

**מערכת לזיהוי מסיכה ומדידת טמפרטורה**

פרויקט גמר

למילוי חלקי של הדרישות לקבל תואר הנדסאי אלקטרוניקה בהתמחות מערכות אלקטרוניות

מאת:

דביר פלח, לביא גריב

מספר תעודת זהות:

211790928, 323021386

בהנחיית:

אהרון ניסנבאום

**הצהרת הלומד**

שם הסטודנט: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ מספר ת"ז: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

שם הסטודנט: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ מספר ת"ז: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

אני הח"מ, מצהיר בזאת כי פרויקט/עבודת הגמר וספר הפרויקט המצ"ב נעשו על ידי בלבד.

הפרויקט מסכם ידע, מיומנות והרגלים שלמדתי במסגרת לימודי ההתמחות במגמה ובאופן עצמאי.

הפרויקט וספר תיעוד הפרויקט נעשו על בסיס ההנחיות שקיבלתי מהמנחה שלי .

מקורות המידע בהם השתמשתי לביצוע פרויקט מצוינים ברשימת המקורות שבסוף הספר.

אני מודע לאחריות שהנני מקבל על עצמי על ידי חתימתי על הצהרה זו שכל הנכתב בה אמת.

חתימת הסטודנט: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

חתימת הסטודנט: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**אישור מנחה הפרויקט/עבודת הגמר**

הריני מאשר שהפרויקט בוצע בהנחייתי, קראתי את ספר הפרויקט ומצאתי כי הוא ראוי להגשה.

שם המנחה:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה:\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**אישור רכז המגמה**

הריני מאשר שדרישות הפרויקט ורמתו מתאימים לדרישות והנחיות משרד החינוך המפורסמים בחוזר המפמ"ר ובאתר המגמה.

שם רכז המגמה: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה \_\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_

# תודה

אחרי חודשים של עבודה אינטנסיבית ושיתוף פעולה אנחנו רוצים להודות לצוות המרצים ממכללת אורט רחובות ובפרט לאייל קודלר – ראש מגמת אלקטרוניקה, לחברים ולמשפחות שלנו על התמיכה, העידוד והאמונה ביכולותינו.

תודה מקרב לב למנחה שלנו, אהרון ניסנבאום. תודה על הרחבת אופקי חשיבה, ועל הליווי הצמוד לאורך כל הדרך – עזרתך לא תסולא בפז.

תודה לדוקטור יורי קלנטרוב, על ההכוונה והעצות המועילות במהלך ביצוע הפרויקט.

תודה לאורן יגב אשר עזר לנו לגבש את הרעיון ולמקד אותנו.

לא היינו מגיעים עד הלום בלעדיכם.

# תוכן עניינים

[תודה 3](#_Toc71802607)

[תוכן עניינים 4](#_Toc71802608)

[רשימת נספחים 6](#_Toc71802609)

[רשימת טבלאות 7](#_Toc71802610)

[תקציר 8](#_Toc71802611)

[תיאור הבעיה או הצורך 8](#_Toc71802612)

[תרשים מלבנים 9](#_Toc71802613)

[פרוטוקולים בהם השתמשנו 10](#_Toc71802614)

[VNC (Virtual Network Computing) 10](#_Toc71802615)

[SSH – Secure Shell 11](#_Toc71802616)

[פרוטוקול I2C 12](#_Toc71802617)

[פרוטוקול SPI 13](#_Toc71802618)

[PWM – Pulse Width Modulation 14](#_Toc71802619)

[רשימת הרכיבים 15](#_Toc71802620)

[Raspberry Pi 4B 15](#_Toc71802621)

[חיישן טמפרטורה – Melexis MLX90640 17](#_Toc71802622)

[חיישן מרחק – HC-SR04 18](#_Toc71802623)

[מצלמה - Logitech C922 Pro 19](#_Toc71802624)

[מנוע סרוו – SG90 20](#_Toc71802625)

[מסך – MHS3528 21](#_Toc71802626)

[רמקול 23](#_Toc71802627)

[שרטוט חשמלי 24](#_Toc71802628)

[Raspberry Pi 4B 25](#_Toc71802629)

[Melexis MLX90640 26](#_Toc71802630)

[Logitech C922 Pro 26](#_Toc71802631)

[HC-SR04 27](#_Toc71802632)

[רמקול 27](#_Toc71802633)

[MHS3528 27](#_Toc71802634)

[SG90 – דגם הדלת 28](#_Toc71802635)

[SG90 28](#_Toc71802636)

[מימוש תוכנה 29](#_Toc71802637)

[train\_mask\_image\_classification\_model.ipynb 29](#_Toc71802638)

[MLX90640.py 32](#_Toc71802639)

[MaskNet.py 33](#_Toc71802640)

[Object\_detection\_webcam\_tflite.py 39](#_Toc71802641)

[door-model.py 40](#_Toc71802642)

[תרשים זרימת התוכנה 41](#_Toc71802643)

[דלת אוטומטית 42](#_Toc71802644)

[תיעוד 43](#_Toc71802645)

[רפלקציה 49](#_Toc71802646)

[רשימה ביבליוגרפית 50](#_Toc71802647)

## רשימת נספחים

[נספח 1: תרשים מלבנים 9](https://d.docs.live.net/1d9cd4b73ee9e262/מסמכים/ספר%20פרויקט%20דביר%20פלח%20ולביא%20גריב.docx#_Toc71802648)

[נספח 2: חיבור תקשורת I2C 12](#_Toc71802649)

[נספח 3: חיבור תקשורת SPI 13](#_Toc71802650)

[נספח 4: Duty Cycle 14](#_Toc71802651)

[נספח 5: פלט וקלט חיישן מרחק HC-SR04 18](#_Toc71802652)

[נספח 6: תהליך לכידת תמונה במצלמה דיגיטלית 19](#_Toc71802653)

[נספח 7: חיבורי המסך מדגם MHS3528 22](#_Toc71802654)

[נספח 8: חיבור 3.5 מ"מ מסוג TRS 23](#_Toc71802655)

[נספח 9: שרטוט חשמלי סופי 24](https://d.docs.live.net/1d9cd4b73ee9e262/מסמכים/ספר%20פרויקט%20דביר%20פלח%20ולביא%20גריב.docx#_Toc71802656)

[נספח 10: התראה על חום גוף גבוה מהמותר 35](#_Toc71802657)

[נספח 11: התראה על אישור כניסה למתחם 35](#_Toc71802658)

[נספח 12: תרשים זרימה תוכנית ראשית 41](https://d.docs.live.net/1d9cd4b73ee9e262/מסמכים/ספר%20פרויקט%20דביר%20פלח%20ולביא%20גריב.docx#_Toc71802659)

[נספח 13: תרשים זרימה דלת אוטומטית 42](#_Toc71802660)

[נספח 14: תהליך תכנות רגיל 45](#_Toc71802661)

[נספח 15: תהליך תכנות בעזרת למידת מכונה 45](#_Toc71802662)

[נספח 16: קטע קוד לבדיקת מהירות המודל 47](https://d.docs.live.net/1d9cd4b73ee9e262/מסמכים/ספר%20פרויקט%20דביר%20פלח%20ולביא%20גריב.docx#_Toc71802663)

[נספח 17: קוד לבדיקת מנוע סרוו 48](#_Toc71802664)

## רשימת טבלאות

[טבלה 1: מפרט Raspberry Pi 4B 15](#_Toc71802665)

[טבלה 2: מפרט חיבורי מסך מדגם MHS3528 22](#_Toc71802666)

[טבלה 3: חיבורי Raspberry Pi 4B במעגל 26](#_Toc71802667)

[טבלה 4: חיבורי חיישן טמפרטורה Melexis MLX90640 26](#_Toc71802668)

[טבלה 5: חיבורי מצלמה Logitech C922 Pro 26](#_Toc71802669)

[טבלה 6: חיבורי חיישן מרחק HC-SR04 27](#_Toc71802670)

[טבלה 7: חיבורי רמקול 27](#_Toc71802671)

[טבלה 8: חיבורי מסך NHS3528 28](#_Toc71802672)

[טבלה 9: חיבורי מנוע סרוו SG90 של דגם הדלת 28](#_Toc71802673)

[טבלה 10: חיבורי מנוע סרוו SG90 28](#_Toc71802674)

# תקציר

החל מחודש נובמבר 2019 מהווה מגיפת COVID-19 נושא חברתי ובריאותי מרכזי במדינת ישראל ובעולם כולו. מאז פרצה מגפת הקורונה אובחנו 839,022 מקרים של הנגיף בישראל; יותר מ-6,378 אנשים נפטרו בעקבות המחלה וסיבוכיה. הנגיף מועבר מאדם אחד לאחר על ידי רסס טיפתי הנוצר כאשר אנשים נושמים, מדברים, משתעלים או מתעטשים – הרסס נישא באוויר ולכן אדם אחד עלול להדביק את כל מי שנוכח באותו החדר. על מנת למנוע את התפשטות הנגיף, משרד הבריאות גיבש הנחיות לפעילות במרחב הציבורי. ההנחיות כללו עטיית מסכת פה ואף, ריחוק חברתי ובידוד במקרה של הופעת סימפטומים המזוהים עם נגיף.

## תיאור הבעיה או הצורך

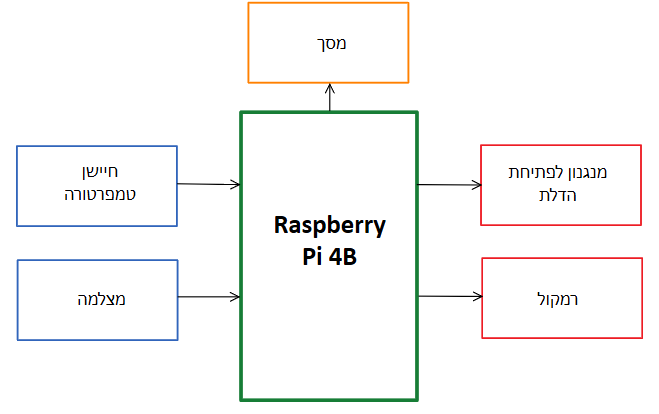
על פי הנחיות משרד הבריאות, יש לעטות מסיכה על הפה והאף, ולבצע בדיקת חום על מנת לוודא שחום הגוף לא עולה על 38 מעלות.

כיום, לא קיימת מערכת אוטומטית למדידת חום וזיהוי מסיכה על פני הלקוח בכניסה למבנה בשוק העסקי. קיימים מדי טמפרטורה אוטומטים בכניסה לחנויות מסוימות אבל הם אינם חוסמים את הגישה למתחם במידה וחום הגוף גבוה מידי או לא נמדדת. האפשרות היחידה לוודא שהלקוח שומר על ההנחיות בעת כניסתו למבנה היא העסקת שומר, שימדוד טמפרטורה בכניסה למתחם סגור ויוודא שהוא עוטה מסיכה על פניו – אפשרות זו אומנם יעילה – אך גם יקרה, ובמידה ואותו שומר נושא את הנגיף עלולה להתבצע הדבקה מהירה יותר (מכיוון שהוא בא במגע עם רוב העובדים והלקוחות).

לאחר שחקרנו את הנושא, החלטנו לפתח מערכת אוטונומית שמבצעת צילום של פני הלקוח ובודקת האם הוא עוטה מסיכה והאם חום גופו תקין. תכנון המערכת מאפשר להתקין אותה על גבי דלתות אוטומטיות סטנדרטיות. המערכת קובעת אם הלקוח מורשה להיכנס למתחם בהתאם להנחיות משרד הבריאות. במידה והלקוח אינו עוטה מסיכה המערכת מציגה על המסך סרטון קצר של הנחיות עטיית המסיכה בצורה תקינה, ומציגה התראה אם חום הגוף של הלקוח גבוה מעל המותר. במידה והלקוח עומד בקריטריונים הנדרשים, המערכת תאפשר לדלת האוטומטית להיפתח.

מטרת המערכת היא להאט את התפשטות הנגיף בכך שלא יתאפשר לאנשים להיכנס למתחמים סגורים אם הם אינם עומדים בהנחיות משרד הבריאות.

## תרשים מלבנים



נספח 1: תרשים מלבנים

# פרוטוקולים בהם השתמשנו

## VNC (Virtual Network Computing)

התוכנה בה השתמשנו בכדי לאפשר שליטה וביצוע פעולות בבקר  
ה-Raspberry Pi דרך המחשב הנגיש, נקראת VNC Viewer מבית RealVNC.

תוכנה זו משתמשת בפרוטוקול VNC – פרוטוקול תקשורת המאפשר שליטה מרחוק על מערכות מחשבים. הוא משדר את הלחיצות על המקלדת והעכבר ממחשב אחד לאחר, ומציג גם כן את המסך של המחשב המרוחק על גבי רשת האינטרנט. בפרויקט שלנו היה צורך בשימוש בתוכנה זו בשלב הפיתוח, משום  
שה-Raspberry Pi אינו נגיש בכל זמן כעמדת עבודה שלמה עם התקני קלט ופלט הכרחיים כגון עכבר, מקלדת ומסך, ולכן העדפנו לחבר את ה-Raspberry Pi לרשת אינטרנט ביתית ולהשתמש בו דרך המחשבים האישיים שלנו.

## SSH – Secure Shell

Secure Shell) SSH) הוא פרוטוקול לתקשורת נתונים מוצפנת בין שני מחשבים המתחברים ברשת משותפת – על גבי רשת מקומית (למשל, ראוטר ביתי) או רשת רחבה (האינטרנט).

השתמשנו בפרוטוקול זה, בנוסף לפרוטוקול VNC, על מנת להפעיל פקודות על  
ה-Raspberry Pi שלא מצריכות ממשק מלא כמו שפרוטוקול VNC מספק, אלא רק את שורת הפקודה ("Terminal").

## פרוטוקול I2C

תקשורת I2C היא תקשורת טורית בין מעבד Master ורכיב Slave.

על פס התקשורת I2C יכולים להתחבר מספר רכיבים שונים. הרכיב המנהל את תהליך התקשורת(המעבד) נקרא Master והרכיבים המתחברים אליו נקראים Slaves.

בתקשורת זו ישנם שני קווים:

קו הנתונים הטורי – SDA (Serial Data) – שהוא דו כיווני ואחראי להעברת הנתונים.

וקו השעון הטורי – SLK (Serial Clock) – שהוא חד כיווני ומופעל על ידי ה-Master, קו שעון שאחראי על תזמון העברת הנתונים.

מהירות מקסימלית:

Standard mode - 100 kbps

Fast mode - 400 kbps

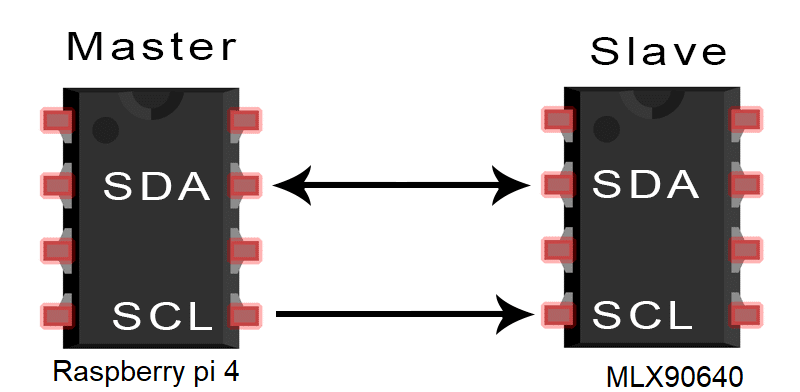
High Speed mode - 3.4 Mbps

Ultra-Fast mode - 5 Mbps

התקשורת הינה תקשורת סינכרונית מפני שיש סנכרון אותות שעון בין הרכיבים.

יש לתקשורת מספר בלתי מוגבל של Masters, ואפשרות לעד 1008 רכיבי Slave.

פרוטוקול זה משמש אצלנו לחיבור חיישן הטמפרטורה MLX90640.



נספח 2: חיבור תקשורת I2C

## פרוטוקול SPI

תקן SPI נועד לאפשר תקשורת טורית בין מכשיר Master אחד למספר מכשירי Slave. התקן הזה לא מגדיר את המידע אלא את האופן והתזמון השידור והקליטה שלו.

קצב השידור בפרוטוקול SPI הוא עד 1 Mbps והמרחק המומלץ בין מכשירים הוא 3 מטר.

בתקשורת SPI ישנם 4 חיבורים של העברת מידע:

SCLK - קו שעון תזמון משותף לכל הרכיבים.

MOSI - מידע שיוצא מה-Master ומגיע ל-Slave.

MISO - מידע שיוצא מה-Slave ומגיע ל-Master.

SS - בחירת רכיב, כשישנם כמה רכיבים מחוברים במקביל ישנו חיבור ייחודי לכל אחד שנקרא Slave select.

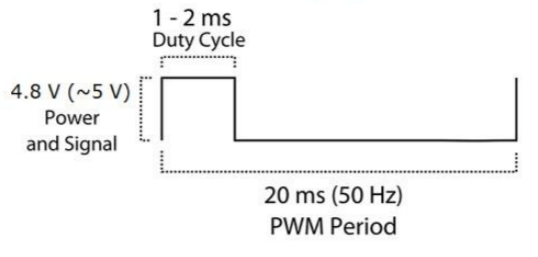


נספח 3: חיבור תקשורת SPI

## PWM – Pulse Width Modulation

בRaspberry Pi 4B קיימים 40 פינים של GPIO, כאשר חלקם תומכים ב-PWM – אפנון רוחב פולס. אפנון רוחב פולס הוא גל ריבועי בעל Duty Cycle ותדר משתנים, שאותם ניתן להגדיר באמצעות התוכנה.

Duty Cycle הוא פרק הזמן היחסי למחזור האות שבו המתח במצב 'גבוה'. למשל באות בו מחזור שלם מתבצע במשך 20ms ופרק הזמן שבו האות בעל מתח 'גבוה' הוא 2ms, ה-Duty Cycle הוא 10%.



נספח 4: Duty Cycle

השתמשו ב-PWM על מנת להפעיל את מנוע הסרוו בעזרת ה-Raspberry Pi.

# רשימת הרכיבים

## Raspberry Pi 4B

ה-Raspberry Pi הוא מחשב לוח-יחיד המכיל את כל הפעולות הנחוצות ממחשב רגיל. הגרסה הראשונה של ה-Raspberry Pi פותחה בבריטניה על ידי קרן  
ה-Raspberry Pi ויצאה בשנת 2012 במטרה לקדם את לימודי ענף מדעי המחשב, במיוחד במדינות פחות מפותחות. המחשב צבר פופולריות רבה ונמכר במקומות רבים בעולם.

מחשב קטן זה מגיע ללא מערכת הפעלה אשר אותה ניתן להתקין עליו וניתן להתקין בכמה אפשרויות. שפת התכנות הראשית בעזרתה ניתן לעבוד עם  
ה-Raspberry Pi היא Python, אך ישנה אפשרות לתמיכה בשפות תכנות נוספות.

בפרויקט זה נעשה שימוש ב-Raspberry Pi מדגם 4B.

|  |  |
| --- | --- |
| Raspberry Pi 4B |  |
| CPU | Quad core 64-bit ARM-Cortex A72 running at 1.5GHz |
| GPU | Broadcom VideoCore VI |
| Memory (SDRAM) | 4GB LPDDR4 |
| USB Ports | 2 × USB 2.0, 2 × USB 3.0 |
| Video Outputs | 2 × micro-HDMI 2.0  1 x 3.5 mm analogue audio-video jack. |
| Ethernet | Gigabit Ethernet |
| On-board storage | microSD slot |
| Wi-Fi | 2.4 GHz and 5 GHz 802.11b/g/n/ac |
| Bluetooth | Bluetooth 5.0, Bluetooth Low Energy |
| Low-level peripherals | 40-pin GPIO header |

טבלה 1: מפרט Raspberry Pi 4B

**CPU –** זהורכיב חומרה בסיסי המצוי בכל מחשב, ותפקידו ביצוע פקודות הנדרשות ממנו ע"פ הפקודות המאוחסנות בזיכרון המחשב.

**GPU –** מעבד אשר נמצא בתוך הכרטיס הגרפי, ומאפשר לחלק את העומסים בעת חישובים גרפיים, נחוץ ביותר בעת טעינת גרפיקות תלת-ממדיות. רכיב זה מאפשר עבודה על מידע גדול של תמונות, ומסוגל להמיר מידע זה להצגה על המסך כמטריצה של פיקסלים. בנוסף מסוגל לבצע חישובים מסובכים וניתן לתכנות בדומה למעבדים אחרים.

**Memory** – זהו שם כללי לזיכרון המחשב המתאפיין ביכולת המעבד לגשת ישירות לכל תא בזיכרון לפי כתובתו, לכתוב בו ולקרוא ממנו.

**USB (2.0/3.0) –** זהו תקן לחיבור בין מחשבים להתקני ציוד היקפי. תקן הUSB הינו חיבור סטנדרטי אחד במחשבים אישיים עבור התקנים שונים, במקום ריבוי סוגי חיבורי המחשב שהיה קיים לפניו.

**Video outputs** – זהו תקן להעברת אותות וידאו ואודיו אל הציוד שהעברת המידע אליו מתבקשת עם חיבורו להתקן

**Etherne**t – היא טכנולוגיה לתקשורת נתונים ברשתות מחשבים מקומיות, פרוטוקול תקשורת אשר אחראי על השכבה הפיזית.

**On Board Storage –** אפשרותהוספת זיכרון אחסון על ידי כרטיס זיכרון, משמש לאחסון מערכת ההפעלה ומידע נוסף אשר נשמר עם הפסקת הספקת האנרגיה אל הבקר.

**Wi-Fi –** זו היא טכנולוגיה המאפשרת למכשירים אלקטרונים העברת נתונים באופן אלחוטי באמצעות גלי מיקרו.

**Bluetooth –** הוא תקן פתוח לתקשורת רדיו שהיא שידור וקליטה של גלי מיקרו באורכי גל קצרים יחסית בקצב נמוך ובהספק נמוך יחסית. התקן זה משמש בעיקר לתקשורת אלחוטית במרחב האישי (טווחים קצרים מאוד).

**GPIO** – צורת תקשורת לחיבור רכיבים אלקטרונים ועבודה עם הבקר, ע"י פינים להעברת תקשורת בינארית דו כיוונית ע"י תכנות עצמאי של המשתמש.  
ה-Raspberry Pi כולל 40 פינים של GPIO.

## חיישן טמפרטורה – Melexis MLX90640

גוף האדם, כמו גם עצמים שונים, פולטים אנרגיה אינפרה-אדומה. קרינה זו נקלטת במצלמת הגלאי ומתורגמת לטמפרטורה באמצעות חישוב מרחק המדידה מהעצם.

חיישן MLX90640 כולל 768 חיישני אינפרה-אדום זעירים במערך בגודל 24 על 32, שיוצרים תמונה תרמית במרחב. החיישן בעל שדה ראייה (FOV) של 55 מעלות על 35 מעלות.

בפרויקט זה חיישן הטמפרטורה משמש על מנת למדוד את חום הגוף של האדם המבקש להיכנס למתחם כדי לוודא שהוא לא עולה על 38 מעלות צלזיוס.

בחרנו בדגם MLX90640 מבית Melexis – מערך חיישני אור אינפרה אדום המשמשים כמצלמה טרמית, ברזולוציה של 32x24.

החיישן מיועד לפעול בטמפרטורה של -40 עד 300 מעלות בטווח דיוק של מעלה אחת.

החיישן צורך מתח הפעלה של 3.3V וזרם הפעלה של mA2.

משתמש בפרוטוקול I2C כדי לתקשר עם הדקי ה-GPIO של ה-Raspberry Pi 4B.

## חיישן מרחק – HC-SR04

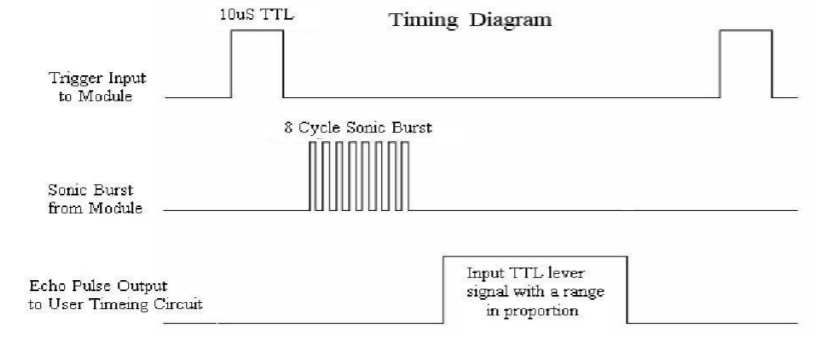
החיישן מודד מרחק על ידי שידור גלי קול לעבר אובייקט מרוחק, המתנה להד המוחזר מהאובייקט, מדידת הזמן שעבר מרגע השידור ועד קליטת ההד וחישוב המרחק לפי משוואת התנועה במהירות קבועה.

החיישן משדר גלי קול בתדר אולטראסוני – תדר שהוא מעבר ליכולת השמיעה של אוזן אנושי. גלי הקול מתקדמים באוויר, פוגעים בעצם שנמצא מול החיישן, ומוחזרים אליו.

לפיכך, חישוב המרחק מתבצע באופן הבא:

כאשר Duration שווה לפרק הזמן הנמדד בין שליחת האות מהדק trigger ועד לקבלתו בהדק echo.

בדף המידע מצוין שיש לספק פולס (עליית שעון בגל ריבועי) לפרק זמן של 10µS להדק trigger, שבעקבותיו החיישן ישלח 8 מחזורים של אות בתדר 40kHz.



נספח 5: פלט וקלט חיישן מרחק HC-SR04

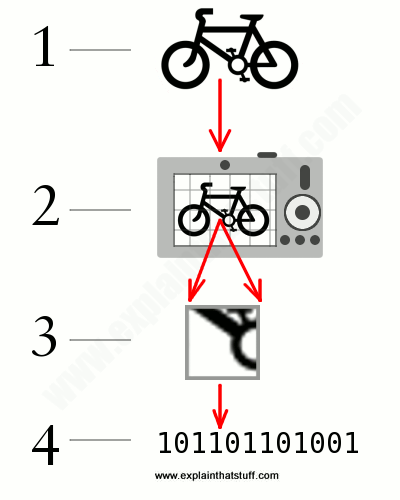
החיישן צורך מתח עבודה של 5V ברגל Vcc, ומשתמש בזרם עבודה של 15mA.  
החיישן בעל טווח מדידה של -40°C עד 300°C ס"מ, עם רמת דיוק של 0.3 ס"מ.  
אנו משתמשים בחיישן המרחק למנגנון לפתיחת וסגירת הדלת.

בחרנו להשתמש בדגם HC-SR04 מפני שהוא זול, יעיל לזיהוי בשינויי מרחק, ויש עליו המון מידע ותוכניות לדוגמא באינטרנט שמאפשרות להבין את אופן פעולתו.

## מצלמה - Logitech C922 Pro

מצלמת רשת היא מצלמה דיגיטלית שמתחברת אל המחשב על גבי פרוטוקול USB 2.0 בכדי לשדר בזמן אמת תמונות וסרטונים. כמו כל מצלמה דיגיטלית היא לוכדת אור דרך עדשה קטנה בחזית באמצעות רשת זעירה של פיקסלים רגישים לאור המובנית בתוך חיישן מצלמה מסוג CMOS. מצלמה זאת פועלת ברזולוציה של 1920x1080.

קרני האור נכנסות לעדשת המצלמה. לאחר מכן, חיישן התמונה מחלק את האור הנקלט למיליוני פיקסלים (ריבועים). החיישן מודד את רמת הבהירות והצבע. לבסוף, רמת הבהירות והצבע של כל פיקסל מועברים לפורמט דיגיטלי של אפסים ואחדות שהמחשב יכול להבין, והמידע מועבר באופן ישיר למחשב ע"י פרוטוקול USB, ובכך מתקבלת תמונה דיגיטלית שניתן להציג על המסך.



נספח 6: תהליך לכידת תמונה במצלמה דיגיטלית

## מנוע סרוו – SG90

מנוע סרוו הוא מנוע שמאפשר שליטה מדויקת על זווית ומהירות הסיבוב.

למנוע סרוו יש 3 הדקים – Vcc ו-GND, המשמשים כמתח מבוא ואדמה בהתאמה, וכניסה נוספת של PWM (Pulse Width Modulation) שבעזרתה שולטים המפתח על פעולת המנוע.

המנוע מדגם SG90 צורך מתח כניסה של 4.8V עד 6.0V, וזרם מקסימלי מדוד של עד 360mA. דגם זה יכול להסתובב בזווית של 0°-180°.

לפי דף המידע של הרכיב, המנוע פועל בתדר של 50Hz, ומזהה דפקי שעון באורך של 0.4ms-2.4ms כאשר פולס באורך של 0.4ms מסובב את המנוע לזווית 0°, ופולס באורך של 2.4ms מסובב את המנוע לזווית של 180°.

בתדר של 50Hz, על מנת לייצר פולס באורך 0.4ms יש להשתמש בDuty Cycle של 2%. על מנת לייצר פולס באורך 2.4ms יש להשתמש בDuty Cycle של 12%.

## מסך – MHS3528

LCD היא טכנולוגית צגים דקים העשויים ממספר מקטעים. בדרך כלל, כל מקטע מהווה מרכיב צבע אחד של פיקסל. יתרונותיו העיקריים, העובי הדק וצריכת האנרגיה הנמוכה, הפכו את הטכנולוגיה למתאימה לשימוש במכשירים אלקטרונים ניידים המופעלים באמצעות סוללה. יתרונות נוספים של עובי קטן, משקל נמוך, רוחב וגודל תצוגה וכן קלות התתקנה, נותנים לטכנולוגיה עדיפות גם לייצור צגי מחשב וטלוויזיות.

בחרנו להשתמש במסך LCD מדגם זה מכיוון שהוא תואם ל-Raspberry Pi, ועלותו הזולה.

אנו משתמשים במסך לצורך הצגת מידע רלוונטי ללקוח המבקש להיכנס למתחם.

למסך יש רזולוציה של 320 על 480, והוא בגודל 3.5 אינץ', עם תאורה אחורית מסוג LED.

המסך משתמש בפרוטוקול SPI, ומתחי ההפעלה שלו הם V3.3 או V5.

זרם ההפעלה שלו הינו 160mA.

תמונה שמכילה טקסט, אלקטרוניקה

התיאור נוצר באופן אוטומטי

נספח 7: חיבורי המסך מדגם MHS3528

מפרט חיבור

|  |  |  |
| --- | --- | --- |
| DESCRIPTION | SYMBOL | NUMBER |
| Power positive (3.3V power input) | V3.3 | 1, 17 |
| Power positive (5V power input) | V5 | 2,4 |
| Power ground | GND | 6 |
| Instruction/data register selection ,low ,level is instruction, high level is data | LCD\_RS | 18 |
| LCD display SPI data input | LCD\_SI | 19 |
| Reset signal, low reset | RST | 22 |
| LCD display SPI clock signal | LCD\_SCK | 23 |
| LCD chip select signal, low level enable | LCD\_CS | 24 |

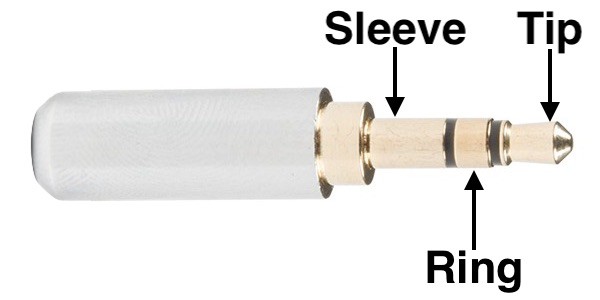
טבלה 2: מפרט חיבורי מסך מדגם MHS3528

## רמקול

רמקול בסיסי מורכב ממגנט קבוע ומסליל של מוליך חשמלי החופשי לנוע בתוך השדה המגנטי הקבוע של המגנט. אל הסליל מחוברת דיאפרגמה – (ממברנה דקה, קשיחה למחצה) בצורה המזכירה חרוט רחב בסיס. שולי הממברנה מחוברים אל מסגרת מתכתית בחיבור גמיש שאיננו מונע את תנועתה.

הצליל מופק כאשר זרם חשמלי עובר בסליל וגורם לשדה אלקטרומגנטי, שעוצמתו וכיוונו יחסיים לעוצמה והכיוון של הזרם. בעקבות זאת, בין השדה האלקטרומגנטי לשדה המגנטי הקבוע נוצרת התנגדות פיזית שגורמת לתנועה בין הסליל למגנט הקבוע. תנועה זאת גורמת לתנועה של הממברנה המחוברת לסליל, שדוחפת אוויר בכיוון ועוצמה יחסיים לכיוון ועוצמת הזרם העובר בסליל. תנודות האוויר המופקות באופן הזה הן בעצם גלי קול.

הרמקול שלנו משתמש בחיבור TRS שבמקור אמור להיות חיבור לסטריאו, לנו יש רק רמקול אחד אז חיברנו אותו רק לערוץ השמאלי בחיבור.



נספח 8: חיבור 3.5 מ"מ מסוג TRS

Tip: חיבור פלוס ערוץ שמאלי.

Ring: חיבור פלוס ערוץ ימני.

Sleeve: אדמה משותפת.

# 

# שרטוט חשמלי

נספח 9: שרטוט חשמלי סופי

## Raspberry Pi 4B

|  |  |  |
| --- | --- | --- |
| **מספר הרגל** | **שם הרגל** | **לאן הרגל מחוברת** |
| 1 | 3.3V | הרגל מספקת מתח ישר של 3.3V למצלמה תרמית MLX90640. |
| 2 | 5V | מספקת מתח ישר של 5V לחיישן מרחק HC-SR04, ולשני מנועי הסרוו SG90. |
| 3 | SDA1/GPIO2 | מחוברת לרגל SDA של מצלמה תרמית MLX90640. |
| 4 | 5V | מספקת מתח ישר של 5V לרגל מס' 2 של מסך מדגם MHS3528. |
| 5 | SCL1/GPIO3 | מחוברת לרגל SCL של מצלמה תרמית MLX90640. |
| 6 | GND/אדמה | מחוברת לרגל הארקה (מס' 6) של מסך מדגם MHS3528. |
| 14 | GND/אדמה | מחוברת לרגל הארקה של מצלמה תרמית MLX90640, חיישן מרחק HC-SR04, ושל שני מנועי הסרוו SG90. |
| 17 | 3.3V | מספקת מתח ישר של 3.3V לרגל מס' 17 של מסך מדגם MHS3528. |
| 18 | GPIO24 | מחוברת לרגל LCD\_RS (מס' 18) של מסך מדגם MHS3528. |
| 19 | SPI\_MOSI/GPIO10 | מחוברת לרגל LCD\_SI (מס' 19) של מסך מדגם MHS3528. |
| 22 | GPIO25 | מחוברת לרגל RST (מס' 22) של מסך מדגם MHS3528. |
| 23 | SPI\_CLK/GPIO11 | מחוברת לרגל LCD\_SCK (מס' 23) של מסך מדגם MHS3528. |
| 24 | SPI\_CE0\_N/GPIO8 | מחוברת לרגל LCD\_CS (מס' 24) של מסך מדגם MHS3528. |
| 32 | PWM0/GPIO12 | מחוברת לרגל PWM של מנוע סרוו SG90 של דגם הדלת. |
| 33 | PWM1/GPIO13 | מחוברת לרגל PWM של מנוע סרוו SG90. |
| 35 | GPIO19 | מחוברת לרגל echo של חיישן מרחק HC-SR04. |
| 36 | GPIO16 | מחוברת לרגל trigger של חיישן מרחק HC-SR04. |
|  | USB Type-C | חיבור USB מסוג C שמשמש לאספקת חשמל ל-Raspberry Pi; צורך כ-3A, 5V. |
|  | 3.5mm TRRS Audio Jack | שקע אוזניות; מחובר לרמקול. |
|  | USB 2.0 | שקע USB מסוג A; מחובר למצלמת רשת מדגם Logitech C922 Pro. |

טבלה 3: חיבורי Raspberry Pi 4B במעגל

## Melexis MLX90640

|  |  |  |
| --- | --- | --- |
| **מספר הרגל** | **שם הרגל** | **לאן הרגל מחוברת** |
|  | VDD | מחוברת לרגל 3.3V (מס' 1) של Raspberry Pi 4B. |
|  | SDA | מחוברת לרגל SDA1 (מס' 3) של Raspberry Pi 4B. |
|  | SCL | מחוברת לרגל SCL1 (מס' 5) של Raspberry Pi 4B. |
|  | GND | מחוברת לרגל הארקה (מס' 14) של Raspberry Pi 4B. |

טבלה 4: חיבורי חיישן טמפרטורה Melexis MLX90640

## Logitech C922 Pro

|  |  |  |
| --- | --- | --- |
| **מספר הרגל** | **שם הרגל** | **לאן הרגל מחוברת** |
|  | USB-A | מחובר לשקע USB 2.0 של Raspberry Pi 4B. |

טבלה 5: חיבורי מצלמה Logitech C922 Pro

## HC-SR04

|  |  |  |
| --- | --- | --- |
| **מספר הרגל** | **שם הרגל** | **לאן הרגל מחוברת** |
|  | VCC | מחוברת לרגל 5V (מס' 2) של Raspberry Pi 4B. |
|  | Trig | מחוברת לרגל GPIO16 (מס' 36) של Raspberry Pi 4B. |
|  | Echo | מחוברת לרגל GPIO19 (מס' 35) של Raspberry Pi 4B. |
|  | GND | מחוברת לרגל הארקה (מס' 14) של Raspberry Pi 4B. |

טבלה 6: חיבורי חיישן מרחק HC-SR04

## רמקול

|  |  |  |
| --- | --- | --- |
| **מספר הרגל** | **שם הרגל** | **לאן הרגל מחוברת** |
|  | + | מחובר ל-Tip (ערוץ שמע שמאלי) של שקע 3.5 מ"מ ב-Raspberry Pi 4B. |
|  | - | מחובר ל-Sleeve (אדמה) של שקע 3.5 מ"מ ב-Raspberry Pi 4B. |

טבלה 7: חיבורי רמקול

## MHS3528

|  |  |  |
| --- | --- | --- |
| **מספר הרגל** | **שם הרגל** | **לאן הרגל מחוברת** |
| 2 | 5V | מחוברת לרגל 5V  (מס' 4) של ה-Raspberry Pi 4B. |
| 6 | GND/אדמה | מחוברת לרגל הארקה (מס' 6) של ה-Raspberry Pi 4B. |
| 17 | 3.3V | מחוברת לרגל 3.3V (מס' 17) של ה-Raspberry Pi 4B. |
| 18 | LCD\_RS | מחוברת לרגל GPIO24 (מס' 18) של ה-Raspberry Pi 4B. |
| 19 | LCD\_SI | מחוברת לרגל SPI\_MOSI (מס' 19) של ה-Raspberry Pi 4B. |
| 22 | RST | מחוברת לרגל GPIO25 (מס' 22) של ה-Raspberry Pi 4B. |
| 23 | SPI\_CLK | מחוברת לרגל SPI\_CLK (מס' 23) של ה-Raspberry Pi 4B. |
| 24 | LCD\_CS | מחוברת לרגל SPI\_CE0\_N (מס' 24) של ה-Raspberry Pi 4B. |

טבלה 8: חיבורי מסך NHS3528

## SG90 – דגם הדלת

|  |  |  |
| --- | --- | --- |
| **מספר הרגל** | **שם הרגל** | **לאן הרגל מחוברת** |
|  | VCC | מחוברת לרגל 5V (מס' 2) של Raspberry Pi 4B. |
|  | PWM | מחוברת לרגל PWM0 (מס' 32) של Raspberry Pi 4B. |
|  | GND | מחוברת לרגל הארקה (מס' 14) של Raspberry Pi 4B. |

טבלה 9: חיבורי מנוע סרוו SG90 של דגם הדלת

## SG90

|  |  |  |
| --- | --- | --- |
| **מספר הרגל** | **שם הרגל** | **לאן הרגל מחוברת** |
|  | VCC | מחוברת לרגל 5V (מס' 2) של Raspberry Pi 4B. |
|  | PWM | מחוברת לרגל PWM1 (מס' 33) של Raspberry Pi 4B. |
|  | GND | מחוברת לרגל הארקה (מס' 14) של Raspberry Pi 4B. |

טבלה 10: חיבורי מנוע סרוו SG90

# מימוש תוכנה

## train\_mask\_image\_classification\_model.ipynb

קובץ זה משמש כתוכנה לאימון מודל סיווג המסיכה שירוץ על גבי  
ה-Raspberry Pi. כאשר מאמנים מודל "למידת מכונה" (Deep Learning) דרוש כוח עיבוד רב, ולכן קובץ זה מיועד לפעול על מחשב ביתי חזק או על גבי שירותי ענן שמציעים מחשבים וירטואליים שמיועדים לכך – לדוגמא: Google Colab.

אנחנו בחרנו להשתמש ב-Google Colab מכיוון שאין לנו גישה למחשב חזק מספיק שמסוגל לבצע את המשימה בצורה טובה, ו-Google מציעים את השירות כשירות חינמי הכולל מחשבים וירטואליים שעליהם מותקנות מראש תוכנות לפיתוח מודלים של למידת מכונה ו-ראייה ממוחשבת (Computer Vision).

Google Colab עובד דרך הדפדפן בפורמט של Jupyter Notebook, שבו בעצם ניתן לפתוח מספר בלתי מוגבל של "תאים" – כל תא משמש כפונקציה בשפת Python, שהמשתנים שלה משותפים לכל המסמך. מבנה התאים מאפשר להריץ כל תא אחד אחרי השני, דבר שעוזר להבנה של התהליך ומפשט אותו. לאחר כל תא יש אפשרות לראות את הפלט של אותן הפקודות, אם יש כזה.

תא ראשון – הורדת נתונים, אתחול משתנים

מטרת התא הראשון הוא לייבא את הספריות שבהן נשתמש בתוכנה, ולהוריד את התמונות והנתונים שבעזרתם נאמן את המודל.

תא שני – עיבוד התמונות ומידע הנלווה

מטרת התא השני היא למיין את כל קבצי התמונות והמידע הנלווה אליהן שהורדנו לתיקיה בתא הראשון, לשני מערכים:

1. מערך מרובע מרובע-ממדים המיצג את התמונות, כאשר המערך החיצוני הוא אוסף של כל התמונות שלנו, וכל תמונה מיוצגת על ידי מערך תלת מימדי בגודל: 160x160x3.

160 על 160 היא הרזולוציה שהמודל שלנו יקבל בכניסה – 160 פיקסלים על 160 פיקסלים – כאשר כל פיקסל מיוצג על ידי 3 ערכים, המבטאים את צבע הפיקסל בפורמט סטנדרטי של RGB (Red, Green, Blue).

1. מערך דו-מימדי, כאשר המערך החיצוני הוא אוסף של הקטגוריות התואמות לכל תמונה שנמצאת במערך הראשון, בפורמט של מערך בגודל 2 נתונים בינאריים, כאשר הנתון הראשון מציין האם הפנים התמונה עוטות מסיכה בצורה תקינה, והנתון השני מציין האם הפנים בתמונה לא עוטות מסיכה בכלל. לדוגמא: [1, 0] מציין כי הפנים בתמונה עוטות מסיכה.

שני המערכים בגודל זהה אחד לשני, ומקבילים זה לזה – הערך הראשון במערך שמציין את הקטגוריה, מתייחס לתמונה הראשונה במערך של אוסף התמונות.

תא שלישי – הגדרת מבנה המודל

תפקיד התא הוא להגדיר את המבנה של המודל שלנו, מה הקלט שלו, ומה הפלט שלו.

מכיוון שפיתוח מודל כזה מצריך ידע וניסיון של שנים בתחום, ביססנו את המודל שלנו על בסיס מוכן של מודל בשם MobileNetV2 שנועד להיות קל ומהיר מאוד על מנת לרוץ על מכשירים בעלי כוח עיבוד מוגבל – למשל ה-Raspberry Pi – על מנת שהמודל יפעל כראוי ולא ייקח למכשיר זמן רב מדי להגיב. למעשה אנחנו משתמשים בשיטה שנקראת "העברת למידה (או "Transfer Learning"), שבה אנחנו עוקפים את השכבה האחרונה של המודל – שהיא השכבה של הפלט שמוציאה את המידע המציין מה הקטגוריה שהוא מסווג אליה את התמונה – ו"מובילים" את המודל אל שאר השכבות שלנו אשר על מנת להוציא פלט של 2 קטגוריות, ולאמן את אותן שכבות למציאת אלגוריתם שיזהה את שתי הקטגוריות שלנו. בשיטה זו אנחנו בעצם משתמשים באלגוריתם המקורי של המודל MobileNetV2 להבדלה מדויקת ומהירה בין הקטגוריות, ורק משנים אותו מעט על מנת שיזהה בסופו את הקטגוריות שלנו – ללא מסיכה, ועם מסיכה – בלי לשנות את אופן הפעולה שלו.

התא גם מגדיר כל אחת מהשכבות במודל הבסיס כשכבה שלא ניתקת לאימון מחדש – על מנת לא לפגוע באלגוריתם המאומן מראש. לאחר מכן, התא מבצע "קומפילציה" – הגדרה לאימון המודל.

תא רביעי – חלוקת המידע

התא הרביעי אחראי לחלק את המידע באופן רנדומלי לשני סטים של צמדי  
תמונה + קטגוריה, כאשר סט אחד הוא בגודל של 20% מהמידע, והסט השני בגודל של 80% מהמידע. לאחר מכן, התא מגדיר פונקציית אוגמנטציה – פונקציה שדרכה עובר המידע למודל, שתפקידה הוא לעבד מעט את תוכן התמונה כך שיהיו הבדלים קלים בין כל פעם שהמודל קורא את אותה התמונה. שלב זה נועד למנוע מהמודל מלהגיע למצב של "אימון יתר", הנגרם כתוצאה מחזרה רבה מדי על אותו סט מדויק של תמונות – בעקבות זאת, המודל ידע להתמודד רק עם התמונות שאיתן הוא אומן בעוד תמונות חדשות יקבלו ערכים שרירותיים או לא מדויקים מספיק. סט המידע בגודל של 20% משמש לבדיקה לאחר כל חזרה על פעולה האימון על מנת שהתוכנה תוכל לבדוק את יכולת המודל לזהות את הקטגוריות שביקשנו ממנו, ובהתאם לכך להתאים את האלגוריתם של המודל ולנסות שוב בחזרה הבאה.

תא חמישי – אימון המודל

תפקיד התא הוא לאתחל את פעולת אימון המודל, המתבצעת במספר שלבים:

1. העברה של מערך התמונות והקטגוריות המקבילות להן דרך פונקציית האוגמנטציה.
2. ניסיון ראשוני לסיווג התמונות דרך שכבות המודל.
3. בדיקה של דיוק המודל אל מול הסט בגודל 20% מהמידע שלנו שאליו הוא לא נחשף מעולם.
4. התוכנה מחפשת הבדלים בין התמונות בהן היא צדקה לתמונות בהן טעתה בסיווג, משנה מעט את האלגוריתם, וחוזרת על התהליך מספר פעמים על מנת למצוא את האלגוריתם הטוב ביותר.

עבור המודל שלנו, מצאנו כי 20 חזרות על הפעולה מביאות את המודל לדיוק של כ-97% אל מול תמונות שלא נחשף אליהן בעבר.

תא שישי – הורדת המודל

תפקיד התא הוא להמיר את המודל לקובץ של מודל בפורמט TensorFlow Lite.

TensorFlow Lite היא פלטפורמה מבית גוגל שכתובה בשפת C++ אליה ניתן להתממשק באמצעות ספריית Python, שמטרתה היא להריץ מודלים באופן יעיל ככל האפשר למכשירים בעלי כוח עיבוד מוגבל (למשל, מכשיר טלפון חכם או Raspberry Pi).

לאחר שמירת קובץ המודל במחשב הווירטואלי, התא מפעיל מוריד את הקובץ למחשב האישי דרך הדפדפן על מנת שנוכל להעביר אותו ל-Raspberry Pi.

## MLX90640.py

קובץ זה משמש להתממשקות וקריאת מידע מחיישן טמפרטורה מדגם MLX90640.

get\_scaled\_temp\_image()

יוצר מערך חד מימדי בגודל 768 ערכים ומאתחל אותם עם ערכים רנדומליים ולאחר מכן מנסה לקרוא 768 ערכים מחיישן הטמפרטורה בעזרת תקשורת I2C. אם הקריאה נכשלת משתמשים במערך הנתונים הרנדומלי על מנת לדמות את הפלט של החיישן.

לאחר מכן משנים את הצורה של המערך המתקבל ל32 על 24 והופכים אותו מימין לשמאל (מכיוון שסדר הנתונים שהחיישן פולט הוא מימין לשמאל, מלמעלה למטה – בעוד מערכים המייצגים תמונות ב-Python מסודרים משמאל לימין, מלמעלה למטה). לאחר מכן מגדילים את התמונה לרזולוציה של 480x640.

מה הפונקציה מקבלת:

הפונקציה לא מקבלת משתנים.

מה הפונקציה מחזירה:

מערך תלת-מימדי המייצג תמונה תרמית בגודל 480x640x1.

temp\_from\_random\_float()

הרכיב מביא לנו מערך של 768 ערכים בין 0 ל1 ומה שהפונקציה הזו עושה היא לוקחת את המערך הזה וממירה אותו למערך עם הטמפרטורה במעלות.

הפונקציה לא מקבלת משתנים.

הפונקציה לא מחזירה משתנים.

temp\_average\_from\_bbox()

הפונקציה מקבלת תמונה תרמית ואת המיקום של הפנים במסגרת ומחזירה את ממוצע הטמפרטורות במיקום של הפנים.

## MaskNet.py

קובץ זה משמש כספריה לתוכנה הראשית שמפעיל ה-Raspberry Pi.

בקבצי Python, התוכן של כל קובץ מתחילתו עד סופו משמש בעצם בתור התוכנית הראשית שלו, שנקראת בשם '\_\_main\_\_'. כאשר מייבאים את קובץ ה-Python כספרייה או פונקציות מתוכו, מתבצעת הפונקציה '\_\_main\_\_'.

'\_\_main\_\_'

הפונקציה הראשית מייבאת ספריות ומצהירה על פונקציות שנמצאות בשימוש בהמשך התוכנה. כמו כן, הפונקציה מאתחלת את מודל זיהוי הפנים ואת מודל סיווג המסיכה.

הפונקציה לא מקבלת משתנים.

הפונקציה לא מחזירה משתנים.

detect\_faces(frame)

תפקיד הפונקציה הוא לזהות פנים אנושיות בכל מסגרת (frame) שנמסר מהמצלמה. הפונקציה מסננת תוצאות בעלות אחוזי ביטחון נמוכים, גוזרת את התמונה, ומבצעת התאמות על מנת להתאים למודל סיווג המסיכה.

מה הפונקציה מקבלת:

מסגרת בודדה שנמסרה מהמצלמה, בפורמט של מערך תלת-מימדי  
בגודל 480x640x3.

מה הפונקציה מחזירה:

1. מערך דו-מימדי (בעל 4 ערכים בכל מערך פנימי, כאשר המערך החיצוני בגודל בלתי מוגבל) של מיקומי הפנים בתמונה, כאשר הפורמט לכל מיקום מייצג שתי נקודות; נקודת התחלה, שהיא הפינה השמאלית העליונית ביותר, ונקודת סיום שהיא הפינה הימנית התחתונה. הנקודות מצוינות לפי צירי X ו-Y ביחס לרזולוציית התמונה.
2. מערך מרובע-ממדים, שמייצג מערך בגודל בלתי מוגבל של מערכים תלת-ממדיים המיצגים תמונות. התמונות מעובדות מראש להתאים לקלט המודל לסיווג המסיכה, כאשר כל אחת מהתמונות היא מערך בגודל 160x160x3.

predict(faces)

הפונקציה מבצעת סיווג של כל התמונות המעובדות בעזרת המודל שפיתחנו לסיווג מסיכה. המודל מקבל תמונה, ומחזיר שני מספרים שמייצגים אחוזי ביטחון, ומשלימים ל-1. המספר הראשון מייצג את קטגוריית "עם מסיכה", והשני מייצג את קטגוריית "ללא מסיכה", בסדר הזה.

מה הפונקציה מקבלת:

מערך מרובע-ממדים, שמייצג מערך בגודל בלתי מוגבל של תמונות מעובדות מראש למודל סיווג המסיכה, כאשר כל אחת מהתמונות היא מערך בגודל 480x640x3.

מה הפונקציה מחזירה:

1. מערך דו-מימדי שגודלו שווה לכמות התמונות שקיבלה הפונקציה, שתוכנו הוא מחרוזות המבטאות את הקטגוריה בעלת אחוזי הביטחון הגבוהים יותר לכל תמונה.
2. מערך דו-מימדי שגודלו שווה לכמות התמונות שקיבלה הפונקציה, שתוכנו הוא מערכים בגודל של שני נתונים המייצגים את אחוזי הביטחון לכל קטגוריה של החיזוי לכל תמונה.

play\_no\_mask\_video()

ניגון קטע וידאו שכולל את סרטון הנחיית משרד הבריאות הישראלי לעטיית מסיכה באופן תקין.

הפונקציה לא מקבלת ולא מחזירה משתנים.

play\_audio(audio)

פונקציה פנימית בתוך הפונקציה 'play\_no\_mask\_video()', שתפקידה היא להשמיע את האודיו במקביל להצגת הוידאו.

מה הפונקציה מקבלת:

הפונקציה מקבלת קטע אודיו מסוג 'AudioSegment'.

מה הפונקציה מחזירה:

הפונקציה לא מחזירה משתנים.

play\_high\_temp\_screen()

הפונקציה מציגה על המסך תמונה שמתרה על כך שהלקוח לא מורשה להיכנס למתחם בעקבות חום גוף גבוהה מדי (ראה נספח 10).



נספח 10: התראה על חום גוף גבוה מהמותר

הפונקציה לא מקבלת משתנים.

הפונקציה לא מחזירה משתנים.

play\_entry\_allowed\_screen()

הפונקציה מציגה על המסך תמונה שמודיעה על כך שהלקוח מורשה להיכנס למתחם (ראה נספח 11).



נספח 11: התראה על אישור כניסה למתחם

הפונקציה לא מקבלת משתנים.

הפונקציה לא מחזירה משתנים.

draw\_boxes\_with\_predictions(frame, locs, pred\_labels, scores, temps)

תפקיד הפונקציה הוא הוא "לצייר" את המלבנים הכוללים את הפנים שזוהו, ומעליהם טקסט הכולל את הקטגוריה ואחוזי הביטחון שלה שניבא המודל לסיווג המסיכה, ואת הטמפרטורה שנמדדה לפנים באותה תמונה.

מה הפונקציה מקבלת:

הפונקציה מקבלת את המסגרת שנמסרה מהמצלמה, בנוסף למערכי הנתונים שאספנו בשלבים קודמים:

1. מערך המיקומים של הפנים.
2. מערך הקטגוריה של כל פנים.
3. מערך אחוזי הביטחון של מודל הסיווג.
4. מערך הטמפרטורה המדודה לכל הפנים שזוהו.

כל המערכים בעלי גודל חיצוני זהה ששווה למספר הפנים שזוהו במסגרת.

מה הפונקציה מחזירה:

הפונקציה מחזירה את המסגרת לאחר השינויים, בפורמט זהה לפורמט של הפריים שנכנס אליה (מערך תלת-מימדי בגודל 480x640x3).

VideoStream

בנוסף לפונקציות שמתוארות לעיל, קובץ זה כולל גם הגדרת מחלקה בשם 'VideoStream', שאחראית על קבלת המסגרות מהמצלמה. ההחלטה לבצע קריאת המסגרות במחלקה נפרדת נובעת מהרצון לבצע קריאה מהמצלמה במקביל לפעולות עיבוד הנתונים (בעזרת ספריה מובנית ב-Python שמשמשת להרצת פונקציות בתהליכונים מקבילים בשם 'threading'), כך שהזמן בין קבלת מסגרת, העיבוד וההצגה ששלה עד לקבלת המסגרת הבאה – מושג שידוע בשם Frames Per Second (או FPS בקיצור) – מתקצר משמעותית.

מחלקות בשפת Python יכולות לכלול את פונקציית '\_\_init\_\_', שמקבילה באופן פעילותה את פונקציית '\_\_main\_\_' בכך שהיא מתבצעת רק פעם אחת (כשמאתחלים משתנה עם אובייקט חדש שתואם לVideoStream), ובדרך כלל משמשת לאתחול משתנים פנימיים.

המחלקה כוללת מספר פונקציות פשוטות.

\_\_init\_\_(self, resolution)

משמשת לאתחול משתנים פנימיים במקרה חדש של VideoStream.

מה הפונקציה מקבלת:

הפונקציה מקבלת את המקרה הנוכחי שתואם למחלקה ואת הרזולוציה הרצויה של כל מסגרת בפורמט של מערך חד-מימדי בעל שני נתונים.

מה הפונקציה מחזירה:

הפונקציה מקבלת את המקרה הנוכחי החדש של VideoStream.

update(self)

מתחילה לולאה אינסופית שרצה כל עוד המשתנה הפנימי בשם 'stopped' של המקרה הנוכחי הוא שלילי. בכל חזרה של הלולאה מתבצעת קריאה של מסגרת חדשה מהמצלמה. המסגרת החדשה נשמרת במשתנה הפנימי בשם 'frame'.

מה הפונקציה מקבלת:

הפונקציה מקבלת את המקרה הנוכחי שתואם למחלקה.

הפונקציה לא מחזירה משתנים.

start(self)

תפקיד הפונקציה הוא ליצור תהליכון (thread) מקביל לפונקציה 'update'.

מה הפונקציה מקבלת:

הפונקציה מקבלת את המקרה הנוכחי שתואם למחלקה.

מה הפונקציה מחזירה:

הפונקציה מחזירה את המקרה הנוכחי שתואם למחלקה.

read(self)

המסגרת האחרונה שנקראה על ידי המקרה הנוכחי של המחלקה, בפורמט של מערך תלת-מימדי.

מה הפונקציה מקבלת:

הפונקציה מקבלת את המקרה הנוכחי שתואם למחלקה.

מה הפונקציה מחזירה:

המסגרת האחרונה שנקראה על ידי המקרה הנוכחי של המחלקה, בפורמט של מערך תלת-מימדי בגודל: x3(רזולוציה). לדוגמא: 480x640x3.

stop(self)

תפקידה הוא לבצעה השמה של ערך 'חיובי' במשתנה הפנימי בשם 'stopped', ובכך לעצור את הלולאה המתבצעת בפונקציה 'update()' ואת הקריאה של מסגרות מהמצלמה.

הערה: כל הפונקציות במחלקה מקבלות את המקרה הנוכחי שתואם למחלקה, על מנת לציין את המקרה שאליו אנחנו פונים. בשונה משפות אחרות, למשל שפות מבוססות C דוגמת C++, 'self' היא לא מילת מפתח מובנית בשפת התכנות, אלא פרמטר רגיל שלשמו אין חשיבות לאופן הפעולה של הפונקציה.

## Object\_detection\_webcam\_tflite.py

זוהי התוכנית הראשית של דגם הפרויקט, שמשתמשת בפונקציות מהקבצים שתוארו לעיל – MaskNet.py, MLX90640.py, ובמודל שיצרנו דרך שירות Google Colab עם הקובץ train\_mask\_image\_classification\_model.ipynb.

הקובץ כולל תוכנית ראשית שאינה מקבלת ואינה מחזירה משתנים, שתפקידה הוא לקלוט מסגרת מהמצלמה, למצוא את הפרצופים במסגרת, לסווג אותם לקטגוריה המתאימה באמצעות המודל שפיתחנו, למדוד את חום הגוף של כל פרצוף לפי מיקומו בתמונה, להציג את התוצאות על המסך, ולבצע מספר פעולות בהתאם לתוצאות:

* אם כל האנשים במסגרת נמדדו עם חום תקין, וזוהו עם מסיכה, המסך מציג הודעה על כך שקיבלו אישור להיכנס והמנגנון לפתיחת הדלת יאפשר לה להיפתח.
* אם לא כל האנשים שבמסגרת זוהו עם מסיכה, יוצג סרטון הכולל את הנחיות משרד הבריאות לעטיית מסיכה באופן תקין.
* אם כל האנשים במסגרת זוהו עם מסיכה, אבל לא לכולם יש חום גוף תקין, תוצג התראה על המסך שמודיעה כי חום הגוף של הלקוח גבוה ממה שההנחיות מתירות.

כל תוכן הפונקציה מתבצע בתוך לולאה אינסופית, על מנת שהתהליך יחזור על עצמו ללא הגבלה.

## door-model.py

קובץ זה משמש כתוכנה המדמה פעולה של דלת אוטומטית אלקטרונית.

בקבצי Python, התוכן של כל קובץ מתחילתו עד סופו משמש בעצם בתור התוכנית הראשית שלו, שנקראת בשם '\_\_main\_\_'. כאשר מפעילים את קובץ ה-Python במתבצעת הפונקציה '\_\_main\_\_'.

התוכנה בודקת בלולאה אינסופית את המרחק מהחיישן ומחליטה אם לפתוח או לסגור את הדלת בעזרת כמה פונקציות פשוטות.

door\_servo\_close()

ברגע שקוראים לפונקציה הזו היא בודקת אם הדלת פתוחה,

אם הדלת פתוחה היא מסובבת את מנוע הסרוו ל90 מעלות בכדי לסגור אותה.

הפונקציה לא מקבלת משתנים. הפונקציה לא מחזירה משתנים.

door\_servo\_open()

ברגע שקורים לפונקציה הזו היא בודקת אם הדלת סגורה,

אם הדלת סגורה היא מסובבת את מנוע הסרוו ל0 מעלות בכדי לפתוח אותה.

הפונקציה לא מקבלת משתנים. הפונקציה לא מחזירה משתנים.

distance()

פונקציה זו משתמשת בחיישן מרחק בכדי למדוד את המרחק מהצד השני של הדלת, ואם יש שינוי במרחק היא פותחת או סוגרת את הדלת בהתאמה.

מה הפונקציה מקבלת:

הפונקציה לא מקבלת משתנים.

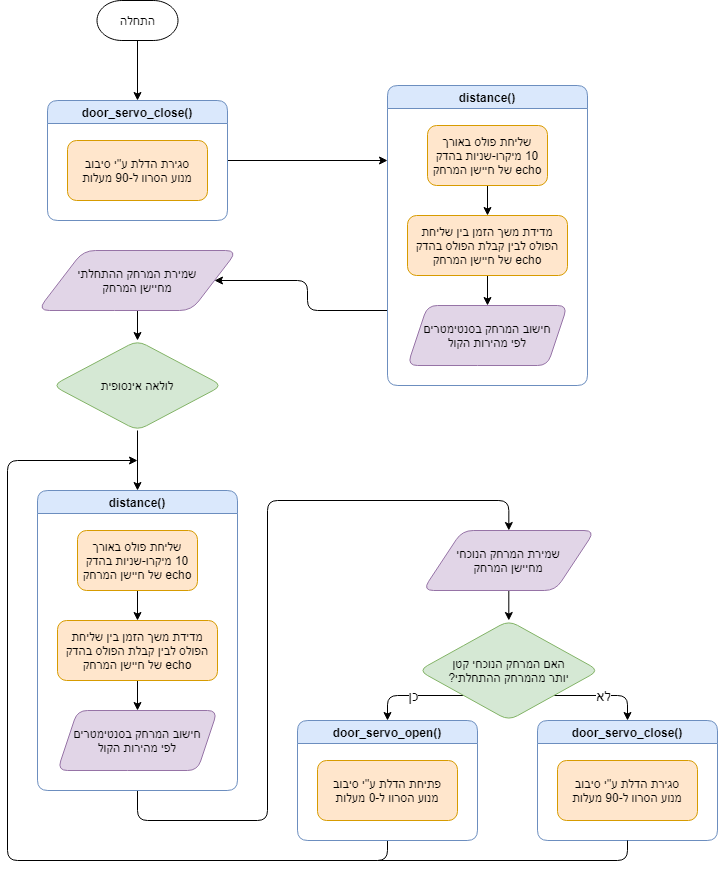
מה הפונקציה מחזירה:

הפונקציה מחזירה את המרחק בסנטימטרים.

# תרשים זרימת התוכנה

נספח 12: תרשים זרימה תוכנית ראשית

## דלת אוטומטית



נספח 13: תרשים זרימה דלת אוטומטית

# תיעוד

**14.10.2020**

פגישה ראשונה עם המנחה שלנו, אהרון ניסנבאום, הפגישה נערכה בזום. היכרות וניסיון למצוא יחדיו רעיונות.

**20.10.2020**

פגישה שנייה עם המנחה בזום. התייעצות לגבי המערכת לזיהוי המסיכה ומדידת טמפרטורה, מבררים לגבי האם הפרויקט בכלל אפשרי.

**23.10.2020**

החלטנו בהחלטה משותפת חברי הקבוצה ללכת על הרעיון של המערכת לזיהוי המסיכה, מתחילים לתכנן את מבנה הפרויקט ולחפש רכיבים.

**25.10.2020**

ביקשנו מרכז המגמה אם אפשר להשאיל Raspberry Pi 4B בשביל הפרויקט, והוא אישר לנו.

**02.11.2020**

החלטנו לבנות דגם של דלת אוטומטית שעובדת בצורה דומה לדלתות הקיימות בכניסה למתחמים שונים. החלטנו לעשות זאת באמצעות חיישן מרחק על מנת לזהות כאשר מישהו מתקרב לדלת, והדלת תיפתח בצורה סיבובית בעזרת מנוע סרוו.

**04.11.2021**

התחלנו ללמוד את שפת התכנות Python בעזרת קורס "Python for Beginners" של חברת Microsoft.

**10.11.2020**

לאחר שהזמנו את הרכיבים, גילינו שרכיב אחד לא קיים במלאי ומאוד קשה להשיג בתקופת הקורונה בישראל, במיוחד בגלל שקצב הייצור של הדגם הספציפי שאנחנו צריכים צנח, והוא חיישן הטמפרטורה MLX90640.

לאחר התייעצות עם המנחה שלנו, הגענו לשתי אופציות כיצד להמשיך:

1. להשתמש בדגם MLX90614, שהוא בעצם חיישן אינפרה-אדום בודד ולא מערך שיכול לשמש במצלמה תרמית. חיסרון בולט נוסף שלו הוא טווח קליטה קצר מאוד, של עד 4 סנטימטרים מרחק. בעקבות החיסרון השני נצטרך להשתמש בבקר נוסף על מנת לשלוט בו, ולהעביר את המידע דרך תקשורת אלחוטית בינו לבין הRaspberry Pi בזמן אמת.
2. לכתוב קוד שאמור להשתמש בחיישן MLX90640, ולמעשה לבצע הדמיה שלו על ידי שימוש בנתונים שמדמים את הפלט מהחיישן.

החלטנו לבצע את המשך הפרויקט השנייה.

**13.11.2020**

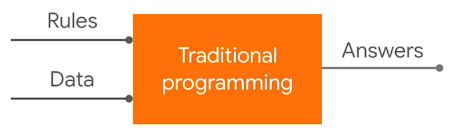
בבסיסו של הפרויקט עומד המודל לסיווג מסיכה, שמטרתו היא להיות מסוגל לקבל תמונה של פנים אנושיות ולהחזיר באופן יחסי לאיזה קבוצה היא משתייכת: עם מסיכה, ללא מסיכה, ועם מסיכה שלא לבושה כראוי.

על מנת לפתח את המודל הזה, התחלנו ללמוד על תחום "למידת מכונה"  
(Machine Learning). קיימות מספר פלטפורמות המאפשרות פיתוח של למידת מכונה בעזרת Python, אך לפי בדיקה שלנו הרבה מפתחים טוענים ש-TensorFlow מבית Google (שנועדה להכיל אוסף של כלים ופלטפורמות הקשורים לראיית מחשב ולמידת מכונה) הכי ידידותית למפתחים שרק נכנסים לתחום – זאת בעקבות Keras, פלטפורמה שקיימת כחלק מ-TensorFlow, שנוצרה בדיוק למטרת פיתוח מודלים פשוטים למתכנתים שרק נכנסים לתחום ללא ידע מוקדם.

**19.11.2020**

התחלנו ללמוד על פיתוח מודלים בסיסיים עם Keras, בעזרת סרטוני הסבר והדרכה של צוות המפתחים של Google.

סדרת סרטונים שמאוד עזרה לנו להבין את הדרך שבה עובדת למידת מכונה היא הרצאה מקוונת מאת לורנס מורוני בשם "ML Zero to Hero", שבה הוא מסביר על נושאים חשובים, כמו למשל ההבדל בין למידת מכונה לתכנות "רגיל" כאשר מנסים לסווג מידע לקטגוריות:



נספח 14: תהליך תכנות רגיל

בתכנות "רגיל" המפתח מגדיר כללים או חוקים מסוימים שלפיהם התוכנה מנסה לפענח ולסווג את המידע, ובכך להוציא תשובה מדויקת ככל האפשר. הבעיה היא שעל מנת למצוא הבדלים בין תמונות למשל, בהן אנחנו יכולים לנסות לזהות תבנית מסוימת של פיקסלים שחוזרים על עצמם, נצטרך לכתוב קוד מאוד ארוך ומורכב – שגם אז יתחשב רק במידע בו יכול המפתח להשתמש לבדיקה בזמן הפיתוח, ולכן התמודדות עם תמונות שלא נבדקו היא לא מובטחת.



נספח 15: תהליך תכנות בעזרת למידת מכונה

לעומת זאת, למידת מכונה מאפשרת לקבל מידע מוכן מראש, ואת התשובות שלו – על מנת להוציא את הכללים או החוקים לסיווג המידע לקטגוריות. המכונה בעצם מבצעת מספר ניחושים לגבי היחסים שבין הערכים בקלט בלולאה שמתבצעת מספר מוגדר של פעמים, ובודקת איזה מהניחושים שלה היו נכונים, ובכך בונה אלגוריתם מותאם.

מורוני מסביר גם על ההשוואה בין תפקיד הנוירונים של מוח האדם לבין נוירונים במודל למידת מכונה, ועל עוד נושאים מעניינים שעזרו לנו בהמשך.

**02.12.2020**

אספנו תמונות רבות הכוללות אנשים עם מסיכה , אנשים עם מסיכה שלא לבושה כראוי, ואנשים ללא מסיכה, וניסינו לפתח מודל למידה עמוקה לסיווג מסיכה בעזרת "העברת למידה" ממודל מסוג VGG16 כמודל בסיס. המודל עבד בצורה מדויקת למדי, עם כ-90 אחוזי ביטחון – אבל לקח לו יותר מדי זמן לספק תוצאות לתמונה בודדת, ולכן הוא לא מתאים לרוץ על מכשיר בעל כוח עיבוד יחסית חלש כמו ה-Raspberry Pi.

**05.12.2020**

החלפנו את מודל הבסיס למודל מסוג MobileNetV2, שנועד למכשירים בעלי כוח עיבוד חלש עם החיסרון של דיוק נמוך יותר. לא הצלחנו להגיע עם המודל לדיוק של יותר מ-60%. בדקנו ומצאנו שבגלל שהמודל נועד לתיעדוף של מהירות תגובה על פני תוצאות מדויקות, הוא לא מבחין בהבדלים קטנים בין תמונות בצורה טובה.

לאחר כמה בדיקות, הגענו למסקנה שקבוצת התמונות בקטגוריה של פנים עם מסיכה שלא לבושה כראוי דומה מאוד לשתי הקטגוריות האחרות, ולכן המודל לא מצליח ללמוד את ההבדלים בין הקטגוריות. בעקבות זאת החלטנו לא להשתמש בקטגוריה הזאת, ולסווג פרצופים לקטגוריה של עם מסיכה או בלי מסיכה. לאחר מכן, המודל הגיע לרמת דיוק של כ-97 אחוזים.

**19.12.2020**

****על מנת לבדוק שהמודל באמת פועל במהירות מספקת על ה-Raspberry Pi, כתבנו תוכנית Python קצרה שמודדת כמה זמן לוקח לו להפיק תוצאות לכל תמונה בממוצע.

נספח 16: קטע קוד לבדיקת מהירות המודל

התוצאות מראות שהזמן הממוצע שלוקח ל-Raspberry Pi לעבד את הנתונים בעזרת המודל הוא כ-55ms, או כ-18 מסגרות בשנייה (frames per second).

**17.1.2021**

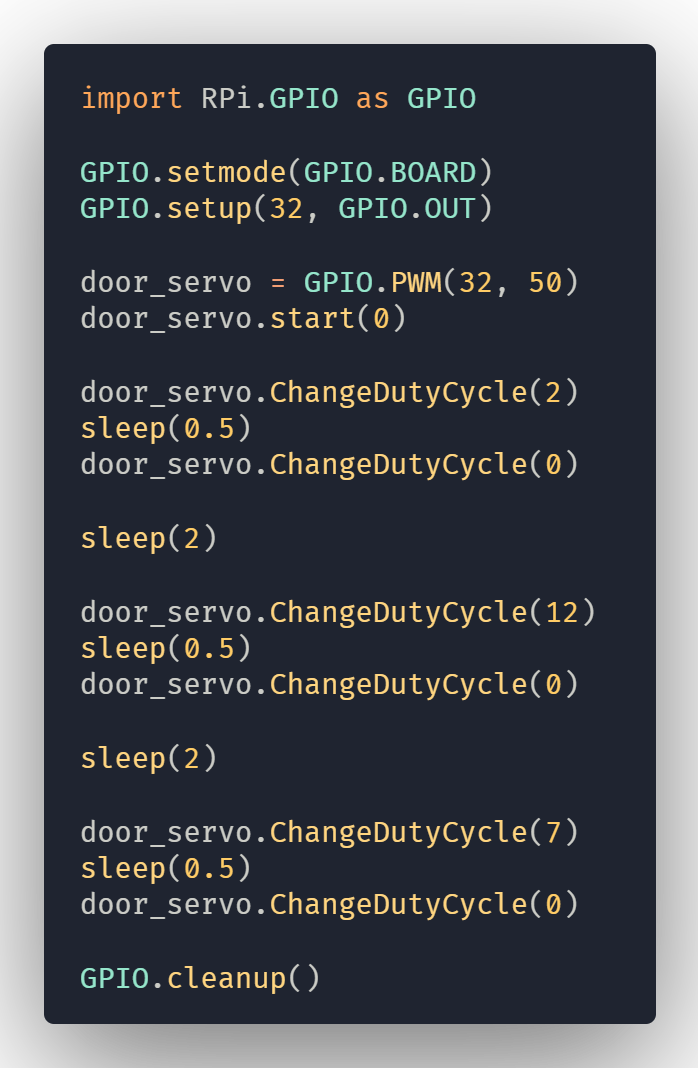
הזמנו מסך ל Raspberry Pi 4B מאתר eBay.

**20.2.2021**

הגיע המסך, קנינו מצלמה מחנות בארץ. בדיקה ראשונית של הקוד ותיקוני באגים בעזרת המצלמה והמסך שעתה הגיעו.

**22.02.2021**

כתבנו קטע קוד לבדיקת מנוע הסרוו.



נספח 17: קוד לבדיקת מנוע סרוו

הקוד מסובב את מנוע הסרוו לזווית של 0°, 180° ו-90° בסדר הזה, עם הפסקה באורך של 2 שניות בין כל סיבוב.

**25.02.2021**

כתבנו את הקובץ 'door-model.py' שמתבסס על קטע הקוד בעזרתו בדקנו את מנוע הסרוו, ביחד עם קוד לחישוב המרחק מהחיישן HC-SR04.

# רפלקציה

העבודה על הפרויקט והספר הייתה מאתגרת, אך למרות הקשיים והבעיות בהם נתקלנו במהלך ביצוע הפרויקט, הצלחנו להתגבר בעבודת צוות.

למדנו הרבה על תחומים שלא היה לנו ידע קודם בהם, למשל Python כשפת תכנות מונחית עצמים, למידת מכונה (Machine Learning), עבודה עם רכיבים אלקטרוניים מסוגים שונים ועוד. כתיבת התוכנה הייתה החלק הכי מאתגר עבורנו, בעיקר בתחום של למידת מכונה, מכיוון שהוא מגיע ממקצוע מדע הנתונים ולא מהנדסת אלקטרוניקה. למרות האתגרים הצלחנו להגיע לדגם מוגמר, שעונה על מטרת הפרויקט.

# רשימה ביבליוגרפית

**מגפת הקורונה**

<https://he.wikipedia.org/wiki/מגפת_הקורונה_בישראל>

<https://www.clalit.co.il/he/your_health/family/Pages/corona_in_israel.aspx>

<https://covid19.apple.com/contacttracing>

<https://govextra.gov.il/ministry-of-health/corona/corona-virus/>

**TensorFlow Lite**

<https://www.tensorflow.org/lite/api_docs/python/tf/lite>

<https://www.tensorflow.org/lite/guide/inference#load_and_run_a_model_in_python>

<https://www.tensorflow.org/lite/inference_with_metadata/lite_support>

<https://www.tensorflow.org/lite/performance/post_training_integer_quant>

<https://www.tensorflow.org/lite/performance/post_training_quantization#full_integer_quantization_of_weights_and_activations>

<https://heartbeat.fritz.ai/running-tensorflow-lite-image-classification-models-in-python-92ef44b4cd47>

**MobileNet V2**

<https://arxiv.org/pdf/1801.04381.pdf>

**זיהוי עצמים בתמונה**

<https://towardsdatascience.com/face-detection-models-which-to-use-and-why-d263e82c302c>

<https://github.com/datitran/object_detector_app/blob/master/object_detection_multithreading.py>

<https://github.com/achen353/Face-Mask-Detector>

<https://arxiv.org/pdf/2101.00784.pdf>

**Python**

<https://www.youtube.com/playlist?list=PLlrxD0HtieHhS8VzuMCfQD4uJ9yne1mE6>

**חיישן טמפרטורה**

<https://www.melexis.com/en/documents/documentation/datasheets/datasheet-mlx90640>

<https://media.digikey.com/pdf/Data%20Sheets/Pimoroni%20PDFs/PIM365_Web.pdf>

<https://www.melexis.com/en/product/mlx90640/far-infrared-thermal-sensor-array>

**Raspberry Pi 4B**

<https://magpi.raspberrypi.org/articles/raspberry-pi-4-specs-benchmarks>

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

<https://www.raspberrypi.org/documentation/usage/gpio/>

**חיישן מרחק**

<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

<https://gabbyshimoni.wixsite.com/arduino-programming/blank-19>

**מצלמה**

<https://www.explainthatstuff.com/webcams.html>

<https://lanoc.org/review/photography/7381-logitech-c922-pro-stream-webcam>

**פרוטוקול I2C**

<http://www.arikporat.com/projects/I2C%20comunication.pdf>

<https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/>

<https://hackstore.co.il/arduino-i2c-spi-protocols/>

**פרוטוקול SPI**

<https://hackstore.co.il/arduino-i2c-spi-protocols/>

<https://cs.hac.ac.il/staff/martin/seminar/תקשורת%20סריאלית%20-%20דביר%20דדון.pdf>

**PWM**

<https://www.idogendel.com/whitebyte/archives/2700>

**LCD**

<http://www.lcdwiki.com/res/MHS3528/MHS-3.5inch_Display_User_Manual_EN.pdf>

<http://www.lcdwiki.com/MHS-3.5inch_RPi_Display>

<https://he.wikipedia.org/wiki/LCD>

**רמקול**

<https://mynewmicrophone.com/how-do-headphone-jacks-and-plugs-work-wiring-diagrams/>

<https://he.wikipedia.org/wiki/%D7%A8%D7%9E%D7%A7%D7%95%D7%9C>

**SSH**

<https://www.easycloud.co.il/guides/how-to-connect-to-your-application-using-ssh/>

**הנחיות משרד הבריאות**

<https://www.gov.il/he/departments/faq/corona-communication-resources>