

עבודת גמר הנדסת תכנה - הגנת סייבר

מסלול 14.50

שנת הלימודים תשע"ט

MotionSense



ניב קונרד, ת.ז. 322595984

מגישים:

רועי שחר, ת.ז. 322372251

מקיף עירוני ח' ראשון לציון (כיתה יב' 5)

בית הספר:

בת-ים (כיתה 13)

שם המוקד:

עודד ניר

שם הרש"צ:

אפריל, 2019

תאריך הגשה:

תוכן העניינים

1	
2 תוכן העניינים	
4 תיאור הפרויקט	1.
6 פתיחה, תיאור כללי והצעת פרויקט	2.
8 מסמך ייזום אפיון וארכיטקטורה	3.
9 ייזום + אפיון	3.1
10 מטרה	3.2
10 סקירה ספרותית	3.3
12 תפקוד	3.4
12 עיבוד	3.5
12 קלט\פלט	3.6
13 חיישנים\מסכי מגע	3.7
13 ממשק משתמש	3.8
13 קוד	3.9
13 תקשורת נתונים	3.10
13 אחסון	3.11
14 אבטחה	3.12
14 צורה ונראות	3.13
14 דרכי שימוש	3.14
14 סביבה	3.15
15 מבנה	3.16
15 מודולים	3.17
15 פיצ'רים	3.18
16 תהליכים	3.19
16 דרישות חומרה\תוכנה	3.20
16 בדיקות	3.21
17 מבט על	3.22
19 פירוט מודולים	3.23
20 טכנולוגיות עיקריות	3.24
21 התאמה לאפיון	3.25
22 מחלקות\מתודות	4.
23 מסמך מפרט התוכנה - Detailed Design	5.

24רקע	5.1
25פירוט מודולים תוכנה	5.2
32מסמך הבדיקות	6.
34הוראות שימוש – התקנה והפעלה	7.
34מדריך התקנה	7.1
37התאמת סביבת העבודה להרצה - קונפיגורציה	7.2
37 User Guide - למשתמש	7.3
47 Admin - למנהל	7.4
48משוב	8.
49קוד הפרויקט	9.
54התייחסות אישית	10.
55הצעות להמשך ושילוב הפרויקט בעולם האמיתי	.11
56הקדשה / תודה	12.
57רשימת מקורות (ביבליוגרפיה)	13.

1. תיאור הפרויקט

הפרויקט הינו תוכנה המאפשרת שליטה על תוכנות במחשב באמצעות תנועות ידיים. בתוכנה מוגדרים מראש כמה סוגים של מחוות ידיים, אשר המשתמש יכול להגדיר עבורם פעולה מתאימה, לכל אחת מהאפליקציות שהוא בוחר, או באופן כללי לשימוש במחשב. בכדי להשתמש בתוכנה, המשתמש מגדיר באמצעות הממשק הגרפי של התוכנה את ה-API (application programming interface) איתו הוא יעבוד. כלומר, הפעולה שתבצע באפליקציה הנבחרת באמצעות מחוות היד המוגדרת. לאחר שלב ההגדרה, ניתן להפעיל את התוכנה דרך כפתור ייעודי בממשק הגרפי ולהעבירה למצב בו היא רצה ברקע. במצב פעולה זה, Process שרץ ברקע של המחשב, מנתח את מחוות הידיים של המשתמש, כפי שהם נקלטים דרך מצלמת המחשב (Web Cam או כל מצלמה אחרת המחוברת למחשב), מנתח אותם באמצעות מודל neural network (עליו נפרט בהמשך המסמך), ומבצע בהתאם להגדרות של המשתמש את הפעולה המתאימה (לחיצה על המקש שהוגדר).

מטרת הפרויקט היא לשמש תחליף פשוט לשימוש יעיל יותר מעכבר ומקלדת לשליטה במחשב באמצעות מצלמת המחשב, ובנוסף לאפשר שליטה נוחה מרחוק במחשב, בניגוד לשליטה באמצעות עכבר אלחוטי שלעיתים אינה נוחה מבלי משטח מתאים.

קהל היעד של התוכנה הוא כל מי שמשתמש במחשב. רוב המשתמשים יוכלו להפיק תועלת מהשימוש בתוכנה, אך היא בעיקר תועיל לאנשים המתקשים בעבודה עם מקלדת ועכבר כמו אנשים מבוגרים. בנוסף התוכנה תתאים לסיטואציות שבהם יש צורך להפעיל את המחשב מבלי לגעת בו כמו בהצגת מצגות.

האתגר המרכזי אתו היינו צריכים להתמודד במהלך העבודה על הפרויקט ושאותו נדרשנו ללמוד הינו תחום ה-Deep Learning. במסגרת פיתוח התוכנה, אנחנו נדרשים לאמן מודל מתמטי וסטטיסטי, כך שהוא יעמוד בדרישות ובאילוצים הבאים:

- יזחה באחוזי דיוק גבוהים איזה תנועת יד נעשתה בסרטון שהועבר למודל.
- שהזמן שייקח למודל לזהות את התנועה יהיה יחסית קטן כך שלא יהיה Delay הנראה בעין גם כאשר המשתמש הוא אינו מפתח.
- שהתוכנה תרוץ על CPU ולא על GPU שאינו זמין בכל מחשב.

בעיה זו היא בעיה מסובכת מכיוון שרוב המודלים הקיימים ברשת מתרכזים ב-Accuracy על פני Latency. כלומר, רוב המודלים הקיימים באינטרנט לא מותאמים לבעיות של real time classification כמו הבעיה במקרה של התוכנה שלנו.

לכן, בנוסף ללמידה של עבודה עם Windows כדי לכתוב את ה-Process נדרשה למידה של תחום ה-Deep Learning בכדי לכתוב את המודל כך שיהיה לו אחוזי דיוק גבוהים בחיזוי התנועה. יתר על כן, נדרשנו גם להתעמק במידע של ה-Dataset עליו אימנו את המודל, וכן ללמוד על אופטימיזציה של מודל כדי לפתור את בעיית ה-Latency הקיימת במשימות של real time classification.

כדי לפתור את בעיה זו אימנו סוגים שונים של מודלים שמתאימים לבעיות שקשורות בסרטונים ובכלל ברצפים שקשורים בזמן (סרטון הוא רצף של תמונות לאורך זמן). לאחר שאימנו בערך עשרה מודלים שונים עם פרמטרים שונים (גדלים שונים של תמונות, אורכים שונים של רצפים) הגענו למסקנה כי המודל שאנחנו משתמשים בו היום, 3D Convolutional networks (סוג של מודל המותאם לעבודה עם תמונות לאורך זמן כאשר שני ממדים הם אורך ורוחב התמונה והמימד השלישי הוא זמן) שאחריהם long short term memory networks (גרסה מקדמת של Recurrent neural network - סוג של רשת נוירונים הקשורה בזכירה של פיצ'רים לאורך זמן ועבודה עם מידע שיש בין חלקיו קשר כמו למשל אזכורים בטקסט או סרטונים) ואחריהם Dense neural network (שכבות neural network רגילות שמטרתם בסופו של דבר להחזיר את תוצאת החיזוי של הרשת הכוללת) הוא סוג המודל האידיאלי הן מבחינת accuracy (מזהה שבע מחוות שונות ביותר מתשעים אחוז דיוק) והן מבחינת מהירות חיזוי (כמעט ולא ניתן לראות דיילי בתוכנה, התוכנה רצה חלק).

בעיה נוספת שנתקלנו בה אך לא הצלחנו לפתור אותה היא כתיבת החלק של החיזוי וביצוע הפעולה, החלק שבפרויקט כתוב כ-process בתור service. במקור כאשר תכננו את הפרויקט החלטנו שהחלק של החיזוי וביצוע הפעולה ייכתב ב-service מכיוון שזהו המימוש המתאים יותר לתהליך שרץ ברקע. לאחר זמן רב של עבודה על כתיבת ה-service הבנו שפעולות מתוך ה-service לא יכולות להשפיע על חלונות שונים מכיוון שמיקרוסופט הגבילה את הגישה של ה-service במערכת ההפעלה (service פועל ב-session 0 שהוא הרמה הכי נמוכה מבחינת הרשאות במערכת ההפעלה windows) ולכן לא נוכל לזהות את התוכנה שרצה ברקע או ללחוץ על מקש במקלדת מתוך ה-service. לאחר זמן רב של מחקר, הבנו שבכדי לפתור את בעיה זו יכולנו או לכתוב driver שיקשר בין ה-service למקלדת או לחפש פריצה במערכת ההפעלה windows שבעזרתה נוכל להעניק ל-service הרשאות יותר גבוהות אך מכיוון שהדגש בפרויקט שלנו הוא על Deep Learning ולא על מערכות הפעלה החלטנו להחליף את ה-service ב-process שרץ ברקע.

2. פתיחה, תיאור כללי והצעת פרויקט

כאשר חשבנו על רעיון לפרויקט, כיוונו לנושא שישלב תחום שמעניין אותנו ביחד עם תוצר סופי שבו נוכל להשתמש בחיי היום יום שלנו. החשיבה לא הייתה בכיוון של בעיה מסוימת אותה רצינו לפתור, או כיוון לקהל יעד ספציפי.

התחום של Deep Learning עניין אותנו מאוד וחיפשנו איך ניתן לשלב את התחום עם תוכנה שתעבוד בסביבת Windows. הנושא שבחרנו מתחום ה-Deep Learning, הוא שימוש במודל רשת נוירונים אשר מנתח את מחוות הידיים. התוכנה אמורה לעבוד על כל מחשב המריץ את מערכת ההפעלה של Windows, ולכן, לצורך הפיתוח, נדרשנו לעשות שימוש נרחב בספריות של Windows כולל ספריות התומכות בקלט ממצלמת המחשב.

כאשר החלטנו על הפרויקט אותו ביצענו לבסוף, ובחנו את מטרותיו וקהל היעד שלו, הבנו שהתוכנה שכתבנו יכולה להועיל לכלל המשתמשים במחשב ובמיוחד למשתמשים שמתקשים לעבוד עם המחשב מכיוון שהוא יקל על השימוש במחשב ויהפוך אותו לטבעי יותר (שליטה במחשב באמצעות חלקי הגוף).

הפרויקט מורכב משלושה חלקים עיקריים:

1. **ממשק גרפי** - חלק זה הוא החלק היחיד מבין שלושת החלקים שהמשתמש רואה. חלק זה נועד כדי לאפשר למשתמש להגדיר כיצד התוכנה תעבוד. כלומר באמצעות הממשק הגרפי יוכל המשתמש להחליט באיזה אפליקציות הוא רוצה שיתבצעו פעולות מסוימות (כמו לחיצה על מקשים במקלדת) ואיזו פעולה תבצע עבור כל מחווה. הממשק הגרפי כתוב ב - C# (WPF).
2. **תהליך (Process)** - חלק זה עובד ברקע, ולכן הוא חלק אשר המשתמש אינו מסוגל לראות אך הוא מסוגל לשלוט בהפעלה ובכיבוי שלו דרך הממשק הגרפי. לאחר שהמשתמש מפעיל באמצעות כפתור את התהליך דרך הממשק הגרפי, התהליך רץ ברקע ומשתמש במודל ה-Deep Learning עליו נפרט בהמשך בכדי לזהות את מחוות הידיים שהמשתמש מבצע ואשר נקלטות בעזרת מצלמת המחשב. לאחר שהתהליך קלט וזיהה את המחווה שהמשתמש ביצע, הוא בודק באיזו אפליקציה נמצא המשתמש כרגע, ובודק בהגדרות שקבע המשתמש את הפעולה המתאימה לאפליקציה ולמחווה שנעשתה. לבסוף אחרי שהתהליך מוצא את הפעולה שהוא צריך לבצע כמו למשל לחיצה על מקש במקלדת, הוא מבצע אותה בעזרת ספריות של Windows.
3. **מודל Deep Learning** - גם את חלק זה המשתמש לא מסוגל לראות והוא נמצא בשימוש בתוך החלק השני שתואר לעיל (התהליך). חלק זה נועד בכדי לקבל מהתהליך סרטון שהמצלמה של המחשב צילמה ולהחזיר כפלט את המחווה שהמשתמש ביצע בסרטון. המודל הוא מודל של רשת

נירונים המורכב משכבות של 3D Convolutional ומשכבות של LSTM (Long short-term) memory). שכבות אלו מתמחות בניתוח של תמונות ושל מידע אשר יש קשר של זמן ביניהם (סרטון הוא רצף של תמונות לאורך פרק זמן). באמצעות שכבות אלו ואימון של המודל עם דוגמאות רבות של סרטונים בהם אנשים מבצעים תנועות ידיים (סרטונים אלו סופקו על ידי ה-Dataset בו השתמשנו - jester של twentybn), המודל יהיה מסוגל לבסוף לזהות את התנועה שבוצעה בסרטון שהוא מקבל מהתהליך. כך התוכנה שלנו יודעת איזה תנועות המשתמש מבצע ולפיהם מתאימה את הפעולות שהיא מבצעת.

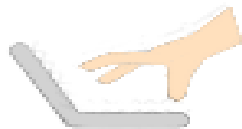
מודל ה- Deep Learning פותח בפיתון תוך שימוש בספריות TensorFlow ו-Keras לכתיבת המודל עצמו באמצעות פונקציות מובנות לבניית רשתות ניורונים ושימוש בספריות OpenCV ו-NumPy לעיבוד המידע בכדי שיתאים למודל.

תחום המחקר העיקרי בפרויקט הוא תכנון ותכנות מערכות מונחות ובאופן ספציפי Deep Learning על סרטונים. כלומר, למידת התיאוריה של שיטת ה-Deep Learning וכיצד לממש את השיטה הזאת בפועל ולשלב אותה בתוך פרויקט.

3. מסמך ייזום אפיון וארכיטקטורה

מסמך אפיון פרויקט

MotionSense



רועי שחר, ת.ז. 322372251

ניב קונרד, ת.ז. 322595984

3.1 ייזום + אפיון

3.1.1 תיאור כללי

הפרויקט הינו תוכנה המאפשרת שליטה על תוכנות במחשב באמצעות תנועות ידיים. בתוכנה יהיו מספר סטים מוגדרים מראש של תנועות שהמשתמש יוכל להגדיר עבורם מקש מתאים בכל אחת מהאפליקציות בהם התוכנה תעבוד.

3.1.2 מטרת הפרויקט

מטרת הפרויקט היא לאפשר עבודה יעילה ומהירה יותר על המחשב תוך מתן הרגשה טבעית יותר לעבודה על ידי שליטה מרחוק באמצעות מצלמת המחשב ותנועות ידיים. כחשבנו על הפרויקט ניסינו למצוא פרויקט שישלב תחום שמעניין אותנו ביחד עם תוצר סופי שבו נוכל להשתמש בחיי היום יום שלנו.

3.1.3 סוגי משתמשים / קהל היעד של היישום

כל האנשים שמשתמשים במחשב יכולים להשתמש בתוכנה זאת ולהפיק תועלת ממנה אך היא בעיקר תועיל לאנשים המתקשים בעבודה עם מקלדת ועכבר כמו אנשים מבוגרים ואנשים שצריכים להשתמש במחשב מבלי לגעת בו כמו בהצגת מצגות. ידע מקדים בתחום כלשהו אינו נדרש. התוכנה תאפשר עבודה נוחה יותר עם יישומי המחשב, כמו PowerPoint, יישומי מדיה ועוד בעזרת השימוש בתנועות הידיים.

3.1.4 ידע, סיכונים ושאלות

החשש העיקרי הוא שנתקע באימון הרשת הניורונים או שהרשת לא תתפקד כפי שרצינו ובסוף נגיע עם תוצר לא גמור. אנחנו נדרשים ללמוד כיצד לבנות רשת ולאמן אותה כדי לבצע את העיבוד של תזוזות הידיים עליהם מתבססת פעולת התוכנה. החלק הכי מסובך וממושך יהיה החלק העיקרי של הפרויקט שאותו לא ידענו מראש כלומר עבודה עם רשתות ניורונים ובכלל עיבוד תמונה באמצעות Deep Learning. אין נושאים לגביהם אנחנו מתלבטים.

3.1.5 פתרונות קיימים

כן ישנם עזרי חומרה כמו עכבר או מסך מגע שמבצעים מה שהפרויקט שלנו מבצע. יש גם בחברות משחקים פרויקטים דומים כמו קינקט באקס בוקס של מיקרוסופט. בעבר הייתה קיימת חברה שעשתה דבר דומה לפרויקט שלנו אך היא נסגרה כי לטענתם לפרויקט אין נתח שוק. באינטרנט ישנם כל מיני תוכנות שעושים דבר דומה לשלנו באמצעות שימוש באמצעים שונים כמו שניתן לראות בסקירה הספרותית.

3.2 מטרה

מטרת הפרויקט היא לשמש תחליף פשוט לשימוש ויעיל יותר ממקלדת לשליטה במחשב ובנוסף לאפשר שליטה מרוחקת ונוחה במחשב בניגוד לשליטה באמצעות עכבר אלחוטי שלעיתים אינה נוחה מבלי משטח מתאים. הפרויקט הזה הוא הדרך הכי טובה לבצע דבר זה כי אין דבר יותר יעיל ופשוט משימוש בידיים שלנו כפי שהטלפונים החכמים הוכיחו. לכן להשתמש בידיים באופן טבעי כמו שאנחנו עושים ביום יום ימקסם את היעילות והפשטות של השימוש ויגשים את המטרה של הפרויקט בצורה הטובה ביותר.

3.3 סקירה ספרותית

- <https://medium.com/@muehler.v/simple-hand-gesture-recognition-using-opencv-and-javascript-eb3d6ced28a0>
זיהוי ידיים עם opencv
- <http://article.sapub.org/10.5923.j.ajis.20170703.11.html>
זיהוי ידיים באמצעות מצלמת מחשב
- <https://www.youtube.com/watch?reload=9&v=oH0ZkfFoeYU>
בפייתון זיהוי מחוות ידיים באמצעות מצלמת מחשב opencv
- <https://www.youtube.com/watch?v=TwYUm7i61JM>
שליטה בעכבר באמצעות מחוות ידיים בעזרת opencv
- <https://www.neurotechnology.com/npointer.html>
אתר לחברה שמתמחה בנושא של הפרויקט

- <https://github.com/AdamCDunlap/webcam-gesture>
קישור לאיחסון גיטהאב שיש בו זיהוי מחוות ידיים בעזרת מצלמת מחשב
- <http://www.techplusme.com/software/control-pc-hand-gesture>
אתר המסביר על שני תוכנות שעושות דבר דומה לפרויקט שלנו
- <https://www.kaggle.com/gti-upm/leapgestrecog>
קישור למאגר מידע לזיהוי ידיים שמכיל עשרים אלף תמונות
- <https://www.kaggle.com/sujay626/gesture-datase>
מאגר מידע לזיהוי מחוות ידיים
- <https://www.kaggle.com/ashish8898/hand-gestures>
קישור לפרויקט של זיהוי ידיים
- <https://www.youtube.com/watch?v=VIPnqL5osYQ>
קישור לסרטון שמציג פרויקט דומה למה שאנו עושים
- <http://sun.aei.polsl.pl/~mkawulok/gestures/>
Database for hand gesture recognition
- <http://www.nlpr.ia.ac.cn/iva/yfzhang/datasets/egogesture.html>
מאגר נתונים של זיהוי ידיים
- <https://medium.com/twentybn/gesture-recognition-using-end-to-end-learning-from-a-large-video-database-2ecbf4659ff>
כתבה על פרויקט בנושא
- <https://www.kaggle.com/benenharrington/hand-gesture-recognition-database-with-cnn/notebook>
מאגר מידע של תנועות יד בתמונות infra red
- <https://www.kaggle.com/benenharrington/hand-gesture-recognition-database-with-cnn/notebook>
מאגר נתונים מחוות ידיים
- <https://www.kaggle.com/gti-upm/leaphandgestuav/home>
Hand gesture recognition dataset is presented, composed by a set of near infrared images and skeletal information acquired by the Leap Motion sensor.

- https://www.kaggle.com/destin369y/interactivehandgesture1/version/4#HandGesture1-6_ddd.zip
database of hand gesture recognition
- <https://zenodo.org/record/1320922#.W9cHbZMzbIV>
מאגר מידע המכיל בו מספר מחוות ידיים
- <https://www.kaggle.com/gti-upm/multimodhandgestrec>
multi modal dataset for hand gesture recognition
- <https://www.kaggle.com/gti-upm/depthgestrecog>
depth based gesture recognition database

3.4 תפקוד

הפרויקט עוקב אחרי תנועות הידיים של המשתמש בהתאם ל-API שיקבע על ידי המשתמש (להסבר נרחב יותר ראה מדריך למשתמש), ינתח את תנועות המשתמש, ויבצע פעולות במחשב כמו המקבילות לשימוש במקלדת. בנוסף באפליקציות מסוימות תתאפשר שליטה מותאמת כדי לגרום לעבודה יותר טובה ונוחה עם אפליקציות אלו.

3.5 עיבוד

המערכת תזדקק לכוח עיבוד רק במהלך שלב האימון של הרשת. לאחר מכן היא תשתמש במשאבי המחשב הקיימים מבלי להשתמש בכוח עיבוד נוסף.

3.6 קלט\פלט

קלט התוכנה הינו תנועות הידיים של המשתמש כפי שנקלטים דרך מצלמת המחשב. פלט התוכנה הוא פעולות במחשב המותאמות לתנועות שהמשתמש ביצע, על פי התצורה שנקבעה על ידו. הצגת הנתונים מתבצעת דרך המסך שכן התוכנה מבצעת פעולות במחשב באופן דומה לעכבר ומקלדת.

3.7 חיישנים\מסכי מגע

התוכנה "חשה" את הסביבה דרך מצלמת המחשב.

3.8 ממשק משתמש

המשתמש מתקשר עם התוכנה באמצעות הידיים והממשק הגרפי שיבוא עם הפרויקט לשליטה מותאמת.

המשתמש צריך מצלמה (מובנית או חיצונית) כדי לתקשר עם התוכנה. לא ניתן לשלוט התוכנה על ידי מכשירים חיצוניים.

3.9 קוד

במהלך הפיתוח נשתמש ב-Jupyter notebook אך התוצר הסופי יהיה ב-PyCharm. הפרויקט ייכתב ב-python 3.6 וב-C#. נשתמש ב-TensorFlow בשביל עיבוד התמונה ובמודלים שונים כדי לבצע פעולת במחשב. יתכן ונשתמש גם ב-OpenCV בשביל ניתוח ראשוני של התמונות. לא נצטרך להשתמש ב-web services.

3.10 תקשורת נתונים

התוכנה ושלב פיתוח התוכנה לא דורשים תקשורת נתונים.

3.11 אחסון

הפרויקט שלנו לא ידרוש הרבה מידע מהמשתמש, עליו לשמור רק את התוכנה (הקוד והספריות הנלוות אליו), מידע זה יאוחסן לוקאלית. בשלב הפיתוח המידע יאוחסן ככל הנראה דרך הענן והוא יכלול בעיקר מאגר נתונים גדול לאימון הרשת וקוד.

3.12 אבטחה

התוכנה לא דורשת אבטחה, שכן היא מבצעת לוקלית פקודות שהמשתמש בחר בדומה לעכבר ומקלדת.

3.13 צורה ונראות

שליטה במחשב באמצעות הידיים.

התוכנה תכיל תפריט שבו מוגדרות מספר סטים של תנועות.

המשתמש יוכל להגדיר מקש מתאים עבור תנועה מסוימת ולבחור אם זה יהיה באופן כללי או באופן ממוקד לתוכנה לבחירתו.

3.14 דרכי שימוש

כדי לגרום לתוכנה לעבוד המשתמש צריך להפעיל את זיהוי הידיים בתוך התוכנה (כפתור הפעלה). לאחר שמשתמש מצבע מחווה עם ידיו הפרויקט מבצע את הפעולה המתאימה. המשתמש רואה על המסך מה קורה בעקבות התנועות שלו (המקש שהגדיר עבור תנועה זו). הפרויקט מבצע מספר פעולות כמספר המקשים שהגדיר המשתמש והוא יכול לתקשר רק עם משתמש אחד.

סדר הפעולות הוא: ראשית המשתמש מבצע מחווה באמצעות היד. לאחר מכן המערכת מעבדת את תנועת היד הזאת ובהתאם ל-api מבצע את הפעולה המתאימה לתנועת היד הזאת. במידה והרצף לא נכון או שהידיים של המשתמש אינן נמצאות בטווח הנראות של המצלמה של המחשב הפרויקט לא יעבוד כמו שצריך. הפרויקט יוכל לשמש כתחליף במחשבי מגע למסך בזמן שהידיים של המשתמש מלוכלכות או רטובות.

3.15 סביבה

הפרויקט הינו תוכנת מחשב אשר תעבוד על כל סוג מחשב.

התוכנה מבצעת את זיהוי הידיים על ידי מצלמה ולכן המחשב צריך להיות עם מצלמה מובנית או מצלמה חיצונית מחוברת בכדי שהתוכנה תעבוד.

לתוכנה יהיו תנאי עבודה מינימליים מבחינת איכות הצילום והתאורה, שיקבעו בהמשך

3.16 מבנה

המבנה של הפרויקט יהיה כך:
קבלת רצף תמונות מהמשתמש -> ניתוח ראשוני של רצף התמונות (אפשרי) -> ניתוח שני של רצף התמונות באמצעות הרשת -> בדיקה האם הרצף תואם לרצף `apic` של התוכנה -> אם כן התוכנה מבצעת את הפעולה המתאימה לרצף, אם לא התוכנה לא מבצעת דבר.
המעבר בין כל הרכיבים יתבצע דרך הזיכרון, מלבד המעבר בין הרכיב ששומר תמונות לצורך אימון לרכיב שמאמן את הרשת שיעשה דרך קבצים.

3.17 מודולים

המודולים העיקריים של התוכנית יהיו:

1. קבלת התמונה
 2. סידור התמונה (כדי שתתאים לתמונות שאימנו את הרשת בעזרתם)
 3. ניתוח התמונה
 4. בדיקת הרצף בהתאם ל-API
 5. ביצוע הפעולה המתאימה
 6. מודל שיהיה מחוץ לפרויקט והמטרה שלו תהיה אימון הרשת
 7. מודל שיבדוק את רשת
 8. שמירת העדפות המשתמש
- הפרויקט יכלול מבני נתונים ביניהם הגרף שיופק מהרשת ואחסון של הרצפים כלומר ה-API שלפיו תעבוד התוכנה.
אין צורך ברכיב תקשורת, אותנטיקציה, רכיב עיבוד ומערכת של שרת ולקוח בתוצר הסופי.

3.18 פיצ'רים

הפיצ'רים המרכזיים של התוכנית הם:

1. שליטה במקלדת המחשב באמצעות תנועות ידיים - הגדרת מחוות עבור מקשים במקלדת.
2. הגדרת מחוות ידיים מותאמות עבור אפליקציות מסוימות או מערכת ההפעלה.

דוגמאות:

- A. שליטה על העברת שקופיות בפאורפוינט באמצעות מחוות ידיים.
B. שליטה על הסאונד במחשב באמצעות מחוות ידיים.
הממשק הכללי מול המערכת יאפשר עבודה עם קיצורים הקיימים כבר ב-windows באמצעות מחוות ידיים.

3.19 תהליכים

התהליכים העיקריים הם:

1. אימון הרשת לזיהוי תנועת ומיקום היד.
 2. שימוש בפלט הרשת ובתמונות לצורך קביעת הפעולה שהתוכנה תצטרך לבצע.
 3. ביצוע הפעולה וקבלת פידבק ממנה.
- הפונקציונליות שלה היא שליטה על הכפתורים באפליקציות שונות באמצעות תוכנות ידיים השימושים העיקריים של התוכנה הם להקל על השימוש במחשב או לשמש תחליף נוח למקלדת במצבים בהם המקלדת או לחלופין מסך המגע לא נוחים לשימוש (פירטנו מצבים כאלו במהלך המסמך). המשתמש חווה רק את התהליך השלישי ברשימה שמניתי כלומר רק התהליך הקשור בפלט שהמחשב נותן למשתמש.

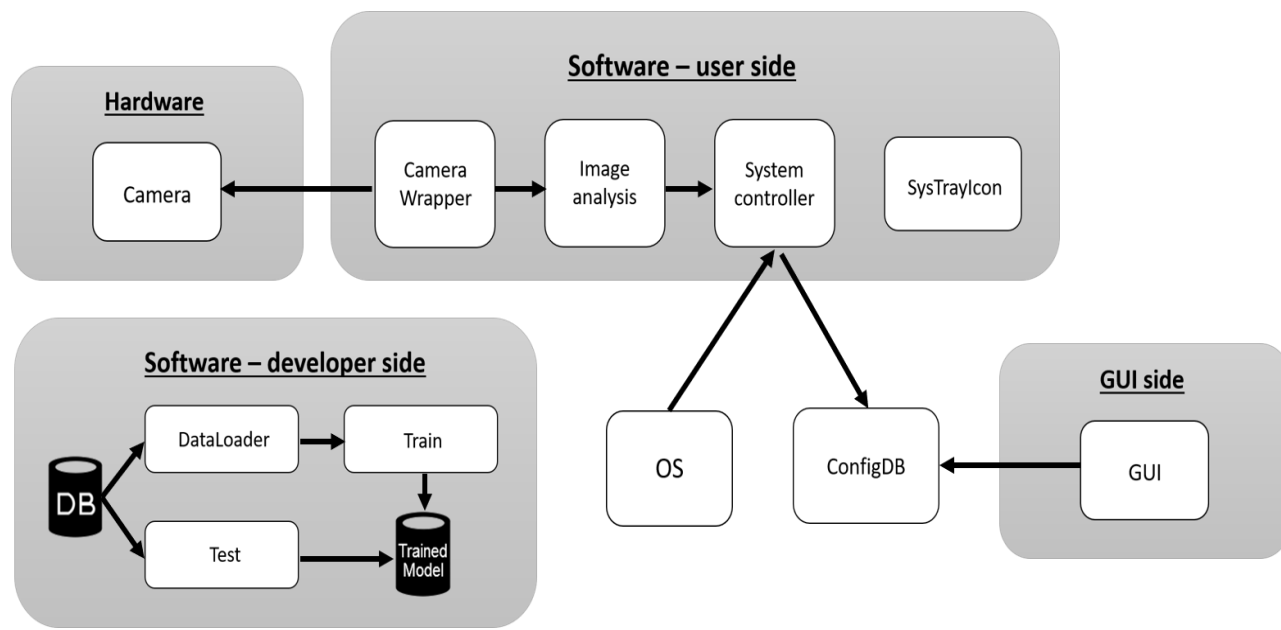
3.20 דרישות חומרה\תוכנה

המחשב שעליו רצה התוכנה יצטרך מצלמה מובנית או מצלמה חיצונית מחוברת.

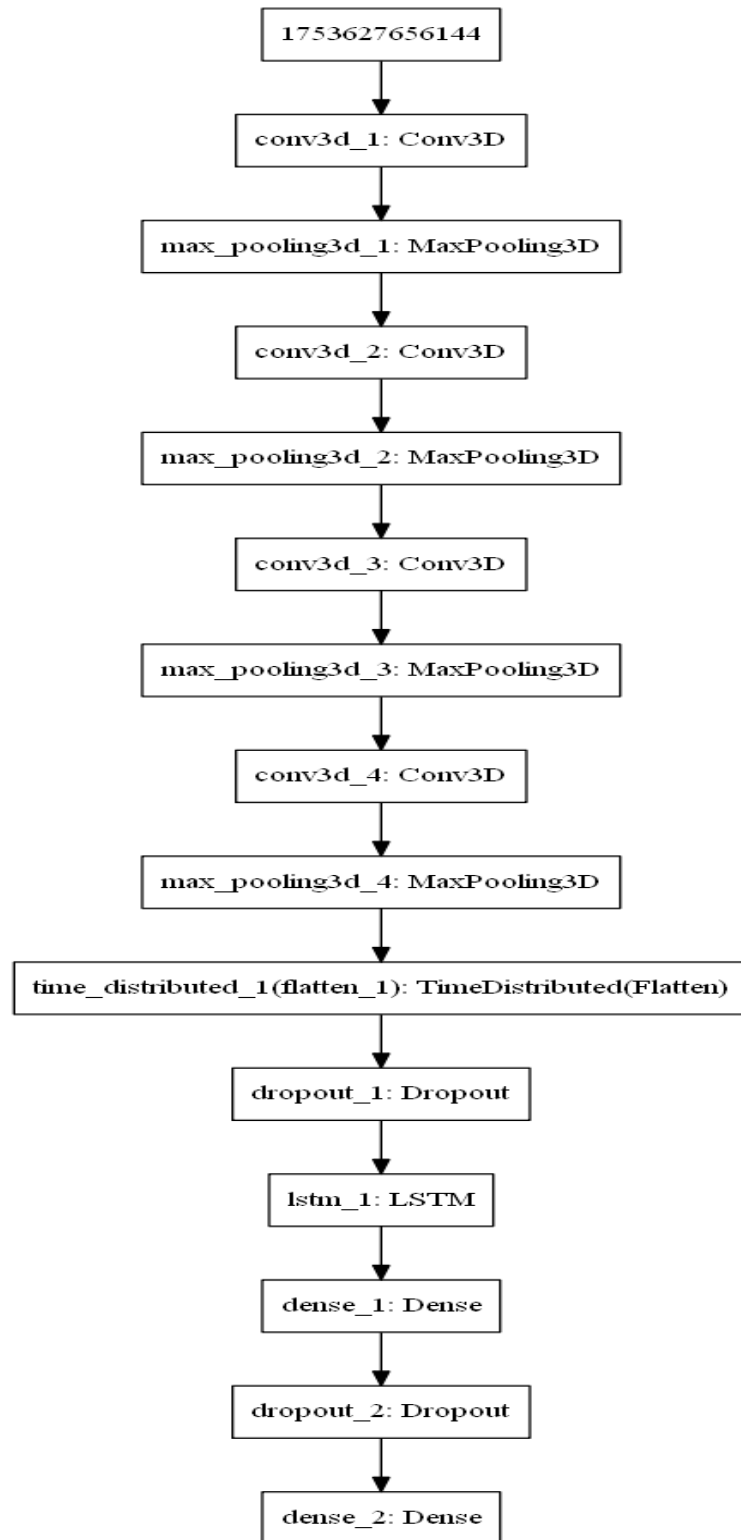
3.21 בדיקות

בדיקת הגדרת מחוות ידיים מותאמות עבור אפליקציות מסוימות או מערכת ההפעלה – אנחנו נבדוק האם לכל אפליקציה עובד ה-API הספציפי והמותאם לה כמו שצריך מבלי לגרום לקריסה של האפליקציה או של התוכנה.
בסוף הפיתוח נבדוק אם התוכנה מבצעת את הדבר הרצוי לפי כל פיצ'ר.
יהיו בדיקות אוטומטיות באמצעות סקריפט, נשמור מצבים מוכרים ואת הפלטים שצריך לצאת ובאמצעותם נבדוק האם המערכת תעבוד כפי שצריך. הסקריפט משתמש באופן אוטומטי בקלטים של תמונות מוכנות מראש שנמצאות ב-database.

3.22 מבט על



תרשים המודל:



3.23 פירוט מודולים

3.23.1 תוכנה

צד מפתח:

- **Data Loader** - רכיב תוכנה האחראי על התאמת מבנה הנתונים **Jester**, איתו המודלים שפיתחנו עובדים למודלים, כלומר טעינה של תמונות ממבנה הנתונים והרכבת רצפים בגדלים המתאימים למודלים.
- **Train** – רכיב תוכנה האחראי על אימון הנתונים בעזרת מאגר המידע של התמונות ורשת נוירונים ועל מעקב אחרי אימון הנתונים ואחוזי ההצלחה, המודול עובד באמצעות **Keras**.
- **Test** - רכיב תוכנה האחראי על בדיקת המודל בעזרת תמונות מקבוצת הבדיקה.
- **Trained model** - המודל המופק מהמרכיבים **train** ו-**test** בעזרתו המערכת תנתח את התמונות בצד הלקוח.
- **DB** - מאגר התמונות בו משתמשים לאימון המודל ולבדיקת המודל.

צד Process:

- **Camera Wrapper** – רכיב תצלום, אחראי על תקשורת נכונה מול המצלמה (מצלמת המחשב או מצלמה המחוברת למחשב).
- **Image Analysis** – רכיב תוכנה האחראי על ניתוח התמונה וסיווג המחווה המתקבלת מהמצלמה בעזרת המודל המאומן.
- **SysTrayIcon** – מאפשרת שליטה בממשק הגרפי דרך ה-**icon tray** של windows.
- **System controller** – רכיב תוכנה שמתקשר עם מערכת ההפעלה באמצעות המידע המועבר מהממשק הגרפי דרך **ConfigDB** ומהתוצאות של ניתוח התמונה ב- **image analysis**.
- **OS** – מערכת ההפעלה אליה שולח **system controller** את הפקודות שמדמות לדוגמא לחיצות מקלדת או עכבר, שינוי ווליום או שליטה בפאורפוינט.
- **ConfigDB** – קובץ **DB** שיימצא במחשב הלקוח ובו יישמרו הגדרות הלקוח.
- **Apps** – קובץ **DB** שיימצא במחשב הלקוח ובו יישמרו שמות ומיקומי האפליקציות שהתוכנה מצאה במחשב וזמינים להגדרה בממשק הגרפי.

צד GUI

- **GUI** – ממשק גרפי למשתמש שכולל בתוכו אפשרות להגדרת מחוות ידיים עבור מקשים במקלדת ותוכנות מתואמות (לדוגמא – **Power Point**).

3.23.2 חומרה

Camera - רכיב חומרה, מצלמת המחשב או מצלמה המחוברת למחשב ומצלמת וידאו של המשתמש בזמן אמת.

3.24 טכנולוגיות עיקריות

שפות פיתוח:

- Python - באמצעות Keras מעל TensorFlow
- C#

סביבות עבודה:

- PyCharm
- Jupyter notebook
- Notepad++

שירותים חיצוניים: שירותי ענן.

הטכנולוגיות העיקריות בהם השתמשנו הם: Deep Learning ועיבוד תמונה.

החלטנו להשתמש דווקא בטכנולוגיות אלו מכיוון שהם הנפוצות בתחום שבחרנו ומכיוון שמעניין אותנו ללמוד בהם ולהשתמש בהם.

פסלנו שימוש ב-TensorFlow בשפות אחרות ובשימוש בספריות יותר מסובכות מכיוון שאנחנו מאמינים שבפיתון הכתיבה תהיה יותר פשוטה ובגלל שספריות אחרות דורשות יותר הבנה מתמטית ממה שיש לנו.

בנוסף על הטכנולוגיות שבחרנו יש הרבה הסברים ודוגמאות באינטרנט כי הן מאוד פופולריות.

3.25 התאמה לאפיון

רכיבים רלוונטיים	פיצ'ר
Camera	קבלת התמונה
Image Analysis	ניתוח התמונה
System controller	בדיקת הרצף בהתאם ל-API
System controller->os	ביצוע הפעולה המתאימה
Train	אימון הרשת
Test	בדיקת הרשת
Gui -> ConfigDB	שמירת העדפות משתמש

4. מחלקות\מתודות

צד פיתוח מודל

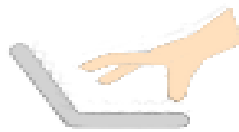
1. מחלקת DataLoader - המחלקה האחראית על טעינת מבנה הנתונים בקוד ויצירת הרצפים (הסרטונים) מהתמונות הבודדות במבנה הנתונים. המחלקה מכילה בתוכה פעולה בונה, פעולה שבונה רצפים, פעולה שמסדרת את התמונות ברצפים כך שיתאימו למודל, פעולה שמסדרת את אורך הרצפים ופעולה שכל פעם מחזירה למודל את הרצף הבא בתור (sequence generator) שיתאים לפעולה המובנית ב-Keras לאימון מודלים מסוג זה הנקראת (fit generator). בנוסף, קיימות בה פעולות האחראיות על עיבוד קבצי ה-csv (מכילים בתוכם מידע בנוגע לסוגי ה-labels ומידע בנוגע לחלוקת הסרטונים ל- train, test, validation ולאיזה תנועה מתבצעת בהם, מה ה-label של כל סרטון) וטעינת הסרטונים ממבנה נתונים לפי המידע בקבצי ה-csv.

צד משתמש

1. מחלקת CameraWrapper - המחלקה עוטפת את מצלמת המחשב, לוקחת ממנה תמונות. המחלקה מכילה בתוכה פעולה בונה ופעולה שאחראית על לקיחת תמונה מהמצלמה.
2. מחלקת ImageAnalysis - המחלקה שומרת רצפים של תמונות ומנתחת איזה מחווה נעשתה בהם באמצעות המודל המאומן. המחלקה מכילה בתוכה פעולה בונה, פעולה שמכינה את התמונות לניתוח, פעולה שמוסיפה את התמונות לרצף, פעולה שמבצעת חיזוי על רצפי התמונות ופעולה שמציגה את תוצאות החיזוי באמצעות גרפיקה.
3. מחלקת SystemController - המחלקה מתקשרת עם מערכת ההפעלה באמצעות המידע המועבר מהממשק הגרפי ומהתוצאות של ניתוח רצף התמונות ומבצעת בהתאם לאפליקציה בה המשתמש נמצא את הפעולות הדרושות במערכת הפעלה כמו למשל לחיצה על מקש במקלדת. המחלקה מכילה בתוכה פעולה בונה, פעולה לקריאת מבנה הנתונים בו מאוחסנים הגדרות המשתמש ופעולה בה המחלקה משתמשת במידע שקיבלה ומתקשרת עם מערכת ההפעלה כדי לבצע את הפעולה הדרושה כפי שתואר לעיל.
4. מחלקת SysTrayIcon - המחלקה יוצרת אייקון של התוכנה ב-icon tray ומאפשרת עבודה עם ה-gui דרכו כמו פתיחה או סגירה של ה-gui (להסבר נרחב יותר ראה מדריך למשתמש). המחלקה מכילה בתוכה פעולה בונה ופעולות שנועדו להצגה ולעדכון של התפריט.

מסמך מפרט התוכנה

MotionSense



רועי שחר, ת.ז. 322372251

ניב קונרד, ת.ז. 322595984

5.1 רקע

הפרויקט מורכב משלושה חלקים עיקריים:

ממשק גרפי שנועד לשמירת העדפות המשתמש, process שרץ ברקע של המחשב ואחראי על לקיחה של תמונות מהמצלמה, בדיקה באיזה אפליקציה נמצא המשתמש ושליחת הפקודה המתאימה למערכת ההפעלה בהתאם להגדרות מהממשק הגרפי, לאפליקציה שבה המשתמש נמצא ולמחווה אשר המשתמש עשה. החלק השלישי הוא החלק שמבצע את הבדיקה איזה מחווה המשתמש עשה בתמונות שה - process לקח, חלק זה הוא מודל רשת נוירונים (מודל deep learning) שמאומן לבדוק איזה מחווה נעשתה על ידי המשתמש מבין 27 מחוות הידועות למודל. הפרויקט נכתב בשתי שפות שונות, שפת python לחלק האחורי (backend) של הפרויקט כלומר ל - process ולמודל ושפת #c לחלק הקדמי(frontend) של הפרויקט כלומר לממשק הגרפי. בחרנו בשפות הללו מכיוון שהתכנות ב- deep learning יותר פשוט ונגיש בשפת פיתון ונהפך לנורמה בשוק העבודה כך שנקבל מכתובה של deep learning בפיתון ערך מוסף. מאותו שיקול גם החלק השני של ה- backend נכתב בפיתון (ה - process). את החלק של הממשק הגרפי כתבנו בשפת #c מכיוון שאנחנו מכירים את השפה הזאת טוב מהלימודים בבית הספר ובגלל שהאופציות לכתיבת ממשק גרפי בה יותר טובות מאשר האופציות שבפיתון. הסביבות בהם כתבנו את הפרויקט הם ++notepad ו- PyCharm עבור החלק שנכתב בפיתון ו- visual studio עבור החלק שנכתב ב- #c. הפרויקט נכתב עבור מערכת הפעלה windows מכיוון שרצינו שהוא יהיה נגיש לכמה שיותר אנשים. הפרויקט מורכב משלושה מודלים חיצוניים:

1. ממשק גרפי (GUI) - ממשק גרפי כתוב ב- #c לשמירת העדפות המשתמש בנוגע לאיזה פעולה לבצע עבור כל תנועה בכל אפליקציה שהמשתמש יבחר.
2. בסיס נתונים (Database) - הפרויקט מורכב ממספר בסיסי נתונים לשמירת מידע כמו הגדרות המשתמש שנקבעו בממשק הגרפי או מיקומי אפליקציות שונות על המחשב של המשתמש, חלקן כתובים בפורמט json וחלקם בפורמט טקסט רגיל.
3. מודל מאומן (deep learning trained model) - הפרויקט משתמש במודל שאימנו בכדי לנתח ולזהות איזו תנועה המשתמש ביצע בסרטון.

5.2 פירוט מודולים תוכנה

5.2.1 צד מפתח

Data Loader - רכיב תוכנה האחראי על התאמת מבנה הנתונים **jester**, איתו המודלים שפיתחנו עובדים למודלים, כלומר טעינה של תמונות ממבנה הנתונים והרכבת רצפים בגדלים המתאימים למודלים.

UI - רכיב ממשק למשתמש בצד המפתח המאפשר מעקב אחרי אימון הנתונים ואחוזי ההצלחה. המודול יהיה סקריפט שאנחנו נפתח שיעבוד עם התוצאות שה Keras מחזיר בהתאם לחיזוי שלנו של התוצאות.

Train – רכיב תוכנה האחראי על אימון הנתונים בעזרת מאגר המידע של התמונות ורשת נוירונים.

Test - רכיב תוכנה האחראי על בדיקת המודל בעזרת תמונות מקבוצת הבדיקה.

Trained model - המודל המופק מהמרכיבים `test` ו `train` בעזרתו המערכת תנתח את התמונות בצד הלקוח.

DB - מאגר התמונות בו משתמשים לאימון המודל ולבדיקת המודל.

5.2.2 צד process

Camera Wrapper – רכיב תצלום, אחראי על תקשורת נכונה מול המצלמה (מצלמת המחשב או מצלמה המחוברת למחשב).

Image Analysis – רכיב תוכנה האחראי על ניתוח התמונה וסיווג המחוזה המתקבלת מהמצלמה בעזרת המודל המאומן.

GUI – ממשק גרפי למשתמש שכולל בתוכו אפשרות להגדרת מחוות ידיים עבור מקשים במקלדת ותוכנות מתואמות (לדוגמא - PowerPoint).

SysTrayIcon – מאפשרת שליטה בממשק הגרפי דרך ה- `icon tray` של windows.

System controller – רכיב תוכנה שמתקשר עם מערכת ההפעלה באמצעות המידע המועבר מהממשק הגרפי ומהתוצאות של ניתוח התמונה.

OS – מערכת ההפעלה אליה שולח `system controller` את הפקודות שמדמות לדוגמא לחיצות מקלדת או עכבר, שינוי ווליום או שליטה בפאורפוינט.

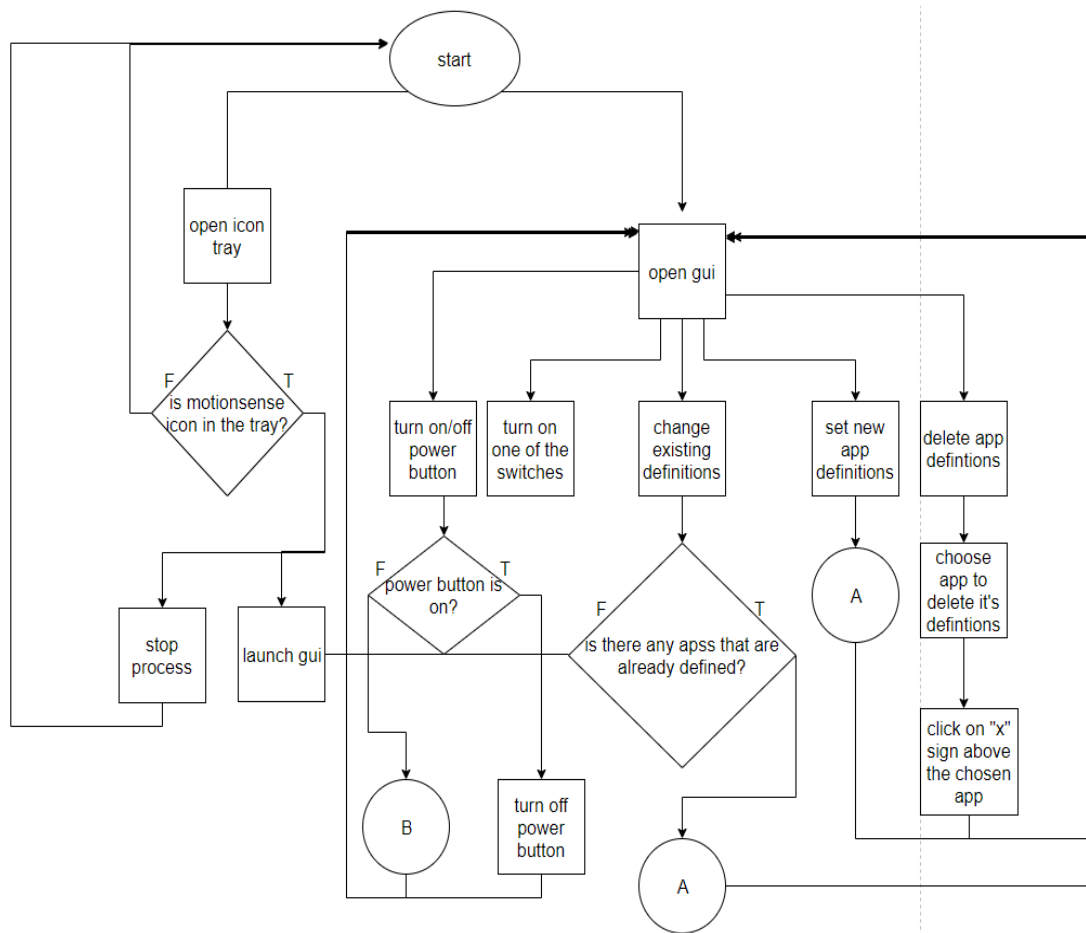
ConfigDB – קובץ DB שיימצא במחשב הלקוח ובו יישמרו הגדרות הלקוח.

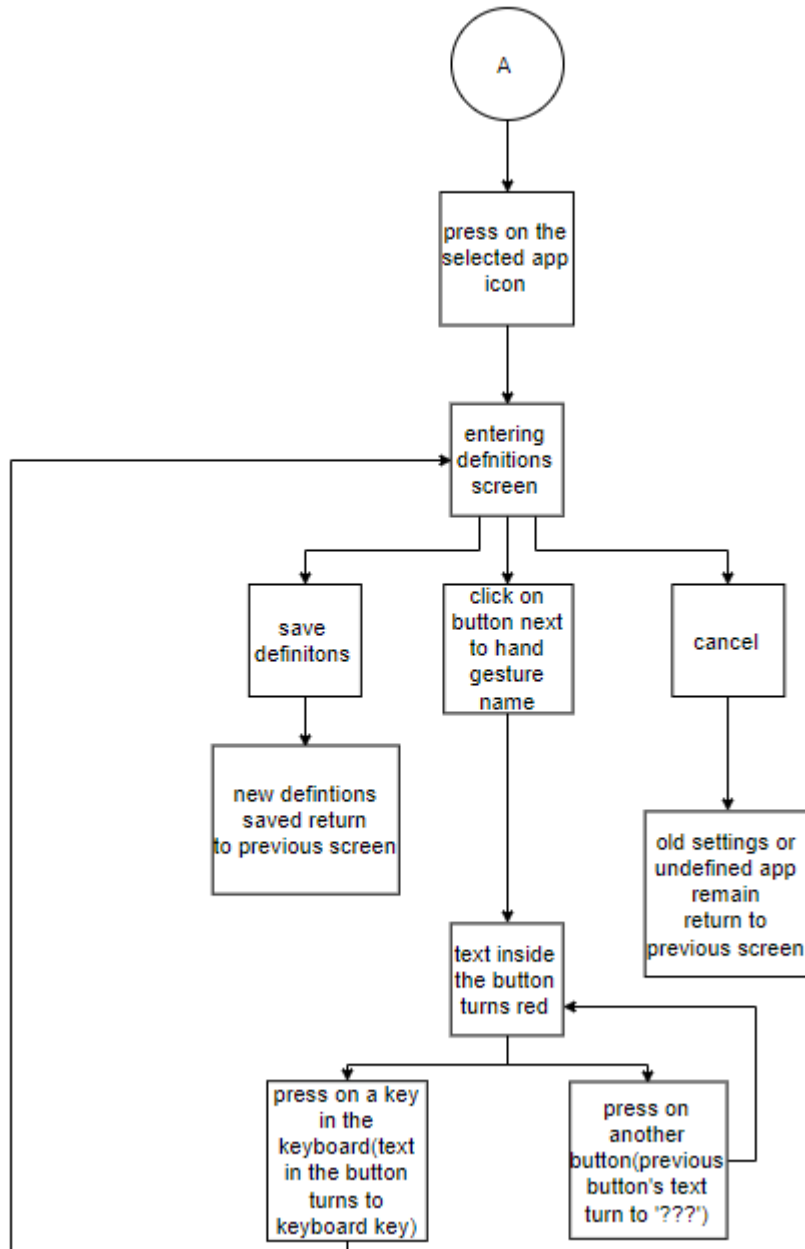
Apps – קובץ DB שיימצא במחשב הלקוח ובו יישמרו שמות ומיקומי האפליקציות שהתוכנה מצאה במחשב וזמינים להגדרה בממשק הגרפי.

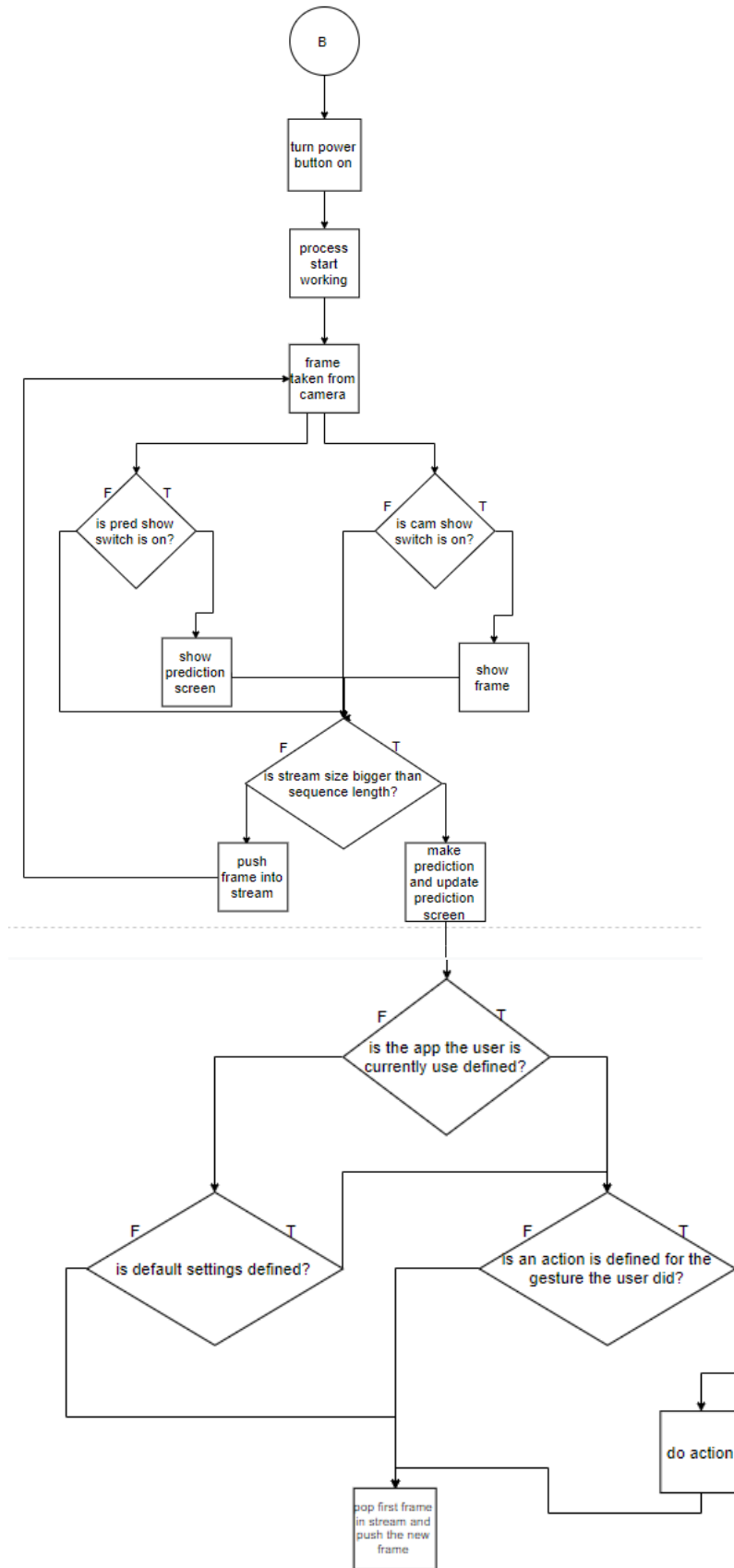
5.2.3 חומרה

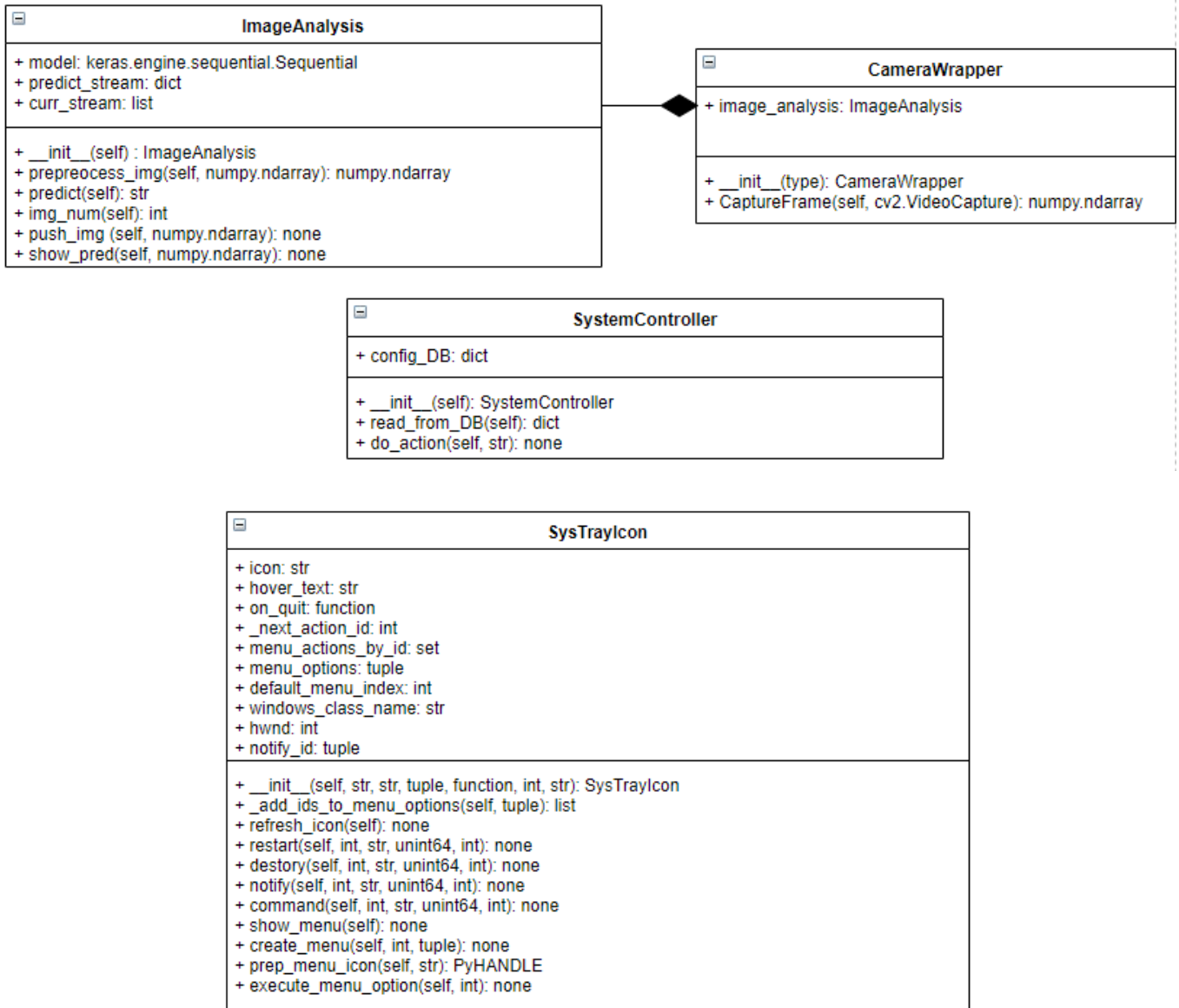
Camera - רכיב חומרה, מצלמת המחשב או מצלמה המחוברת למחשב ומצלמת וידאו של המשתמש בזמן אמת.

5.2.4 מבנה הפעולה









1. המחלקה CameraWrapper - המחלקה לוקחת תמונות מהמצלמה באמצעות הפונקציה CaptureFrame ומחזירה אותם לפונקציה הייצונית StartPreview (הפונקציה לא נמצאת במחלקה).
2. המחלקה ImageAnalysis - המחלקה מקבלת מהפונקציה הייצונית StartPreview תמונות אותן היא מוסיפה ל- curr stream באמצעות הפונקציות push_img ו- preprocess_img (פונקציה שמטרתה לסדר את המערך המייצג את התמונה למען חיזוי) ולאחר שאורכו של curr stream הוא כאורך הרצף שנקבע מראש מופעלת הפונקציה predict

באמצעות model ומוחזר מחרוזת שאומרת איזה מחווה המשתמש ביצע ברצף התמונות שעליו נעשה החיזוי.

3. המחלקה SystemController - המחלקה קוראת מה- Database שנקרא ConfigDB את ההגדרות שהמשתמש הגדיר ב- gui (אפליקציה: מחוות יד: מקש לביצוע) , היא מקבלת (בפונקציה do_action) מהפונקציה החיצונית StartPreview מחרוזת שערכה הוא התנועה שהמשתמש ביצע, היא בודקת באמצעות הספריות של windows באיזה אפליקציה המשתמש נמצא ואז באמצעות המידע שהיא קראה מה- Database היא מחליטה איזה פעולה לבצע (על איזה מקש לדמות לחיצה) ומבצעת זאת באמצעות הספריות של windows.
4. המחלקה SysTrayIcon – המחלקה יוצרת את האייקון של התוכנה ב- icon tray ומאפשרת שליטה בממשק הגרפי באמצעותו. באמצעות הספריות win32api, win32con, subprocess, ו- win32gui_struct המחלקה מתקשרת עם מערכת ההפעלה, דואגת שיהיה אייקון ב- tray icon ומאפשרת פתיחה של הממשק הגרפי או סגירה שלו.
5. מבני הנתונים – ConfigDB (נמצא ב- definitions.json) מכיל את הגדרות המשתמש שנקבעו בממשק הגרפי, כלומר עבור כל אפליקציה שהוגדרה איזה מקש לבצע כאשר מתבצעת מחווה מסוימת. Apps (נמצא ב- apps.txt) רשימת האפליקציות והמיקומים שלהם שנסרקו על ידי קובץ ה- getApp.exe המופעל מתוך ה- process.

דף מפתח - data

DataLoader
<pre> + data_dir: str + video_dir: str + n_videos: dict + train_df: pandas.core.frame.DataFrame + validation_df: pandas.core.frame.DataFrame + labels: list + n_labels: int + label_to_int: dict + int_to_label: dict + seq_length: int </pre>
<pre> + __init__(self, str, int, dict, list): DataLoader + get_labels(self, list): none + get_labels_list(self): list + load_video_labels(self, str): pandas.core.frame.DataFrame + @staticmethod: reduce_labels(pandas.core.frame.DataFrame, dict): pandas.core.frame.DataFrame + sequence_generator(self, str, int, tuple(int,int)): tuple(numpy.ndarray, numpy.ndarray) + build_sequence(self, str, int): numpy.ndarray + @staticmethod: random_sample(pandas.core.frame.DataFrame, int): pandas.core.frame.DataFrame + @staticmethod: preprocess_image(numpy.ndarray): numpy.ndarray </pre>

1. המחלקה DataLoader - היא טוענת לתוך מבנה נתונים של ספריית pandas את הסרטונים המסווגים כ - 'train' או 'validation' בסוגים מסוימים של תנועות (labels) לפי הערך שניתן בפונקציית ה-main. בנוסף היא לוקחת את המידע שנטען מהזיכרון ומחברת אותה לרצפים של תמונות (סרטונים) בהתאם לאורך שנקבע בפונקציית ה-main ומחזירה את הרצפים האלה באופן איטרטיבי ל-fit generator המאמן את המודל.

5.2.6 תקשורת

לא קיימת תקשורת בפרויקט זה (אין ממשק שרת ולקוח).

5.2.7 סביבה

מבחינת צד המפתח יש צורך ב- visual studio ו- PyCharm או ++notepad. ב- visual studio צריך להיות #c.

גרסת ה-visual studio יכולה להיות מ-2013 והלאה. בנוסף python 3 שהספריות python- pywin32, TensorFlow, Keras, NumPy, OpenCV מותקנות בו, ישנה עדיפות לגרסאות הכי חדשות בכל אחת מן הספריות הללו וניתן להוריד אותם באמצעות ה-pip של python 3.
מבחינת צד המשתמש אין צורך בשום התקנה מעבר להתקנה של ה-setup - שלנו מכיוון שהוא מבצע את כל ההתקנות הדרושות באופן אוטומטי.

6. מסמך הבדיקות

1. תיאור הקונפיגורציה של הפרויקט בזמן הבדיקות השונות , setup - (הגדרות של חומרה) מערכת הפעלה windows, מצלמת רשת או מצלמה מובנית, מעבד intel-i5.
2. תיאור הבדיקות (קלט, פלט) לתרחישים אופייניים - (אנשים שונים מבצעים את כל הפעולות) מושיבים מספר אנשים במראה שונה ונותנים להם לבצע במשך מספר דקות את 27 התנועות שהמודל צריך לזהות בקצב משתנה ורואים כיצד המערכת מגיבה.
3. תיאור תרחישים (קלט ופלט) למקרי קצה - (בחושך, תנועות מהירות, תנועות איטיות, צבע עור שונה)
 - יש מספר תרחישים בהם הזיהוי של התנועות עלול להיות בעייתי:
 1. חושך - כאשר הסביבה חשוכה, המצלמה לא מצלמת באופן ברור את התנועות ולכן הזיהוי עלול להיות לא נכון.
 2. תנועות מהירות/איטיות - התוכנה לא מותאמת לזהות תנועות איטיות מדי מכיוון שאורך התנועה צריך להיות במקסימום 40 פריימים, מעבר לכך זה יהיה בעייתי לכן תנועות איטיות הן בעייתיות לתוכנה. גם תנועות מהירות הן בעייתיות מכיוון שהמערכת תתקשה לזהות את התנועה אם היא הייתה רק על מספר פריימים בודדים.
 3. צבע עור מיוחד - מכיוון שהמודל שאחראי על הזיהוי הוא מודל של machine learning אז אנחנו לא יכולים לצפות את התגובה שלו למקרים שונים כמו למשל במקרה הזה, אם צבע העור של המשתמש יהיה מיוחד אז התוכנה עלולה לא לזהות את הידיים שלו כמו שצריך וכתוצאה מכך תהיה בעיה בזיהוי התנועה.
 4. ידיים בגודל לא רגיל - אם למשתמש יש ידיים גדולות מאוד או קטנות מאוד כמו למשל תינוק אז התוכנה עלולה לא לזהות את הידיים שלו כמו שצריך ותיווצר בעיה בזיהוי התנועה.
4. תיאור תרחישים המייצרים שגיאות וכיצד המערכת אמורה להגיב ומדווחת עליהם. - (זיהוי לא נכון, קריסה של אפליקציה)
 - ישנם שלושה תרחישים בהם התוכנה לא מבצעת את פעולתה כראוי אך בשני המקרים הללו המערכת לא תקרוס אלא תמשיך את עבודתה כרגיל לאחר פרק זמן משתנה.
 1. זיהוי לא נכון של תנועה - כאשר התוכנה מזהה לא נכון, תנועה היא תבצע את הפעולה הלא נכונה אבל לאחר מכן היא תחזור ותמשיך את פעילותה כרגיל.

2. קריסה של אפליקציה - כאשר האפליקציה שבה המשתמש נמצא קורסת אז עד שהיא לא תיסגר התוכנה לא תצליח להמשיך את פעילותה כרגיל משיקולים של מערכת ההפעלה.
3. קריסת הממשק הגרפי של התוכנה - התוכנה ממשיכה את פעילותה כרגיל כי היא אינה תלויה בממשק הגרפי להמשך ביצוע פעילותה, תפקידו היחיד הוא קביעת הגדרות המשתמש והפעלת התהליך מעבר לכך התוכנה פועלת ברקע מבלי צורך בממשק הגרפי.
5. פירוט בדיקות שבוצעו:
 1. בדיקה על מחשב נייד עם מצלמה כדי לבדוק האם התוכנה עובדת כראוי בלי באגים – הבדיקה הצליחה.
 2. בדיקה על מחשב נייד בלי מצלמה כדי לבדוק האם התוכנה מקפיצה שגיאה – הבדיקה הצליחה.
 3. בדיקה על מחשב נייד עם מצלמה כדי לבדוק האם התוכנה עובדת כראוי בלי באגים – הבדיקה הצליחה.
 4. בדיקה על מחשב נייד בלי מצלמה כדי לבדוק האם התוכנה מקפיצה שגיאה – הבדיקה הצליחה.
 5. בדיקה על מחשב נייד בלי שפות תוכנה מותקנות – הבדיקה הצליחה.
 6. בדיקה על מחשב נייד בלי שפות תוכנה מותקנות – הבדיקה הצליחה.
 7. בדיקות מרחקים – התרחקות והתקרבות למצלמה בכדי למצוא את המרחק הרחוק ביותר והקרוב ביותר מהם המודל מזהה: המרחק הרחוק ביותר הוא מטר בערך, והמרחק הקרוב ביותר עשרה סנטימטר בערך.
- לפי הבדיקות התוכנה עבדה באיכות גבוהה יותר על מחשבים בעלי CPU אך לא בהבדל ניכר וניתנת לשימוש על כל מחשב עם מצלמה ו-intel i5 CPU או מתקדם יותר.

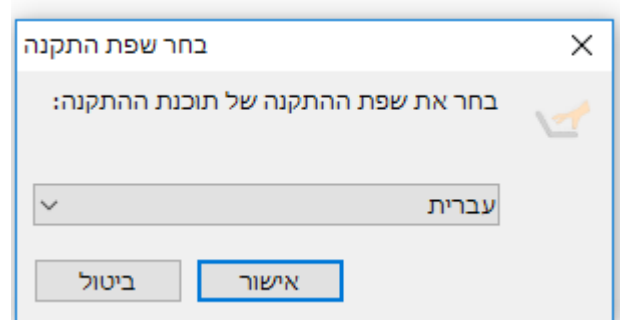
7. הוראות שימוש – התקנה והפעלה

7.1 מדריך התקנה

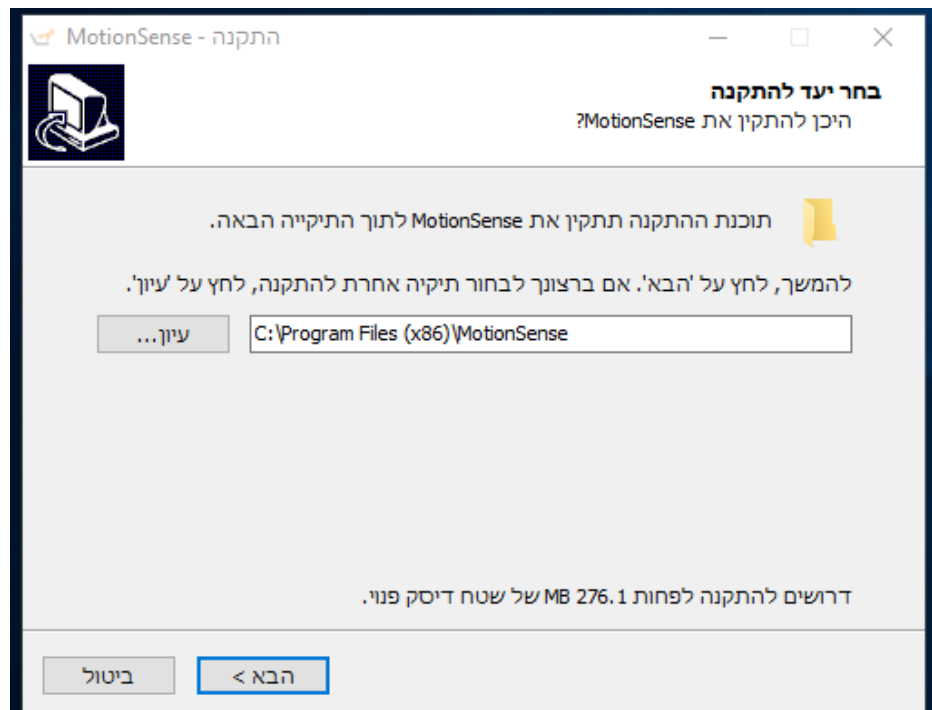
הפרויקט מגיע בקובץ MotionSense_setup.exe בדיוק כמו שהשם מציינ הפרויקט מגיע עם מדריך התקנה מובנה:

שלב ראשון - לחיצה על קובץ ההתקנה ולחיצה על אשר לתוכנה לבצע שינויים במחשב שלי.

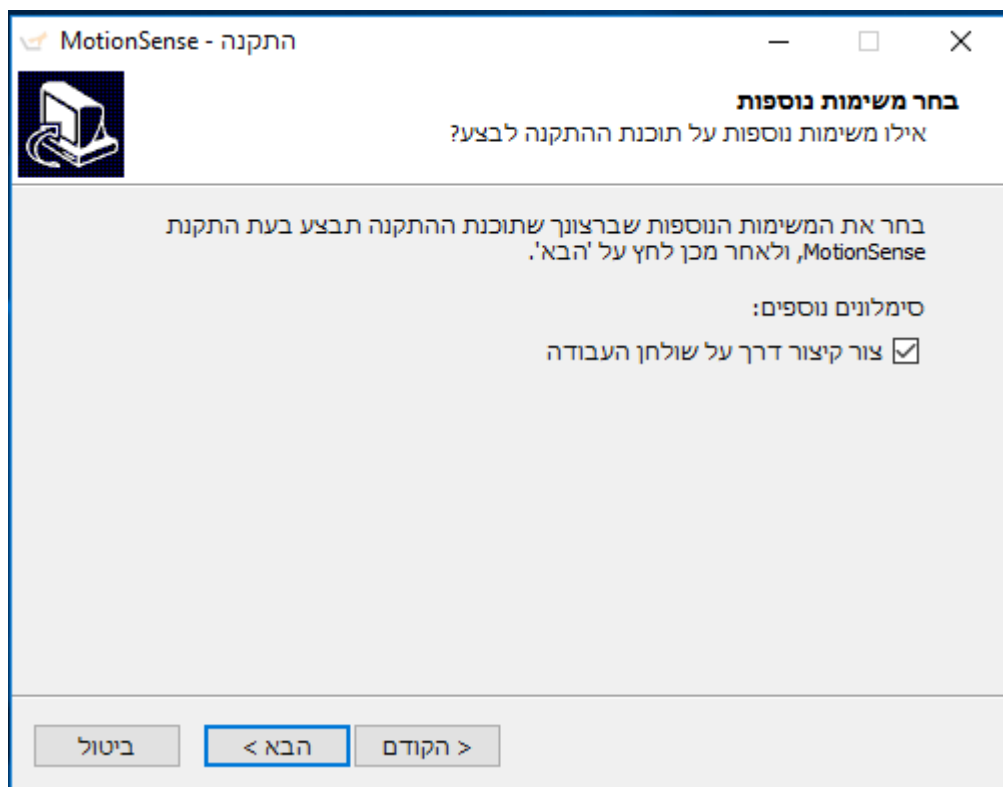
שלב שני - בחירת שפת התקנה



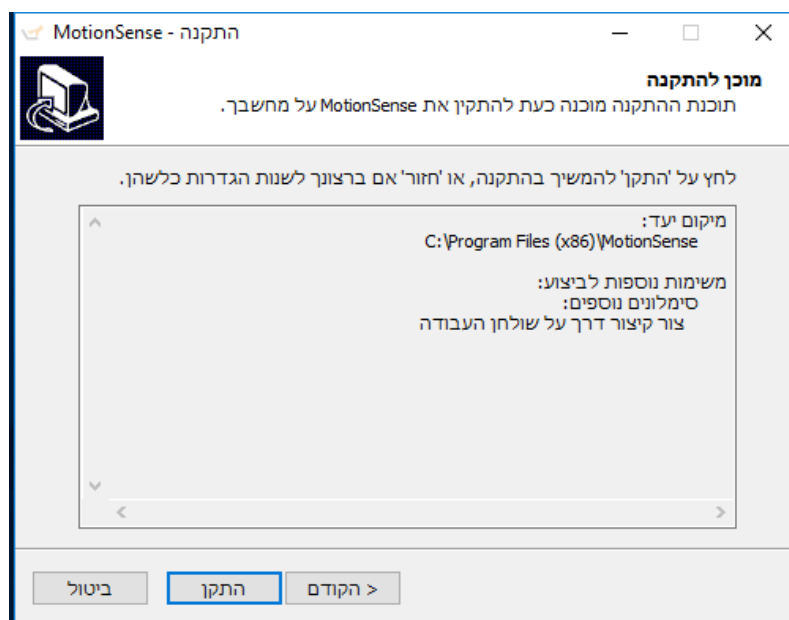
שלב שלישי - בחירת מיקום התקנה



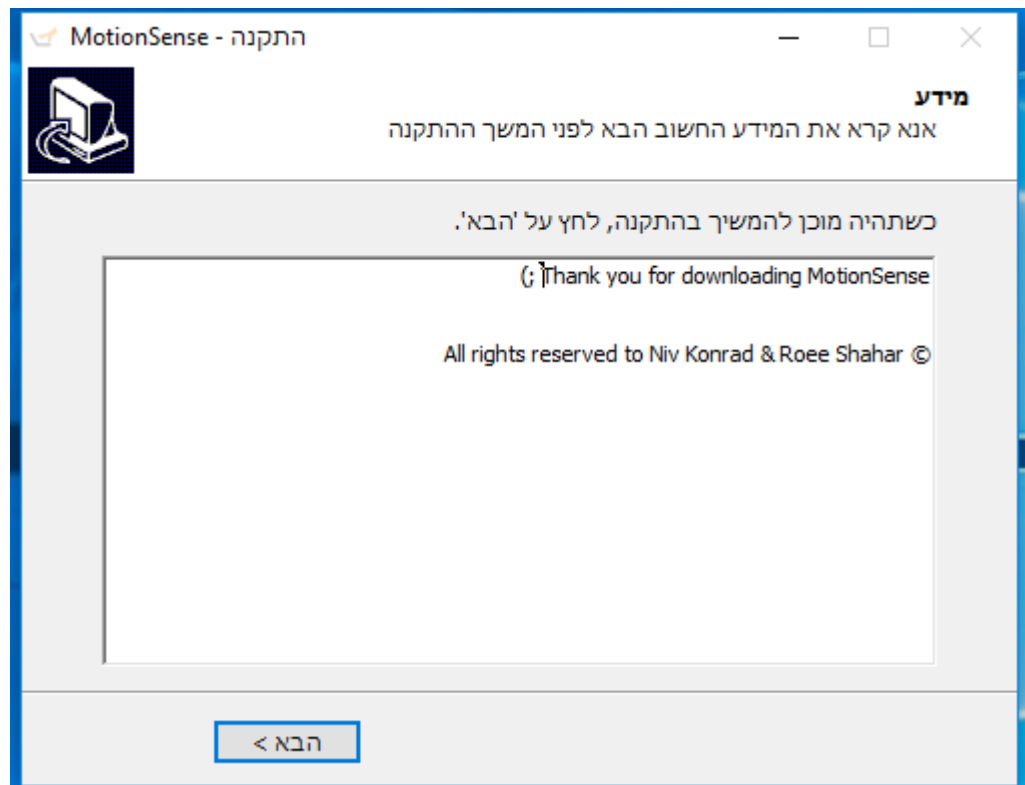
שלב רביעי - בחירת אפשרויות נוספות



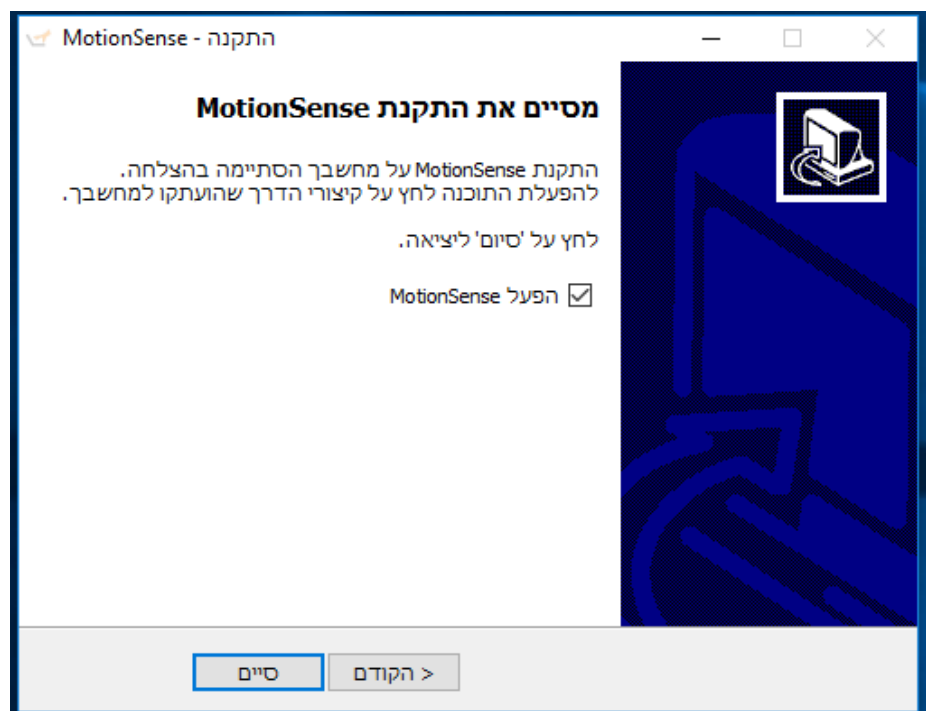
שלב חמישי - התקנה



שלב שישי - קרדיטים



שלב שביעי - סיום התקנה



תיאור התקנת הפרויקט עצמו וכן תיאור הנדרש כדי להתקין את חבילת התוכנה הנדרשות להרצת הפרויקט (לא כולל python ומחלקות builtin בפייתון).

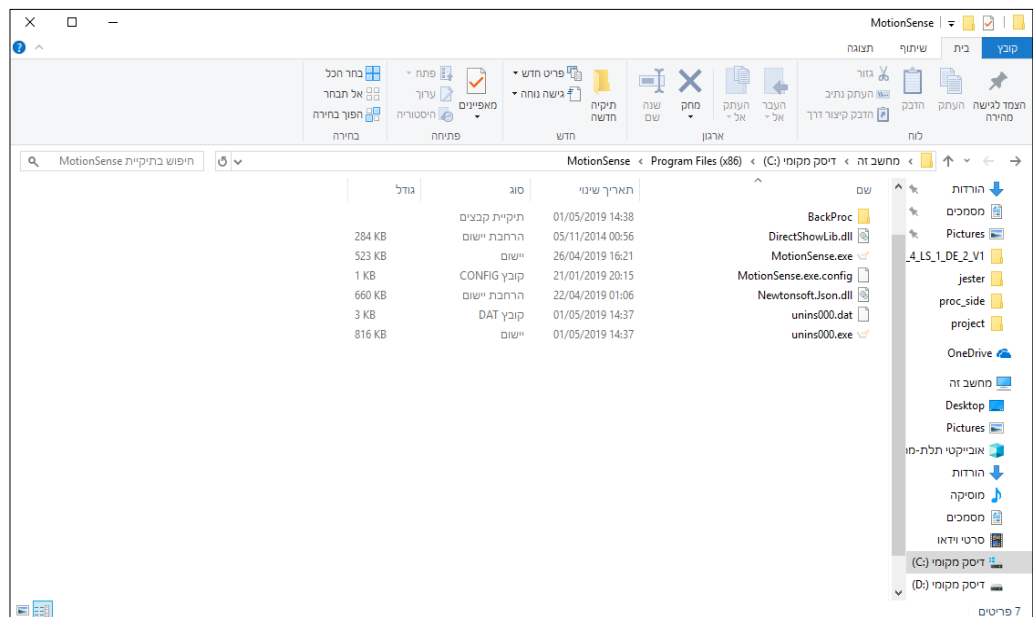
7.2 התאמת סביבת העבודה להרצה - קונפיגורציה

אין הכנות שצריך לבצע לפני השימוש במערכת.

7.3 מדריך האפליקציה למשתמש - User Guide

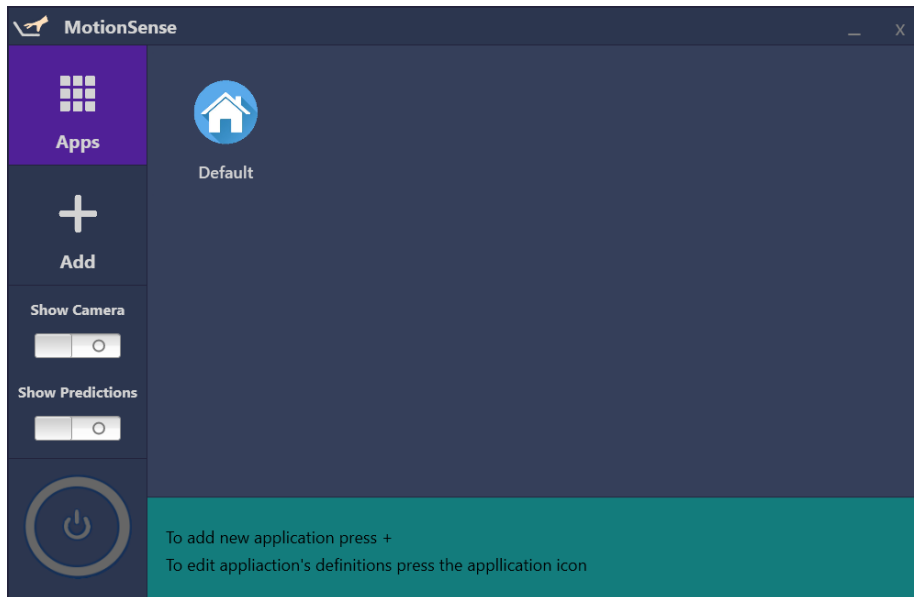
7.3.1 הגדרת המערכת

שלב ראשון - כניסה לתוכנה מתבצעת על-ידי לחיצה על icon התוכנה בדף הבית או ישירות על-ידי הפעלת הקובץ MotionSense.exe המותקן בספרייה שהוגדרה בשלב ההתקנה.
בתור ברירת מחדל התוכנה תתקן במיקום: "C:\Program Files (x86)\MotionSense".



שלב שני – הגדרות התוכנה.

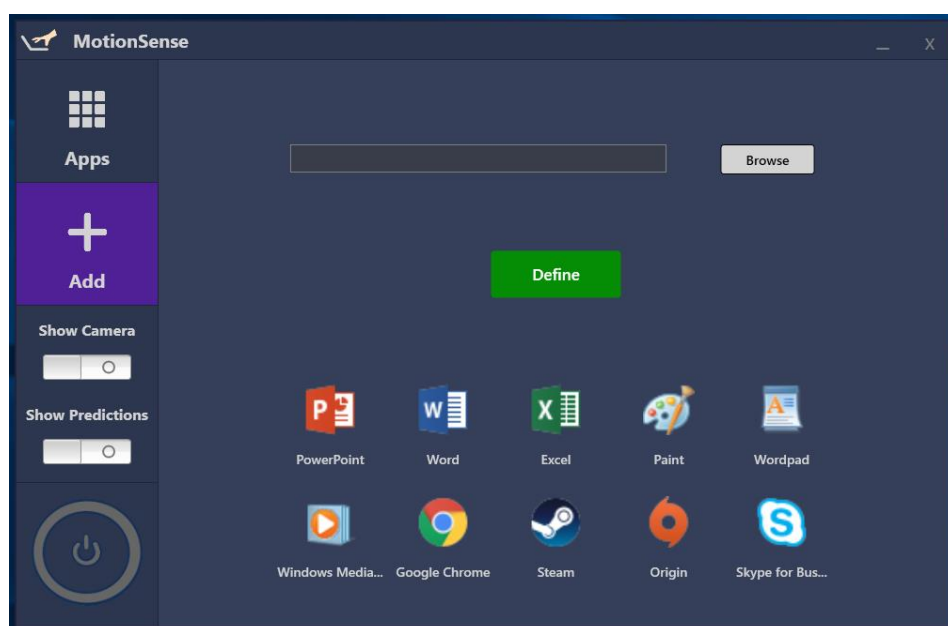
בפעם הראשונה שהתוכנה עולה, עדיין אין אפליקציות מוגדרות לשימוש ויש צורך להגדיר אותם.



על מנת להגדיר אפליקציות יש ללחוץ על האייקון + (עמוד ה- Add).

שלב שלישי - לחיצה על האייקון + מובילה אותנו לדף הגדרת האפליקציות, יש שלוש אפשרויות:

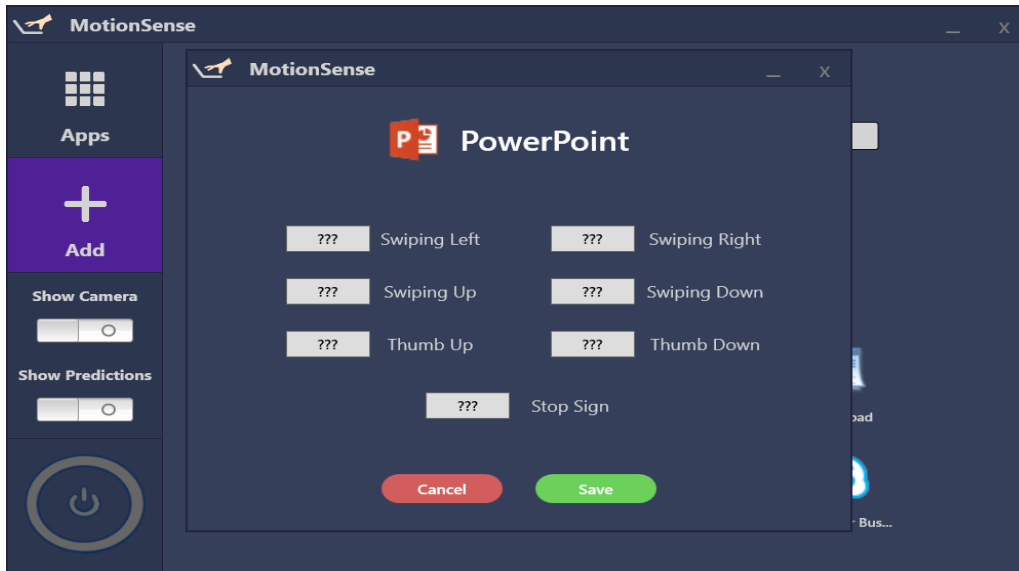
1. הגדרת אפליקציה מרשימת האפליקציות המוצעות למטה על ידי לחיצה על האייקון של אחת מהם.
2. הגדרת אפליקציה על ידי הכפתורים Browse ו- Define המאפשרים למשתמש לנווט ב file explorer למיקום של קובץ .exe. ולהגדיר עבורו מחוות ידיים.
3. בחירת אפליקציה והגדרתה תוך שימוש בשורת החיפוש.



להלן ההוראות להגדרה של כל אחת משלושת האופציות:

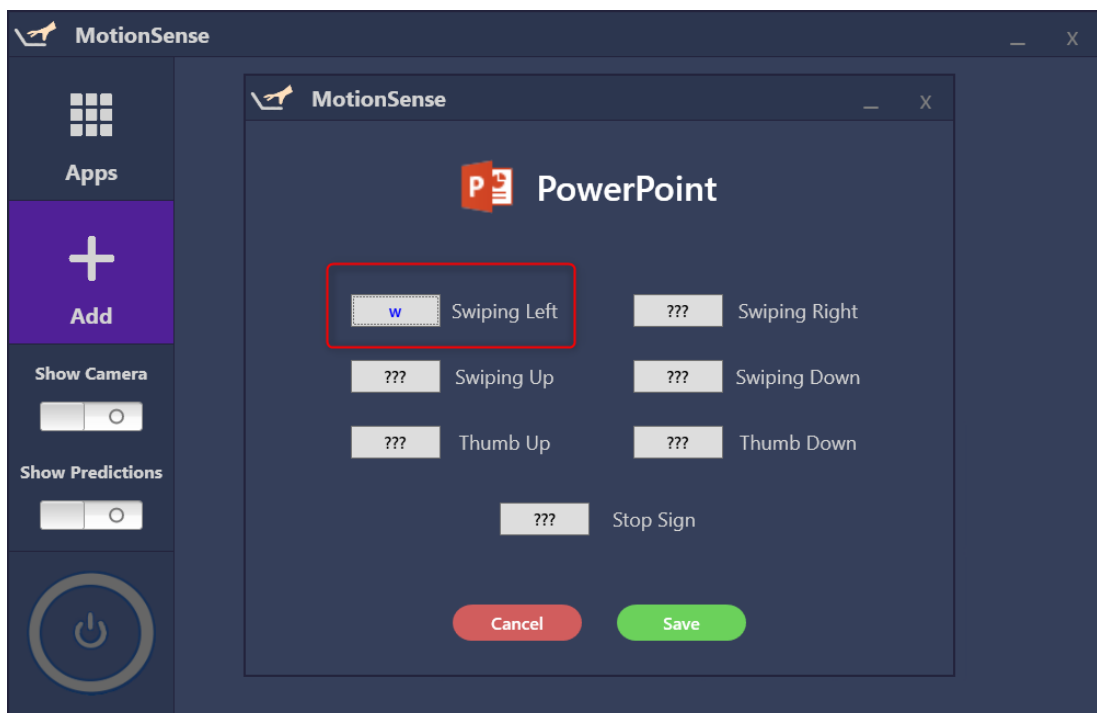
הגדרת אפליקציה תוך שימוש ברשימת האפליקציות:

לחיצה על אחד מהאייקונים של האפליקציות בתחתית העמוד תפתח את חלון ההגדרות הבא:



רשימת המחוות הניתנות לשימוש מוגדרות מראש, וכפי שניתן לראות, בשלב ההגדרה, לצד כל מחווה יש כפתור ובתוכו רשום '???'.

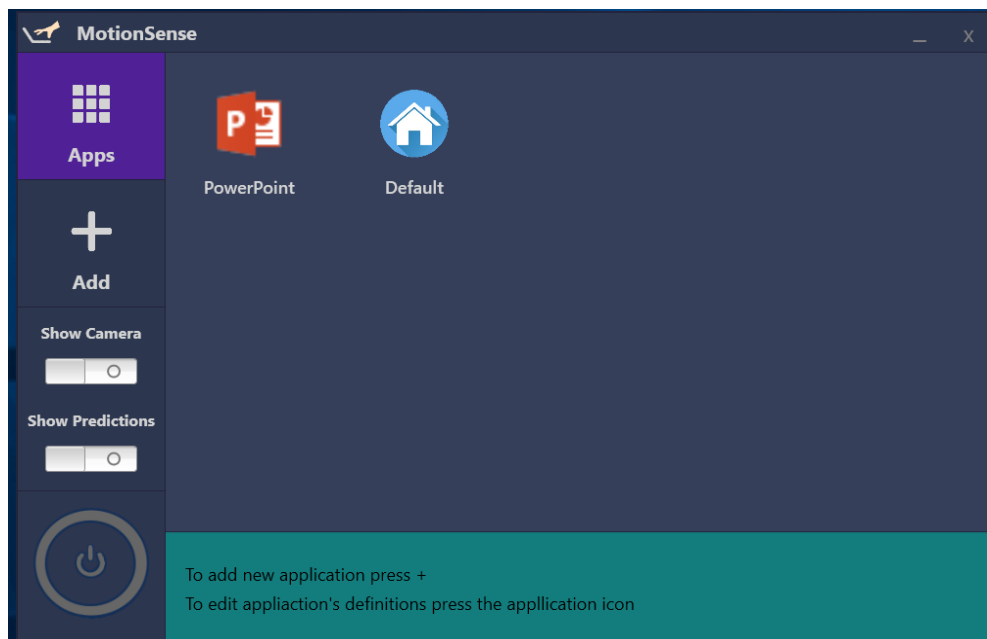
לחיצה על כפתור זה ישנה את צבעו לאדום ותוכנו ישתנה ל - <???.>. כלומר שהוא במצב הגדרה. נלחץ על המקש במקלדת אשר נרצה שיתבצע כאשר אנו מבצעים את המחווה המתאימה באפליקציה



הנבחרת ומקש זה יופיע בצבע כחול כתוכן הכפתור. במקרה שלנו נבחר את הכפתור של הזזת היד שמאלה ונלחץ על האות 'w'.

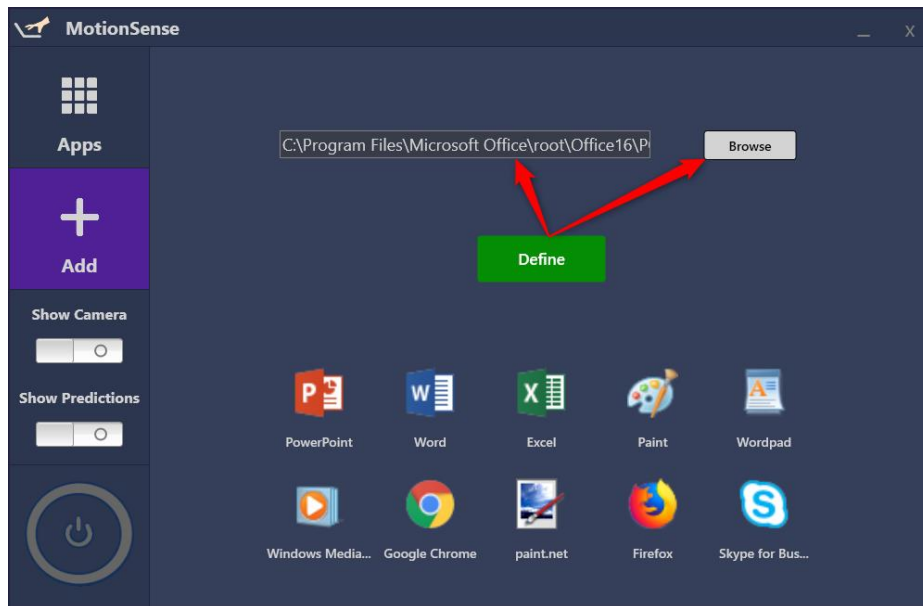
ניתן לראות שהאות 'w' המסמלת את המקש שעליו לחצנו הופיעה ליד השם של התנועה. כלומר, הגדרנו שבאפליקציה PowerPoint כאשר המשתמש יבצע הזזה של היד שמאלה יתבצע המקש 'w'.

לחיצה על כפתור ה- Save תשמור את ההגדרות הנוכחיות לאפליקציה. כאשר נלך לחלון Apps נראה כי האפליקציה PowerPoint הוגדרה.



הגדרת אפליקציה על ידי הכפתורים Browse ו- Define:

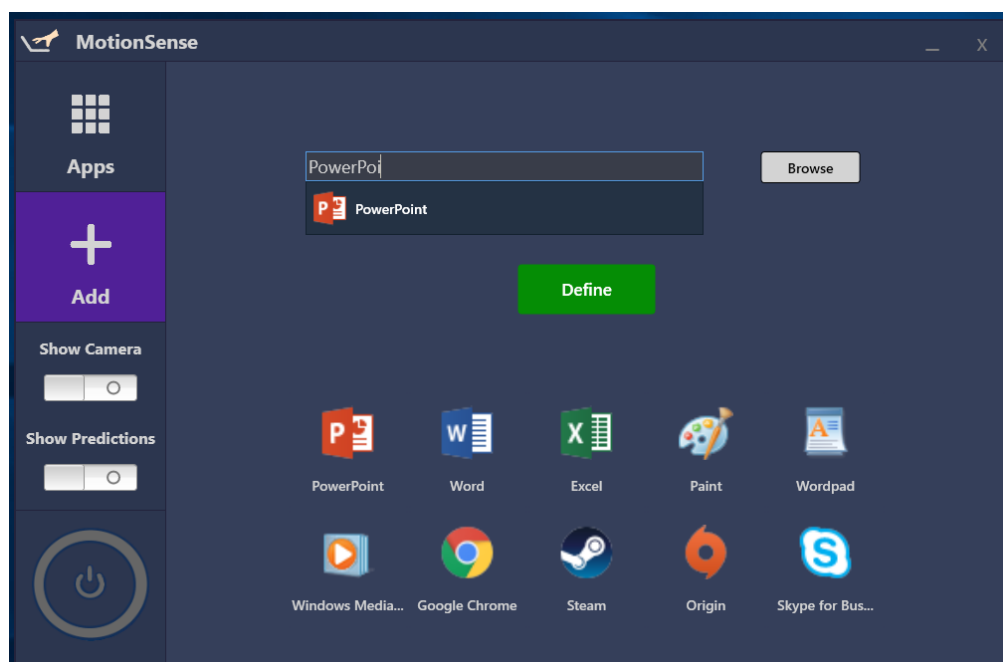
בחלון ה- Add, לחיצה על כפתור ה-Browse תפתח את ממשק ה-file explorer. נווט לקובץ ה- Exe הרצוי ובחירתו, ייתן לנו אפשרות להגדיר את אפליקציית ה-Exe שנבחרה. אם לדוגמא נרצה להגדיר בדרך זו את PowerPoint - נווט ב- file explorer לספרייה 'C:\Program Files (x86)\Microsoft Office\root\Office16\POWERPNT.EXE' שם נמצא הקובץ POWERPNT.EXE שהוא בעצם אפליקציית ה-PowerPoint. לאחר בחירתו, חלון החיפוש יתמלא בשם הקובץ.



כעת, משנבחר הקובץ המייצג את האפליקציה, נלחץ על Define על-מנת להגדירה. שלב הגדרת האפליקציה ושמירת ההגדרות זהה לשלבי ההגדרה כפי שפורטו בסעיף הקודם

בחירת אפליקציה והגדרתה תוך שימוש בשורת החיפוש:

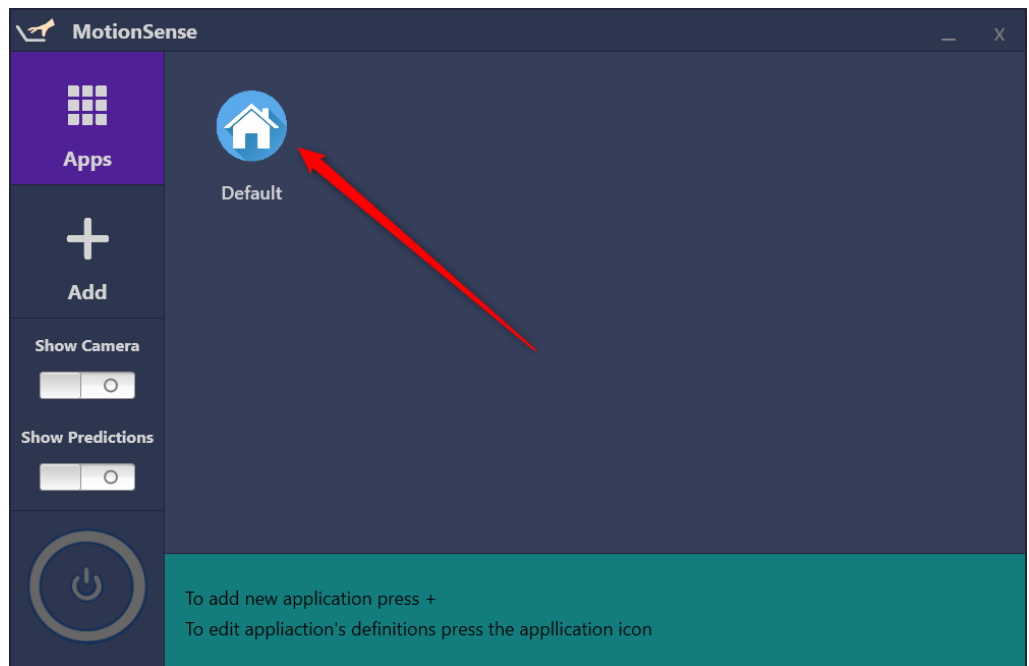
בחלון ה-App, כאשר מקלידים בשורת החיפוש את מסוימת, המערכת מחפשת אפליקציות שהותקנו במחשב ואשר ניתן להגדירם. תוצאות החיפוש והאפליקציות התואמות, מופיעות ברשימה מתחת לשורת החיפוש.



כך למשל אם נרצה לחפש את תוכנת PowerPoint שמותקנת אצלנו במחשב, נקיש 'Power' ותופיע התוכנה PowerPoint עם האייקון המתאים בתוצאות החיפוש.

נבחר באפליקציה הרצויה, במקרה זה PowerPoint ונלחץ על הכפתור Define.
שלב הגדרת האפליקציה ושמירת ההגדרות זהה לשלבי ההגדרה כפי שפורטו לעיל.

הגדרת ה- Default
כדי להגדיר את הגדרות ה- Default שימשו לכל האפליקציות שלא הוגדר להם הגדרות נלך קודם אל המסך Apps:



ונפעל כפי שהוסבר בשלבים קודמים. כלומר, נלחץ על האייקון של ה- Default ונגדיר בחלון שנפתח בפנינו את המקשים/קומבינציות המקשים שאנחנו רוצים שיתבצעו כאשר תנועה מסוימת מתבצעת על ידינו. לאחר שקבענו default מכאן ואילך כאשר ה-process מופעל אך המשתמש נמצא באפליקציה שלא הוגדרו לה הגדרות, התוכנה תפעל על פי הגדרות ה-default.

7.3.2 שינוי ומחיקת הגדרות

שינוי הגדרות לאפליקציה מוגדרת:

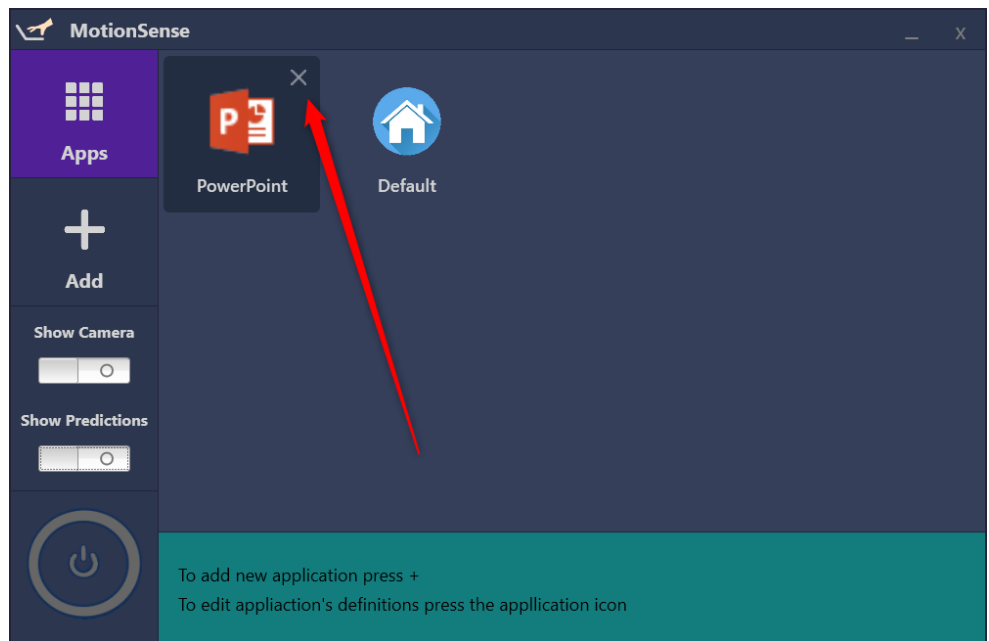
על מנת לשנות את ההגדרות לאפליקציה שהגדרנו (במקרה זה PowerPoint) נלחץ על האייקון שלה בעמוד Apps. עמוד ההגדרות של האפליקציה יפתח ומכאן ניתן לשנות הגדרות של מחוונים בפקודות אחרות.

לחיצה על כפתור ה-Save תשמור את ההגדרות החדשות לאפליקציה.

מחיקת הגדרת אפליקציה:

נלך לחלון Apps. נעמוד על האייקון של האפליקציה עם העכבר.

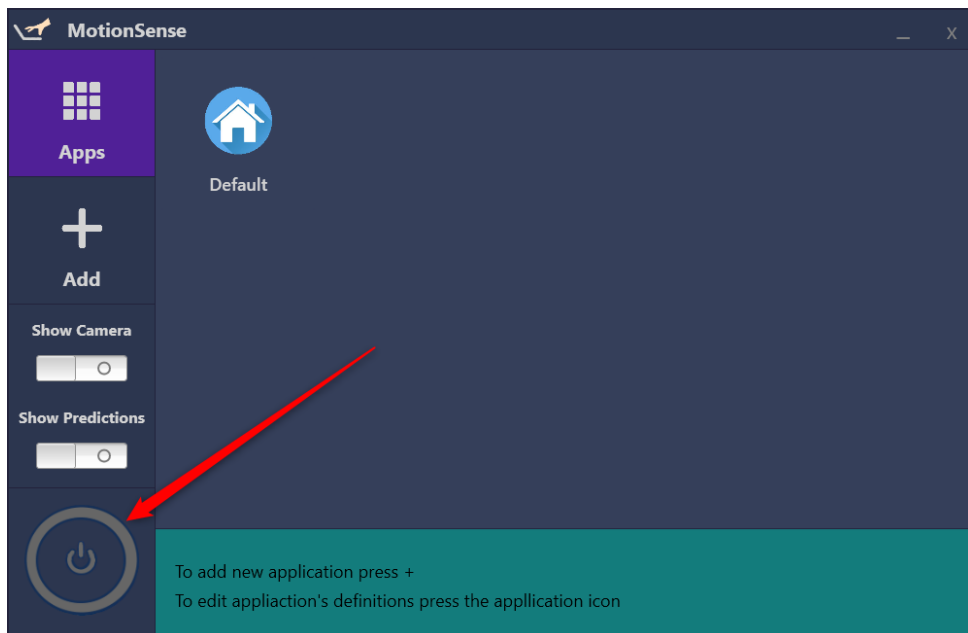
לחיצה על ה-X שמופיע מעל האייקון של האפליקציה תמחק את הגדרות האפליקציה מהמערכת ואייקון האפליקציה יעלם מהמסך.



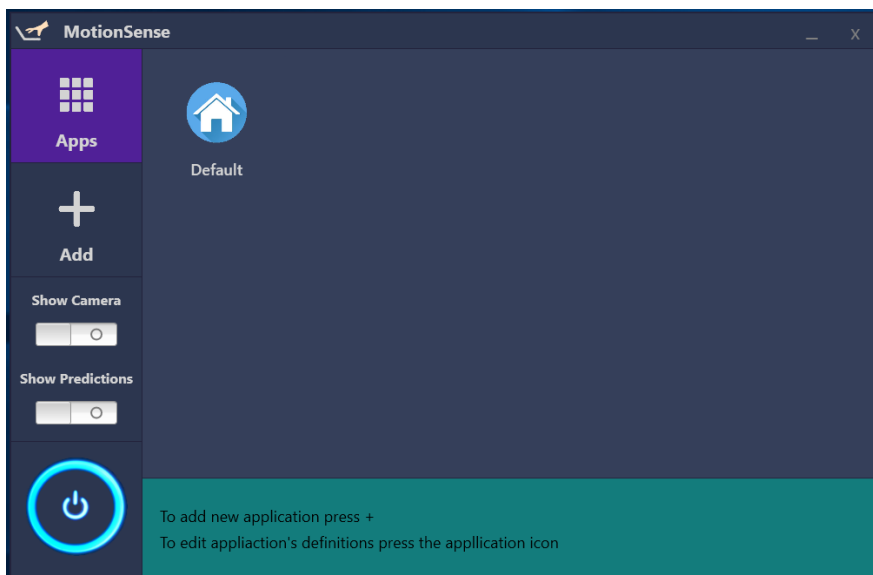
בכדי להגדיר את האפליקציה שוב, יש לחזור על פעולות ההגדרה כפי שמופיע בסעיף "הגדרת המערכת" בפרק זה.

7.3.3 הפעלת המערכת

בכדי להפעיל את המערכת ולהתחיל את פעולת ה- Process שמנתח את התנועות ומבצע את הפעולות באפליקציות בהתאמה צריך ללחוץ על כפתור ה-power המופיע בצד השמאלי התחתון של המסך.



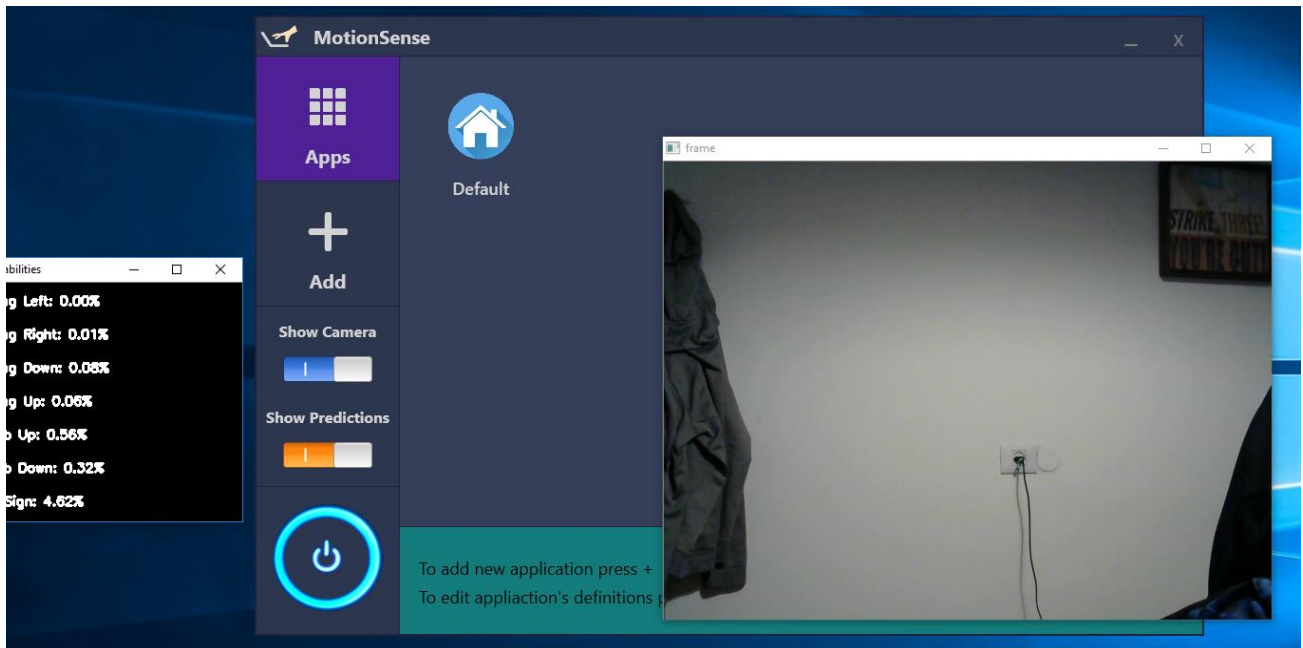
כאשר לוחצים על הכפתור הוא משנה את צבעו לכחול ולאחר כמה שניות פעולת ה- process מתחילה.



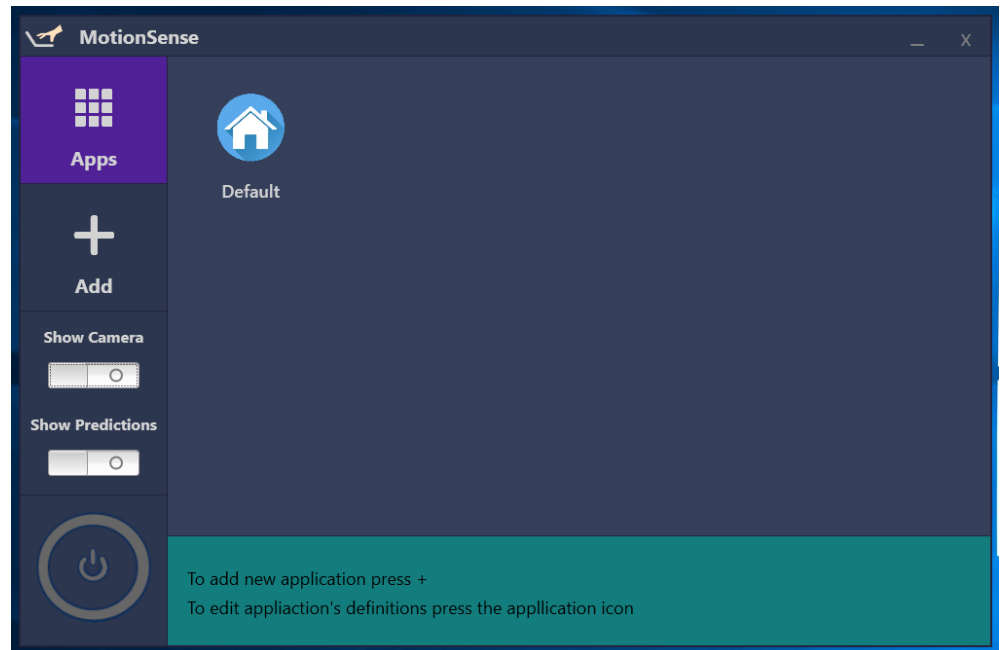
ישנם מספר דרכים לזהות אם פעולת ה-process התחילה:

- ברוב המקרים ידלק אור במצלמה של המשתמש ברגע שה-process יתחיל לפעול. אך במצלמות מסוימות אין נורה שנדלקת כאשר הן מופעלות לכן נוכל או לחכות בסביבות עשרים שניות כך שנדע בהכרח שפעולת ה-process התחילה או שנוכל להדליק את אחד משני המתגים הנראים בתמונה:
1. מתג show camera - כאשר מדליקים את מתג זה הוא מאפשר לך לראות את הסרטון שהמצלמה מצלמת, הוא מתחיל לצלם כאשר פעולת ה-process מתחילה כך שהוא יכול לשמש כאינדיקטור לתחילת פעולת ה-process. בנוסף הוא מועיל בכדי לסדר את המצלמה כך שהמשתמש יצלם כמו שצריך את חלק הגוף העליון שלו כדי שהמודל יוכל לחזות באופן מדויק יותר את התנועות שהמשתמש מבצע.
 2. מתג show predictions - כאשר מדליקים את מתג זה נפתח חלון שבו רשומות המחוות ולידן מד שמציג בזמן אמת בכמה אחוז המודל בטוח כי הוא זיהה שהמחוות הללו מתבצעות בסרטון. חלון נפתח כאשר ה-process מתחיל לפעול לכן גם הוא יכול לשמש אינדיקטור לפעולת ה-process.

דוגמא של פעולת ה-process ששני המתגים דלוקים:

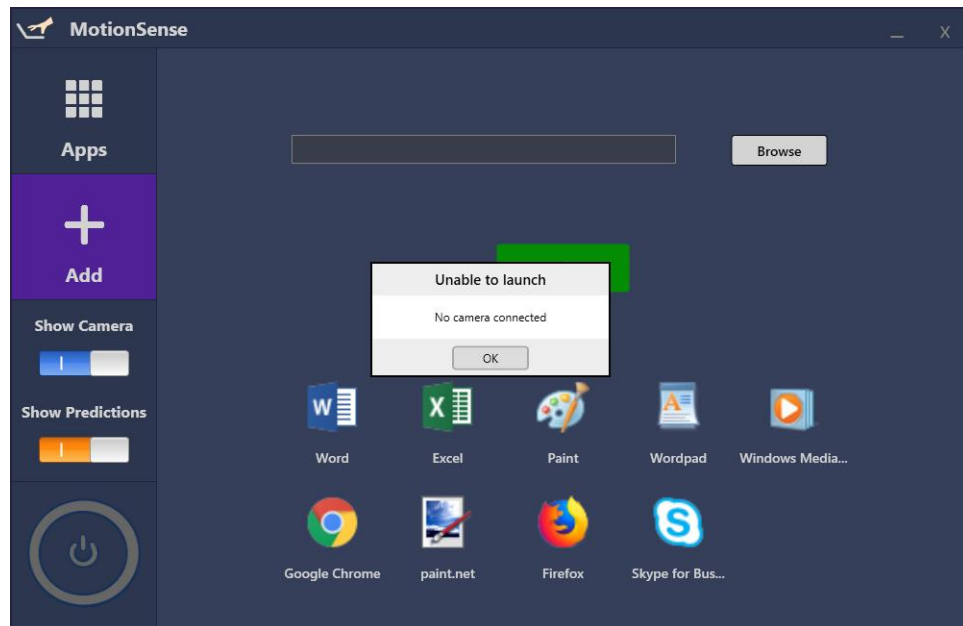


כמו שניתן לראות הדלקנו את שני המתגים והמצלמה מצלמת את הקיר לכן לא מתבצעת מחווה בסרטון וליד כל מחווה בחלון show predictions האחוזים נמוכים יותר אפילו מחמש אחוז. כעת ניתן ללחוץ שוב על הכפתור בכדי לסגור את החלונות הללו ואת ה-process:

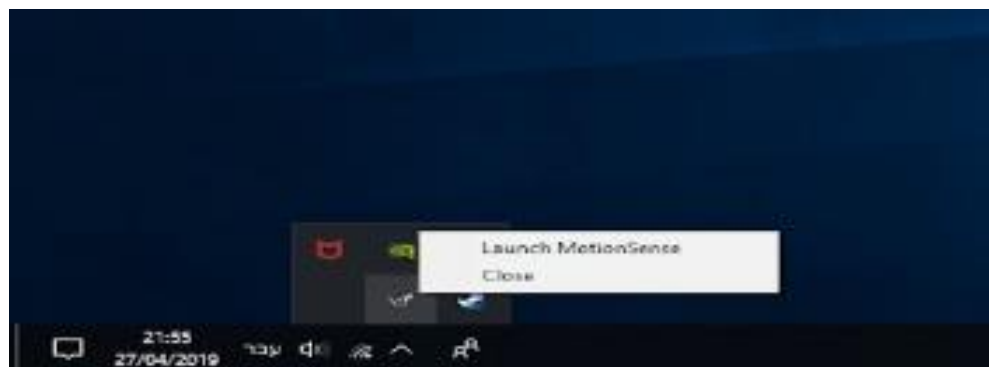


ישנם מספר דברים נוספים שיש לציין:

- פעולת ה- Process הינה נפרדת מפעולת הממשק הגרפי לכן אפשר להפעיל את ה- Process מהממשק הגרפי ולאחר שהוא מתחיל לפעול לצאת מן הממשק הגרפי וה- Process עדיין ימשיך לפעול. בנוסף כאשר המשתמש יפתח את הממשק הגרפי שוב הכפתור שמראה על פעולת ה- Process ישאר דולק מכיוון שגם ה- Process עדיין דולק.
- ישנו זמן המתנה של כמה שניות בין הלחיצה על כפתור ההדלקה לבין תחילת פעולת ה- Process כפי שהוצג לעיל.
- כאשר לא מחוברת מצלמה למחשב אך המשתמש עדיין מנסה להדליק את ה- Process קופצת הודעת שגיאה שמצלמה לא מחוברת



ישנה אפשרות להפעיל את האפליקציה או לסגור אותה גם מה- icon tray:



7.4 מדריך האפליקציה למנהל - Admin

אין אפשרויות מיוחדות למנהל האפליקציה Admin.

8. משוב

בעקבות העבודה על הפרויקט למדנו מספר דברים. ראשית למדנו כיצד לקחת רעיון ולהפוך אותו לפרויקט באמצעות תהליך של תכנון וביצוע. בנוסף, למדנו איך לבצע תכנון נכון של פרויקט לפי חלוקת עבודה וזמן (במקרה של פרויקט זה בשיטת עבודה הקרויה ספרינטים) וכיצד לבצע תהליך של למידה עצמית של חומר מהאינטרנט באמצעות מאמרים, קורסים ודוגמאות קוד.

הפרויקט הצית בנו עניין הן בתחום של Deep Learning, והן בתחום של פיתוח ושימוש ב-windows, והפך אותנו לאנשי תוכנה טובים יותר בעלי מגוון רחב יותר של כישורים.

הפרויקט השיג את המטרות שהצבנו לעצמנו - ביצוע תוכנה שתעבוד ונוכל להשתמש בה ולהפיק ממנה תועלת גם לאחר שסיימנו את הפיתוח למטרות אישיות. המטרה הנוספת של הפרויקט הייתה להעשיר את הידע בתחומים שונים ובעיקר בתחום ה-Deep Learning.

מכיוון שכל המטרות שהצבנו לעצמנו הושגו, ונוסף עליהם למדנו עוד דברים שיועילו לנו בעתיד כגון: אופן תכנון נכון של פרויקט.

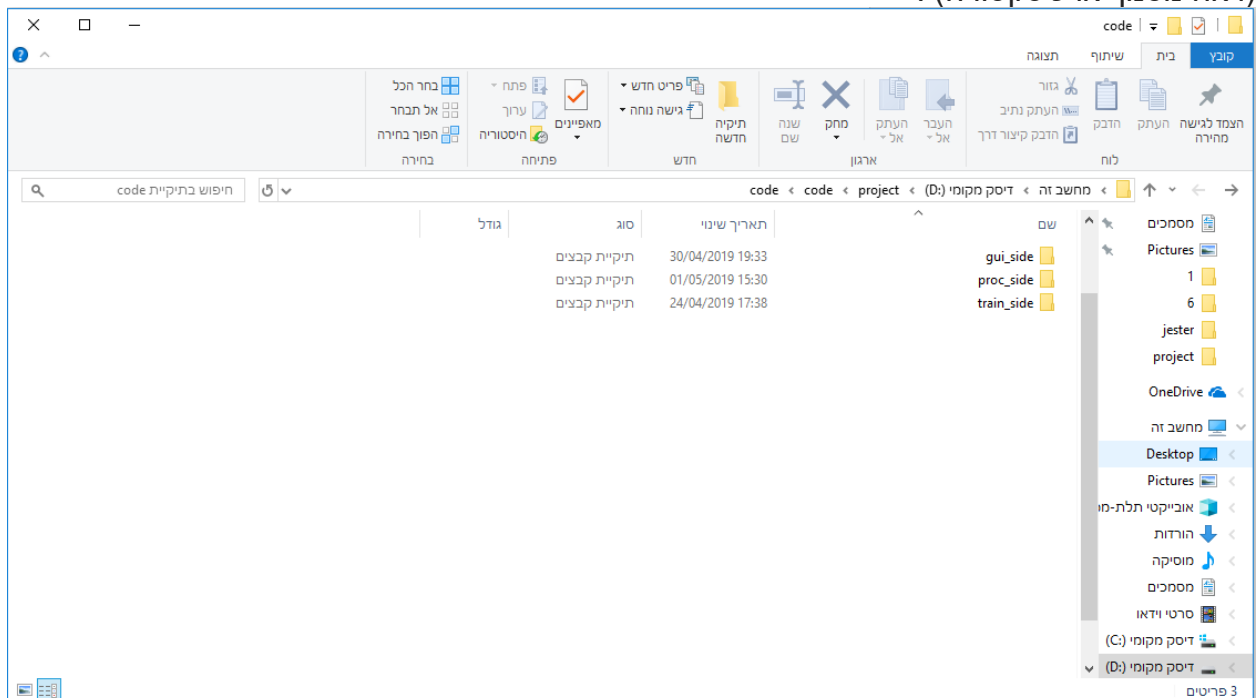
לסיכום - תהליך העבודה על הפרויקט היה תהליך מועיל ומהנה, שתרם ליכולות שלנו בעתיד בתור אנשי תוכנה.

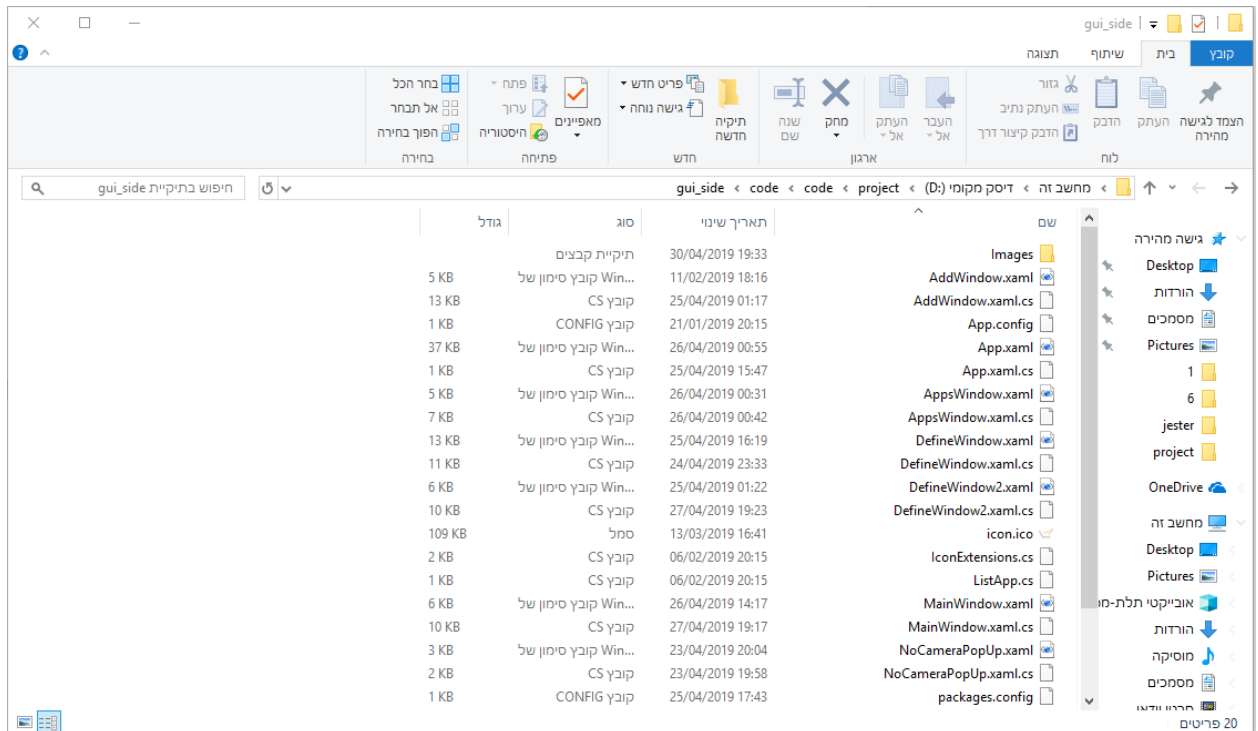
9. קוד הפרויקט



code.rar

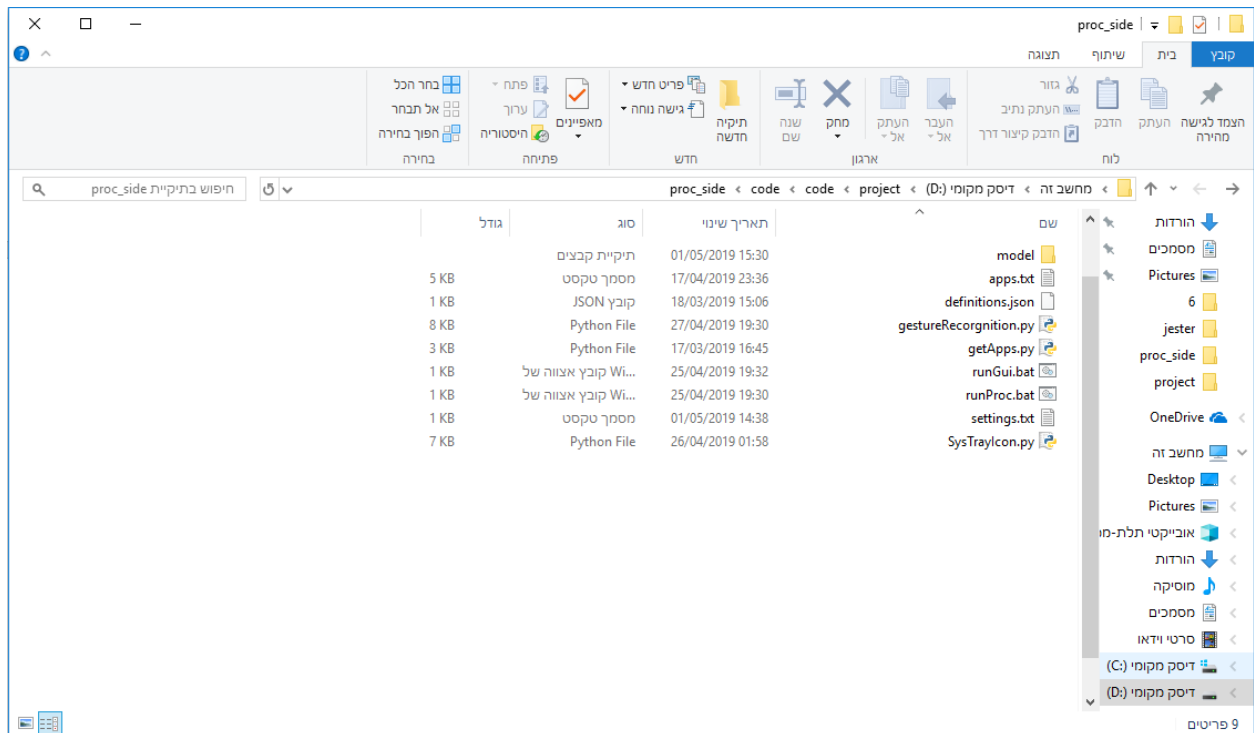
הקוד של הפרויקט בקובץ ה-rar מחולק לשלושה תיקיות ראשיות לפי החלקים הראשיים של הפרויקט (ראה מסמך ארכיטקטורה):





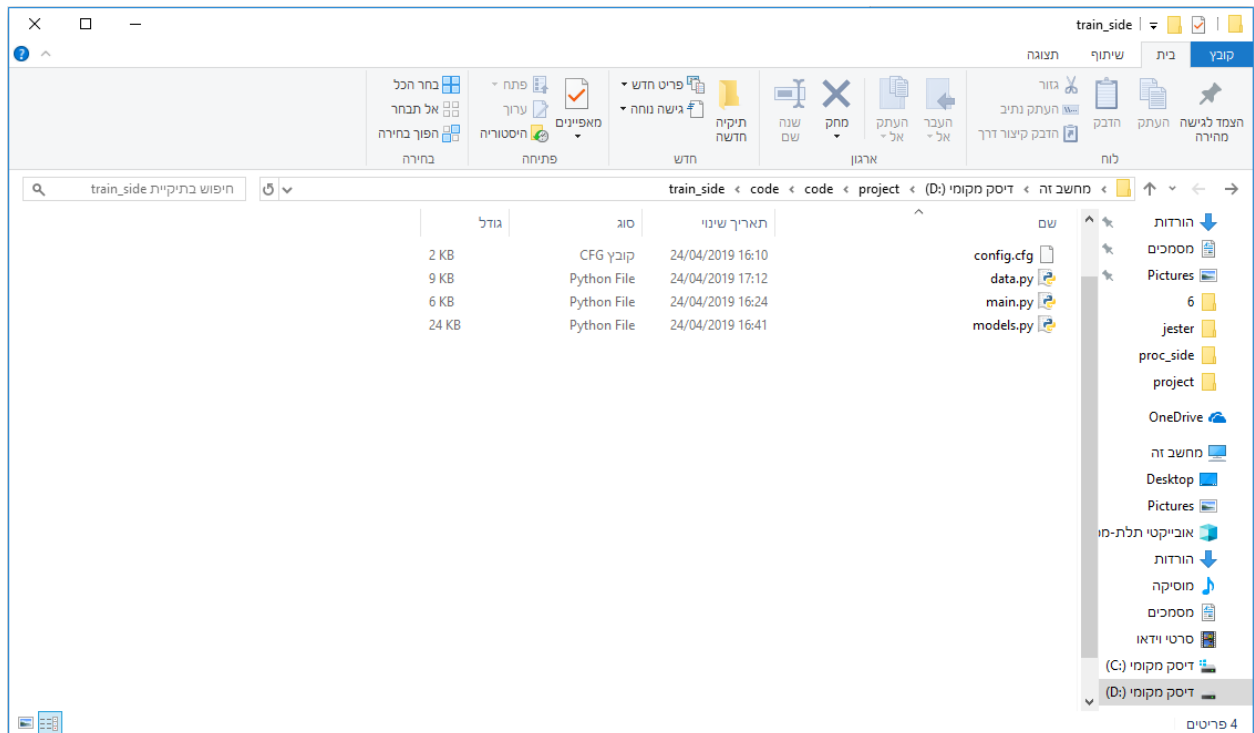
1. **gui_side** – החלק של הממשק הגרפי בפרויקט, חלק זה נכתב בשפת #c בשימוש ב- visual studio. בתוך התיקייה gui_side ישנם קבצי xaml וקבצי cs, אלו הם קבצי הקוד של הממשק הגרפי. בנוסף יש קובץ config שמסדר את קונפיגורצית המערכת (החבילות של #c בהם הממשק הגרפי משתמש) וקבצי תמונות בתוך התיקייה Images שבהם הממשק הגרפי משתמש. קבצי הקוד מורכבים מקבוצות של זוגות קבצים בעלי אותו שם, כאשר אחד בעל סיומת xaml ומסמן את העיצוב הגרפי של חלון בממשק הגרפי והאחר בעל סיומת xaml.cs ומכיל בתוכו את הקוד שדואג לפעולות שיהיה אפשר לעשות בתוך החלון בממשק הגרפי. פירוט הקבצים:
- App – החלון הבסיסי שנפתח כאשר יוצרים ממשק גרפי ב- visual studio מגדיר את הפעולות שקוראות בעת פתיחת הממשק הגרפי.
 - AddWindow – חלון הוספת האפליקציות בממשק הגרפי (ראה מדריך למשתמש).
 - AppsWindow – חלון האפליקציות המוגדרות (ראה מדריך למשתמש).
 - DefineWindow – חלון הגדרת המקשים לאפליקציה (ראה מדריך למשתמש).
 - MainWindow – הרקע לחלון הראשי.
 - NoCameraPopUp – חלון השגיאה שקופץ במקרה שמצלמה לא מחוברת למחשב (ראה מדריך למשתמש).

- IconExtensions – קובץ קוד בשפת c# שנועד להחזיר את התמונה של האייקון של כל אפליקציה כך שהמשתמש יראה בחלונות השונים את התמונות של האייקונים של האפליקציות (ראה מדריך למשתמש).
- ListApp – קובץ קוד בשפת c# שמגדיר מחלקה בשם ListApp שתכונתיה הם שם האפליקציה והאייקון של האפליקציה, מחלקה זאת נמצאת בשימוש כאשר אנחנו רוצים להציג את האפליקציה ביחד עם השם שלה בממשק הגרפי (ראה מדריך למשתמש)
- icon.ico – האייקון של האפליקציה בשולחן העבודה וב icon tray של windows.
- pacakages.config – קובץ האחראי על קונפיגורצית המערכת, חבילות שנעשה בהם שימוש במערכת.
- Images – תיקייה המכילה קבצי תמונות שונים שנעשה בהם שימוש בממשק הגרפי.



2. **proc_side** – החלק של ה-process בפרויקט, חלק זה נכתב בשפת פיתון בשימוש ב-notepad++ ו-PyCharm. בתוך התיקייה proc_side יש שלושה קבצי פיתון, שתי קבצי טקסט שאחד מהם משמש כקובץ הגדרות והאחר כבסיס נתונים, קובץ json שמשמש כבסיס נתונים, שתי קבצי bat ותיקייה שבה יש קובץ hdf5 שהוא קובץ המודל שאומן ב-Keras ואחראי על חיזוי מחוות היד בסרטון. פירוט הקבצים:

- apps.txt – קובץ שנוצר כתוצאה של הרצת קובץ הפיתון getApps והוא משמש כבסיס נתונים המכיל בתוכו מיקומים ושמות של אפליקציות ברחבי המחשב שאותם יהיה ניתן להגדיר בממשק הגרפי.
- defintions.json – קובץ שמשמש כמבנה נתונים ובו שמורים ההגדרות לכל אפליקציה באופן הבא (שם האפליקציה: סוג מחוות יד: מקש לביצוע) מסודר בפורמט של מילון בתוך מילון בפיתון.
- gestureRecorgnition – תוכנה שכתובה בפיתון שהוא עיקר צד ה-process, בקובץ זה נוצר ה-process, נבדק באיזה אפליקציה המשתמש נמצא כרגע (באמצעות winapi), מתבצע החיזוי של איזה תנועה נעשית בסרטון על ידי המשתמש (באמצעות המודל) ונשלחת ההוראה למערכת ההפעלה בנוגע לאיזה פעולה עליה לבצע על ידי התאמה של האפליקציה שבה המשתמש נמצא לתנועה שהמודל זיהה בסרטון ולבסיס הנתונים defintions.json.
- getApps – סקריפט שכתוב בפיתון שאחראי על סריקת המחשב בחיפוש אחר אפליקציות אשר יהיה ניתן להגדיר להם פעולות בממשק הגרפי (הסריקה מתבצעת על תיקיות מסוימות בהם נשמרות אפליקציות ברוב המקרים ב-windows). הסקריפט שומר את המידע שמצא בקובץ apps.txt.
- runGui.bat – קובץ שמשמש ב-SysTrayIcon בכדי להפעיל את הממשק הגרפי מתוך ה-icon tray של windows.
- runProc.bat – קובץ שמשמש ב-SysTrayIcon בכדי להפעיל את ה-process מתוך ה-icon tray של windows.
- Settings.txt – קובץ טקסט שבו מוגדר האם לפתוח או לא את החלונות שמראים את הסרטון ואת אחוזי החיזוי כאשר ה-process רץ (ראה מדריך למשתמש)
- SysTrayIcon – קובץ פיתון שתפקידו לדאוג לכך שיהיה את האייקון של האפליקציה MotionSense ב-icon tray של windows, ולהגדיר את הפעולות שיהיה ניתן לבצע ממנו.
- model – תיקייה שמכילה את הקובץ 3DCNN_LSTM_RGB.hdf5 שהוא קובץ המודל המשמש לחיזוי של תנועה בסרטון (ראה הרכב מודל במסמך ארכיטקטורה).



3. **train side** – החלק של ה- train בפרויקט, חלק זה נכתב בשפת פיתון באמצעות NumPy, TensorFlow ו- Keras בשימוש ב- notepad++ ו- Jupyter notebook. בחלק זה אימנו את המודל באמצעות בסיס הנתונים jester של twentybn המכיל מעל ל- 14 אלף סרטונים ו- 27 סוגים שונים של תנועות ידיים. פירוט הקבצים בחלק זה:

- **config.cfg** – קובץ שבו מוגדר ההגדרות שעל פיהם המודל מתאמן כמו על איזה סוגים של תנועות המודל צריך להתאמן ועל איזה אורך של סרטונים להתאמן.
- **data** – סקריפט פיתון ובו מחלקה אחת (ראה פירוט מחלקות בסעיף 4 של המסמך) שתפקידה הוא לשלוף את המידע ממבנה הנתונים jester שהורדנו לתיקיות, לעבד אותו כך שיתאים למודל ולהחזיר אותו ברצפים כל batch בטיפוס שנקרא generator כך שהמודל יוכל להתאמן על המידע באמצעות פונקציית fit של Keras.
- **models** – קובץ פיתון ובו מספר פונקציות שבכל אחת מהם מוגדר סוג שונה של מודל שאותו ניסינו לאמן על המידע, המודלים השונים בנויים באמצעות הפונקציות של Keras.
- **Main** – סקריפט בפיתון שמשמש במודל שנוצר ב- models ובמידע שהותאם ב- data בכדי לאמן את המודל, לשמור תמונה של המבנה שלו, לשמור אותו ולהדפיס את התוצאות שלו לקובץ.

10. התייחסות אישית

עבודת הצוות הייתה טובה, דאגנו לדבר בינינו באופן יומי ולקיים מפגשים פנים מול פנים פעם בשבוע. החלקים הטובים בעבודת הצוות היו התקשורת בינו והתמיכה ההדדית. החלקים הקשים בעבודת הצוות היה לתאם את הזמנים בינו. בתחילה הם האטו את קצב העבודה אך לאחר זמן מסוים התרגלנו אליהם והם לא השפיעו יותר. נתקלנו באתגרים שנבעו מחוסר ידע בנושאים מסוימים כמו במערכת הפעלה או ב- deep learning כדוגמת בעיית הדיילי בחיזוי של המודל עליה הסברנו כבר. כדי לפתור את הבעיות הללו פנינו אל אנשים שיש להם ידע בנושאים הקשורים לבעיות וגם פנינו לאינטרנט. האתגרים שינו את חלוקת העבודה בספרינטים של הפרויקט ואת הזמן שלקח לסיים אותם. בתכנון שלנו מראש קיוונו שנוכל לנתח את כל 25 המחוות במודל שלנו אך לבסוף עשינו מודל שמנתח רק 7 מחוות. הפער נבע מחוסר הידע שלנו בנוגע לאימון של מודל deep learning. למדנו לעבוד עם services ב- windows, services הם process שרצים ברקע וניתן להפסיק או להפעיל את פעולתם אך הם תמיד מותקנים על המחשב. בתחילה חשבנו לכתוב את החלק שבגרסה הסופית הוא process בתור service אך לאחר הבעיה שהוסברה קודם לכן (ראה תיאור פרויקט) עברנו מ- service ל – process. נהיית לעבוד על חלק זה במיוחד מכיוון שתמיד עניין אותי לעבוד עם הממשק של windows ולהבין יותר לעומק את הפעולה של מערכת ההפעלה הזו. למדתי כיצד לעבוד על פרויקט גדול, לתכנן זמנים בצורה נכונה, ללמוד חומר חדש ולהוציא בסופו של דבר מוצר סופי. ישנם פתרונות נוספים באמצעותם אפשר לממש תוכנה דומה לפרויקט שלנו אך אני מאמין שהפתרון שלנו הוא הפתרון האידיאלי למימוש התוכנה מכיוון שאחוזי הדיוק והמהירות שבה הפרויקט שלנו עובד כמו כן חווית המשתמש הם אידיאליים בעיניי.

11. הצעות להמשך ושילוב הפרויקט בעולם האמיתי

ישנם מספר שיפורים שניתן לעשות בתוכנה:

1. אפשרות של עבודה עם עכבר ומקשים מיוחדים במקלדת כמו volume up. כרגע התוכנה עובדת על רוב מקשים במקלדת מלבד מספר מקשים מיוחדים לכן הרחבה זאת תשפר את חווית המשתמש ואת אופציות השימוש בתוכנה.
2. הוספת אפשרות של personalization על ידי הוספת תנועות - לתת את האופציה למשתמש להוסיף תנועות חדשות על ידי צילום של סרטונים ושליחה שלהם למודל כך שהוא ילמד אותם (חיבור של צד המפתח בתוכנה ל-GUI), על ידי כך הרשת גם תלמד את המראה של האדם ותעבוד באופן יותר מותאם אליו מאשר אל אנשים אחרים.

הפרויקט כתוב כך שהוא יתאים לעולם האמיתי ולכן אין צורך לבצע התאמות כדי להסב אותו מפרויקט לימודי לפרויקט שמתאים לעולם האמיתי.

12. הקדשה / תודה

ברצוננו להודות למדריכי התכנית ולתכנית מגשימים על שנתנו לנו את הכלים ואת הידע להתמודד עם פרויקט בהיקף שכזה.

במיוחד נרצה להודות לראש הצוות שלנו והמדריך של שנת הפרויקט שלנו במגשימים עודד ניר, על כך שתמך בנו, עזר לנו וחיפש לנו עזרה במקרים שבהם לא יכל לעזור ובאופן כללי שימש כמעין תחליף למנטור שלא היה לנו. כמו כן נרצה גם להודות לגיל שאליו עודד הפנה אותנו על העצה שלו שעזרה לנו לפתור את בעיית ה-latency ב-prediction.

בנוסף אנחנו מודים גם למירון קונרד ולשמעון שחר על התמיכה והעצות לאורך הדרך שגרמו לכך שהפרויקט יהיה כמו שהוא היום.

13. רשימת מקורות (ביבליוגרפיה)

- <https://medium.com/@muehler.v/simple-hand-gesture-recognition-using-opencv-and-javascript-eb3d6ced28a0>
זיהוי ידיים עם opencv
- <http://article.sapub.org/10.5923.j.ajis.20170703.11.html>
זיהוי ידיים באמצעות מצלמת מחשב
- <https://www.youtube.com/watch?reload=9&v=oH0ZkfFoeYU>
בפייתון זיהוי מחוות ידיים באמצעות מצלמת מחשב opencv
- <https://www.youtube.com/watch?v=TwYUm7i61JM>
שליטה בעכבר באמצעות מחוות ידיים בעזרת opencv
- <https://www.neurotechnology.com/npointer.html>
אתר של חברה המתמחה בנושא של הפרויקט
- <https://github.com/AdamCDunlap/webcam-gesture>
קישור לגיטהאב שיש בו זיהוי מחוות ידיים בעזרת מצלמת מחשב
- <http://www.techplusme.com/software/control-pc-hand-gesture>
אתר המסביר על שני תוכנות שעושות דבר דומה לפרויקט שלנו
- <https://www.kaggle.com/gti-upm/leapgestrecog>
קישור למאגר מידע לזיהוי ידיים המכיל עשרים אלף תמונות
- <https://www.kaggle.com/sujay626/gesture-datase>
מאגר מידע לזיהוי מחוות ידיים

- <https://www.kaggle.com/ashish8898/hand-gestures>
קישור לפרויקט של זיהוי ידיים
- <https://www.youtube.com/watch?v=VIPnqL5osYQ>
קישור לסרטון שמציג פרויקט דומה למה שאנו עושים
- <http://sun.aei.polsl.pl/~mkawulok/gestures/>
Database for hand gesture recognition
- <http://www.nlpr.ia.ac.cn/iva/yfzhang/datasets/egogesture.html>
מאגר נתונים של זיהוי ידיים
- <https://medium.com/twentybn/gesture-recognition-using-end-to-end-learning-from-a-large-video-database-2ecbf4659ff>
כתבה על פרויקט בנושא
- <https://www.kaggle.com/benenharrington/hand-gesture-recognition-database-with-cnn/notebook>
מאגר מידע של תנועות יד בתמונות infrared
- <https://www.kaggle.com/benenharrington/hand-gesture-recognition-database-with-cnn/notebook>
מאגר נתונים מחוות ידיים
- <https://www.kaggle.com/gti-upm/leaphandgestuav/home>
Hand gesture recognition dataset is presented, composed by a set of near infrared images and skeletal information acquired by the Leap Motion sensor.
- https://www.kaggle.com/destin369y/interactivehandgesture1/version/4#HandGesture1-6_ddd.zip
database of hand gesture recognition

- <https://zenodo.org/record/1320922#.W9cHbZMzbIV>
מאגר מידע המכיל בו מספר מחוות ידיים
- <https://www.kaggle.com/gti-upm/multimodhandgestrec>
multi modal dataset for hand gesture recognition
- <https://www.kaggle.com/gti-upm/depthgestrecog>
depth based gesture recognition database