הטכניון – מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל

Laboratory of Networked Software Systems

**בית חכם עם רשת חברתית**

אביב תשע"ח

מגיש: אלכסיי חרומוב מנחה: אורן קלינצקי

תוכן עניינים

[1. מבוא 3](#_Toc526863340)

[2. אתגרים בפיתוח אפליקצייה חכמה 4](#_Toc526863341)

[3. SDKs & APIs 5](#_Toc526863342)

[facebook-jssdk 5](#_Toc526863343)

[FB-RECOG 5](#_Toc526863344)

[BEAUTIFULSOUP 5](#_Toc526863345)

[FLASK 5](#_Toc526863346)

[OPENCV4 + NUMPY 6](#_Toc526863347)

[4. מבט על 7](#_Toc526863348)

[5. ארכיטקטורה ומימוש 8](#_Toc526863349)

[6. סיכום 13](#_Toc526863350)

[7. ביבליאוגרפיה 13](#_Toc526863351)

[8. נספחים 13](#_Toc526863352)

1. מבוא

העולם סביבנו כבר השתנה, הטכנולוגיה כבר נמצאת בבית של כל אחד ואחת מאיתנו, כולנו נושאים אותה בכיס, ואפילו עונדים אותה במקום שעון קוורץ. אז למה לא להכריח אותה להציג לנו מראש את מה שאנחנו אוהבים?

ניקח דוגמה: אדם קם בבוקר, רוצה לראות מה חדש בעולם החדשות, הספורט או הסדרות האהובות עליו, הוא כבר עשה "לייק" לעמודים האהובים שלו בפייסבוק, נשאר רק שהמחשב יציג לו את העדכונים החדשים מתוכם!

כאן הבית החכם נכנס לפעולה: נעמיד מחשב 3raspberry pi עם מצלמה ומסך מחוברים אליה, המצלמה תקלוט ותזהה את האדם העומד מולה, המסך יציג את התוכן האהוב עליו מאחד עמודי פייסבוק.

כעת נניח שאדם קורא תוכן והוא רוצה לגלול מטה או שהוא דווקא רוצה להחליף עמוד, למה "להתאמץ" בלחיצת כפתורי מקלדת \ עכבר, אם אפשר פשוט לעשות תנועות ידיים באוויר?

נכון! המערכת תזהה את תנועות הידיים ותפעל לפיהם.

1. אתגרים בפיתוח אפליקצייה חכמה

בפיתוח האפליקציה עמדו לפנינו אתגרים רבים. ראשית התממשקות עם SDKs שונים (יפורטו בהמשך), עמידה בדרישות אבטחה של פייסבוק ולבסוף שימוש במחשב raspberry pi3 לתפעול האפליקציה.

* האתגר הכי גדול בפיתוח התגלה דווקא לאחר ניסוח הפרוייקט:

מתוך ויקיפדיה:

**קיימברידג' אנליטיקה** (ב[אנגלית](https://he.wikipedia.org/wiki/%D7%90%D7%A0%D7%92%D7%9C%D7%99%D7%AA" \o "אנגלית): **Cambridge Analytica**) הייתה [חברה פרטית](https://he.wikipedia.org/wiki/%D7%97%D7%91%D7%A8%D7%94_%D7%A4%D7%A8%D7%98%D7%99%D7%AA) [בריטית](https://he.wikipedia.org/wiki/%D7%91%D7%A8%D7%99%D7%98%D7%A0%D7%99%D7%94) ששילבה [כריית נתונים](https://he.wikipedia.org/wiki/%D7%9B%D7%A8%D7%99%D7%99%D7%AA_%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D), מסחר בנתונים, [ניתוח מידע](https://he.wikipedia.org/wiki/%D7%A0%D7%99%D7%AA%D7%95%D7%97_%D7%9E%D7%99%D7%93%D7%A2) ואסטרטגיה תקשורתית לצורך השפעה על תהליכי [בחירות](https://he.wikipedia.org/wiki/%D7%91%D7%97%D7%99%D7%A8%D7%95%D7%AA) במדינות ברחבי העולם[...] בשנת [2016](https://he.wikipedia.org/wiki/2016) הייתה החברה מעורבת במסע [הבחירות לנשיאות ארצות הברית](https://he.wikipedia.org/wiki/%D7%94%D7%91%D7%97%D7%99%D7%A8%D7%95%D7%AA_%D7%9C%D7%A0%D7%A9%D7%99%D7%90%D7%95%D7%AA_%D7%90%D7%A8%D7%A6%D7%95%D7%AA_%D7%94%D7%91%D7%A8%D7%99%D7%AA_2016) של [דונלד טראמפ](https://he.wikipedia.org/wiki/%D7%93%D7%95%D7%A0%D7%9C%D7%93_%D7%98%D7%A8%D7%90%D7%9E%D7%A4) ובקמפיין ה[ברקזיט](https://he.wikipedia.org/wiki/%D7%91%D7%A8%D7%A7%D7%96%D7%99%D7%98) ליציאת [בריטניה](https://he.wikipedia.org/wiki/%D7%91%D7%A8%D7%99%D7%98%D7%A0%D7%99%D7%94) מ[האיחוד האירופי](https://he.wikipedia.org/wiki/%D7%94%D7%90%D7%99%D7%97%D7%95%D7%93_%D7%94%D7%90%D7%99%D7%A8%D7%95%D7%A4%D7%99" \o "האיחוד האירופי). תפקודה של קיימברידג' אנליטיקה בקמפיינים הללו הוא נושא לחקירות פליליות מתמשכות בשתי המדינות, בעיקר בנוגע לשיטות בהן היא נוקטת לצורך מיקוד בוחרים.]..] ב-[17 במרץ](https://he.wikipedia.org/wiki/17_%D7%91%D7%9E%D7%A8%D7%A5) [2018](https://he.wikipedia.org/wiki/2018), דיווחו העיתונים "[ניו יורק טיימס](https://he.wikipedia.org/wiki/%D7%A0%D7%99%D7%95_%D7%99%D7%95%D7%A8%D7%A7_%D7%98%D7%99%D7%99%D7%9E%D7%A1)" ו"[האובזרבר](https://he.wikipedia.org/wiki/%D7%94%D7%90%D7%95%D7%91%D7%96%D7%A8%D7%91%D7%A8)" שקיימברידג' אנליטיקה עשתה שימוש עסקי במידע אישי מ[פייסבוק](https://he.wikipedia.org/wiki/%D7%A4%D7%99%D7%99%D7%A1%D7%91%D7%95%D7%A7), שמלכתחילה נאסף על ידי חוקר חיצוני למטרות אקדמיות. בתגובה, אסרה פייסבוק על קיימברידג' אנליטיקה לפרסם בפייסבוק.

האירועים הנ"ל רגמו לפייסבוק לבצע שינוי משמעותי באבטחה. החל מאפריל לא ניתן לתת הרשאות publish\_actions לאפליקציות – הרשאה שהיה בה צורך להעלאת תמונה לקיר של המשתמש הראשי – כלומר הבית.

* אתגר נוסף שעמד לפנינו היה שליפת תמונה מתוך streaming של מצלמה והעברתה לשרת לצורך זיהוי. פתרון שלבסוף לא נלקח משום שלא היה יעיל: לעשות THREAD בצד שרת של המצמלה, ופעם בחצי שנייה לעצוןר אותו, לעשות שליפת תמונה ואז להמשיך את הSTREAMING. הפתרון כמובן לא טוב, אין סיבה לעצור את הזרמת המצלמה בשביל לקחת תמונה, זה עלול לגרום לתקיעות ולפעולות מיותרות של התוכנה.

הפתרון שכן נלקח לבסוף הוא שימוש במצלמה מצד הלקוח, שירות הSTREMING מבוצע ישירות בצד לקוח ע"י JAVASCRIPT, פעם בחמי שנייה הסקריפט ייקח SCREENSHOT מתוך המצלמה, בלי לעצור את הזרמת הנתונים למסך, ויישלח את הלינק הזמני של התמונה לצד שרת. השרת ייגש ללינק הזמני, ישמור את התמונה ולאחר מכן יעבד אותה – בין אם מדובר בזיהוי פנים ובין אם מדובר בזיהוי תנועות.

* האתגר האחרון שעמד לפנינו היה זיהוי תנועות ידיים. על מנת לזהות את התנועות השתמשנו באלגוריתם הבא: לקחנו 2 תמונות עוקבות בהפרש של חצי שנייה ועשינו על שתיהן השוואה של תנועה. את הפירוט של האלגוריתם ניתן למצוא בפרק "ארכיטקטורה ומימוש".

1. SDKs & APIs

# facebook-jssdk

רץ בצד לקוח ומתממשק עם javascript.

SDK שאנחנו מקבלים מפייסבוק, אחראי על יצירת חלון עם עמוד פייסבוק באפליקציה שלנו, אשר מכיל התוכן שאנחנו מבקשים, במקרה זה – אחד העמודים האהובים על המשתמש הנוכחי. נשים לב שהעמוד חייב להיות PUBLIC, מכיוון שאנחנו ניגשים אליו דרך המשתמש הראשי, שייתכן ולא עשה לו לייק.

# FB-RECOG

להשלים

# BEAUTIFULSOUP

רץ בצד שרת וכתוב בפייתון. הSDK אחראי על PARSING של עמוד HTML שאנחנו נותנים לו (עפ"י URL). במקרה שלנו, אנחנו מזינים לאובייקט של BEAUTIFULSOUP כתובת של עמוד הלייקים של המשתמש הנוכחי, האובייקט ניגש לעמוד, ומפרסר אותו. באחריותנו לקחת את הקובץ XML שיצר הפרסר, ולמצוא בתוכו את האובייקט (CLASS) אשר מכיל בתוכו לינק לעמוד אהוב.

נמצא את כל העמודים האהובים של המשתמש, ונחזיר אותם כמילון של שם + לינק למי שקרא לSDK.

הערה חשובה: התוכן שמוחזר מדף HTML מוחזר בתוך הערה, טרם פרסור הHTML, עלינו להוריד את סימני ההערה, על מנת שהBEAUTIFULSOUP יוכל למצוא את תגיות XML הדרושות.

# FLASK

API בפייתון שאחראי על הרצת אתר במתודולוגיה הסטנדרטית של שרת-לקוח. במקרה שלנו השרת הינו מחשב וירטואלי שאנו מפעילים טרם הרצת האפליקציה.

הAPI הוא זה שמאפשר לנו ליצור תקשורת בין הלקוח והשרת, ע"י פנייה של הלקוח ללינקים מוגדרים מראש. לדוגמה פנייה של הלקוח לכתובת: /my\_link"" תגרום לקריאה הבאה של הפונקצייה בצד שרת:

@app.route(**"/my\_link"**, methods=[**'GET'**,**'POST'**])  
**def** getHandGesture():

…

ניתן להעביר מידע מלקוח לשרת בעזרת METHOD: 'POST', השרת יקרא את המידע בעזרת גישה לארגומנטים, לדוגמה:

path = request.args.get(**'image\_src'**)

התקשורת בין הצדדים היא בעזרת סטנדרט JSON (ראה נספח). מקור לFLASK:

http://flask.pocoo.org/docs/1.0/

# OPENCV4 + NUMPY

ספריות של PYTHON אשר אחראיות על עיבוד תמונה, בעזרת ספריות אנו מפענחים את תנועות הידיים של המשתמש. לאלגוריתם של הםענוח, ראה פרק "ארכיטקטורה ומימוש".

1. מבט על

האפליקציה עובדת בצורת תכן סטנדרטית של צד לקוח וצד שרת:

Response uploaded picture ID  
+ Recognized face

Ask for picture upload   
+ face recognition

Non-stop streaming from camera to client-side

צד שרת

צד לקוח

Facebook



כל חצי שנייה: צלם תמונה ושלח בקשה לצד שרת.  
אם שנייה עגולה שלח בקשה לזיהוי פנים, אחרת שלח בקשה לזיהוי תנועה.  
פעל לפי תשובת השרת שהתקבלה: דפדוף אנכי, שינוי עמוד או שינוי זיהוי.

אם הגיע בקשת זיהוי פנים: העלאה את התמונה הנוכחית לפייסבוק, ובקש זיהוי.  
אם הגיע בקשת זיהוי תנועה, השווה את 2 התמונות האחרונות. בכל מקרה שלח תשובה ללקוח.

1. ארכיטקטורה ומימוש

צד לקוח:

כפי שנאמר תוכנית הMAIN שלנו היינה לולאה אינסופית, אשר נקראת כל חצי שנייה, היא נראית כך:

**var *interval*** = 500; *// miliseconds***var *cycles\_till\_recognition*** = 2000 / ***interval***; *// we want recognition once in 2 seconds***var *calls\_counter*** = 0;  
**var *first\_cycles\_run\_counter*** = 0;  
  
**function** *run\_with\_timer*() {  
  
 **if** (***first\_cycles\_run\_counter*** < 5) { *// async - takes time for first picture to load* ***first\_cycles\_run\_counter***++;  
 *get\_screenshot*();  
 } **else** {  
 *get\_screenshot*();  
 **if** (***calls\_counter*** === 0) {  
 *get\_person\_name*();  
 } **else** {  
 *get\_gesture*();  
 }  
 ***calls\_counter*** = (***calls\_counter*** + 1) % ***cycles\_till\_recognition***;  
 }  
  
 *setTimeout*(*run\_with\_timer*, ***interval***);  
}  
  
$(***document***).**ready**(*run\_with\_timer*());

לאחר האתחול הראשוני, אשר לוקח כמה מחזורי קריאה מכיוון שהוא אסינכרוני, הפונקצייה נכנסת למעגל של לקיחת תמונה ושליחתה לשרת. כל 2 שניות תתבצע בקשה לזיהוי פנים, ובכל פעם אחרת (סה"כ 3 פעמים ב2 שניות) תהיה בקשה לזיהוי תנועה.

עבור מימוש כל אחת מהבקשות לFLASK, עם JSON QUERY נשתמש במבנה הבא:

$.getJSON(  
 {  
 **url**: **<url which describes the wanted request>,**  
 **data**: {**<url to image if required>** },  
 success: **function** (result) {  
 **<commands in case of successful request>**

}  
 }  
 });

הפונקציות הבאות ישתמשו במבנה זה:

**function** *get\_next\_page*(direction)

* הפונקציה נקראת כאשר המשתמש הנוכחי דפדף ימינה או שמאלה, הצד-לקוח יבקש מהשרת את העמוד הבא (או הקודם) מרשימת העמודים האהובים על המשתמש.

**function** *get\_person\_name*()

* הפונקציה נקראת כל שניה עגולה, ומעבירה לצד שרת את התמונה הנוכחית במצלמה, וזאת עבור זיהוי פנים של המשתמש.

**function** *get\_gesture*()

* הפונקציה נקראת כל חצי שניה מהצד-לקוח לצד שרת, ומבקשת ממנו זיהוי תנועת יד לפי 2 תמונות אחרונות שצולמו.

פונקציית הזרמת תמונה ללא הפסקה (STREAMING):

***navigator***.**mediaDevices**.getUserMedia(***constraints***).then(*handleSuccess*).catch(*handleError*);  
**function** *handleSuccess*(stream) {  
 ***video***.**srcObject** = stream;  
}  
**function** *handleError*(error) {  
 ***console***.error(**'Error: '**, error);  
}

פונקציית לקיחת SCREENSHOT רגעי מהמצלמה:

**function** *get\_screenshot*()  
{  
 ***canvas***.**width** = ***video***.**videoWidth**;  
 ***canvas***.**height** = ***video***.**videoHeight**;  
 ***canvas***.getContext(**'2d'**).drawImage(***video***, 0, 0);  
 ***img***.**src** = ***canvas***.toDataURL(**'image/png'**);  
}

צד שרת:

צד שרת נכתב בפייתון, מתוך בחירה שרירותית.

השרת מכיל פונקציות תגובה לבקשות URL של FLASK API, לדוגמה:

@app.route(**"/getPrevFrame"**, methods=[**'GET'**])  
**def** getPrevFrame():  
 **return** \_get\_liked\_page(likes\_list\_index-1)

ונשים לב שניתן להעביר מידע מהלקוח לשרת בצורה הבאה:

path = request.args.get(**'image\_src'**) *# image link passed from javascript*

פונקציות עזר:

**def** \_login(session, email, password):

**def** \_get\_liked\_page(next\_index):

**def** \_create\_likes\_list():

**def** \_get\_hand\_gesture(img\_name\_first, img\_name\_second):

**def** \_compare\_images\_for\_gesture(first\_avg, second\_avg):

וכמובן פונקציית MAIN שמריצה את שרת הFLASK הוירטואלי (app.run(debug=**True**)).

זיהוי בן אדם:

להשלים

זיהוי עמודים מועדפים והעברתם לשרת:

זיהוי עמודים מועדפים של החבר יתבצע בעזרת הפונקצייה: \_create\_likes\_list. הפונקצייה מכירה את האדם שכרגע מול האפליקציה (הוא כבר זוהה), ולכן היא יכולה להתחבר כמשתמש הראשי לפייסבוק, ולהגיע דרך URL לעמודים הלייקים של המשתמש הנוכחי (בעזרת \_login):

likes\_url = **'https://www.facebook.com/'** + dotted\_name\_of\_curr\_friend +**'/likes?lst=1198688678%3A843054236%3A1535896155'**response = session.get(likes\_url, cookies=cookies, allow\_redirects=**False**)

מכאן נשתמש בSDK של BEAUTIFUSOUP שעליו הסברנו קודם, על מנת להוציא מעמוד הלייקים את השמות והלינקים של העמודים האהובים, ולעשות מהם רשימה.

כל פעם שהצד לקוח יבקש לינק לעמוד הבא או הקודם מרשימת הלייקים, העמוד המבוקש יחזור אליו מהרשימה הנ"ל, בעזרת הפונקציה: \_get\_liked\_page.

זיהוי תנועות ידיים:

עלינו לזהות 4 תנועות ידיים: למעלה, למטה, משיכה לימין ומשיכה לשמאל. הדבר מתבצע בעזרת הפונקצייה: \_get\_hand\_gesture, אשר מקבלת 2 תמונות, מפעילה על כל אחת מהן את האלגוריתם הבא, ולאחר מכן שולחת את התוצאות ל: \_compare\_images\_for\_gesture.

האלגוריתם:

* תמונה מקורית:



* הפוך את התמונה לגווני שחור – לבן:



* טשטש את התמונה מעט על מנת להמנע מרעשים:



* קיטוב תבעי התמונה לשחור אם הצבע בין 0 ל100, וללבן אם הצבע בין 101-255:



* וכעת ניתן לחשב את ממוצע הצבע הלבן בתמונה, למשל: [285, 261].
* \_compare\_images\_for\_gesture מקבלת 2 ממוצעים של צבעים לבנים ב2 תמונות. נזכור שהתמונות הן במרווח זמן של חצי שנייה. נוכל לראות לאיזה כיוון זז ממוצע הצבע הלבן בתמונה השנייה. לפי כיוון התזוזה המקסימלית נדע אם תנועת היד הייתה בכיוון מעלה \ מטה \ ימינה \ שמאלה.

1. סיכום

ניתן לראות שהתממשקות עם פייסבוק היא אפשרית, עם עמידה באתגרי אבטחה. זיהוי פנים ותנועות ידיים מבוצע בצורה טובה אפילו עם מצלמה פשוטה יחסית.

המסקנה היא שניתן ליצור בית חכם ונוח לשימוש כפי שהוגדר ביעדים.

1. ביבליאוגרפיה
2. נספחים

JSON:

מתוך ויקיפדיה: <https://he.wikipedia.org/wiki/JSON>

**JSON** ראשי תיבות של **J**ava**S**cript **O**bject **N**otation) ) הוא [פורמט](https://he.wikipedia.org/wiki/%D7%A4%D7%95%D7%A8%D7%9E%D7%98_%D7%A7%D7%95%D7%91%D7%A5) טקסטואלי, הקריא לאדם, המיועד להעברת מבני מידע המורכבים מזוגות של מפתח-ערך. השימוש העיקרי של הפורמט הוא להעברת מידע בין שרת לצרכן כתחליף לפורמט XML.