

הטכניון - מכון טכנולוגי לישראל
הפקולטה להנדסת חשמל ע"ש אנדרו וארנה ויטרבי
המעבדה לראייה ומדעי התמונה

דו"ח סיכום פרויקט: א'

מערכת לפתיחת דלתות מבוססת ESP32-CAM

ESP32-cam door security system

מבצעים:

Eden Ishay
Yael Tzadik

עדן ישי
יעל צדיק

Johanan Erez

מנחה: יוחנן ארז

סמסטר רישום: חורף תשפ"ג

תאריך הגשה: מאי, 2024

תוכן עניינים

1	מבוא	1
1.1	מטרת הפרוייקט	1
2	חומרה	1
2.1	סקירת רכיב Esp32-CAM	1
2.1.1	חיבור רשת	2
2.2	בחירת סביבת עבודה	3
2.2.1	ARDUINO	3
2.2.2	ESP-IDF on VSCODE	3
3	עבודות קודמות	4
3.1	Esp32-CAM camera webserver v1.0.4	4
3.2	משקפיים לחולי אלצהיימר	7
4	פתרון מוצע	7
4.1	מבנה כללי	7
4.1.1	הגדרות חיבורי הרכיב	8
4.1.2	הפעלת המערכת – חיבור רכיב לרשת	9
4.1.3	הפעלת המערכת – חיבור רכיב למחשב	9
4.1.4	תיאור ריצה פעילה	10
4.2	אלגוריתם המערכת המרכזי	10
4.3	אלגוריתמים וכלי עזר	11
4.4	שיטות נוספות לשיפור המערכת	18
4.5	בחינת ביצועים	18
5	רעיונות שנפסלו	19

20	6. תוצאות ומסקנות.....
21	7. הצעות לפרוייקט המשך.....
22	רשימת מקורות.....

רשימת איורים

2	איור 1 – רכיב ESP32-CAM
4	איור 2 – ממשק camera webserver
5	איור 3 - תוצר רשת MTMN
6	איור 4 - דוגמא לשתי תמונות פנים של אותו אדם, ללא חיוך
7	איור 5 - דיאגרמת בלוקים רכיבי המערכת המוצעת
9	איור 6 - תרשים זרימה חיבור מערכת לרשת
9	איור 7 - צילום מסך טרמינל bluetooth בעת חיבור לרכיב
9	איור 8 – חלון התחברות לשרת מציאת landmarks
10	איור 9 - תרשים זרימה אלגוריתם המערכת המוצעת
11	איור 10 - תרשים מערכת MTMN
11	איור 11 - תמונות לדוגמא, בחינת ביצועים MTMN
13	איור 12 - מפת הנקודות של אלגוריתם facemesh
14	איור 13 - מפת נקודות מצומצמת עבור הפתרון המוצע
16	איור 14 - מרחקים בין נקודות לזיהוי חיוך לפי Nadir Trapsida
17	איור 15 - מרחקים בין נקודות לזיהוי חיוך

רשימת טבלאות

2	טבלה 1 – נתוני הרכיב ESP32-CAM
6	טבלה 2 – בחינת ביצועים אלגוריתם ESP-WHO
12	טבלה 3 – בחינת ביצועים אלגוריתם MTMN
13	טבלה 4 – השוואה בין ספריית MediaPipe ו OpenFace
15	טבלה 5 - השוואה בין 468 נקודות ל 20 נקודות
18	טבלה 6 – השוואה בין ESP-WHO לבין האלגוריתם שלנו
20	טבלה 7 – השוואה בין האלגוריתמים השונים עבור זווית של 0 מעלות

תקציר

בפרויקט זה פיתחנו מערכת ניידת לפתיחת דלתות המורכבת מכרטיס האלקטרוני esp32-camera וסיוע מחשב. המערכת המבוססת על אלגוריתמים קיימים של זיהוי פנים מספריות פתוחות בפייתון ושימוש באלגוריתם חדש למניעת זיוף על ידי הצגת תמונה כתחליף לאדם המזוהה. המערכת מבססת על מאפיינים ותכונות המובנות בקוד המקור בדוגמא לזיהוי פנים שפותחה לסביבת Arduino IED המומלצת להתממשקות עם הרכיב. אלגוריתם זיהוי הפנים בו השתמשנו מתבסס על נקודות ציון הנפרשות על ידי רשת נוירונים ושימוש בספריית Mediapipe בפייתון והשוואת ממוצע המרחקים שלהן מנקודה נבחרת במרכז הפנים. מציאת הנקודות והחישוב המתמטי הארוך של הממוצעים מתבצעת על מחשב המקושר לרכיב דרך רשת Wi-Fi לתמיכה ביכולות החישוב המוגבלות של הרכיב. תקשורת עם הרכיב היא לפי פרוטוקול HTTP. לאחר החישובים, נבדקת התאמה למאגר תמונות של האנשים המורשים לכניסה. אם האדם זוהה נבדקת האפשרות לזיוף על ידי בדיקת פעולה דינאמית של חיוך ורק לאחר מכן ניתן אות אישור לפתיחת הדלת.

Abstract

In this project we developed a mobile door opening system consisting of the esp32-camera electronic card and computer assistance. The system is based on existing facial recognition algorithms from open libraries in Python and the use of a new algorithm to prevent forgery by presenting an image as a replacement for the identified person.

The system is based on characteristics and features built into the source code in the facial recognition example developed for the Arduino IED environment recommended for interfacing with the component. The face recognition algorithm we used is based on landmarks spread by a neural network and the use of the Mediapipe library in Python and comparing the average of their distances from a selected point in the center of the face. The finding of the landmarks and the long mathematical calculation of the averages is carried out on a computer linked to the component via a Wi-Fi network to support the limited calculation capabilities of the component. Communication with the component is according to the HTTP protocol.

After the calculations, compatibility with an image database of the people authorized to enter is checked. If the person has been identified, the possibility of forgery is checked by testing the dynamic action of a smile, and only then a confirmation signal given to open the door.

A half-page summary of the project in Hebrew.

1. מבוא

בפרויקט זה פיתחנו מערכת לפתיחת דלתות על ידי שימוש באלגוריתמים קיימים לזיהוי פנים באמצעות הרכיב ESP32-cam ותוך התחשבות במגבלות החומרה שלו. האלגוריתם הקיים בדוגמאות הקוד של מפתחי הרכיב בסביבת Arduino Ide קורס לאחר ניסיון זיהוי של יותר מ-3 אנשים (מתוך 5 אפשריים). המערכת שלנו מהווה שיפור משמעותי ואותה מימשנו על ידי קישור הרכיב באמצעות פרוטוקול הרשת HTTP למחשב קרוב שמסוגל להריץ אלגוריתם לזיהוי פנים המבוסס נקודות ציון על הפנים (בדומה לקוד המקור של המפתחים). במהלך הפרויקט כולו השתדלנו להיצמד ליכולות הרכיב ולמגבלותיו החומרתיות.

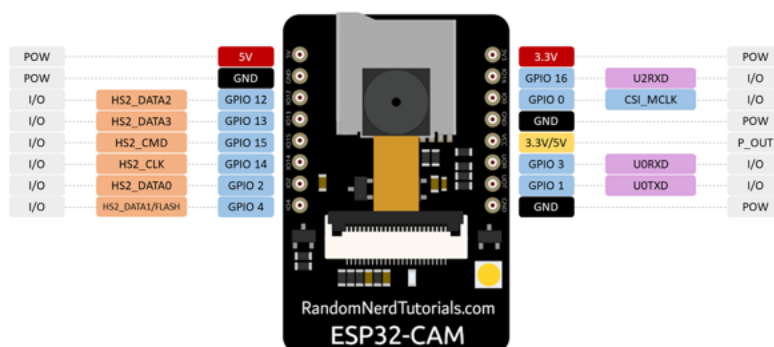
1.1. מטרת הפרויקט

בניית מערכת לזיהוי פנים לצורך פתיחת דלתות המתבססת על אלגוריתמים קיימים ופיתוח אלגוריתם למניעת זיופים תוך שימוש בחומרה הקיימת ברכיב ESP32-cam. עקב מגבלות החומרה הפרויקט אינו מתמקד בפיתוח אלגוריתם למניעת זיופים אלא באינטגרציה בין הרכיב למחשב חיצוני המאפשר פעולה תקינה של המערכת.

2. חומרה

2.1. סקירת רכיב Esp32-CAM

רכיב ESP32 Camera הוא רכיב חשמלי מתקדם שפותח על ידי חברת Espressif Systems על פי היצרן זהו פתרון אידיאלי עבור יישומי IoT, אלקטרוניקה לבישה ומערכות בית חכם. הלוח משלב שני מעבדי 32 סיביות LX6. הוא מכיל ארכיטקטורת צינור בת 7 שלבים, חישן על-שבב, חישן הול, חישן טמפרטורה פנימי וכן הלאה, וטווחי התאמת התדר העיקריים שלו מ-80 מגה-הרץ עד 240 מגה-הרץ. הרכיב ESP32 Camera מתאים במיוחד ליישומים המצריכים יכולות צילום ותמונה מותאמות אישית ומרוחקות ותומך במצלמות OV2640 או OV7670, ומצויה בו ממשקי מודולציה מתוחברים אליו. יתרה מכך, הרכיב מגיע עם ממשקי רשת אלחוטית מובנים, Wi-Fi | Bluetooth מסורתיים, מה שמאפשר תקשורת אלחוטית חלקה ויעילה ואף מוסיפה אפשרויות רחבות לשליטה ושדרוג. נתוני הרכיב לפי היצרן [1]



איור 1 – רכיב ESP32-CAM

טבלה 1 – נתוני הרכיב ESP32-CAM

clock speed	160MHz
SRAM	520 KB
PRAM	4MB
Supported cameras	OV2640 and OV7670
Bluetooth	4.2 BR/EDR and BLE standards
Wi-Fi	802.11b/g/n/e/i
Security	WPA/WPA2/WPA2-Enterprise/WPS
Supported TF card	Max of 4GB
Serial Port Baud-rate	115200 bps
Image Output Format	JPEG(OV2640only),BMP, GRAYSCALE
Spectrum Range	2412 ~2484MHz
Power supply range	3.3V – 5v
Flash lamp	Included at several intensities

2.1.1. חיבור רשת

תקני ה-Wi-Fi הנתמכים של הרכיב מקבוצה 802.11 פותחו על ידי הוועדה לתקני LAN/MAN של ארגון IEEE [2] ומוגבלים לשימוש בתדרים ללא רישוי באזור 2.4 GHz אשר חשוף להפרעות מתנורי מיקרוגל וטלפונים. משפחת תקנים זאת משמשת גם ליצירת קשר נקודה לנקודה אלחוטי. פרוטוקול האבטחה של הרכיב מאפשר גם חיבור לרשת מסוג WPA2-Enterprise. הרכיב מאפשר גם חיבור לפי פרוטוקול Bluetooth אך לא ניתן להשתמש ב-Wi-Fi ו Bluetooth במקביל.

2.2. בחירת סביבת עבודה

קיימות שתי סביבות עבודה אשר תומכות בעבודה עם הרכיב [3] :

1. ARDUINO

2. ESP_IDF – אנחנו בדקנו את הגרסה שמגיעה כהרחבה לסביבת העבודה VS-CODE

במהלך העבודה על הפרויקט התנסנו בשתי סביבות העבודה הללו.

2.2.1. ARDUINO

- שימוש בשפת תכנות C/C++
- סביבת עבודה פשוטה יותר, מתאימה למתחילים.
- מבצעת את פעולות הקישור והקומפילציה מאחורי הקלעים.
- הנחת ברירת מחדל, עבודה עם מעבד יחיד.
- הסביבה מאפשרת גישה למגוון רחב של ספריות כולל ספריות המתאימות לעבודה עם הרכיב במספר גרסאות.
- סביבת עבודה מוכרת וותיקה עם קהילה גדולה של מפתחים ותיקוד רב.
- מציעה שיטת debugging פשוטה לשימוש באמצעות הדפסות, וניהול הזיכרון נקבע לפי אחת ממספר אפשרויות קבועות מראש.

2.2.2. ESP-IDF on VSCODE

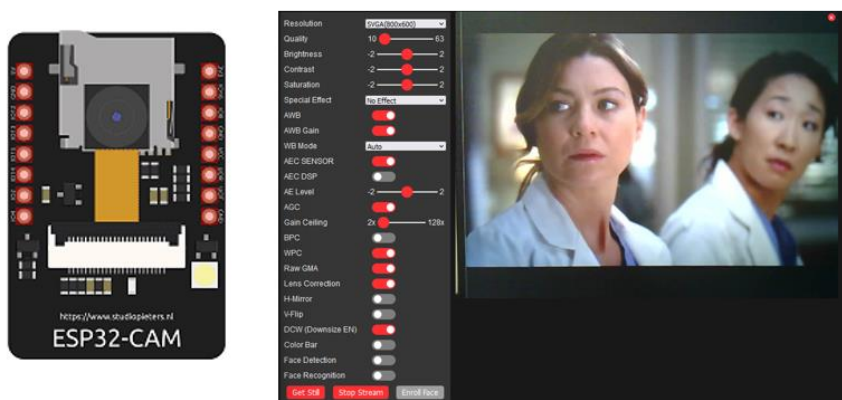
- שימוש בשפת תכנות C/C++
- סביבת עבודה מורכבת יותר, מתאימה למפתחים מתקדמים עם ניסיון בעבודה עם מיקרו בקרים.
- הסביבה מאפשרת יותר גמישות בקישור הספריות, אך הקישור, הקומפילציה וצריבת הקוד לרכיב צריכים להתבצע באופן ידני.
- הנחת ברירת מחדל, עבודה עם מספר מעבדים.
- הסביבה מאפשר גישה למגוון רחב אפילו יותר של ספריות ופיצ'רים ספציפיים לפיתוח בתחום IoT
- סביבת עבודה עם קהילה מצומצמת יותר ומקצועית יותר, עם זאת חלקים גדולים ממנה לא זמינים בשפה אנגלית.
- מציעה כלים מתקדמים יותר לניהול זיכרון על גבי הרכיב ודיבוג.
- קומפילציה וקישור חלקיים ובהתאם לשינויים שהוכנסו לקוד, ולכן תהליך הצריבה מהיר יותר.

אף על פי ש ARDUINO היא מעטפת של סביבת ESP-IDF, לא ניתן לקשר ספריות שנוצרו עבור IDF לתוך פרויקט ARDUINO, ולכן לא ניתן לשלב עבודה על שתי הפלטפורמות בפרויקט אחד. [1] שנכתבה במיוחד עבור רכיב ESP32. (Joachim, n.d.) בפועל התברר כי הספרייה הזאת נכתבה עבור רכיב שונה מאת אותו יצרן ובגרסה הנוכחית שלה לא ניתן לקשר אותה לרכיב הספציפי הזה. בנוסף, הספריות הבסיסיות שנכתבו על ידי Espressif עדיין נמצאו בשלבי פיתוח והעבודה איתן הייתה מאתגרת במיוחד. ללא השימוש בספריית OpenCV מצאנו שמגוון הספריות המוצא על ידי ARDUINO הוא מספק, ומכיוון שהעבודה עם סביבה זאת פשוטה יותר ואמינה יותר הוחלט לעבור להשתמש בה. ESP-IDF היא סביבה שעדיין נמצאת בשלבי פיתוח, ייתכן שבעתיד או עבור רכיב שונה, סביבת הפיתוח הזאת תהיה עדיפה.

3. עבודות קודמות

3.1. Esp32-CAM camera webserver v1.0.4

מערכת לדוגמא נתונה על ידי מפתחי הרכיב. זוהי המערכת שאנחנו שואפים לשפר. הרכיב מתחיל שרת בפרוטוקול HTTP בכתובת IP פנויה ברשת ודרכו מציג את ממשק המערכת.



איור 2 – ממשק camera webserver

המערכת מסוגלת לזהות נוכחות של אדם בתמונה, ולעקוב אחר מיקום הפנים שלו בתוך טווח הראייה של המצלמה. האלגוריתם מזהה מיקום של 5 נקודות פיצ'ר על גבי הפנים של המטרה ולשמור אותם עבור 7 אנשים, נקרא להם משתמשים מורשים. המערכת מסוגלת לזהות עבור מטרה חדשה האם היא מכירה אותו, כלומר האם הוא אחד משבעת האנשים השמורים, או האם מדובר באדם זר. אלגוריתם הזיהוי הזה מכונה ESP-WHO [2] עושה שימוש במספר אלגוריתמים:

- *MTMN face detection* – מודל מאומן מראש לצורך זיהוי נוכחות של אדם בשדה הראיה של המצלמה וחמש נקודות על גבי הפנים: עין ימנית, עין שמאלית, מרכז האף, פינה ימנית של הפה, פינה שמאלית של הפה.



איור 3 - תוצר רשת MTMN

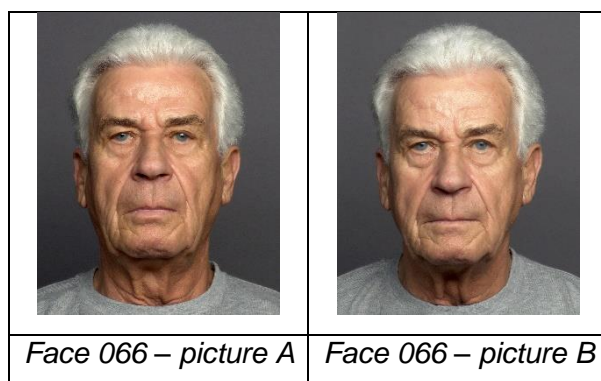
- *Cosine distance face recognition* – האלגוריתם מחשב את *cosine distance* בין ווקטור הנקודות הנוכחי לבין ווקטור הנקודות של שבעת האנשים השמורים ומחליט לפי פרמטרים קבועים מראש האם האדם שמולו הוא אחד מהשבעה או מטרה זרה.

חסרונות של המערכת:

- איטית – אלגוריתם *ESP-WHO* מאט משמעותית את ריצת הרכיב.
- זיהוי המטרה מוגבל ל 7 משתמשים מורשים, בפועל המערכת התחילה לקרוס אחרי שהוכנסו 5 משתמשים מורשים.
- זיהוי נקודות הפיצ'ר מוגבל ל 5 נקודות בלבד ופועל עבור מבט ישיר לתוך המצלמה בלבד.
- שמירת הנתונים נעשית על גבי זיכרון נדיף ויש צורך להזין אותם מחדש בכל פעם שהרכיב מפעיל את עצמו מחדש.
- המערכת אינה מבדילה בין אדם אמיתי לבין תמונה שלו.
- גרסאות חדשות יותר של המערכת הפסיקו לתמוך ברכיב *esp32-cam*. ניתן לאלץ אותן לפעול עבור הרכיב אך היצרן מזהיר שזמן החישוב הוא מעל 15 שניות לפריים.

ביצועים:

למערכת הוצגו כ 50 תמונות פנים של 25 אנשים (2 תמונות פר אדם) כל האנשים צולמו כאשר הם מביטים ישירות למצלמה ולא מחייכים.



איור 4 - דוגמא לשתי תמונות פנים של אותו אדם, ללא חיוך

טבלה 2 – בחינת ביצועים אלגוריתם ESP-WHO

<i>Fps while face detection active</i>	1.4fps
<i>Fps while face detection and recognition active</i>	0.9 fps (with 5 enrolled)

	0° angle	30° angle
<i>Correct recognition rate</i>	34%	68%
<i>Face recognition false positive rate (Authorizing unauthorized person)</i>	83%	40%
<i>Face recognition true positive rate (Recognizing authorized person correctly)</i>	100%	100%
<i>Correlation of angled and not</i>	52%	

תהליך הבדיקה בוצע על מספר מצומצם של אנשים. 25 מועמדים בסך הכל, מתוכם 5 בעלי הרשאות כניסה. הסיבה לכך היא שהבחנו שעבור מספר גדול יותר של אנשים הרכיב נוטה להתחמם ולקרוס. כאשר ביצענו את ההשוואה בין אלגוריתם זה לבין האלגוריתם המשופר שמרנו על אותו סטנדרט בדיקה.

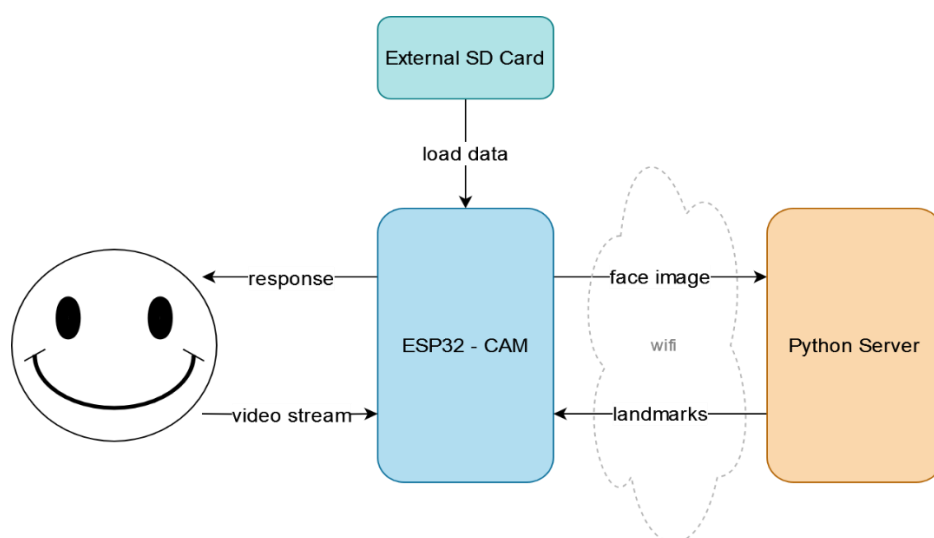
3.2. משקפיים לחולי אלצהיימר

A face recognition application for Alzheimer's patients using ESP32-CAM and Raspberry Pi [5]

מאמר זה מציע אפליקציה לזיהוי פנים בזמן אמת עבור אנשים החיים עם אלצהיימר. המערכת מזהה אוטומטית אנשים ושולחת משוב קולי למשתמש. המערכת משלבת את הרכיבים ESP32-CAM ו-Raspberry Pi כאשר התמונה נרכשת באמצעות הרכיב ESP32-CAM ונקלטת על ידי הרכיב Raspberry Pi שם מבוצע אלגוריתם זיהוי נוכחות פנים באמצעות על יד המודל MobileNetV2 וזיהוי האדם על ידי DenseNet-16. אחוזי הדיוק עבור זיהוי אדם במערכת זו מוערכים להיות מעל 91%.

4. פתרון מוצע

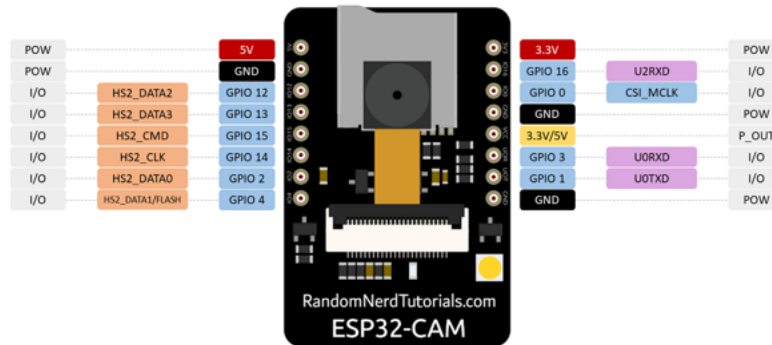
4.1. מבנה כללי



איור 5 - דיאגרמת בלוקים רכיבי המערכת המוצעת

הגדרות חיבורי הרכיב

4.1.1.



את חיבור הרכיב למקור מתח ולמחשב לצריבת התוכנה ביצענו בהתאם להנחיות המפתחים ולפי דוגמאות לפרויקטים קודמים שמצאנו באינטרנט. ההגדרות שבחרנו הן ברירת המחדל של הדוגמה CameraWebServer של

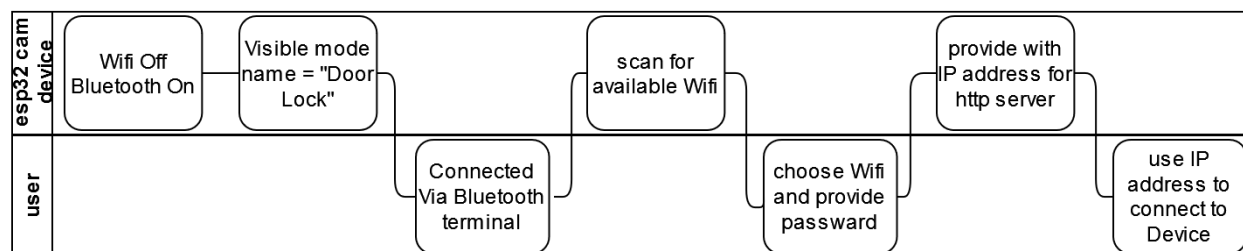
ארדואינו:

```
#define PWDN_GPIO_NUM    32
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM     0
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22
```

4.1.2. הפעלת המערכת – חיבור רכיב לרשת

בעת הפעלת הרכיב או בעת הפעלה מחדש הרכיב זמין לפתיחת קשר Bluetooth תחת השם "Door Lock" ניתן לתקשר עם הרכיב בשלב זה באמצעות Bluetooth terminal. אלגוריתם זה מבוסס על האלגוריתם שנבנה על ידי "robotzero1" [3] תהליך החיבור מפורט באיור 6 ואיור 7.



איור 6 - תרשים זרימה חיבור מערכת לרשת

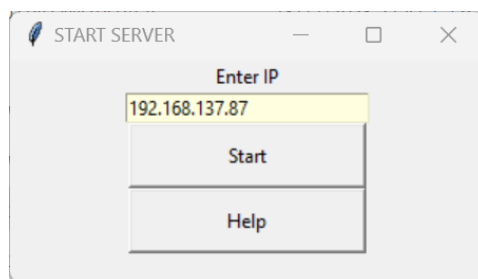
```

01:19:32.908
01:19:32.939 Scanning Wi-Fi networks
01:19:34.389 2 networks found
01:19:35.369 1: LAPTOP (Strength:-66)
01:19:35.369 2: HOTBOX 4-87C8 (Strength:-75)
01:19:35.369 Please enter the number for your Wi-Fi
01:19:39.478 1
01:19:40.516 Please enter your Wi-Fi password
01:19:44.094 12345678
01:19:45.117 Please wait for Wi-Fi connection...
01:19:51.545 ESP32 IP: 192.168.137.139
01:19:51.651 Bluetooth disconnecting...
01:19:52.681 Connection lost
01:20:42.456 Connecting to DoorLockControler ...
01:20:49.072 Connected
01:20:49.112
  
```

איור 7 - צילום מסך טרמינל bluetooth בעת חיבור לרכיב

4.1.3. הפעלת המערכת –חיבור רכיב למחשב

על מנת לחבר את הרכיב לשרת שיבצע את חישוב נקודות ה landmark יש לתת לשרת את כתובת ה IP של הרכיב.



איור 8 – חלון התחברות לשרת מציאת landmarks

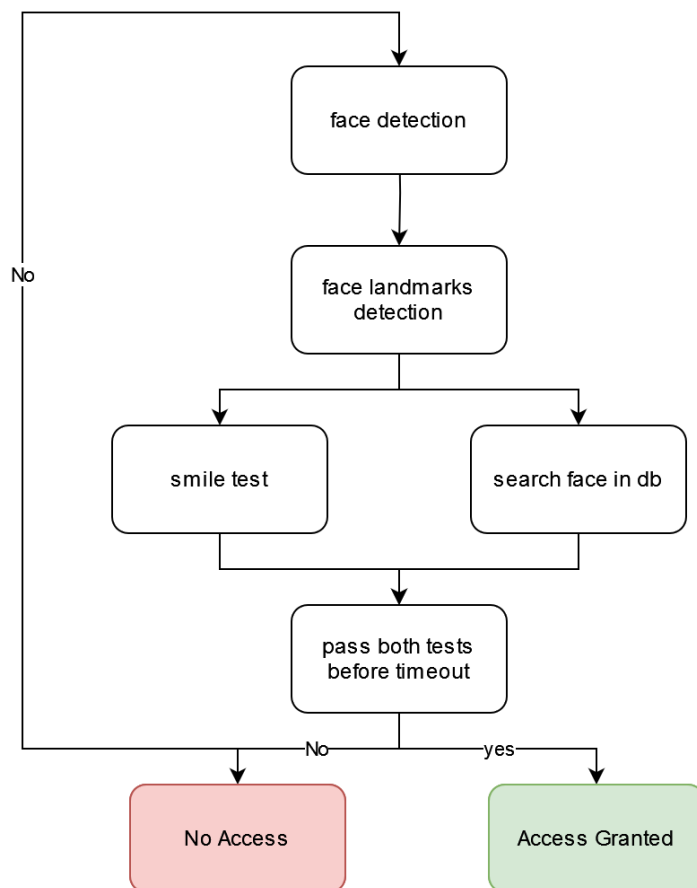
הרכיב מקשר את מערכת הקבצים של כרטיס הזיכרון החיצוני וטוען מתוכו את המידע של המשתמשים המורשים.

תיאור ריצה פעילה

4.1.4.

הרכיב עושה שימוש במצלמה על מנת לקלוט זרם רציף של תמונות. כאשר הרכיב מזהה נוכחות של פנים בשדה הראייה שלו הוא שולח את התמונה לשרת python השרת מחלץ נקודות Landmark מתוך הפנים בתמונה ושולח חלק מהן בחזרה לרכיב הרכיב מחשב האם הפנים הן פנים עם הרשאת כניסה ומנסה לזהות חיוך הרכיב מחזיר חיווי של הצלחה או כישלון.

4.2. אלגוריתם המערכת המרכזי



איור 9 - תרשים זרימה אלגוריתם המערכת המוצעת

מצב ההמתנה של הרכיב הוא המתנה לפנים בשדה הראייה.

חיפוש הפנים ממשיך להתבצע כל עוד לא התקבלה תשובה. אם מזהה חוסר רציפות בנוכחות פנים בשדה הראייה של המצלמה האלגוריתם, הרכיב חוזר לנקודת ההתחלה. השלב הבא באלגוריתם הוא חיפוש נקודות פיצ'ר. חלק זה מתבצע על גבי השרת באמצעות הספרייה face mesh ונקודות אלו נשלחות בחזרה אל ה ESP32. על גבי הרכיב מבוצעות שתי פעולות:

- זיהוי פנים באמצעות אלגוריתם השוואת המרחקים של נקודות פיצ'ר על גבי הפנים
- זיהוי דינמי של חיוך באמצעות מעקב אחר 10 תמונות הפנים האחרונות.

אם שני המבחנים נתנו תשובה חיובית במסגרת זמן של 10 שניות החל מרגע זיהוי נוכחות של פנים, האלגוריתם מזהה זיהוי חיובי ונותן חיווי של הצלחה. אחרת הוא נותן חיווי של כישלון.

4.3. אלגוריתמים וכלי עזר

זיהוי נוכחות פנים באמצעות [1] – דגם מצומצם לזיהוי פנים אנושי, אשר בנוי סביב ארכיטקטורות MobileNetV2 ו MCTNN ותוכן במיוחד עבור רכיבים קטנים.

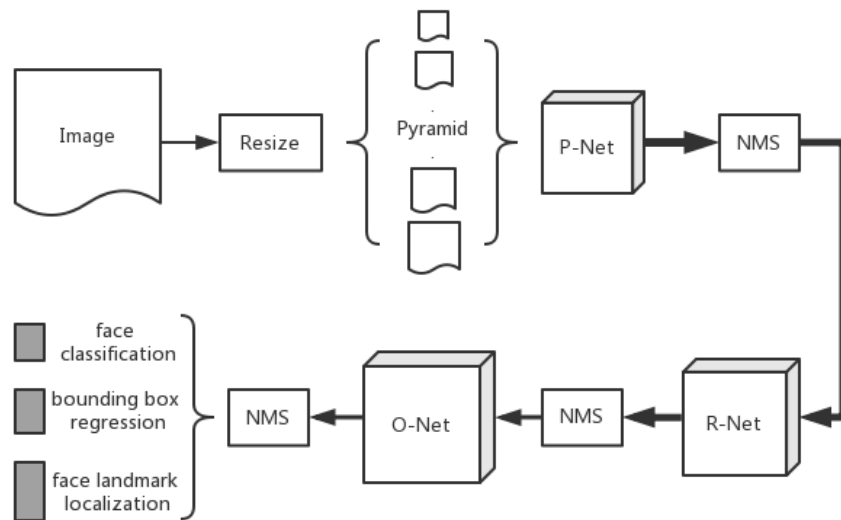
הרשת מורכבת משלושה חלקים עיקריים:

1. רשת הצעות P-net – מציעה תיבות תחומות של מועמד בתמונה.

2. רשת R-net – מסן את התיבות מהרשת הקודמת.

3. רשת פלט O-net – מציגה את התיבה הנבחרת הסופית

על מנת להקטין את המודל הוחלפו הבלוקים המוכרים בארכיטקטורת CTNN בבלוקים מתוך MobileNetV2



איור 10 - תרשים מערכת MTMN

תוצאות על גבי הרכיב:

לרכיב הוצגו 25 תמונות פנים ו 10 תמונות שאינן פנים כולל תמונות חיוך מצויר

Hallway (ot a face)	Smile icon image	Face number 004	Face number 066

MTMN - תמונות לדוגמא, בחינת ביצועים 11 איור

טבלה 3 – בחינת ביצועים אלגוריתם MTMN

100%				אחוזי הצלחה בגילוי פנים תוך פחות מ 3 שניות	
<1%				אחוזי גילוי שווא	
~15 sec				זמן ממוצע עד כשלון זיהוי בפעם הראשונה עבור אדם שאינו זז (תמונה מקובעת)	
1.4fps				FPS בזמן פעולת האלגוריתם לבדו	
center of left Eye				סטיית תקן מיקום landmark עבור אדם שאינו זז במשך 100 frames (תמונה מקובעת)	
0.55 pixel	y-axis	0.54 pixel	x-axis		
center of right eye					
0.48 pixel	y-axis	0.52 pixel	x-axis		
center of nose					
1.11 pixel	y-axis	0.65 pixel	x-axis		
Left mouth corner					
1.01 pixel	y-axis	1.1 pixel	x-axis		
Right mouth corner					
0.98 pixel	y-axis	0.56 pixel	x-axis		

סטיית תקן מיקום landmark עבור אדם שאינו זז במשך 100 frames (תמונה מקובעת)

סטיית תקן של מיקום 5 הנקודות נבדקה לפי מיקום כל נקודה בנפרד בלבד וחושבה באופן הבא:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

כאשר הערך x_i הוא ערך הפיקסל בציר x או ציר y מעוגל לפי round סטנדרטי.

המטרה הייתה להבין האם רמת הדיוק היא פחות מפיקסל יחיד. ניתן לראות שהתקבלה רמת דיוק גבוהה יותר עבור נקודות העיניים מאשר עבור הנקודות בקצוות הפה והאף.

חילוץ נקודות face landmark points באמצעות face Mesh – לאחר שהוחלט לבצע את שני השלבים הבאים באלגוריתם באמצעות נקודות landmark על גבי מחשב חיצוני [4]:

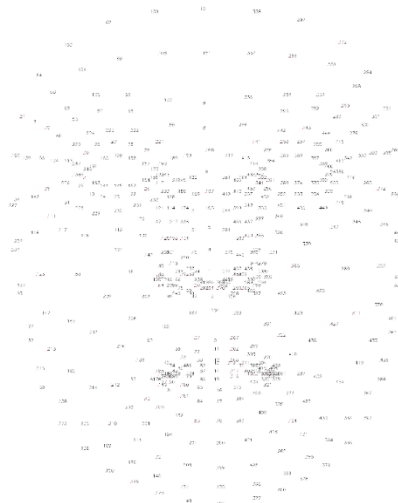
- ספריית OpenFace
- ספריית MediaPipe – אלגוריתם face mesh

טבלה 4 – השוואה בין ספריית OpenFace ו MediaPipe

ספריית MediaPipe	ספריית OpenFace	
468	68	מספר נקודות
3.2-5.1	2.8-4.1	סטייה ממוצעת
10ms or 100fps	1ms or 1000fps	מהירות ממוצעת
תמיכה ב 3D נקודות מהצורה (x,y,z)	אין תמיכה ב 3D נקודות מהצורה (x,y)	3D

לאחר ההשוואה הוחלט להשתמש באלגוריתם FaceMesh במהלך הרצת גרסות ניסיון של המערכת נמצא שהגורם המגביל הראשי של קצת פעולת המערכת הוא זמן השליחה של landmark points מהשרת לרכיב לכן הוחלט להגביל את מספר הנקודות להיות 20. גם לאחר הגבלת מספר הנקודות קצב העבודה של הרכיב לצורך face detection הוא נמוך מ 3fps ולכן אין משמעות לאיזו ספרייה מהירה יותר. המודל שניתן באמצעות הספרייה FaceMesh מאפשר גמישות גבוהה יותר בבחירת הנקודות ודיוק גבוה יותר ולכן הוא נבחר.

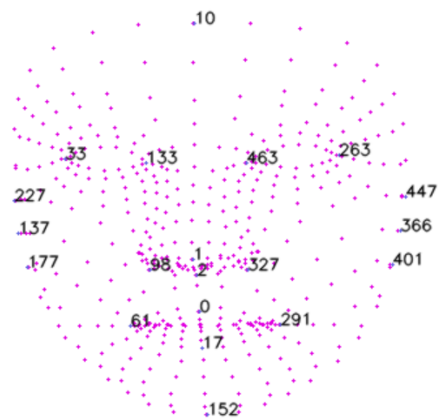
מספר הנקודות המקורי שהאלגוריתם מספק הוא 468 והמפה נתונה באופן הבא:



איור 12 - מפת הנקודות של אלגוריתם facemesh

כפי שראינו במאמר מאת Yu-jin Hong קיימת שיטה קלה לזיהוי פנים באמצעות תמונה באיכות ירודה המבוססת על השוואת מרחקים של נקודות ציון אנטומיות בין תמונת פנים לתמונת רפרנס. הנקודות המשוות שייכות לאזורים בהם נבדלים אנשים שונים. לכן הוחלט לצמצם את מספר הנקודות שמצאנו באמצעות רשת הנוירונים המאומנת של ספריית facemesh לרשימת האזורים שנמצאו היעילים ביותר לפי המאמר [10]:

מספר הנקודה באלגוריתם	מיקום הנקודה	מספר הנקודה באלגוריתם	מיקום הנקודה
1	מרכז האף	227	קצה עליון אذن שמאל
2	תחתית האף	137	מרכז אذن שמאל
327	צד ימין אף	177	קצה תחתון אذن שמאל
98	צד שמאל אף	447	קצה עליון אذن ימין
0	מרכז שפה עליונה	366	מרכז אذن ימין
17	מרכז שפה תחתונה	401	קצה תחתון אذن ימין
291	קצה שפה ימני	33	קצה שמאל עין שמאל
61	קצה שפה שמאלי	133	קצה ימין עין שמאל
10	מרכז קו מצח	463	קצה שמאל עין ימין
152	קצה סנטר	263	קצה ימין עין ימין



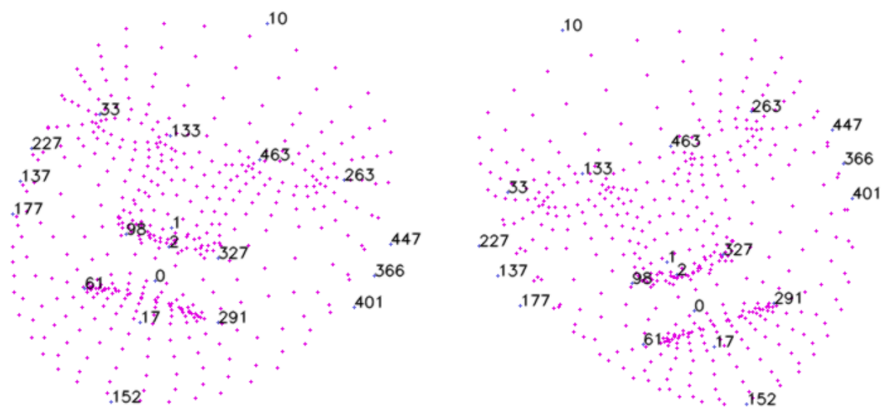
איור 13 - מפת נקודות מצומצמת עבור הפתרון המוצע

ההבדל בביצועים עבור רשת FaceMesh:

טבלה 5 - השוואה בין 468 נקודות ל 20 נקודות

20 נקודות	468 נקודות	
~0.2 sec	> 3sec	זמן חישוב ממוצע
1.62 fps	0.25 fps	מהירות תקשורת עם הרכיב
60%	84%	אחוזי דיוק

זיהוי זהות פנים באמצעות אלגוריתם השוואת מרחקי נקודות הציון האנטומיות - חיסרון אחד של אלגוריתם זיהוי נוכחות הפנים שנבחר הוא חוסר היכולת של הרכיב לזהות פנים כאשר המבט לא מופנה ישירות למצלמה. האלגוריתם כן מסוגל לזהות פנים בהטיה של הראש לצדדים לכן הוחלט להתמקד בבעיה זאת. האלגוריתם מבצע חישוב של המרחק בין הנקודה 1 שהיא מרכז האף לשאר הנקודות על גבי הפנים ומשתמש בווקטור המידע הזה לצורך ההשוואה, לכן האלגוריתם אינו רגיסי להטיה של הראש מצד לצד.



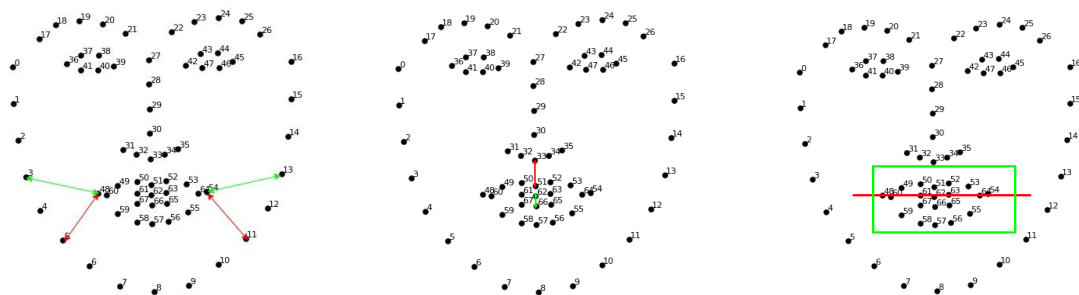
ביצועי האלגוריתם עבור 468 נקודות:

	0° angle	30° angle
Correct recognition rate	84%	72%
Face recognition false positive rate (Authorizing unauthorized person)	15%	25%
Face recognition true positive rate (Recognizing authorized person correctly)	80%	60%
Correlation of angled and not	80%	

ביצועי האלגוריתם עבור 20 נקודות:

	0° angle	30° angle
Correct recognition rate	60%	44%
Face recognition false positive rate (Authorizing unauthorized person)	45%	32%
Face recognition true positive rate (Recognizing authorized person correctly)	100%	60%
Correlation of angled and not	60%	

זיהוי חיוך – השיטה שנבחרה לזיהוי חיוך מבוססת על כתבה מאת Nadir Trapsida [1]. במערכת שהוצגה על ידי נאדיר נמדדו מרחקי פנים נחות וחיוך הוגדר בתור גדילה של מרחקים במרחק בין קצוות הפה האופקיים והלסת בנוסף למרחק בין השפתיים ובין השפתיים לאף ומיקום מרכז הפה ביחס לקו השפתיים.



איור 14 - מרחקים בין נקודות לזיהוי חיוך לפי Nadir Trapsida

הבעיות שאנחנו מצאנו בשיטה הזאת:

- השיטה נועדה לזיהוי חיוך סטטי, לא לשינוי דינמי ממבט ניטרלי לחיוך.
- השיטה משתמשת בקצוות הלסת ועבור אנשים מסוימים, בעיקר אנשים צעירים, קצוות הלסת נעלמים מאחורי הלחיים בזמן חיוך.

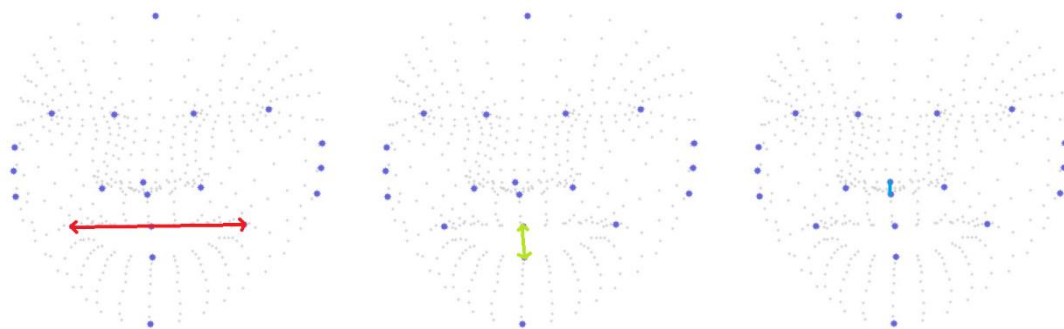
אנחנו מציעות שיפור לשיטת החישוב:

הרכיב ישמור את 20 המרחקים האחרונים בין קצוות הפה האופקיים ואת 20 המרחקים האחרונים בין השפתיים בטור.

את ההשוואה בין קצוות הפה לקצוות הלסת והלחיים החלפנו במדידת השינוי במרחק בין קצוות הפה האופקיים.

את ההשוואה בין מרחק השפה העליונה לתחתית האף החלפנו במדידת המרחק האנכי בין השפתיים.

המרחקים האלו מנורמלים למרחק בין נקודה 1 לנקודה 2 כפי שניתן לראות במפת הנקודות במצומצמת, כדי להתגבר על ההבדלים בין מרחק הפנים והמצלמה.



איור 15 - מרחקים בין נקודות לזיהוי חיוך

הטור הזה מתעדכן בכל frame ומתאפס כאשר לא מופיעות פנים תמונות בשדה הראייה. בכל frame שמצולם האלגוריתם מחשב את המרחקים הממוצעים עבור החצי הראשון של הטור ועבור החצי השני של הטור ומשווה ביניהם.

חיוך מוגדר כגדילה של 20% במרחק קצוות הפה האופקיים וגם גדילה של 100% במרחק השפתיים. את המרחק בין השפתיים הגדרנו כגדל ב-100% כיוון שרצינו לזהות את פעולת פתיחת הפה כלומר מעבר בין מרחק אפסי למרחק נמדד.

לצורך קביעת אחוזי גדולת המרחק בין קצוות הפה האופקיים ביקשנו מ 10 אנשים לבצע 3 פעולות:

1. לחייך ללא שיניים
2. לחייך עם שיניים
3. לפהק

הצלחה הוגדה בתור זיהוי של חיוך עם שיניים בלבד.

סף מינימום גדילה במרחק שפתיים אופקיים	מזהה חיוך עם שיניים	מזהה חיוך ללא שיניים כחיוך	מזהה פיהוק כחיוך	אחוזי הצלחה
10%	100%	0%	100%	66.66%
50%	20%	0%	0%	73.33%
30%	60%	0%	0%	86.66%
20%	90%	0%	10%	93.33%

כלומר אחוזי הצלחה האופטימליים התקבלו עבור גדילה של 20% במרחק קצוות הפה.

4.4. שיטות נוספות לשיפור המערכת

הקטנת משקל התמונה הנשלחת – לפני שליחת תמונה מהרכיב לשרת, התמונה עוברת קידוד בפרוטוקול JPEG [11].

על מנת להקטין יותר את גודל הקובץ הנשלח, בעת זיהוי נוכחות פנים בתוך שדה הראיה הרכיב צובע באפור x8080800 את כל התמונה פרט לאזור בו זוהו פנים. ממבחנים שבצענו במעבדה הצלחנו להקטין את גודל הקובץ הנשלח ב 65% עבור פנים במרחק 45 סנטימטרים מהמצלמה כאשר היא מספקת רזולוציה CIF ולהאיץ את זמן השליחה.

4.5. בחינת ביצועים

את ביצועי הרכיב בדקנו ממספר זוויות על מנת להבין כיצד תגיב המערכת לשינויים קטנים בזווית האדם הניגש אליה ובכך לדמות מצב מציאותי עד כמה שניתן.

טבלה 6 – השוואה בין ESP-WHO לבין האלגוריתם שלנו

	Original algorithm		Our algorithm	
	0° angle	30° angle	0° angle	30° angle
Correct recognition rate	34%	68%	60%	44%
Face recognition false positive rate (Authorizing unauthorized person)	83%	40%	45%	32%
Face recognition true positive rate (Recognizing authorized person correctly)	100%	100%	100%	60%
Correlation of angled and not	52%		60%	

ניתן לראות כי יש ירידה משמעותית באחוזי הזיהוי השגוי עבור האלגוריתם שפיתחנו. לאלגוריתם המקורי יש נטייה ברורה לזיהוי חיובי של האדם שמולה ולכן אחוזי הזיהוי החיובי השגוי שלו גבוהים.

5. רעיונות שנפסלו

הקטנת משקל התמונה הנשלחת באמצעות חיתוך התמונה – תחילה נבחן הרעיון לפיו בעת זיהוי נוכחות פנים, הרכיב ישלח לשרת תמונה מוקטנת של הפנים בלבד, בפועל התברר שהרשת FaceMesh אשר מחלצת את נקודות ה landmark מתקשה לזהות את המיקומים הנכונים כשאר התמונה ניתנת לה בגדלים משתנים, ולכן הוחלט לצבוע את הרקע במקום לחתוך אותו.

האצת החיפוש על ידי חיפוש לפי צבע עיניים – במקור נבחנה האפשרות להשתמש ב 5 נקודות הציון המקוריות שמסופקות על ידי הרשת MTMN על מנת לחלץ את צבע הקשתית. המטרה הייתה לחלק את המאגר ל 3 קבוצות קטנות יותר ולטעון מתוך כרטיס הזיכרון לרכיב רק את המועמדים הרלוונטיים. כבדיקה מקדימה חושבה היציבות של נקודות הציון של העיניים באמצעות חישוב סטיית התקן הממוצעת של המיקום שלהן. אף על פי שסטיית התקן יצאה נמוכה במידה מספקת, בשימוש בפועל הרכיב ביצע עודף של קריאות מתוך כרטיס הזיכרון מה שהאט את הריצה משמעותית. בעיה נוספת שעלתה משימוש באלגוריתם כזה נבעה ממקור מאגר תמונות הפנים. ספי הצבע חושבו לפי אנטרופיה מקסימלית של צבע הקשתית של המועמדים במאגר אך מכיוון שמקור המאגר במדינה אירופאית בה לאחוז גבוה יותר באוכלוסייה יש עיניים בהירות, שני צבעי הסף יצאו כחולים ולכן לא התאימו לאוכלוסייה הישראלית.

זיהוי מסך באמצעות ברק Flash – בתחילה בחנו את האפשרות של מניעת זיהוי פנים על גבי מסך על ידי בדיקת האור החוזר מהמסך. רעיון זה נפסל כיוון הרכיב לא מצליח להכיל את הספרייה OpenCV ולתמוך בה או בגרסאות מוקטנות שלה, ולכן לא מצליח לבצע עיבוד תמונה מתקדם הנדרש לביצוע הרעיון.

שימוש בזוויות שונות על מנת למדוד מרחקים גם בציר העומק – אחת מהמטרות של הפרויקט היא לבנות אלגוריתם שיאפשר להבדיל בין הדפס פנים לבין אדם אמיתי מול המצלמה. לצורך זה נבחנה האפשרות למצוא את נקודות הפנים במספר זוויות כדי לחשב הבדלי עומק בין הנקודות. רעיון זה נפסל לאחר שהתברר כי אלגוריתם ESP-WHO אינו מסוגל למצוא נקודות landmark בדיוק מספק, במיוחד עבור פנים שמוצגות בזווית. ספריית face-mesh אמורה לספק נקודות גם בציר Z, כלומר ציר העומק אבל לאחר עבודה עם הספרייה הוחלט שגם נקודות אלו לא ברמת דיוק גבוהה. לאחר הוספת האלגוריתם לבדיקת חיוך הוחלט שאין צורך באלגוריתם נוסף להבדיל בין פנים אמיתיות לתמונה.

6. תוצאות ומסקנות

בתחילת הפרויקט ראינו את הפוטנציאל הגדול הטמון ברכיב Esp32-camera ובייחודיות שלו בכך שהוא מכיל מעבד ומצלמה ואף תומך בתקשורת WIFI .

לאחר בדיקת התוצאות וניתוחן הגענו למסקנה כי הרכיב אינו מתאים לתפקד כמערכת עצמאית כיוון שיכולות החישוב שלו מוגבלות מאוד. הרכיב מתאים מאוד להיות החלק במערכת שמקשר את הקלט מהמצלמה שבו ומכשירים חיצוניים שמסוגלים לחשב ולנתח את המידע שהוא מעביר. בנוסף היכולת הקישורית של הרכיב טובה מאוד ולכן הוא מתאים למערכת בה משתמשים במספר רכיבי ESP32-camera ומחשב ראשי המנתח את כלל המידע שהם קולטים.

בנוסף לאחר בירור מעמיק גילינו כי החברה המפתחת של הרכיב חסמה עבור גרסאות מתקדמות שלה את האפשרות להשתמש ביכולת זיהוי הפנים המקורית שרצה על הרכיב עצמו.

את הפרויקט שלנו כתבנו כך אם וכאשר יהיה רכיב חדש בעל יכולות חישוב חזקות יותר ניתן יהיה להריץ את המערכת עליו כיחידה עצמאית ללא תמיכה במחשב. כלומר, רוב החישוב מתבצע על הרכיב עצמו למעט החישוב הכבד של חיפוש נקודות הציון על הפנים המתבצע במחשב חיצוני. בהינתן רכיב חזק יותר, ניתן להעביר את פונקציית החיפוש לקוד הרץ על הרכיב עצמו.

טבלה 7 – השוואה בין האלגוריתמים השונים עבור זווית של 0 מעלות

	<i>Original algorithm – ESP WHO</i>	<i>Our algorithm</i>	
		<i>20 landmarks</i>	<i>468 landmarks</i>
<i>Correct recognition rate</i>	34%	60%	84%
<i>Face recognition false positive rate (Authorizing unauthorized person)</i>	83%	45%	15%
<i>Face recognition true positive rate (Recognizing authorized person correctly)</i>	100%	100%	80%

בעת העבודה על הפרויקט שאפנו ל 2 מטרות עיקריות, להעלות את רמת הדיוק של אלגוריתם זיהוי הפנים ולהוסיף אפשרות תמיכה במספר רב ככול האפשר של משתמשים. על מנת להעלות את רמת הדיוק החלפנו את הספרייה אשר מזהה את נקודות הציון על גבי הפנים לספרייה שמספקת יותר נקודות ברמת דיוק גבוהה יותר,

אולם התברר שלרכיב אין את כוח החישוב לעבד 468 נקודות בזמן סביר או את המספיק מקום בזיכרון הנדיף כדי להכיל יותר ממספר אנשים בודד בכל פעם. בעת מעבר ל20 נקודות ציון הצלחנו להאיץ את החישוב פי 15 ולהגדיל את כמות האנשים שניתן להשוות ממספר בודד לכמה עשרות אך במחיר רמת הדיוק של האלגוריתם.

האמצעי למניעת זיוף בו בחרנו להשתמש הוא זיהוי חיוך. חיוך דינמי של אדם קשה יותר לזיוף על ידי חיתוך דף מודפס עם פנים המאגר.

במהלך הפרויקט עבדנו עם שישה רכיבי ESP32-cam. שני רכיבים התחממו יתר על מידה והמצלמה הפסיקה לעבוד. שני רכיבים נוספים אין ניתנים לצריבה חוזרת. אנחנו מסיקות שהרכיב אינו עמיד ואינו אמין לשימוש ממושך.

7. הצעות לפרויקט המשך

אנו ממליצות להמשיך לפתח את הפרויקט הנוכחי על גבי רכיב אחר בעל יכולות חישוב חזקות יותר ובכך להוספת בדיקות זיוף נוספות וניתוח נקודות ציון רבות יותר ובכך לייעל את מערכת הזיהוי. לאורך הדו"ח הצענו מספר הצעות ורעיונות של אלגוריתמים שנפסלו עקב מגבלות הרכיב ואותן ניתן לנסות על גבי רכיב חדש.

בנוסף, האלגוריתם שבו השתמשנו להשוואת מרחקי הנקודות הממוצע וזיהוי האדם, נעשה עם אלגוריתם פשוט ומצומצם שניתן להרחיבו לאלגוריתמים מוכחים ומורכבים יותר כגון fisher linear discriminate אך זאת רק עם רכיב בעל יכולות חישוב רחבות יותר

בנוסף ניתן לנסות ליצור מערכת חדשה המשתמשת בקישוריות של שני רכיבי ESP32-camera המבצעים מינימום חישובים ומקושרים למחשב חזק המנתח את הקלט שלהן משתי זוויות שונות. בכך מנצלים את היתרונות המובהקים של הרכיב ונמנעים מהחסרונות שלו.

רשימת מקורות

- [1] dfrobot, "ESP32-CAM Development Board," 15 8 2019. [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf.
- [2] IEEE, "IEEE 802.11TM WIRELESS LOCAL AREA NETWORKS," [Online]. Available: <https://www.ieee802.org/11/>.
- [3] ESPBoards, "espboards/esp-idf-vs-arduino-core," 18 3 2023. [Online]. Available: <https://www.espboards.dev/blog/esp-idf-vs-arduino-core/>.
- [4] B. Joachim, "github/esp32-opencv," 6 8 2021. [Online]. Available: <https://github.com/joachimBurket/esp32-opencv>.
- [5] shootao, "github/shootao/esp32-who," 2019. [Online]. Available: <https://github.com/shootao/esp32-who>.
- [6] W. H. N. S. Z. D. B. A. Thair A. Kadhim, "A face recognition application for Alzheimer's patients using," *IEEE*.
- [7] robotzero1, "github/robotzero1/esp32-bluewifi," 31 1 2020. [Online]. Available: <https://github.com/robotzero1/esp32-bluewifi>.
- [8] zhouyangyale, "github/M5Stack-Camera/MTMN," 2019. [Online]. Available: https://github.com/m5stack/M5Stack-Camera/blob/master/face_qr/components/esp-face/face_detection/README.md.
- [9] V. A. S. a. M. B. N. Alexander V. Savin, "Comparison of Facial Landmark Detection Methods," *IEEE*, 2021.
- [10] .Y.-J. Hong, "Facial Identity Verification Robust to Pose Variations and Low," Hoseo University,, 2022
- [11] N. Trapsida, "Smile detection for image moderation," 15 12 2020. [Online]. Available: <https://medium.com/decahlondigital/smile-detection-for-image-moderation-9b06b695f642>.
- [12] R. Neelamani and R. d. Queiroz, "JPEG compression history estimation for color images," *IEEE*.
- [13] N. C. R. M. & L. U. Ebner, "FACES—A database of facial expressions in young, middle-aged, and older women and men: Development and validation. Behavior Research Methods, 42, 351-362. doi:10.3758/BRM.42.1.351," 2010.

