותר את ההתאמה של שאר הנקודות היא זו שנבחרת).


איור 1- תיאור נקודות עניין בפריים

<u>טיפול בקצוות שחורים</u>

אחד הבעיות ששמנו לב שקורות בפתרון שאנו מציעים הוא שע"י המרה למישור התמונה הנוכחי נשארים אזורים שחורים בתמונה ,זאת בשל ההתאמה כיווץ והזזה של מישור התמונה למישור אחר באופן המשאיר אזורים מתים.

הדרך שמצאנו לפתור את בעיה זו היא לעשות stitching בין התמונות ולמלא באזורים השחורים המתים פריימים חלקים מהפריים הקודם.

ע מנת ליישם פתרון זה כתבנו את הפונקציה fixBoundary אשר מה שהיא עושה הוא להפעיל את הטרנפורנמציה על הקודקדים של הפריים הנוכחי להבין לאן הם ממופים ואז להשתמש בפוליגן שנוצר ע"י קודקודים אלו כמסכה לפריים הנוכחי ובשאר האזורים המתים לשים פיקסלים מהפריים שאליו מיפנו.

הלכה למעשה הנסיון לעשות המרה בין פריים לפריים יצר איתו המון רעש עקב התזוזה של התמונה הפריים הנוכחי והכנסה של מסגרת התמונה באזורים השחורים (ראה איור 2 ) ולכן הפתרון שיישמנו היה לבחור 3 פריימים לאורך הסרטון במרווחי דגימה שווים ולעדכן את מציאת ההומגרפיה של התמונות שאחריהם בוידאו ביחס אליהם(ראה איור 3)


איור 2- הפריים בזמן t לאחר הפעלת הומוגרפיה עליו


איור 3- לאחר ביצוע ההומגרפיה ותיקון הקצוות השחורים.

**חיסור רקע- background subtraction**

בחלק הזה אנחנו מקבלים :

קלט- סרטון מיוצב (**Stabilized Video**) .

פלט- **binary video** שמזהה את האובייקט (**foreground**).

המטרה של הבלוק הזה היא למצוא את הסרטון הבינארי, כך שאובייקט (foreground) מסומן ב 1 והרקע מסומן ב 0.

על מנת לבצע זאת אנו נדרשים לבצע סגמנטציה לאובייקט, לתת תיוג לכל פיקסל בתמונה לרקע או אובייקט .

לשם כך אנו בונים פונקציות פילוג לרקע ולאובייקט , שבאמצעותם נוכל להחליט את הסיכוי של כל פיקסל בתמונה להיות רקע או אובייקט.

**1. שלב הoffline**

1.א מציאת הרקע -**background**

בשלב ראשון אנו מוצאים ייצוג לרקע בסרטון ,פעולה זאת מתבצעת ע"י לקיחת 50 פריימים אקראים מהסרטון ומציאת הרקע בעזרת הערך החיצוני של כל פיקסל במיקום מסוים לאורך כל הפריימים שנדגמו. מכיוון שרוב הזמן ערכי הפיקסלים מכילים את הרקע, ערך החיצון יהיה תואם לצבע של הרקע ,וכך נוכל לקבל יצוג לרקע בסרטון.


איור 4- הרקע לאחר החציון

1.ב בניית היסטוגרמות(תהליך זהה לרקע ואובייקט לאחר חילוץ תמונה)

לפני התחלת תהליך העיבוד אנחנו בונים בoffline , פונקציות פילוג לערכי הצבעים ברקע ובאובייקט על בסיס מספר פריימים שאנו דוגמים באופן אקראי מהסרטון.

הפונקציה buildhist אשר מקבלת תמונה, אחראית ליצירת ההיסטוגרמה על בסיסה,( תהליך מציאת התמונה לרקע ולאובייקט הנשלחת לפונקצייה הוא שונה עבור כל אחד ויובהר בסעיף הבא).

ההיסטוגרמה תשמש לבניית כלל החלטה ראשוני בשלב הonline , לכל פיקסל לתיוג לרקע ואובייקט כפי שיוסבר בהמשך.

לצורך הבנייה מתבצעים השלבים הבאים :

תחילה אנחנו עושים resize לגודל של התמונה פי 4 על מנת לחסוך בזמן העיבוד , וכמו כן אנחנו עושים קוונטיזציה מ255 ל64 לכל אחד מערוצי הצבע.

לאחר מכן אנחנו בונים היסטוגרמה בגודל של 64,64,64 (לכל אחד מערוצי הצבע) וממלאים אותה בהתאם לערכים של הצבעים בסרטון .

כיוון שמימד ההיסטוגרמה הוא גדול מאוד והסיכוי שפיקסל ספיציפי יפול בתא בהיסטוגרמה הוא נמוך אנחנו עושים 3d blur באמצעות guissian על פני הערכים בהיסטוגרמה שיאפשר החלקה של פונקציית הפילוג על פני תאים שכנים .
בשלב האחרון אנחנו מנרמלים את ההיסטוגרמיה לסכום של אחד על חלוקת בסכום הערכים על מנת שתוכל להוות פונקציית הסתברות .

הערות :

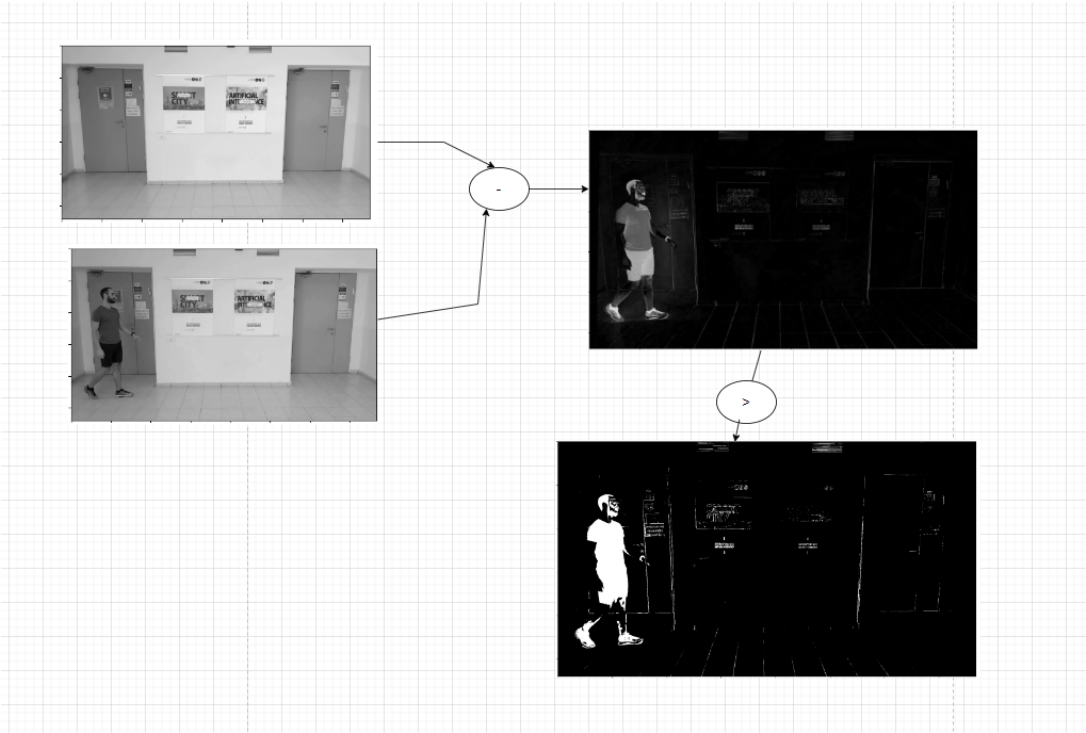
-כיוון שלopncv אין מימוש לblur 3d אנו עושים שימוש בספריה scipy.ndimage.gussian_filter אשר מבצעת טשטוש גאוסי ע"י שימוש בתכונת הספריבליות של גרעין גאוסי ( מבצעת 1d לאורך הערוצים אחד אחרי השני ) .

-בחרנו להשתמש במרחב של הצבעים *RGB* , מכיוון שההתפלגות של הצבע באובייקט שונה מהרקע.
בנוסף ניסינו גם להשתמש במרחבים שונים , אבל רמת סטורציה והתאורה לא מפרידה בין האובייקט לרקע לאורך כל התמונה לכן לא הצלחנו להשתמש במרחב נוסף ליעל את הזיהוי.

<u>1ג.בניית היסטוגרמה לאובייקט **(foreground).$P_f$**- $p(c(x)\backslash fg)$ .</u>

כאמור על מנת למצוא פונקצית פילוג של האובייקט אנו דוגמים באקראי ב offline פריימים מהוידאו ועל בסיסים מחשבים את ההיסטוגרמה במקרה של היסטוגרמה לאובייקט אנו דוגמים 10 אחוז מהפריימים בוידאו.
לאחר מכן על מנת לחלץ את המסכה אנו ממירים את הפריימים לגווני אפור ועושים חיסור בין הפריים הנוכחי לרקע שמצאנו קודם בערך מוחלט, - בשיטה זו רוב הפיסקלים שהם רקע צריכים להיות בקירוב 0 ואילו פיקסלים בהם נמצא האדם אמורים להיות עם ערך חזק יותר ולכן נוכל לחלץ מסכה לאדם בקירוב ע"י ערך סף- מעשית בגלל תזוזות לאורך הסרטון יש פיקסלים שאינם מתאפסים ברקע ,

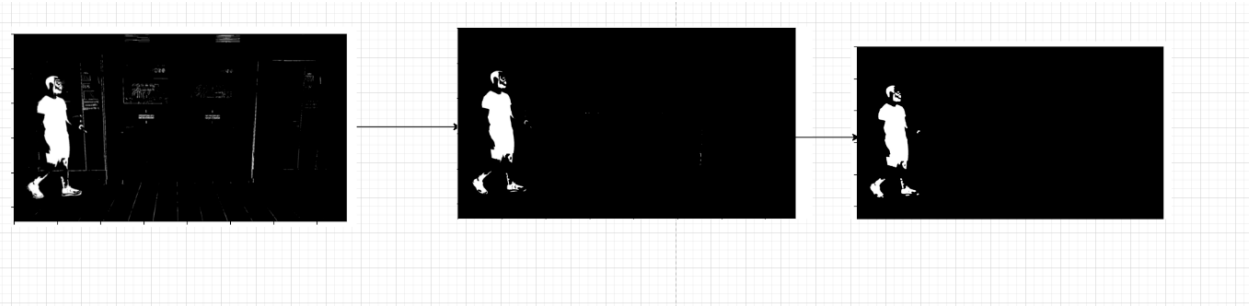
בעיה נוספת שנוצרת היא עקב ההמרה לגווני אפור יתכן כי לשתי צבעים שונים יהיה את אותו ערך אפור ועל אף שהם שונים זה מזה בrgb באפור הם יקזזו זה אתזה – אפשר להבחין בתופעה זו על פני הפיקסלים שעל האדם ,ובשל זאת נדרשות פעולות עיבוד תמונה נוספות.
יש לציין שנעשו נסיונות למצוא מרחקים שונים על מנת להבדיל בין הרקע לפריים -כמו להמיר לhsv ולחסר על הקורדינטיה של ה hue או להעביר למרחב lab ,כמו כן התבצע גם ניסון למצוא מרחק אוקלידי על פני rgb אך התוצאות לא נראו מספיק טובות ולכן נבחר לבצע את החיסור בgray.




איור 5- הפרש בין רקע לפריים וערך סף.

פעולות עיבוד התמונה הנוספות שבוצעו הן:

1. הפעלת טשטוש גאוסי עם kernel 7*7- על מנת למצע את רעשים בתמונה עם פיקסלים שהם 0 מסביבים ובכך משמעותית להוריד להם את הערך.
2. לקיחת ערך סף נוסף- לנקות פיקסלים של רעש אחרי הפעלת הגאוסיאן
3. פעולות מורפולגיות של opening- כדי לנקות רעשים אחרונים שנשארו אחרי הסינון ולנסות לסגור חורים באובייקט



איור 6- משמאל לימין הפעלת גאוסיאן-ערך סף נוסף – פעולות מורפולגיות.

לאחר שקיבלנו את המסכה הנ"ל אנו שמים בפיקסלים שבהם המסכה היא 1 את ערכי האובייקט ומוסרים את התמונה הזאת לפונקציה שמחשבת היסטוגרמה.

**ניתן לשים לב** שהאובייקט הוא לא מלא אך נקי מרעש, לכן כדי לתפוס שטחים אחרים של האובייקט תוך כדי תנועותו ברקע מה שאנו עושים הוא להרכיב היסטוגרמה שהתאים בה יהיו הערכי **מקסימום** של כל התאים משאר ההיסטוגרמות.

<u>2.שלב ה*online*-מציאת המסיכה הבינארית ווידאו של האובייקט ללא הרקע.</u>

בשלב זה אנו רצים על פני הפריימים לאורך הוידאו ובכל פריים מוצאים את מפת ההסתברות של כל פיקסל להיות רקע ולהיות אובייקט. כלל החלטה כאן הוא כלל החלטה בסייאני עם *prior* זהה לרקע ואובייקט(אין העדפה להיות רקע או אובייקט -כאמור ההיסטוגרמות מנורמלות)

$$\boldsymbol{p}(\mathrm{fg}\backslash\mathrm{c(x)}) = \frac{p(c(x)\backslash fg)}{p(c(x)\backslash fg) + p(c(x)\backslash bg)}$$

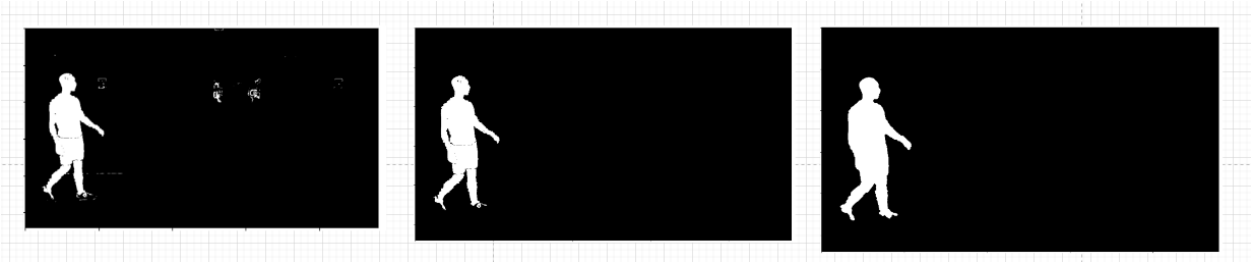
תמונת הבינארית תהיה:

$$\boldsymbol{Binary = threshold\left(P_{fg\backslash c}, \theta\right)} \qquad \boldsymbol{\theta} \in [\mathbf{0}, \mathbf{1}]$$

ערך הסף שנבחר הוא $\boldsymbol{\theta = 0.95}$

לאחר הפעלת כלל ההחלטה על הפיקסלים הבחנו שיש פיקסלים רועשים בעלי גווני צבע דומים לאובייקט לכן על מנת לעצב את המסכה הסופית ביצענו את הפעולות הבאות:

1. *getConnectedComponnet* מקבל תמונה בינארית וקושר את הפיקסלים לקבוצות לפי רמות צבע קרובות – על מנת שפקסל יהיה קשיר הוא נדרש ל*conectivity-8* כלומר ש8 השכנים שלו יהיו בעלי ערך זהה לשלו -255.
ולקיחת *max* על ה*label* עם כמות הפיקסלים הגדולה ביותר- ההנחה כאן היא שבמסכה הנוכחית האובייקט הקשיר הכי גדול בתמונה יהיה האדם.
2. פעולה של *closing* -כלומר *dilation* ואז *erosion* כדי לסגור את החורים על פני האובייקט בוצע באמצעות מסכה של **MORPH_ELLIPSE** בגודל (5,5).



איור 7- משמאל לימין הפריים לאחר ערך הסף- הפעלת connected component - הפעלת closing לסגירת חורים.

לבסוף שומרים את המסכה הסופית בוידיו בשם **Binary.avi**, ולפי המסכה הסופית אחרי העיבוד מחלצים את הפיקסלים של הפריים המקורי אחרי שעשינו הגדלה למסכה לגודלה המקורי, ושומרים אותה בוידאו בשם **Extracted.avi**.

$$\boldsymbol{Extract = BW \cdot Image_{RGB}}$$

# **Matting*:***

קלט- binary video,background image and stabilized video
פלט-matted video,alpha ,alpha-unstable and trimap
המטרה  של הבלוק הזה הוא לסדר את השפות של האובייקט ולמנוע גלישת צבע מהרקע המקורי.

את שלב זה ניתן לחלק ל3 שלבים- מציאת פונקציות פילוג הסתברות ונגזרות שלהן באמצעות *kde*
יצירת *trimap* באמצעות מפות מרחק של הסתברות, מציאת ערכי *a* על מנת למצוא את האופן הנכון ביותר לעשות *blending* בין פיקסלים בקצוות  האובייקט לבין המסיכה.

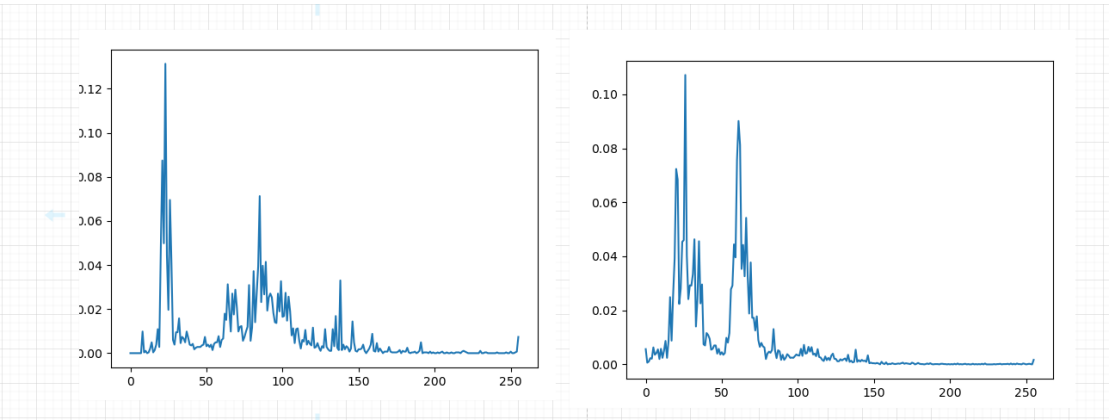
<u>הקדמה לשלבים – מציאת נקודות ברקע ובאובייקט- *scribbles*</u>
השיטה שבה נשתמש מתבססת על היכולת לשייך פיקסל לסט נתון של פיקסלים בעלי תיוג מסוים על פני סט אחר ,במקרה שלנו לשייך את הפיקסלים בתמונה לסט פיקסלים ידועים המוגדרים כרקע וסט פיקסלים ידועים המוגדרים כאובייקט .
הפיקסלים הלא ידועים כאמור הם אלו שעל השפה .
על מנת ליצור אזור כזה של אי וודאות ולדעת לעצב אותו יותר טוב עלינו להוציא אותו מתוך המפה הבינארית ולהבדיל אותו משאר הפיקסלים המייצגים אובייקט.
לשם כך העברנו גרעין גאוסי בגודל (11,11 ) על מנת שמיצע על פני הפיקסלים בשפה יחד עם פיקסלים מהרקע שערכם 0 ויקטין את ערכם ביחס לפיקסלים בתוך האובייקט אשר ישארו גם לאחר הפעלת הגואסין בעלי ערך של255  (זאת כאמור כי כל הפיקסלים הם בעלי ערך 255 וסכום המקדמים בגרעין הוא 1) .

לאחר מכן  הגדרנו שתי מסיכות `bg_mask` `fg_mask` שמבוססות על הסרטון הבינארי `binary_frame_gray`*:*

```
fg_mask[binary_frame_gray >= 240] = 255
bg_mask[binary_frame_gray <= 10] = 255
```

<u>חישוב פונקציות פילוג הסתברות ונגזרתיהן-</u>
על מנת למצוא בהמשך את מפות המרחקים של הפיקסלים מה*scribbles* , אנו נדרשים למצוא תיאור של פונקציות הפילוג של הפיקסלים ברקע ובאובייקט ,נגזרות של פונקציות אלו ישמשו אותנו כדי לחשב מרחק *geodesic* בין הפיקסלים בתמונה לבין הסט פיקסלים של הרקע והאובייקט.

את פונקציות ההסתברות אנו מייצרים באמצעות *kde* כלומר מייצרים גאוסיאן עבור כל נקודת דגימה של צבע ועושים סכום משוקלל של גאוסיאנים-כאמור נקודות הדגימה הן על בסיס המסכות של ה*scribbles*.
את ה*kde* אנחנו עושים על מרחב ה*value* לאחר המרה של קורדינטות של *hsv* .
על מנת להקטין את העלות החישובית של יצירת מפות הסתברות כאלו לכל פריים אנו עושים *resize* לגודל של התמונה פי 4 ובנוסף מסתכלים על חלון המכיל פיקסלים רק באזור של האובייקט.
ה*kde* עצמו מחושב לפי פונקציית ספריה של *scipy* בהינתן הנקודות.




איור 8- משמאל לימין – פונקציות הפילוג של האובייקט (שמאל) ושל הרקע(ימין)

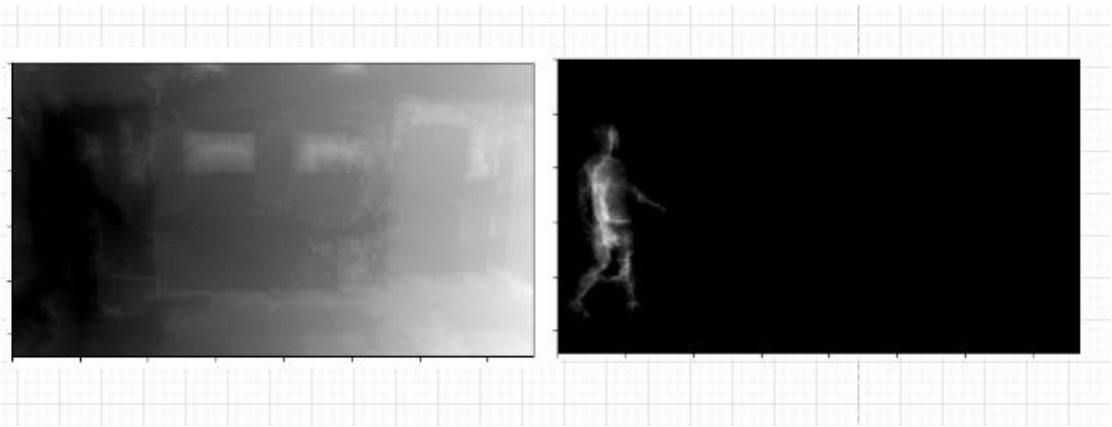
לאחר מציאת פונקציות הפילוג אנו מוצאים 2 מפות הסתברות של הרקע והאובייקט כלומר לכל פיקסל בפריים *v* של התמונה אנו שמים את הערך של ההסתברות בהתאם לצבע שלו.
ולאחר מכן לפי כלל בייס כמו בחלק של ה*backgeound substraction* מוצאים את ההסתברות שפיקסל מסוים הוא רקע או אובייקט לפי הרקע שלו -כלומר

$$\boldsymbol{p}(\mathrm{fg}\backslash\mathrm{c(x)}) = \frac{p(c(x)\backslash fg)}{p(c(x)\backslash fg) + p(c(x)\backslash bg)}$$

לאחר מציאת מפות הפילוג אנו מחשבים את הגודל של הנגזרות שלהן וממשיכים לשלב של מציאת מפת המרחקים על בסיס מפת הנגזרות של ההסתברות.

מפת המרחקים מחושבת לפי מטריקה של מרחק *geodesic* -כלומר מציאת המרחק הקצר ביותר בין פיקסל לבין סט פיקסלים (רקע או אובייקט) כאשר המרחק נקבע לפי סכום המשקולות בדרך מהפיקסל לסטים .
המשקולות כאן מייצוגיות ע"י ערכי הגרדיאנטים של הנגזרות במפות ההסתברות- כלומר אם יש שינוי בהסתברות לאורך דרך מסוימת כלומר אנו עוברים מאזור יותר סביר עבור סט מסוים לאזור פחות סביר עבור סט מסוים נשלם יותר.
החישוב של מפת המרחקים מבוצעת לפי פונקציה שנקראת *GDT* אשר מה שהיא עושה היא מאתחלת את הפיקסלים שלא חושב המרחק אליהם בערך גדול מאוד ופיקסלים באזור ה*scribbles* ב0 ומבצעת סריקה על התמונה ב4 כיוונים (סריקת *raster*) כאשר כל פיקסל מעדכן את המרחקים באופן סיבתי (כלומר כל פיקסל במקום ה*i,j* מעדכן את שכניו במקום ה*i-1,j* ו *i,j-1*) .

על מנת לייעל את החישובים של חישוב המרחק(אשר לצערנו להבדיל משפות אחרות לא קיים ב*opencv* *out of the box*) ,אנו משתמשים בספרייה שנקראת *numba* אשר עושה אופטמזציה לביצוע של לולאות.
השימוש בספריה היה הכרחי כיוון שזמן הריצה לכל חישוב מרחק בלעדיו הוא ארוך מאוד וחישוב מפת המרחקים מבוצעת לכל פריים בסרטון.(איור 9)

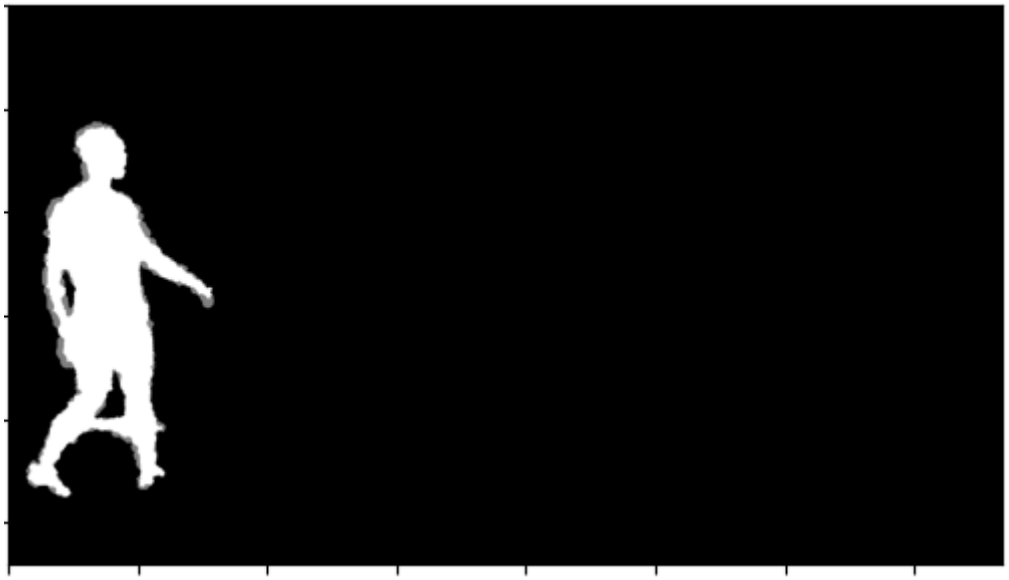


איור 9 – מפות מרחקים – מפות מרחקים מהאובייקט (שמאל) ומהרקע(ימין)

ניתן לראות שעבור פיקסלים שהם חלק מהאובייקט במפת המרחקים ברקע הם לבנים כלומר המרחק שלהם יותר גדול ואילו שאר הפקסילים בצבע שחור כלומר מרחק קטן וההפך קורה במפת האובייקט.

כעת נשתמש במפת המרחקים על מנת למצוא את ה *trimap*.
עבור פיקסלים בעלי מרחק קטן מהאובייקט יקבלו 255 ופקסילים בעלי מרחק קטן לרקע יקבלו 0 ,
פיקסלים שהם גבוליים כלומר ההפרש קטן מ $\rho$ יקבלו 128 ובכך קבלנו את סרטון ה*trimap* :

```
trimap[(fg_gdf-bg_gdf) > rho] = 0
trimap[(bg_gdf-fg_gdf) > rho] = 255
trimap[abs(bg_gdf-fg_gdf) <= rho] = 0.5*256
```



איור 10 -trimap

***alpha matting***:
באמור מטרת שלב זה הוא למצוא את ערכי *alpha* שיאפשרו לעשות*blending* לאובייקט לרקע חדש.
ערך *a* יהיה 1 עבור פיקסלים ב*trimap* שהם 255 ו0 עבור פיקסלים שהם 0 .

ועבור הפיקסלים האפורים נחשב את ערך של $\alpha$ בעזרת הנוסחאות :

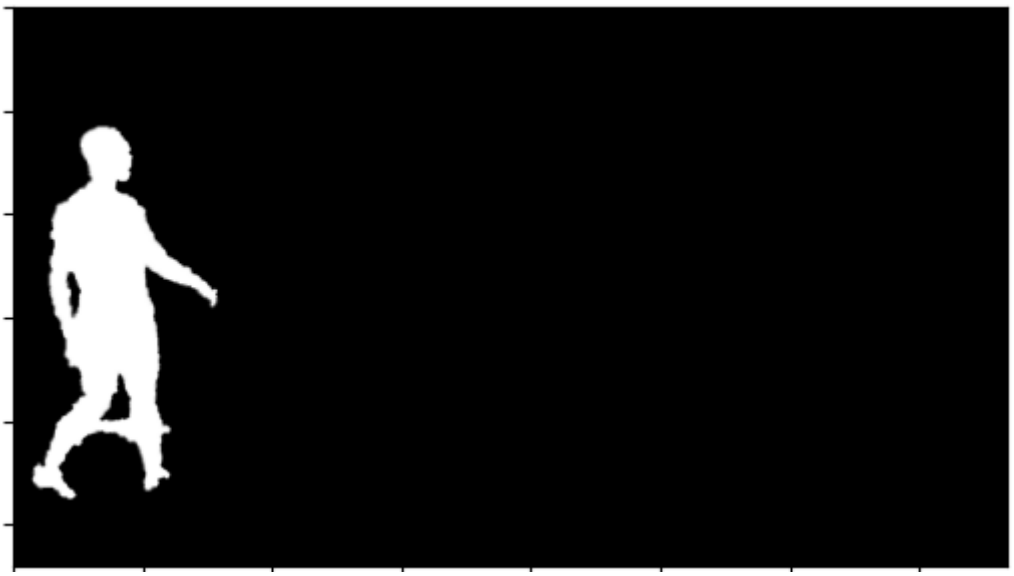
$$W_{fg} = \left(fg_{gdf}\right)^{-r} * prob_{fg}$$
$$W_{bg} = \left(bg_{gdf}\right)^{-r} * prob_{bg}$$

$$alpha = \frac{W_{fg}}{(W_{fg} + W_{bg})}$$

כאן *r* נקבע לערך של 1.2
ואז נעדכן את התמונה בעזרת המשואה כאשר *bg* היא תמונת הרקע החדשה ו *img_frame* היא התמונה המקורית:

$$frame = alpha * img_{frame} + (1 - alpha) * bg$$

ניתן לראות שכאשר $\alpha = 0$ נקבל את ערכי הפיקסלים ברקע החדש וכאשר $\alpha = 1$ נקבל את הפיקסלים מאובייקט .



איור 11 -$\alpha\ map$

איור 12 -התמונה הסופית אחרי bledning

## ***:Tracking***

קלט- *matted video,binary video*

פלט- *tracked video*

בשלב זה, נייצר פונקציה לעקיבת האובייקט בסרטון *matted video*. בכדי לחסוך את זמן הריצה וגם נקבל תוצאה מדויקת, בזכות התמונה הבינארית שקיבלנו משלב חיסור רקע, העקיבה מתבצעת באמצעות מציאת הסגמנט הכי גדול בתמונה אחרי שמצאנו את כל הסגמנטים בתמונה בעזרת פונקציית *connectedcomponentsWithstats*(), מהפונקציה אפשר לשלוף מידע לגבי כל סגמנט כמו מרכז(*c_x,c_y*), רוחב (*width*) ואורך (*height*). ואחר כך בונים את החלון לפי הפרמטרים שקיבלנו מעיבוד התמונה הבינארית. בשיטה הזאת נקבל חלון אדפטיבי לכל פריים המכיל את האובייקט לאורך הוידאו.

הערה: ניתן להריץ גם בשיטה שלמדנו בכיתה והגשנו בתרגיל בית 3 , העדפנו לעבוד לפי שיטה המוצעת למעלה בשביל לקבל תוצאה מהירה ששומרת על רמת הדיוק של העקיבה.

איור 13 -התמונה הסופית עם החלון העוקב