

## פרויקט גמר בתכנ הנדסי

### מערכת לזיהוי בני אדם

הוגש לשם מילוי חלקי של הדרישות לקבלת התואר

בוגר במדעים " B.Sc. " בהנדסת חשמל ואלקטרוניקה

תאריך הגשה: 30/9/2025

### תקציר:

פרויקט זה בוצע במסגרת תכן הנדסי פנימי (פרויקט גמר) ב"מכללת בראודה" בכרמיאל.

מטרת הפרויקט היא לפתח מערכת חכמה לזיהוי ומעקב אחר בני אדם בזמן אמת, המשלבת טכנולוגיות מתקדמות של למידת מכונה, עיבוד תמונה, ושליטה במצלמה ממונעת.

הפרויקט מהווה אב-טיפוס למערכת אבטחה אינטראקטיבית וניידת, המבוססת על פלטפורמת Raspberry Pi 4B. המערכת מאפשרת זיהוי אוטומטי של אדם בפריים, ביצוע מעקב פיזי באמצעות סרוו, שליחת התראות אוטומטיות (כולל מייל), הקלטת וידאו, והצגת ממשק משתמש גרפי מלא.

המערכת מיועדת ליישומים כמו אבטחת בתים, מעקב אחר כניסת אנשים לאזורים רגישים, ניטור סביבתי חכם, ואף לשילוב בכלים בהם שדה ראייה מוגבל או יש מגבלות בטיחות. הרעיון לפיתוחה עלה מתוך צורך גובר למערכות בטיחות חכמות שיכולות לזהות בני אדם גם בתנאים של שדה ראייה מוגבל או צורך במעקב רציף – בצורה קומפקטית, יעילה, ובמחיר נמוך יחסית.

מערכת זו משתמשת במצלמה המחוברת ל-Raspberry Pi, במודל זיהוי עצמים מסוג TFLite MobileNet-SSD, ובמנוע סרוו שנשלט בהתאם למיקום האדם בתמונה. בנוסף, כוללת המערכת חיישן מרחק, רכיב התראה קולית (באזר), קלט מהמשתמש באמצעות GUI, ואפשרות לשליחת התראות בזמן אמת.

## תוכן עניינים

4.....	1. מבוא:
5.....	2.1 הגדרה פונקציונאלית :
5.....	2.2 מפרט פונקציונלי:
6.....	3.2 מפרט טכני:
7.....	2.4 תרשים מלבנים:
8.....	2.5 עקרון פעולת המערכת:
9.....	3. מטלות:
9.....	3.1 מטלות הנדסיות:
9.....	3.2 שלבי תכנון – תכנון אב:
20.....	3.3 תרשים זרימה של המערכת:
21.....	3.4 סכמה חשמלית ותרשים זרימה של המערכת:
22.....	3.5 בעיות הנדסיות:
23.....	3.6 הוראות הפעלה:
24.....	4. סיכום ודיון:
24.....	4.1 סיכום:
25.....	4.2 עמידה במטלות:
25.....	4.3 הצעות לשיפור :
26.....	5. ביבליוגרפיה:
27.....	6. נספחים:

## תוכן טבלאות

6.....	טבלה 1: מפרט טכני
25.....	טבלה 2: עמידה במטלות

## תוכן איורים

7.....	איור 1: תרשים בלוקים
11.....	איור 2: תקשורת CSI-2
12.....	איור 3: תקשורת I2C
13.....	איור 4: תרשים זרימה עבור HC-SR04
13.....	איור 5: דיאגרמת זמנים HC-SR04
14.....	איור 6: גרף תזמון סיגנל PWM למנוע סרוו
15.....	איור 7: תרשים זרימה עבור מצבי פעולה של מנוע סרוו
17.....	איור 8: תרשים זרימה של אלגוריתם המעקב
18.....	איור 9: ממשק המשתמש הגרפי (GUI)
20.....	איור 10: פרוטוקול SMTP
20.....	איור 11: תרשים זרימה של המערכת
21.....	איור 12: סכמה של מעגל הפעלה ל-IR LED במערכת הצילום

21.....	איור 13 : סכמה חשמלית של מעגל בקרת IR-CUT Filter
22.....	איור 14 : סכמה חשמלית של המערכת
24.....	איור 15 : מסך ה-GUI של המשתמש

## רשימת קיצורים :

SMTP-Simple Mail Transfer Protocol  
 GUI- Graphical User Interface  
 FPS- Frames Per Second  
 TFLite- TensorFlow Lite  
 SSD (Single Shot Detector)(מכונה למידת מכונה)  
 SSH- Secure Shell  
 VNC- Virtual Network Computing  
 LDR- Light Dependent Resistor  
 CSI- Camera Serial Interface  
 IC- Inter-Integrated Circuit  
 PWM- Pulse Width Modulation  
 TLS- Transport Layer Security  
 SBC- Single Board Computer  
 LED- Light Emitting Diode

## 1.מבוא:

בעידן המודרני, מערכות אבטחה חכמות הפכו לכלי מרכזי בהגנה על אנשים ורכוש, תוך שילוב טכנולוגיות מתקדמות לזיהוי בזמן אמת. בפרויקט זה פותחה מערכת חכמה מבוססת Raspberry Pi 4B, המסוגלת לזהות נוכחות של בני אדם, לעקוב אחר תנועתם, לשלוח התראות ולהקליט את המתרחש, כל זאת תוך שימוש בממשק משתמש נגיש ויכולת תפעול בזמן אמת. המערכת מתאימה ליישומים מגוונים, בהם אבטחת אזורים רגילים, סיוע במעקב אחר תנועה במתקנים חכמים, או שילוב בכלים אוטונומיים הזקוקים ליכולת תגובה מהירה לסביבה אנושית.

הצורך במערכת מסוג זה נובע מהחשיבות הרבה בזיהוי מדויק של תנועה אנושית בזמן אמת, במיוחד כאשר אין השגחה אנושית רציפה או כאשר מדובר בסביבה מאתגרת לזיהוי. המערכת עושה שימוש בלמידת מכונה (TFLite), עיבוד תמונה (OpenCV ו-picamera2), שליטה על מנוע סרוו לצורך תנועה פיזית של המצלמה, וממשק גרפי (Tkinter) המאפשר שליטה פשוטה ונוחה. שילוב רכיבים אלה יוצר פתרון הנדסי חכם, יעיל, ומתואם לצרכים המשתנים של סביבות אבטחה מודרניות.

## 2. תיאור המערכת:

### 2.1 הגדרה פונקציונאלית :

המערכת נועדה לאפשר זיהוי, מעקב והתראה על נוכחות של אדם סמוך למצלמה באמצעות שילוב של עיבוד תמונה, ממשק גרפי, למידת מכונה ורכיבי חומרה.

### 2.2 מפרט פונקציונלי:

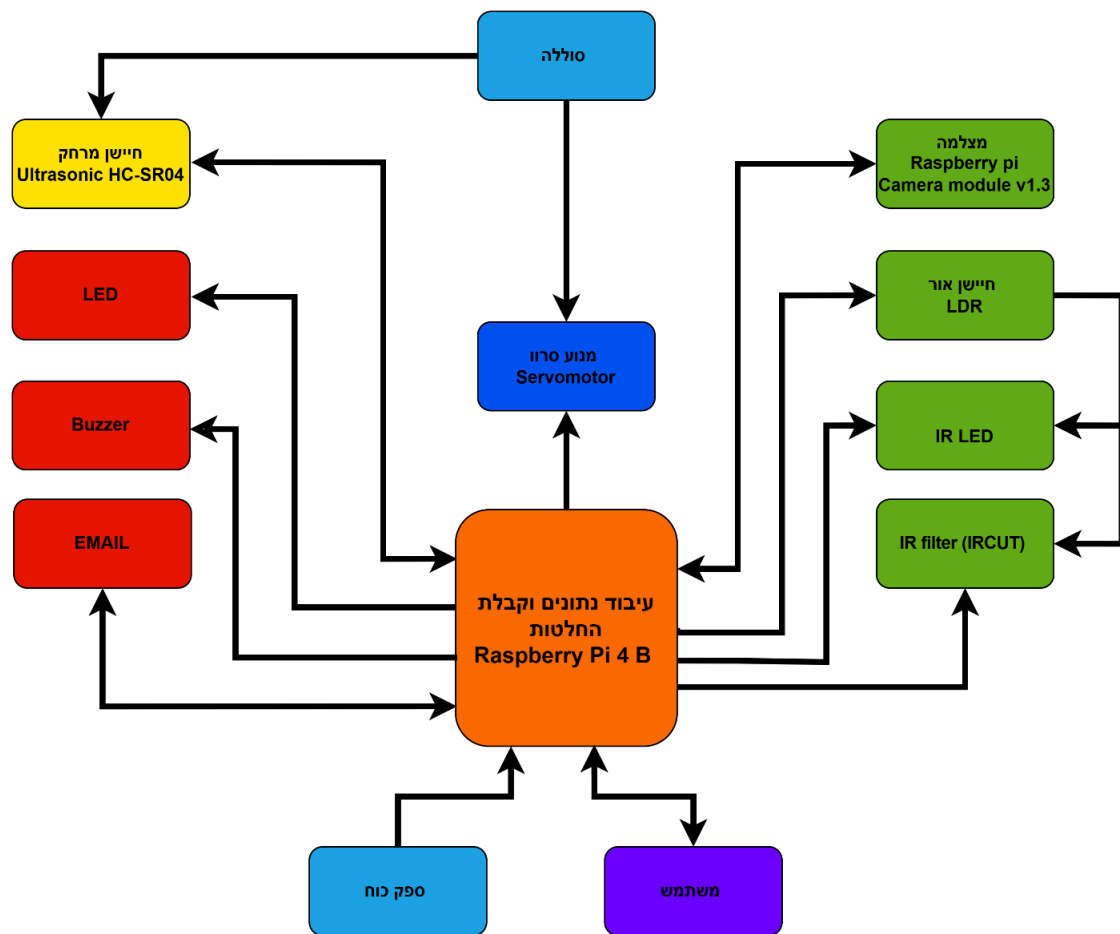
- יכולת עיבוד נתונים וקבלת החלטות.
- יכולת זיהוי בני אדם
- יכולת מעקב אוטומטי
- יכולת סריקת הסביבה ידנית/אוטומטית
- יכול מדידת מרחק המשתמש במצב מעקב והצגתו במסך
- יכולת הפעלת התראה בזמן אמת
- יכולת שליחת התראה למשתמש בעזרת פרוטוקול SMTP
- יכולת שליטה ובקרה גרפית
- יכולת קליטת מצב תאורה ופעולה במצב לילה
- יכולת צילום והקלטה

## 2.3 מפרט טכני:

מכלולי המערכת	תיאור המכלול	פרמטרים
מכלול הבקרה, עיבוד הנתונים וקבלת ההחלטות-  Raspberry Pi 4 Model B	המכלול אחראי על תפעול לוגיקת המערכת בזמן אמת, עיבוד הנתונים מהמצלמה, חיישן המרחק ושאר הרכיבים, וקבלת החלטות בהתאם לנתונים שהתקבלו.	מערכת הפעלה LINUX 64-bit זכרון SSD חיצוני 4 GB RAM מעבד Cortex-A72 מהירות שעון המעבד – 1.5 GHZ
מכלול צילום - Raspberry Pi Camera Module OV5647 v1.3	צילום הסביבה בזמן אמת ושמירת וידאו מהמצלמה לאחסון חיצוני	Resolution : 5 megapixel Night vision: Support IR light 1W IR filter (IRCUT): Auto switch
מכלול תצוגה	הצגת הנתונים ושליטה	ממשק גרפי (GUI) על גבי מסך
מכלול סריקה	יכולת תנועה של המצלמה לצורך ביצוע סריקה אוטומטית או ידנית של הסביבה ולטובת מעקב אחר אנשים באזור רחב יותר.	MG996R Servo Motor  <ul style="list-style-type: none"> <li>Stall torque: 9.4 kgf·cm (4.8V), 11 kgf·cm (6 V)</li> <li>Running Current~ 500mA.</li> <li>Operation Frequency 50HZ</li> <li>Maximum Rotation Angle 180°</li> </ul>
מכלול התראה	שליחת הודעת התראה למשתמש והפעלת התראה קולית וחזותית בעת זיהוי של אדם שנמצא קרוב למצלמה.	Buzzer LED שליחת התראה בעזרת פרוטוקול SMTP
מכלול מרחק - ultrasonic sensor hc-sr04	חישוב המרחק מכל אובייקט מסוים ובמצב מעקב נקבל את המרחק של האדם מהמצלמה.	Working Voltage 5 V Working Current 15mA Working Frequency 40Hz Range ~ 4m-2cm

טבלה 1: מפרט טכני

## 2.4 תרשים מלבנים:



איור 1 : תרשים בלוקים

מקרא:

צבע כתום : מכלול הבקרה, עיבוד הנתונים וקבלת ההחלטות-

צבע ירוק : מכלול צילום

צבע אדום : מכלול התראה

צבע כחול : מכלול סריקה

צבע צהוב : מכלול מרחק

צבע תכלת : מקורות מתח

## 2.5 עקרון פעולת המערכת:

המערכת מבוססת על יחידת עיבוד מרכזית מסוג Raspberry Pi 4B, המחוברת למודול מצלמה. מטרת המערכת היא לזהות אנשים בזמן אמת ולהציג את המידע למשתמש בצורה חכמה ואינטואיטיבית.

### תהליך הזיהוי:

- המצלמה מצלמת וידאו חי, ונתוני התמונה מועברים ל-Raspberry Pi.
- כל תמונה מנותחת באופן עצמאי על ידי מודל למידת מכונה קל ויעיל מסוג MobileNetV1-SSD, הפועל על גבי ה-Raspberry Pi באמצעות TensorFlow Lite.
- כאשר אדם מזוהה, נוצרת סביבו Bounding Box.
- המערכת מחשבת את שטח התיבה כדי להעריך את קרבת האדם למצלמה (ככל שהתיבה גדולה יותר – האדם קרוב יותר).

### תגובה לזיהוי:

- כאשר האדם רחוק, המסגרת סביבו תהיה ירוקה.
- כאשר האדם קרוב, המסגרת תהיה אדומה עם כיתוב "WARNING" על המסך.
- במצב של קרבה, המערכת תפעיל:
  - א. התראה קולית (זמזום)
  - ב. התראה חזותית (לד)
  - ג. שליחת מייל למשתמש דרך פרוטוקול SMTP
  - ד. תקליט את המתרחש ותשמור אותו בזיכרון SSD חיצוני

### שליטה וממשק משתמש:

- שליטה על המצלמה דרך ממשק גרפי (GUI) עם 3 מצבים:
  - א. סריקה ידנית – המשתמש שולט ישירות על כיוון המצלמה.
  - ב. סריקה אוטומטית – המצלמה נעה מצד לצד לסריקת הסביבה.
  - ג. מעקב אוטומטי – כאשר אדם קרוב מזוהה, המצלמה מסתובבת אוטומטית לעברו.
- אפשרות למשתמש להקליט את המתרחש.
- אפשרות למשתמש לבחור לאיזה מייל לשלוח את ההתראה.
- אפשרות לשים במצב השתק את הזמזום.

### מדידת מרחק:

- המערכת מציגה מרחק בזמן אמת מכל אובייקט.



- במצב מעקב, יוצג גם המרחק המדויק של האדם מהמצלמה באמצעות חיישן אולטרסוני (HC-SR04).

### 3. מטלות:

#### 3.1 מטלות הנדסיות:

- כתיבת הצעת פרויקט.
- עיבוד תמונה ובדיקת מודלים שונים, במטרה להשיג FPS גבוה ביותר
- תכנון ובניית ממשק גרפי ידידותי למשתמש (GUI).
- פיתוח אלגוריתם לזיהוי מטרות והערכת מרחקן מהמצלמה.
- תכנון מערכת התראה משולבת .
- תכנון מערכת לתנועת המצלמה.
- מימוש המד מרחק.
- אינטגרציה של כלל רכיבי החומרה והתוכנה למערכת אחת מתואמת.
- בדיקות מערכת בתנאי סביבה שונים להערכת ביצועים ועמידות.
- איתור תקלות ותיקונם.
- כתיבת ספר פרויקט + מצגת.

#### 3.2 שלבי תכנון – תכנון אב:

##### סביבת פיתוח:

נבחרה סביבת פיתוח Visual Studio Code בשפת Python על גבי מערכת ההפעלה Linux (מערכת Raspberry Pi OS), עקב היכרות מוקדמת עם השפה והניסיון שנצבר בקורסים קודמים.

Python כוללת מגוון רחב של ספריות יעודיות לעיבוד תמונה וללמידת מכונה (OpenCV, TensorFlow Lite, Numpy, Tkinter) אשר משתלבות בצורה נוחה עם מצלמת ה-Raspberry Pi ומאפשרות פיתוח מהיר, יעיל ואינטואיטיבי.

## בחירת מודל:

לשם ביצוע זיהוי בזמן אמת, נבחר מודל MobileNetV1-SSD, מודל קל ויעיל המותאם למערכות בעלות משאבים מוגבלים כמו Raspberry Pi. המודל מאומן לזיהוי אובייקטים, ובפרויקט זה הוגדר לזיהוי בני אדם בלבד. MobileNetV1-SSD משלב בין שתי גישות עיקריות:

- א. MobileNet – רשת נוירונים קלה ומודולרית, אשר משתמשת בטכניקת קונבולוציה יעילה (Depthwise Separable Convolution) כדי להפחית באופן משמעותי את מספר החישובים הנדרשים, תוך שמירה על דיוק סביר.
- ב. SSD – שיטה לזיהוי אובייקטים בתמונה ב"זיהוי יחיד", ללא צורך בשלבים מרובים. המודל מסוגל לזהות מספר אובייקטים בפריים אחד במהירות גבוהה.

המודל פועל על תמונות בגודל  $300 \times 300$  פיקסלים, ומחזיר תוצאות הכוללות את מיקום האובייקט (Bounding Box), ציון הסתברות, וסיווג הקטגוריה (במקרה זה – "אדם"). המימוש בפרויקט נעשה באמצעות TensorFlow Lite, ספרייה המאפשרת הרצה של מודלי למידת מכונה בצורה אופטימלית על מערכות משובצות מחשב. השימוש במודל זה אפשר שמירה על ביצועים גבוהים (FPS), תגובה מהירה, ותאימות לחומרה הקיימת.

## בקרים:

המערכת נשענת על Raspberry Pi 4B (SBC) כבקר ראשי, אשר מבצע את כלל פעולות העיבוד, הבקרה והאינטגרציה בין רכיבי המערכת השונים.

הבחירה ב-Raspberry Pi 4B נובעת ממאפייניו החזקים:

- מעבד מרובע ליבות (Quad-Core) – מאפשר הרצת מספר תהליכים במקביל, תומך בביצועים גבוהים הדרושים לעיבוד תמונה ולמידת מכונה.
- זיכרון RAM בנפח 4GB – מאפשר טעינת מודל למידת מכונה, עיבוד תמונה והפעלת ממשק משתמש במקביל בצורה יציבה ומהירה.
- חיבוריות מלאה – כולל Wi-Fi, יציאות USB ו-GPIO לשליטה ברכיבים חיצוניים.
- תמיכה בסביבות פיתוח מתקדמות – כגון Python, OpenCV, TensorFlow Lite ועוד.
- יכולת שליטה מרחוק – דרך פרוטוקולי VNC, SSH, וממשק גרפי (GUI).

כל אלו יחד מאפשרים גמישות תכנותית גבוהה והתאמה מיטבית של חומרה ותוכנה בזמן אמת.

## תכן חומרה:

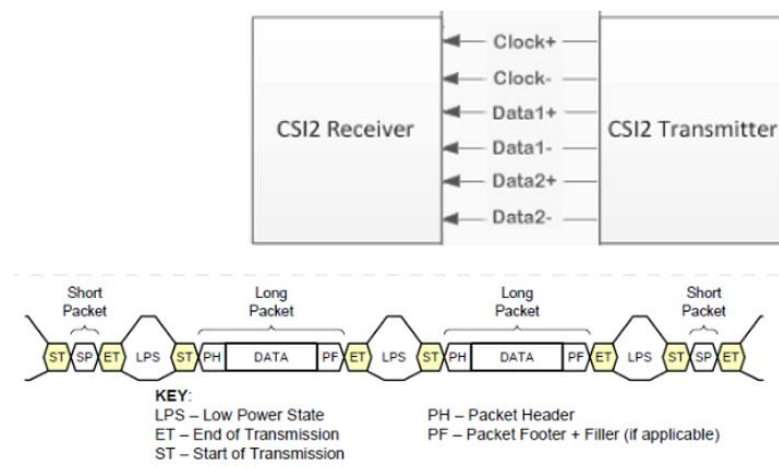
- זמזום (Buzzer) ו-LED- משמשים כאמצעי התרעה ויזואליים וקוליים מקומיים כאשר מזוהה אדם קרוב מדי למצלמה. הפעלת הזמזום ונורת ה-LED נעשית על ידי הבקר (Raspberry Pi) באמצעות ממשקי GPIO, כאשר מתקבל "מצב אזהרה".

השימוש ברכיבים אלה מאפשר תגובה מידית גם ללא חיבור לרשת או שליחת מייל – מה שהופך את המערכת ליעילה גם במצבים של שימוש מקומי או באזורים מבודדים.

- מצלמה מסוג OV5647+ IR-Cut Filter & IR LEDs – מצלמת 5 מגה-פיקסל ייעודית ל-Raspberry Pi. המצלמה שנבחרה היא OV5647 IR – מצלמת 5 מגה-פיקסל ייעודית ל-Raspberry Pi. מדובר בגרסה הכוללת פילטר IR-Cut נשלף באופן אוטומטי באמצעות מנגנון מכני-אלקטרומוגנטי (שכולל סליל), וכן תאורת אינפרא-אדום (IR LEDs) המאפשרת ראייה בתנאי תאורה משתנים, כולל חשיכה מוחלטת.

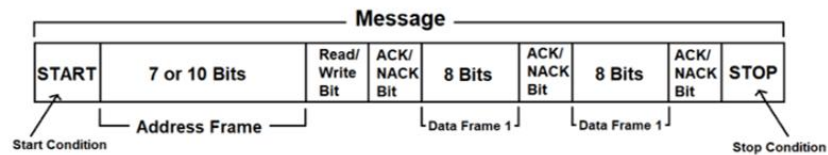
השליטה בהפעלת התאורה ובמעבר בין מצבי היום והלילה מתבצעת באמצעות חיישן LDR, שחיישניו משנים את התנגדותם בהתאם לעוצמת האור. בכך, המצלמה מתאימה עצמה אוטומטית לתנאי תאורה סביבתיים. המצלמה מאפשרת צילום וידאו בזמן אמת, גם בתנאי תאורה נמוכים, ומספקת למערכת היכולת לזהות בני אדם גם בשעות הלילה.

- חיבורים ופרוטוקולי תקשורת:  
המצלמה מתממשקת עם ה-Raspberry Pi דרך ממשק CSI, אשר מבוסס על פרוטוקול MIPI CSI-2. פרוטוקול זה מאפשר העברת תמונה (פיקסלים) מהמצלמה ל-Raspberry Pi בצורה מהירה ואפקטיבית, לצורך עיבוד בזמן אמת.



איור 2: תקשורת CSI-2

והבקר מתממשק למצלמה באמצעות פרוטוקול I<sup>2</sup>C  
תפקידו לשלוח פקודות בקרה מה-Raspberry Pi למצלמה.



איור 3 : תקשורת I2C

## • חיישן מרחק HC-SR04

ה-HC-SR04 הוא חיישן מרחק אולטרסוני בעל טווח בין 2cm ל-400cm, אשר משמש למדידת מרחקים בין החיישן לאובייקטים הנמצאים מולו.

החיישן פועל באמצעות שליחת פולס של  $10\mu s$  ל-Trigger בעקבות כך החיישן שולח גם קולי בתדר של כ-40 קילו-הרץ (8 מחזורים), ומעלה את ה-Echo לרמת לוגי "1", משך הזמן שבו הפין Echo נמצא ברמה לוגית "1" מייצג את זמן המעבר של הגל קול מהחיישן אל האובייקט וחזרה לחיישן, לפי כך נוכל לחשב את המרחק של המצלמה מאובייקט מסוים באופן הבא :

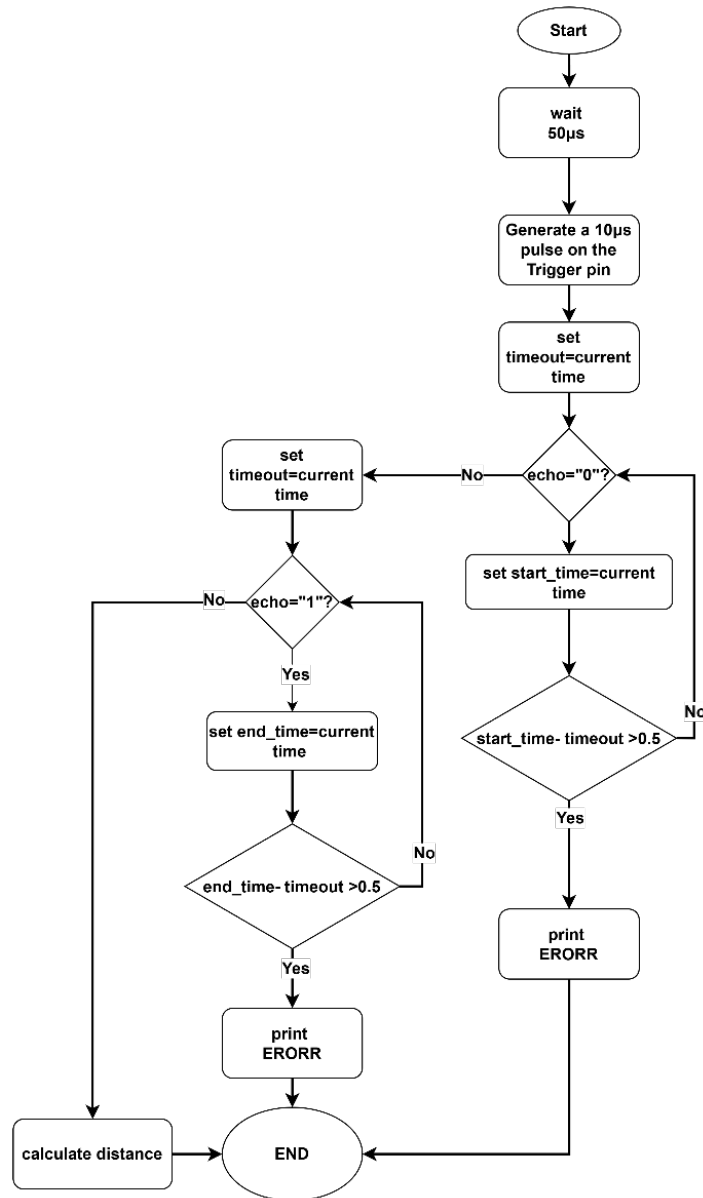
$$\Delta t = t_F - t_S$$

$$distance = \frac{\Delta t \cdot V}{2} = \frac{\Delta t \cdot 34300}{2}$$

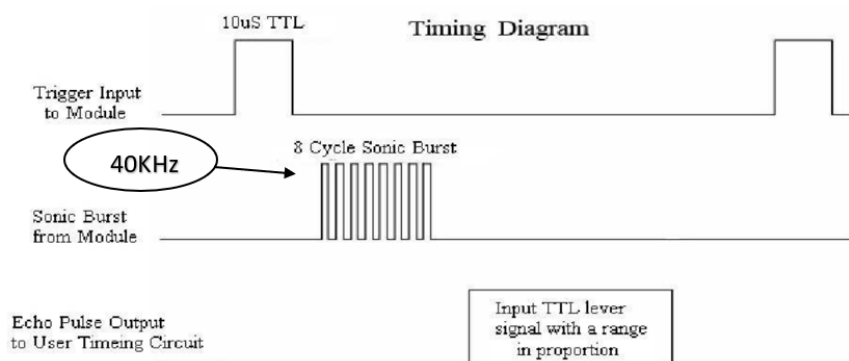
$\Delta t$  - הזמן מרגע יציאת הגל ועד חזרתו לחיישן

$V$  - מהירות הקול באוויר ביחידות [cm/sec]

$distance$  - תוצאה המרחק היא ביחידות [cm]



איור 4 : תרשים זרימה עבור HC-SR04



איור 5 : דיאגרמת זמנים HC-SR04

• מנוע סרוו MG996R-

מנוע הסרוו MG996R משמש בפרויקט להטיית המצלמה בזוויות שונות, במטרה לאפשר סריקה רחבה של הסביבה ויכולת מעקב אחר אדם המתקרב למצלמה.

המנוע מאפשר סיבוב של עד  $180^{\circ}$  מעלות, בעל מומנט מרבי של  $10 \text{ kgf} \cdot \text{cm}$ , ופועל במתח של  $6 \text{ V}$  וולט.

בחירת מנוע זה נבעה מהצורך במנוע עמיד וחזק מספיק כדי להתמודד עם העומס של תנועת המצלמה, תוך שמירה על תגובה מהירה ומדויקת לפי דרישות המערכת. הסרוו מספק תנועה חלקה ויציבה, שתורמת לדיוק בזיהוי ובמעקב.

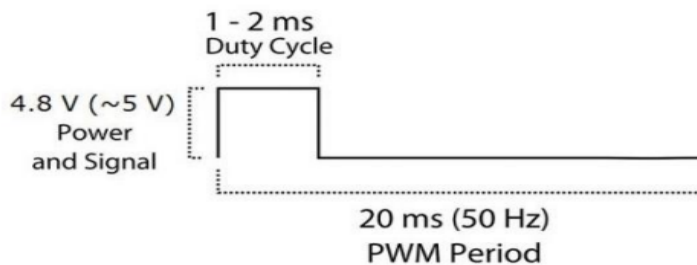
המנוע נשלט באמצעות אות PWM, הפועל בתדר של  $50 \text{ Hz}$ . המיקום הזוויתי של הסרוו נקבע לפי רוחב הפולס:

א. פולס של  $1 \text{ ms}$  מייצג זווית של  $90^{\circ}$  שמאלה ( $-90^{\circ}$ )

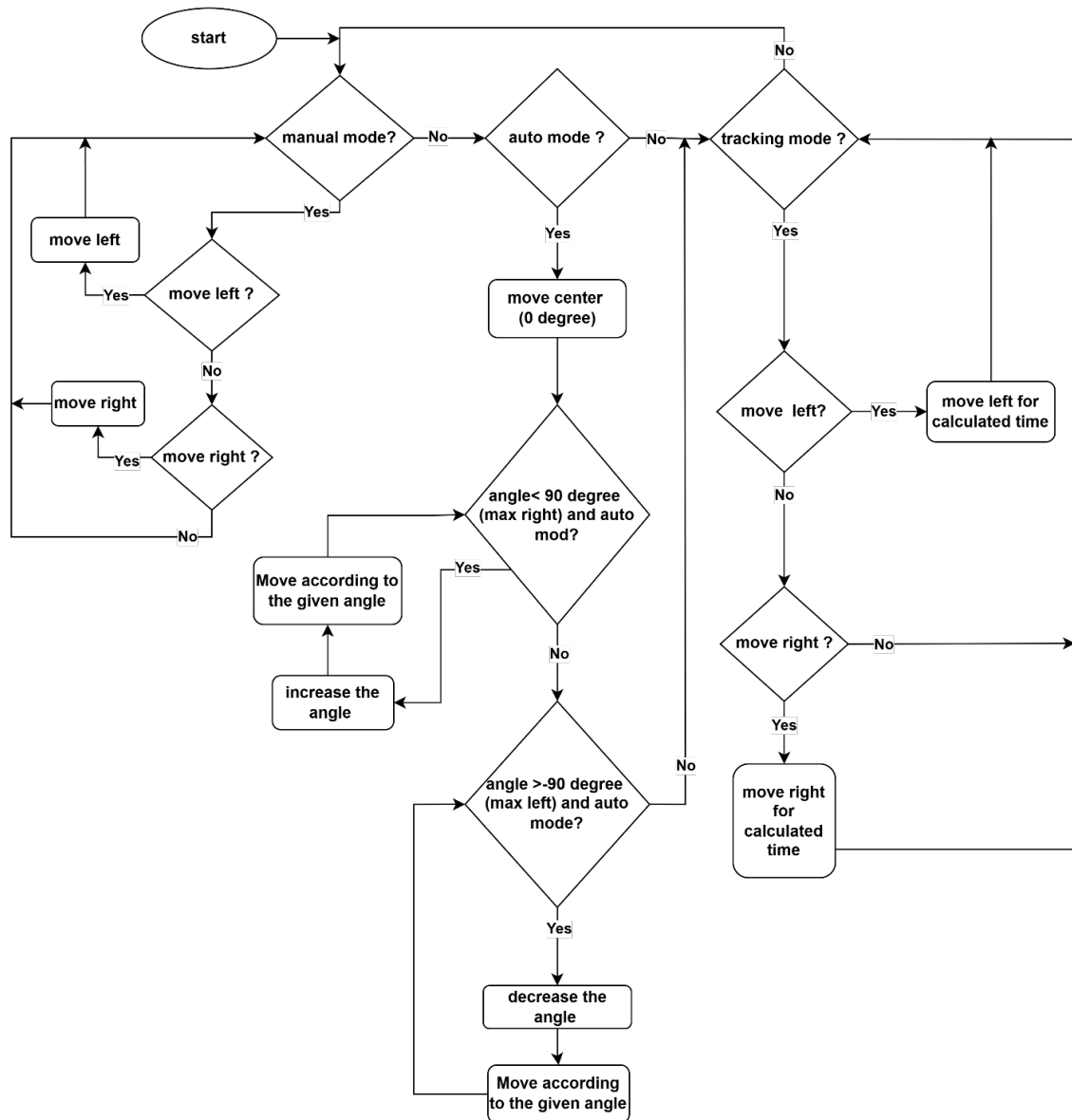
ב. פולס של  $1.5 \text{ ms}$  מייצג זווית אמצע ( $0^{\circ}$ )

ג. פולס של  $2 \text{ ms}$  מייצג זווית של  $90^{\circ}$  ימינה ( $+90^{\circ}$ )

באמצעות שינוי רוחב הפולסים, ניתן לשלוט על כיוון המצלמה בזמן אמת – הן במצב סריקה אוטומטית, מעקב והן לפי בחירת המשתמש דרך הממשק הגרפי.



איור 6: גרף תזמון סיגנל PWM למנוע סרוו



איור 7 : תרשים זרימה עבור מצבי פעולה של מנוע סרוו

## תכן תוכנה:

- קוד עיבוד תמונה ולמידת מכונה:

הוא החלק המרכזי של המערכת, האחראי על זיהוי בזמן אמת של בני אדם בתמונה המצולמת ממצלמת ה-Raspberry Pi. בתחילת הקוד מוגדרים פרמטרים התחלתיים לעיבוד התמונה, הכוללים:

א. רזולוציית התמונה (480x480) לתצוגה.

ב. פורמט צבע RGB בעל 3 ערוצים.

ג. הפעלת המצלמה והגדרת תצורתה.

בהמשך נטען מודל למידת מכונה בפורמט TFLite מסוג MobileNetV1 SSD שאומן מראש לזיהוי עצמים, וניתן לו קלט בגודל קבוע של 300x300 פיקסלים. הקוד מגדיר את משתני הקלט והפלט של המודל לצורך ביצוע חישובים בהמשך.

בעת ריצת התוכנית, המערכת מבצעת את הצעדים הבאים:

א. לכידת פריים מהמצלמה.

ב. שינוי גודל התמונה לגודל הקלט של המודל (300x300)

ג. המרת הפורמט למבנה שהמודל דורש

ד. הרצת זיהוי וקבלת תוצאות הזיהוי, הכוללות:

- תיבות תחומות (bounding boxes)
- מזהי מחלקות (class IDs)
- ורמת ביטחון (confidence scores)

אם זוהה אובייקט מסוג "person" עם רמת ביטחון מעל 50%, נוצרת מסגרת סביב האדם. בנוסף, מחושב שטח התיבה – ואם הוא גדול מסף שנקבע מראש, המערכת מפעילה מצב אזהרה שכולל הצגת טקסט "WARNING" על המסך והדגשת המסגרת בצבע אדום. לבסוף, כל פריים מעובד ומומר לפורמט המתאים לתצוגה ב-GUI של המשתמש.



- קוד מעקב : חלק זה אחראי על מעקב אחר אדם שנמצא קרוב למצלמה, במטרה למרכז אותו בתמונה על ידי סיבוב המצלמה באמצעות סרוו. בעת זיהוי אדם בתמונה, מתבצע חישוב של נקודת המרכז של המסגרת (Bounding Box) באופן הבא :

$$X_{center} = \frac{X_{max} - X_{min}}{2}$$

חישוב זה מייצג את מיקום האובייקט בממד האופקי (X), ונותן אינדיקציה האם האדם נמצא במרכז התמונה או שיש להזיז את המצלמה.

המערכת מגדירה מראש נקודת אמצע של התמונה מסביב לנקודה זו קיים טווח של  $\pm 25$  פיקסלים, שבתוכו האדם נחשב ממורכז ואין צורך להזיז את המצלמה.

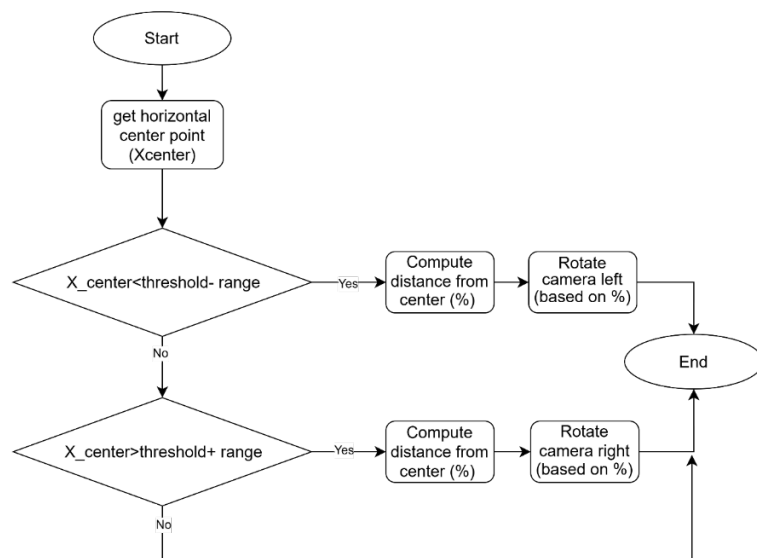
אם מרכז המסגרת (center\_x) חורג מטווח זה :

מבוצע חישוב אחוזי הסטייה מהמרכז באופן הבא :

$$p = \begin{cases} 1 - \frac{X_{center}}{threshold - range}, & X_{center} < threshold - range \\ \frac{X_{center}}{threshold + range} - 1, & X_{center} > threshold + range \end{cases}$$

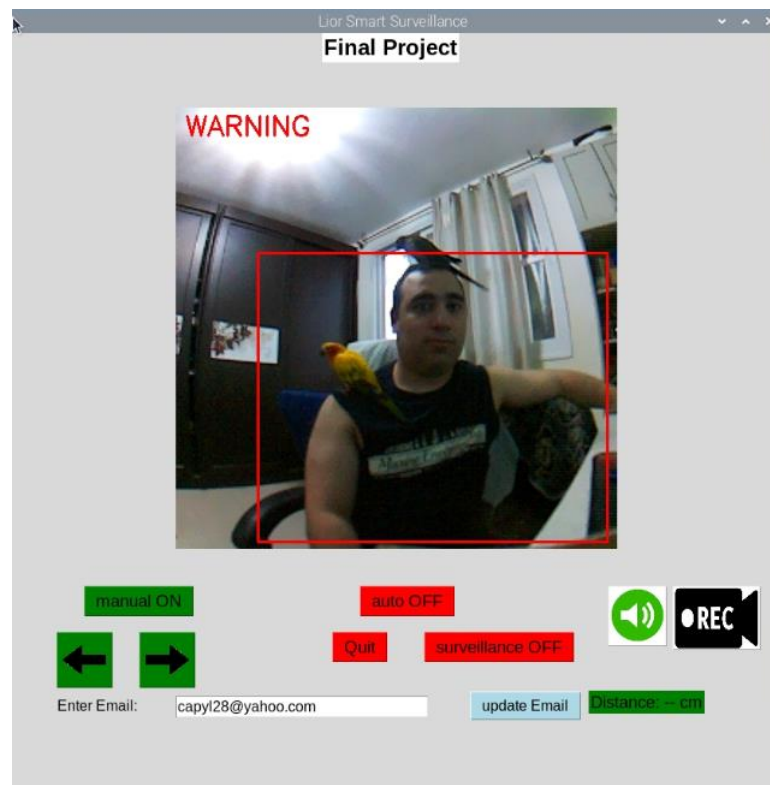
$$time = const \cdot (1 + P)$$

בהתאם לחישוב זה מתבצע סיבוב של הסרוו (ימינה או שמאלה) על ידי קריאה לפונקציה שאחראית לתנועת הסרוו, כך ניתן לשלוט בצורה דינמית על תנועת המצלמה כדי לשמור על האדם בתוך מרכז שדה הראייה, מבלי שהמשתמש יתערב.



איור 8 : תרשים זרימה של אלגוריתם המעקב

- קוד של ממשק גרפי GUI : זהו החלק שמציג למשתמש את הנתונים שהתקבלו ומאפשר לו לשלוט על המערכת חלק זה אחראי על תצוגה גרפית ידידותית למשתמש ועל שליטה בזמן אמת במצבי המערכת. הממשק הגרפי נבנה באמצעות ספריית Tkinter, ומאפשר למשתמש להפעיל ולכבות מצבים שונים, להכניס פרטי תצורה, ולהציג את התמונה מהמצלמה בלייב.  
מטרות הממשק :
  - הצגת וידאו בזמן אמת מהמצלמה.
  - שליטה ידנית על תנועת המצלמה (ימינה/שמאלה).
  - הפעלה/כיבוי של מצבי פעולה שונים (ידני, אוטומטי, מעקב).
  - הפעלה/כיבוי של הזמזום.
  - התחלת/עצירת הקלטה.
  - הזנת כתובת אימייל לקבלת התראות.
  - הצגת מרחק מהאובייקט הקרוב.



איור 9 : ממשק המשתמש הגרפי (GUI)

• הקלטת וידאו :

מערכת ההקלטה אחראית על תיעוד מצולם של האירועים, בין אם ביוזמת המשתמש (הקלטה ידנית), ובין אם בזיהוי אוטומטי של נוכחות אדם קרוב למצלמה.

לצורך ההקלטה, נעשה שימוש במקודד מסוג MJPEG וביכולת ההקלטה של ספריית Picamera2, אשר תומכת בהקלטה בפורמט MJPEG לקובץ.

הבחירה במקודד MJPEG בוצעה מכיוון שהוא דורש משאבי עיבוד נמוכים יחסית לעומת מקודדים אחרים (כגון H.264), ואינו מצריך דחיסה מורכבת בזמן אמת. כל פריים מקודד כקובץ JPEG עצמאי, דבר שמפשט את תהליך ההקלטה, מאפשר עיבוד מהיר, ומפחית את העומס על המעבד.

• ניהול תהליכונים (Threads) עם מנעולים (Locks) :

בפרויקט זה נעשה שימוש בתהליכונים (Threads) על מנת לאפשר הרצה מקבילית של רכיבי המערכת, כגון :

א. עיבוד תמונה וזיהוי בזמן אמת.

ב. שליטה במנוע הסרוו לשליטה אוטומטית.

ג. הפעלת הבאזר והלד.

ד. הקלטת וידאו.

ה. רענון ממשק המשתמש.

ו. שליחת מייל למשתמש.

שימוש בתהליכונים מאפשר תגובה מהירה ויעילה של המערכת תוך כדי שמירה על ביצועים סבירים גם כאשר מספר פעולות מתבצעות בו־זמנית.

שימוש במנעולים (Locks) :

כאשר מספר תהליכונים פועלים במקביל, קיימת סכנה שגישה בו־זמנית לאותה פונקציה או משתנה תוביל ל־Race Conditions – מצבים לא צפויים שעלולים לגרום לתקלות.

לשם כך השתמשתי ב־Locks, שמונעים מתהליכונים לגשת לאותו משאב קריטי בו־זמנית. כך מובטח שרק תהליכון אחד בכל רגע נתון יכול להריץ את אותה פעולה, מבלי לפגוע בשאר מרכיבי המערכת.

• שליחת מייל דרך פרוטוקול SMTP :

המערכת כוללת מנגנון לשליחת התראה במייל כאשר מזהה אדם קרוב מדי למצלמה. לצורך כך, נעשה שימוש בפרוטוקול SMTP (simple mail transfer protocol) - שהוא פרוטוקול סטנדרטי לשליחת דוא"ל באינטרנט. בפרויקט הוגדר חשבון דוא"ל שולח (Yahoo) וההתחברות מתבצעת באמצעות אימות TLS מאובטח.

היעד.

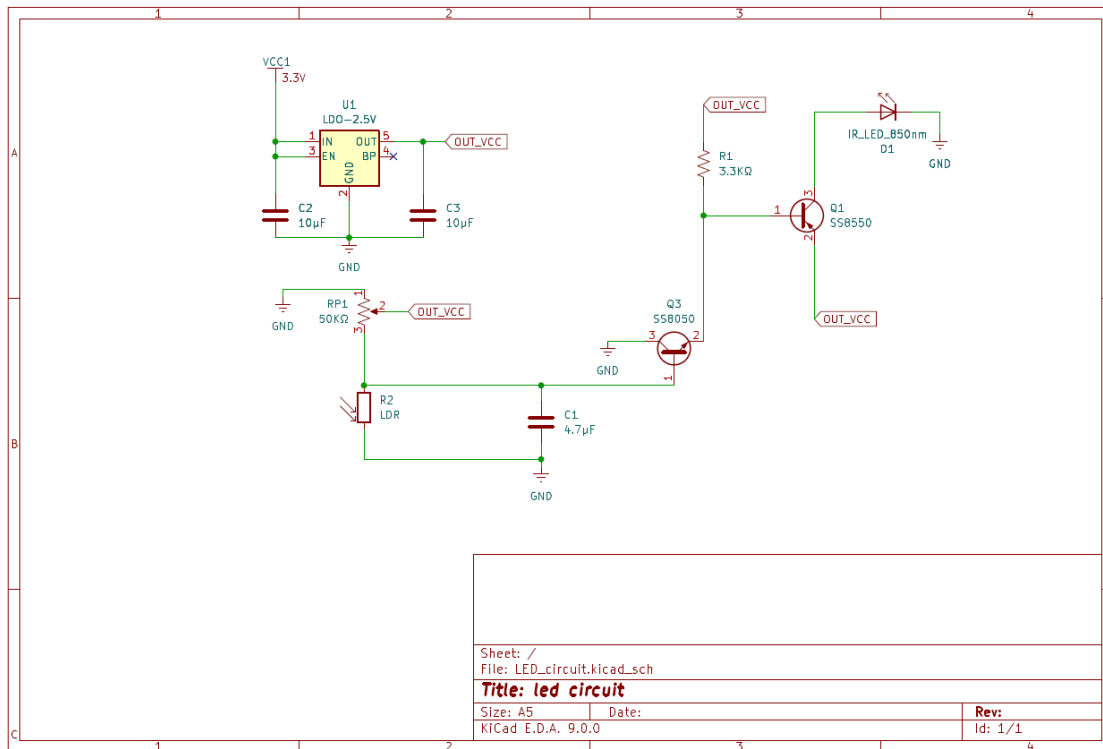


```

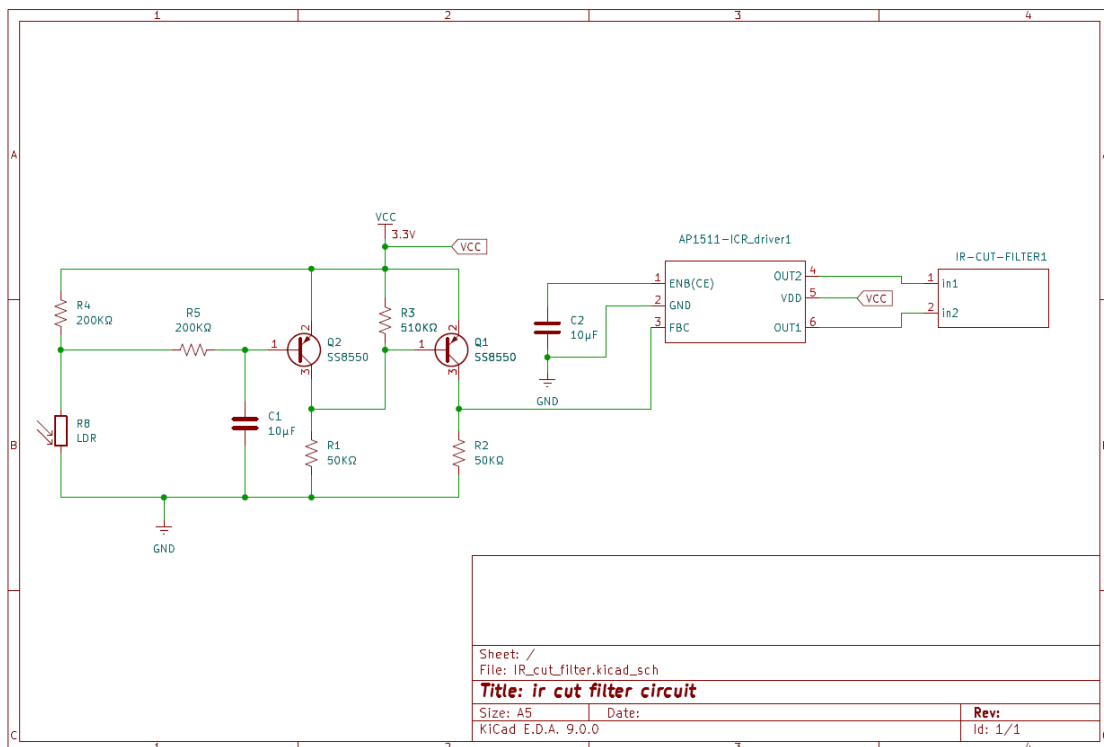
graph TD
    Start([התחלה]) --> Step1[אתחול יציאות וכניסות לבקר]
    Step1 --> Step2[אתחול מודל הזיהוי והמצלמה]
    Step2 --> Step3[אתחול מערכת GUI-ה]
    Step3 --> Decision1{כיבוי המערכת?}
    Decision1 -- כן --> Step4[כבה מערכת]
    Step4 --> End([סוף])
    Decision1 -- לא --> Step5[מדוד מרחק מהאובייקט]
    Step5 --> Decision2{האם נבחרה פעולה מסוימת בממשק GUI?}
    Decision2 -- כן --> Decision3{האם נעשה שינוי (ידי, אוטומטי, מעקב)?}
    Decision3 -- כן --> Step6[שנה למצב הנבחר]
    Step6 --> Decision1
    Decision3 -- לא --> Decision4{שינוי כתובת מייל?}
    Decision4 -- כן --> Step7[בצע שינוי]
    Step7 --> Decision1
    Decision4 -- לא --> Decision5{האם הפעלה?}
    Decision5 -- כן --> Step8[הפעל הקלטה]
    Step8 --> Decision1
    Decision5 -- לא --> Decision6{העברת הזמון למצב השתק?}
    Decision6 -- כן --> Step9[העבר את הזמון למצב השתק]
    Step9 --> Decision1
    Decision6 -- לא --> Decision7{האם יש אדם שקרוב למצלמה?}
    Decision7 -- כן --> Decision8{מצב מעקב פועל?}
    Decision8 -- כן --> Step10[הפעל מעקב]
    Step10 --> Decision1
    Decision8 -- לא --> Decision9{העברת הזמון למצב זמון ולד?}
    Decision9 -- כן --> Step11[הפעל זמון ולד]
    Step11 --> Decision1
    Decision9 -- לא --> Decision10{האם הפעלה?}
    Decision10 -- כן --> Step12[הפעל הקלטה למשך 10 שניות]
    Step12 --> Decision1
    Decision10 -- לא --> Step13[בצע סימון המסגרת באדום והתראה על המסך]
    Step13 --> Decision1
    Decision7 -- לא --> Decision11{מצב מעקב פועל?}
    Decision11 -- כן --> Step14[חזור למצב סריקה הקודם (אוטומטי, ידי)]
    Step14 --> Decision1
    Decision11 -- לא --> Decision1
  
```

איור 11: תרשים זרימה של המערכת

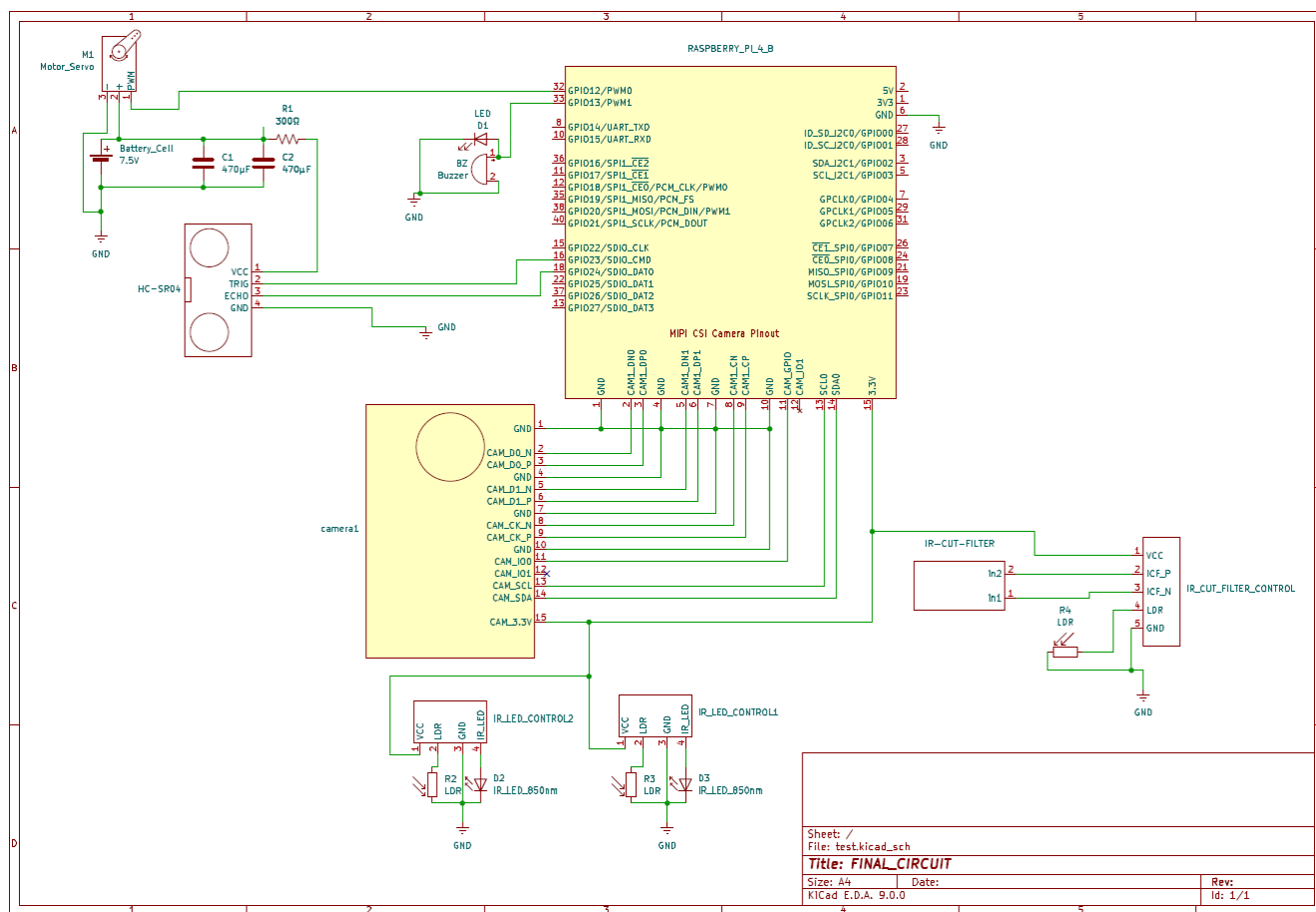
### 3.4 סכמה חשמלית ותרכיבים זרימה של המערכת:



### איור 12 : סכמה של מעגל הפעלה ל-IR LED במערכת הצילום



### איור 13 : סכמה חשמלית של מעגל בקרת IR-CUT Filter



איור 14 : סכמה חשמלית של המערכת

### 3.5 בעיות הנדסיות:

בעיה 1 : במהלך בדיקות המערכת זוהו עיכובים בהצגת הווידאו, בעיקר כאשר מופעלות מספר פונקציות במקביל (כגון זיהוי, שליחה, מעקב).

פתרון : המערכת חולקה לתהליכונים (Threads) כאשר פעולות קריטיות כגון סריקה אוטומטית, שליחת מיילים והקלטה הופעלו כ-Thread נפרד, על מנת לא לחסום את עיבוד התמונה המרכזי. כך נשמרה פעולה רציפה ובזמן אמת.

בעיה 2 : זוהו מקרים של הפרעות כאשר כמה תהליכונים ניסו לגשת לאותה פונקציה/משאב, מה שגרם לשיבושים בביצוע פעולות כמו התראה קולית (זמזום) או שליחת מייל.

פתרון : יושומו מנעולים (Locks) על פונקציות קריטיות כדי להבטיח גישה בלעדית של Thread אחד בכל זמן נתון. כך נמנעה גישה מקבילית לא מבוקרת ונפתרו מצבי Race.

בעיה 3 : הפעלת (stop\_recording) גרמה לא רק לעצירת ההקלטה אלא גם לכיבוי המצלמה, דבר שגרם לשגיאות והפסקת התוכנית.

פתרון : בוצע ניתוח של ספריית Picamera2 והוחלפה הפונקציה (stop\_encoder) בלבד, פעולה שעוצרת את ההקלטה מבלי לכבות את המצלמה

בעיה 4 : הסרוו המניע את המצלמה הציג רעידות וחוסר יציבות בתנועה.

פתרון : הוספת קבלים לייצוב מתח ההזנה של הסרוו, מה שצמצם את הרעידות ומנע הפרעות חשמליות בזמן סיבוב.

בעיה 5 : בעת מעקב אחר אדם, המצלמה הסתובבה מהר מדי או לא הספיקה לעקוב, וכתוצאה מכך האדם יצא מפריים.

פתרון : בוצע חישוב של אחוז סטיית מרכז האדם ממרכז התמונה, ועל פי ערך זה חושב זמן הסיבוב של הסרוו. כך תנועת המצלמה הפכה מדויקת יותר בהתאם למרחק האופקי של האדם מהמרכז.

### 3.6 הוראות הפעלה :

1. חבר את המערכת למקור חשמל.

2. למשתמש יש אפשרות לצפות במתרחש דרך :

- VNC (פרוטוקול גישה מרחוק)
- האתר של Raspberry Pi
- חיבור הבקר למסך דרך כבל HDMI

2. הפעל את התוכנית הראשית על ידי הרצת קובץ ההפעלה (.sh executable).

3. עם תחילת הריצה תיפתח ממשק המשתמש (GUI) הכולל תצוגה בזמן אמת מהמצלמה.

4. בממשק ניתן לבחור בין שלושה מצבי סריקה :

- מצב ידני : מאפשר למשתמש לשלוט על תנועת המצלמה ימינה ושמאלה באמצעות לחצני החצים.
- מצב אוטומטי : המצלמה תנוע ימינה ושמאלה במחזור קבוע ללא התערבות המשתמש.
- מצב מעקב : כאשר אדם מזוהה כקרוב למצלמה, היא תעקוב אחריו אוטומטית. אם אין אדם קרוב, המצלמה תחזור למצב הקודם (ידני או אוטומטי).

5. יוצג על המסך המרחק בין המצלמה לבין האובייקט או האדם שזוהה בעת מעקב.

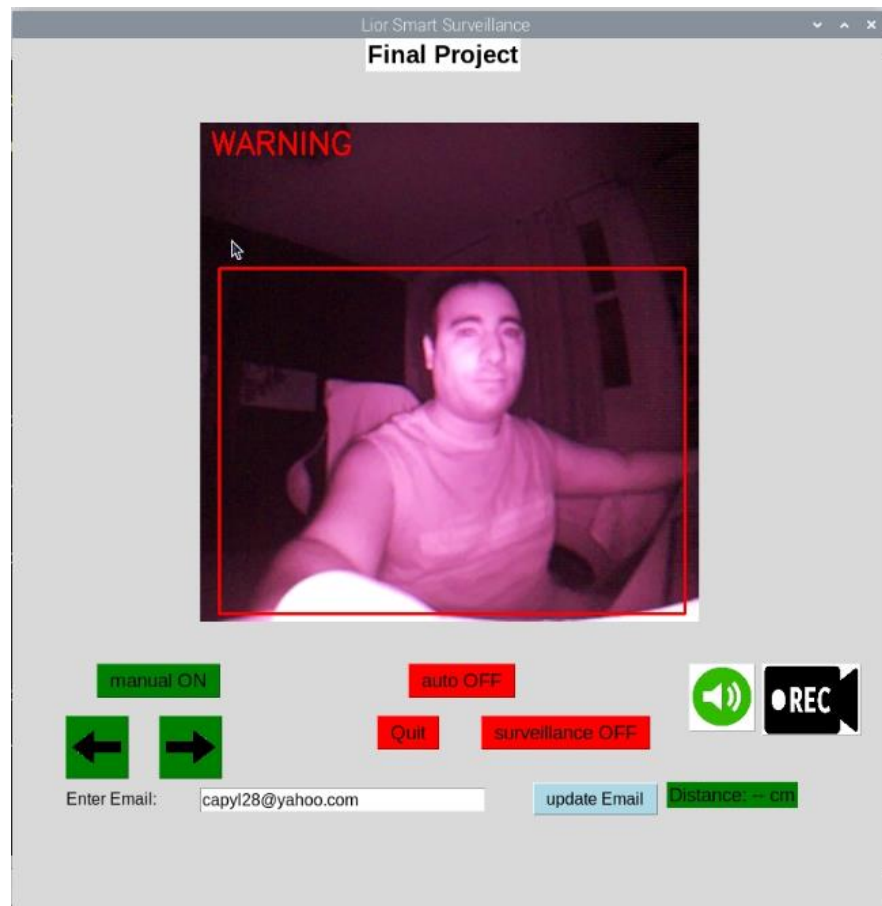
6. ניתן להשתיק את ההתראה הקולית על ידי לחיצה על סמל הרמקול.

7. המשתמש יכול להזין כתובת דוא"ל יעד לקבלת התראות בזמן אמת.

8. ניתן להתחיל או לעצור הקלטת וידאו באמצעות לחיצה על סמל ההקלטה.

9. בסיום השימוש יש ללחוץ על כפתור QUIT, על מנת לוודא סיום בטוח של כל התהליכונים וסגירת המערכת בצורה תקינה.

להלן מסך התצוגה המוצג למשתמש בעת הפעלת המערכת :



איור 15 : מסך ה-GUI של המשתמש

#### 4. סיכום ודיון:

##### 4.1. סיכום

פרויקט זה מתאר מערכת חכמה לזיהוי ומעקב אחר בני אדם בזמן אמת, שפותחה על גבי פלטפורמת Raspberry Pi 4B, ומשלבת טכנולוגיות מתקדמות כגון למידת מכונה, עיבוד תמונה, מנוע סרוו, חיישן מרחק, ממשק משתמש גרפי (GUI) ועוד. מטרת המערכת היא לשדרג את היכולות של מצלמה סטנדרטית, ולאפשר לה לזהות אנשים, לעקוב אחריהם פיזית, לשלוח התראות, ולהקליט את הסביבה – באופן חכם, מהיר ומבוקר.

מטרת הפרויקט הייתה ליצור מערכת אינטראקטיבית ונוחה לתפעול, המשלבת שליטה ידנית ואוטומטית, תוך תגובה למצבים משתנים כמו הופעת אדם קרוב למצלמה. ליישום זה יישומים פוטנציאליים רבים, כגון מערכות אבטחה, ניטור סביבות רגישות ועוד.

בפרויקט נעשה שימוש במודל TFLite מסוג MobileNetV1-SSD, שאפשר זיהוי מדויק של אנשים בפריים. לאחר הזיהוי, המערכת מבצעת חישוב של מרכז המסגרת (Bounding Box) על מנת לעקוב אחר האדם בתנועה, כאשר תנועת המצלמה נשלטת ע"י מנוע סרוו באמצעות חישוב סטייה מהמרכז. בנוסף, המערכת כוללת:

- אפשרות להקלטת וידאו בפורמט MJPEG (לשימוש יעיל במשאבי המעבד),



- שליחת התראות קוליות, חזותיות ודוא"ל דרך פרוטוקול SMTP,
- וממשק גרפי מבוסס Tkinter לשליטה מלאה במערכת.

בפרויקט זה השתמשתי בשפת Python וברכיבים שונים. לצורך בניית המערכת חקרתי והעמקתי בלמידה של ספריות מתקדמות כמו TensorFlow Lite, Picamera2, OpenCV, אשר היוו חלק מרכזי בתכנון ובמימוש הפרויקט.

הפרויקט היווה עבורי אתגר הנדסי מעניין ומורכב, שדרש ממני יכולת למידה עצמית, התמודדות עם איתור ופתרון בעיות מגוונות, ויישום פתרונות יצירתיים תחת מגבלות חומרה ותזמון.

לסיכום, תהליך התכנון והביצוע של הפרויקט תרם לי רבות בהבנת התהליכים הדרושים ביישום ופיתוח אבטיפוס של המוצר. הכלים והידע שרכשתי בפרויקט, ובמהלך הקורסים השונים בתואר, העניקו לי את היכולת להתמודד עם משימות מורכבות אלו והיוו אבן דרך משמעותית עבורי בהפיכתי למהנדס חשמל ואלקטרוניקה.

#### 4.2 עמידה במטלות:

מטלות	ביצוע
כתיבת הצעת פרויקט	בוצע
זיהוי ועיבוד תמונה בזמן אמת	בוצע
יצירת ממשק גרפי (GUI)	בוצע
סריקה ידנית ואוטומטית של האזור	בוצע
התקנת הסרבו פיזית ותכנותו באמצעות בקר Raspberry Pi	בוצע
אינטגרציית המערכת	בוצע
אופטימיזציה לזיהוי בתנאי תאורה נמוכים	בוצע
יכולת התראה של המערכת על קרבה של המטרה למצלמה	בוצע
כתיבת אלגוריתם תקשורת ושליחת נתונים בין הבקר למחשב	בוצע
כתיבת אלגוריתם למעקב + זיהוי מטרות ומרחקם מהמצלמה	בוצע
כתיבת ספר פרויקט	בוצע
הכנת מצגת	בוצע

טבלה 2 עמידה במטלות

#### 4.3 הצעות לשיפור :

- שילוב מאיץ חומרה (כגון Coral USB או דומה) – לשיפור מהירות הרצת המודל ותגובה בזמן אמת, כך תהיה תמיכה במודלים מדויקים ומתקדמים יותר כמו YOLO.
- שדרוג המצלמה ושיפור יכולות ראיית לילה – שימוש במצלמה איכותית יותר לזיהוי מדויק בתנאים משתנים, עם תמיכה ב-IR לטווחים ארוכים ותאורת אינפרא-אדום חזקה יותר.
- הוספת אפליקציה או Web App – לשליטה קלה ונוחה מרחוק על ידי המשתמש.

- [1] Arducam, Raspberry Pi Camera Pinout. [Online]. Available:  
<https://blog.arducam.com/raspberry-pi-camera-pinout/>
- [2] Raspberry Pi Foundation, Raspberry Pi 4 Reduced Schematics. [Online].  
Available: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-reduced-schematics.pdf>
- [3] Raspberry Pi Foundation, Camera Module Documentation. [Online]. Available:  
<https://www.raspberrypi.com/documentation/accessories/camera.html>
- [4] pigpio – Python GPIO Library. [Online]. Available:  
<https://abyz.me.uk/rpi/pigpio/python.html>
- [5] Netron, Neural Network Model Visualizer. [Online]. Available: <https://netron.app/>
- [6] Raspberry Pi Foundation, Picamera2 Manual. [Online]. Available:  
<https://datasheets.raspberrypi.com/camera/picamera2-manual.pdf>
- [7] Raspberry Pi Foundation, Camera Software Documentation. [Online]. Available:  
[https://www.raspberrypi.com/documentation/computers/camera\\_software.html](https://www.raspberrypi.com/documentation/computers/camera_software.html)
- [8] Raspberry Pi Foundation, Raspberry Pi GPIO Documentation. [Online]. Available:  
<https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#gpio>
- [9] E. Juras, "TensorFlow Lite Object Detection Model Performance Comparison," EjTech, Dec. 10, 2022. [Online]. Available: <https://www.ejtech.io/learn/tflite-object-detection-model-comparison>.
- [10] Google AI, TensorFlow Lite Runtime for Microcontrollers. [Online]. Available:  
<https://ai.google.dev/edge/litert/microcontrollers/python>
- [11] TkDocs, Tkinter GUI Tutorial. [Online]. Available:  
<https://tkdocs.com/tutorial/index.html>
- [12] Python Software Foundation, Tkinter Library Documentation. [Online].  
Available: <https://docs.python.org/3/library/tk.html>

[13] Pillow Library, ImageTk Reference. [Online]. Available:

<https://pillow.readthedocs.io/en/stable/reference/ImageTk.html>

[14] OpenCV, "Image Processing (imgproc) Tutorial," OpenCV Documentation, Jan. 9, 2025. [Online]. Available:

[https://docs.opencv.org/4.x/d7/da8/tutorial\\_table\\_of\\_content\\_imgproc.html](https://docs.opencv.org/4.x/d7/da8/tutorial_table_of_content_imgproc.html)

[15] OpenCV Tutorial Series, YouTube, Oct. 7, 2019. [Online]. Available:

[https://www.youtube.com/playlist?list=PLMoSUbG1Q\\_r\\_sc0x7ndCsqdIkL7dwrnNF](https://www.youtube.com/playlist?list=PLMoSUbG1Q_r_sc0x7ndCsqdIkL7dwrnNF)

[16] Raspberry Pi Foundation, Raspberry Pi 4 Model B Specifications. [Online].

Available: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>

[17] Programming Knowledge, "Tkinter Python GUI Tutorial for Beginners,"

YouTube, Mar. 14, 2019. [Online]. Available: [https://www.youtube.com/watch?v=-GhzipvIXIM&list=PLS1QulWo1RIY6fmY\\_iTjEhCMsdtAjgbZM&index=2&ab\\_channel=ProgrammingKnowledge](https://www.youtube.com/watch?v=-GhzipvIXIM&list=PLS1QulWo1RIY6fmY_iTjEhCMsdtAjgbZM&index=2&ab_channel=ProgrammingKnowledge)

## 6.נספחים:

בקובץ zip מצורפים הקבצים הבאים :

א. הצעת הפרויקט והדוח הסופי בקובץ PDF.

ב. תוכנת המערכת (Software Package) - תיקייה המכילה :

- קוד המקור המלא בשפת Python
- משאבים גרפיים המשמשים ל-GUI
- קבצי המודל של למידת המכונה.
- תיקיית RECORD לשמירת קבצי הווידאו מההקלטות
- קובץ הרצה (.sh) - מאפשר הפעלה מלאה של המערכת בלחיצה אחת, בדומה לקובץ EXE במערכת Windows.

ג. שרטוטים חשמליים.

ד. דפי נתונים של הרכיבים העיקריים בפרויקט.

ה. ארכיטקטורת המודל בקובץ PDF.

ו. סרטון הדגמה של המערכת בפעולה.