

# רנ-קו אונליין PROFILE



מכללה: סמינר בנות אלישבע  
שם סטודנט: תמר גיגי  
ת"ז סטודנט: 324817162  
שם המנחה: גב' שולמית ברלין  
תאריך הגשה: כ' סיוון

## הוראות לכתיבת ספר פרויקט

- הצעת פרויקט
- מבוא
  - הרקע לפרויקט
  - תהליך המחקר
  - סקירת ספרות
- מטרות ויעדים
  - מטרות
  - יעדים
- אתגרים
- מדדי הצלחה למערכת
- תיאור המצב הקיים
- רקע תיאורטי
- ניתוח חלופות מערכתי
- תיאור החלופה הנבחרת
- אפיון המערכת
  - דרישות המערכת
  - מודול המערכת -תהליכים מרכזיים
  - אפיון פונקציונלי וביצועים עיקריים
  - אילוצים
- תיאור הארכיטקטורה
  - תיאור שלבי העבודה
  - תיאור רכיבים
  - ארכיטקטורת רשת
  - תיאור פרוטוקולי רשת
  - שרת – לקוח
  - תיאור הצפנות
- ניתוח ותרשים UML/Use cases של המערכת המוצעת
  - תרשים UML
  - רשימת useCase
  - תיאור useCasen
  - מבנה הנתונים בהם השתמשתי בפרויקט
  - עץ מודולים ותרשים מחלקות
  - תיאור המחלקות המוצעות

- תיאור התוכנה
  - סביבת פיתוח
  - שפות תוכנה
- אלגוריתמים מרכזיים
- קוד התוכנית – על פי סטנדרטים בליווי תיעוד
  - Server
  - Client
- תיאור המסד נתונים
- תיאור מסכים
  - תפקידי המסכים
  - תיאור מסך פתיחה
  - כל מסכי המערכת בליווי הסברים (כולל הספרים על הכפתורים)
  - הודעות למשתמש
- מדריך למשתמש
- בדיקות והערכה
- ניתוח יעילות
- אבטחת מידע
- מסקנות
- פיתוחים עתידיים
- ביבליוגרפיה

## הצעת פרויקט – יד הנדסת תוכנה

סמל מוסד: 189084

שם מכללה: בנות אלישבע

שם הסטודנט: תמר גיגי

ת"ז הסטודנט: 324817162

שם הפרויקט: שינוי פרופיל נוסע ברב קו מפרופיל נוער לפרופיל סטודנט

### תיאור הפרויקט

אפליקציה המקבלת מהמשתמש מסמכים מתאימים הרלוונטיים עבור שינוי פרופיל ברב-קו. התוכנה תבדוק אם המשתמש רשאי לשנות פרופיל לפי בדיקות של נתונים מסוימים מתוך המסמכים שהעלה, כגון זיהוי שם זהה בכל המסמכים, תאריך תקין ועוד, במידה והנתונים נכונים המשתמש יוכל להמשיך בהטענת הפרופיל, במידה ולא, תוצג לו הודעה מתאימה.

### הגדרת הבעיה האלגוריתמית

זיהוי אותיות ומספרים בעזרת מודל מתחום הבינה המלאכותית

זיהוי נתונים משותפים לכל המסמכים

זיהוי נכונות נתונים

### רקע תיאורטי בתחום הפרויקט

משתמשים רבים צריכים לשנות את פרופיל הנוסע שלהם ברב קו.

כיום הפעולה נעשית בדרך הבאה. ניתן להגיש את המסמכים באתר החברה. המסמכים נבדקים על ידי פקיידים וכשנמצאו תקינים הפקיד משנה את הפרופיל. מסיבה זו יש להמתין זמן מה עד לקבלת תשובה.

האפליקציה שאכתוב תזהה ותבדוק את הנתונים באופן אוטומטי ותאשר לשנות את הפרופיל מה שיחסוך בזמן המתנה ובכוח עבודה.

### תהליכים עיקריים בפרויקט

- קבלת המסמכים מהמשתמש
- זיהוי הנתונים וההשוואות ביניהם
- תגובה למשתמש-אישור או שגיאה

### תיאור הטכנולוגיה

צד שרת: Python

צד לקוח: React

### מסד נתונים

מאגר תמונות של אותיות ומספרים

### פרוטוקולי תקשורת

http

## לוחות זמנים

- אוקטובר - תכנון הפרויקט
  - נובמבר - לימוד האלגוריתם
  - דצמבר - כתיבת האלגוריתם
  - ינואר - ניסוי ובדיקות
  - פברואר-מרץ - בניית ממשק ופונקציות נוספות
  - אפריל - ספר פרויקט
  - מאי- הגשת הפרויקט
- חתימת הסטודנט: תמר גיגי
- חתימת רכז המגמה:
- אישור משרד החינוך:

## מבוא

### הרקע לפרויקט

בשנים האחרונות התפתחה שיטת הבינה המלאכותית בעולם התכנות, וכשחשבתי על רעיון לפרויקט היה ברור לי שאני רוצה להתמחות בנושא הזה שכולל בתוכו אלגוריתם מורכב וגם טכנולוגיה חדשה ומרתקת.

התלבטתי באיזה תחום לממש את הבחירה שלי, ואז שמעתי מהרבה סטודנטים שמתלוננים על עיכוב בשימוש ברב-קו, מאחר ובכל שנה הם צריכים לשנות את הפרופיל מחדש לפרופיל סטודנט ותהליך זה אורך זמן רב, בכדי להחליף לפרופיל סטודנט נדרש להביא את המסמכים הבאים: כרטיס סטודנט, אישור לימודים, תעודת זהות וכרטיס רב-קו בשביל שידקו את זכאותם לשינוי הפרופיל. כרגע ישנם שני דרכים לעשות זאת: האחד על ידי הגעה למרכז שרות, השני דרך האתר או האפליקציה.

דרך המרכז שרות יש לחכות זמן רב בתור לנציג שיוכל לעבור על טפסים ולשנות להם את הפרופיל, ודרך האתר או אפליקציית רב-קו אונליין יש להמתין זמן רב לאישור מהחברה, מאחר וגם שם נציג עובר על הקבצים, וכל זה בזמן שפעולה זו יכולה להיות קלה ומהירה. אז חשבתי לפתח תוכנה שתפענח את הקבצים הנדרשים ותדע להחליט אם הקבצים חוקיים וניתן לאשרם.

הלקוח הוא כל סטודנט שצריך להגדיר את הפרופיל שלו כסטודנט. הלקוח יסרוק או יצלם את תעודת הזהות שלו, כרטיס סטודנט בתוקף, אישור לימודים ורב-קו, המערכת תעשה סריקה על הקבצים ותבדוק את תוכנם באמצעות עיבוד התמונה ושליחה לרשת ניוירונים שמפענחת את תוכן הקבצים, במידה והקבצים שצורפו נכונים הלקוח יוכל להמשיך בתהליך, ובמידה ולא הלקוח יקבל על כך תשובה ויצטרך לשלוח מחדש את הקבצים הנכונים.

### תהליך המחקר

הבעיה המרכזית בפרויקט היא זיהוי תוכן המסמכים המצורפים לבקשה לשינוי פרופיל. לצורך זה יש לזהות את הכתב המופיע בטפסים ולהבין את תוכנו.

ניתן היה לפתור את הבעיה על ידי שימוש בספריות של עיבוד תמונה הקיימות בפיתון, שבאופן אוטומטי מעבדות תמונה ומוציאות מתוכה את הכתוב – אפשרות זו נפסלה משום שרציתי לדעת לעומק את עניין הבינה המלאכותית וכיצד עובד אימון רשת ניוירונים ולא להשתמש בספריות מוכנות שמסתירות את הקוד העומד מאחוריו.

בסופו של דבר החלטתי לכתוב את הקוד כך, בתחילה לשלוח לפונקציה שמוצאת את מיקומי המילים בתמונה, לאחר מכן לשלוח את כל רשימת המיקומים לפונקציה שחותכת אותם מהתמונה, ולבסוף לחתוך כל מילה לאותיות, כאשר בכל תמונה יש אות אחת בלבד, ואז לשלוח

כל אות לרשת נוירונים (שאימנתי בעצמי) המזהה אותה. אחר כך צריך לצרף את האותיות למילים ולשורות ולבדוק שהתוכן הכתוב חוקי. כמו כן, היה ניתן לכתוב את הקוד במגוון שפות כמו ++c, java או #c אך אני בחרתי את פיתון מכיוון שהיא השפה הכי מפותחת בנושא וגם כי היא נכנסה מאד לשוק העבודה ורציתי להתמקצע בה ולהכיר אותה יותר לעומק.

## סקירת ספרות

- <https://github.com/> GitHub
- [/https://stackoverflow.com](https://stackoverflow.com) Stack Overflow
- <https://reshetech.co.il/machine-learning-tutorials/all-the-tutorials> לימוד למידת מכונה
- [/https://opencv.org](https://opencv.org) ספרית OpenCV
- ספרית NumPy
- <https://numpy.org/doc/stable/reference/generated/numpy.ndarray.tolist.html>
- [/https://keras.io](https://keras.io) ספרית Keras
- [/https://www.tensorflow.org](https://www.tensorflow.org) ספרית TensorFlow
- <https://pyimagesearch.com/2020/08/24/ocr-handwritinrecognition-with-opencv-keras-and-tensorflow> עיבוד תמונה
- <https://reshetech.co.il> רשת-טק
- [/https://programmingwithmosh.com](https://programmingwithmosh.com) חיבור Python ל React
- [/https://www.geeksforgeeks.org](https://www.geeksforgeeks.org) Geeks For Geeks
- <https://he.reactjs.org/tutorial/tutorial.html> מדריך ב React
- [/https://nanonets.com/blog/ocr-with-tesseract](https://nanonets.com/blog/ocr-with-tesseract) ספרית pytesseract
- [/https://pillow.readthedocs.io/en/stable](https://pillow.readthedocs.io/en/stable) ספרית Pillow

## מטרות ויעדים

### מטרות

- בדיקת תקינות הטפסים בצורה קלה ומהירה
- החזרת תגובה ללקוח בצורה המהירה ביותר
- התנסות בפיתוח והכרת ספריות משמעותיות בצורה יסודית יותר
- למידת הנושא Machine Learning מתוך נושא Artificial intelligence לעומק ויצירת מודל של רשת נוירונים
- לשפר וליעל את היכולת של הלימוד העצמי בתחומים שונים

### יעדים

- תכנון המערכת בצורה טובה וכתיבת אלגוריתם יעיל ומסודר
- איסוף תמונות רבות של אותיות ומספרים על מנת לאמן רשת נוירונים אמינה ביותר
- אימון מספר רשתות נוירונים ובחירת הרשת הנותנת את התוצאות המדויקות ביותר
- בנית אתר נעים לעין, שנותן למשתמש תוצאות מהירות
- במקרה שהקבצים לא חוקיים החזרת הסבר על הקבצים שלא היו תקינים

## אתגרים

בפיתוח המערכת ניצבו בפני אתגרים רבים:

ראשית כל היה עלי ללמוד את כל הנושא של למידת מכונה, לדעת לאמן את המחשב להבין דברים שמוח אנושי מבין אך אינו יודע לתת לכך כללים ברורים.

האתגר הגדול הוא אימון רשת הנוירונים ולהצליח להגיע לרשת נוירונים טובה שמביאה את התוצאה הנכונה ביותר. לשם כך היה עלי לבנות את שכבות הרשת בצורה יעילה.

אתגר נוסף הוא בנית Data Base ובו כמות גדולה של תמונות של אותיות ומספרים, על מנת לגרום לרשת להיות מדויקת כמה שיותר. לצורך כך יצרתי קובץ עם הרבה מאוד סוגים של פונטים שלבסוף הימרתי אותם לקובצי png, וחתכתי מתוכם את האותיות. לאחר מכן הפעלתי על מאגר התמונות פונקציה נוספת המעבה את התמונות, עד שלבסוף היו בידי למעלה מ 50,000 תמונות!!!

אתגר נוסף שהיה לי הוא עיבוד התמונה ומציאת מיקום האותיות על מנת לחתוך אותם ולשלוח לרשת לזיהוי.

בנוסף לכל זה כתיבת הקוד בפיתוח שהצריך לימוד עצמי הייתה אתגר לא קטן. לכאורה שפת האם שלי היא ++C, אני מכירה לעמוק. וכמובן שהיה לי יותר נוח לכתוב בה את הפרויקט



, אך דווקא בגלל זה העדפתי להתמקצע בצורה יסודית פיתון כדי לרכוש בצורה יסודית שפה נוספת שפופולארית היום בשוק.

### מדדי הצלחה למערכת

המערכת שלי מזה כ 99.98% אך אם המערכת תצליח לזהות ולהשוות נכון את הנתונים בדיוק של 90%+ מהקבצים זה יהווה הצלחה של המערכת. זיהוי של פחות מכך יורה על חוסר הצלחה.

### תיאור המצב קיים

מבדיקה שערכתי קיימת אפשרות באפליקציה ובאתר לשנות את הפרופיל, אולם בשלב של שליחת הטפסים לבדיקה, הבדיקה נעשית על ידי נציגים שעוברים על הטפסים ונותנים את האישור.

### רקע תיאורטי

זיהוי האותיות :

ניתן לזהות אותיות בתמונה באמצעות אלגוריתם של עיבוד תמונה. באופן זה הופכים קודם את התמונה לשחור- לבן, ממירים אותה למטריצת פיקסלים ואז ניתן לבדוק אם קיימות צורות של אותיות או ספרות במטריצה ע"פ הפיקסלים הצבועים בשחור.

הבעיה בשיטה הזאת היא החוסר דיוק , אם האות קצת מטושטשת או חתוכה בקצוות העיבוד לא יהיה אמין ולכן לא נוכל להסתמך על זה, לכן בחרתי בבינה מלאכותית שמניבה תוצאות נכונות גם במקרים אלו.

בחירת רשת הניורונים המתאימה

ניתן להשתמש בסוגים שונים של רשתות ניורונים , חלקם אפילו מובנים בשפה (למשל), אך אני בחרתי ברשת שאימנתי בעצמי מהסיבה שהיא מביאה לדיוק מירבי וכן כי רציתי ללמוד לפתח רשת ניורונים.

### ניתוח חלופות מערכתי

ישנם אפשרויות נוספות לקיצור זמן ההמתנה לאישור הטפסים ע"י הוספת נציגי שרות במשך כל שעות היום לקיצור התורים ולזמינות במשך כל היום.

### תיאור החלופה הנבחרת

הדרך שבחרתי בה היא בנית אתר שהמשתמש מעלה בו את הקבצים לבדיקה, קבצים אלו נשלחים לשרת, ושם לאחר עיבוד התמונה ושליחת האותיות לרשת ניורונים הנתונים ישלחו לפונקציית חיפוש והשוואות הנתונים ותבדוק את נכונותם- תבדוק שהטפסים שהועלו באמת

מתאימים (אישור לימודים זה אישור לימודים, תעודת זהות זה תעודת זהות וכו'), מספר הזהות והשם זהים בכל הקבצים, תאריך של השנה הנוכחית באישור לימודים. במידה והקבצים תקינים והנתונים נכונים יחזור למשתמש האישור והוא יוכל להמשיך בפעולה, במידה ולא, תחזור הודעה על כך למשתמש והוא יצטרך לשלוח מחדש את הקבצים הנכונים.

## אפיון המערכת

### דרישות המערכת

- כתיבה בסטנדרטים גבוהים ומקצועיים, באופן מסודר תוך הקפדה על תיעוד בכל שלב
- מעל 99% של זיהוי האותיות והספרות
- בניית ממשק נעים למשתמש
- כתיבת אלגוריתם יעיל ככל האפשר.

### מודול המערכת -תהליכים מרכזיים

- חלק ראשון- הכנת הנתונים (התמונות) לרשת הניורונים:
  - איסוף תמונות רבות של אותיות – על ידי חיתוך אותיות מהתמונה
  - הפיכתם לצבעי שחור- לבן
  - שינוי גודל התמונה על מנת שיוכלו לעבד אותה ולזהות את הכתוב בה בצורה הטובה ביותר
  - עיבוי מאגר התמונות על ידי פונקציה של עיבוד תמונה
  - חלוקת התמונות לשלושה תיקיות validate, train, test: באופן רנדומלי.
  - כאשר בכל תיקיה ישנם 42 תיקיות שכל אחת מהם מכילה תמונות של אות, תו או ספרה אחת.
- חלק שני – אימון הרשת:
  - בניית הרשת ואימונה על התמונות שנאספו ועובדו
  - בדיקת אמינות הרשת
- חלק שלישי- תהליך הפרויקט ופעולותיו בצד השרת:
  - קבלת התמונות מהclient-
  - הפיכת התמונה לשחור לבן ושינוי הגודל התמונה
  - חיתוך התמונות לפי שדות
  - חיתוך השדות למילים
  - חיתוך המילים לאותיות
  - שליחת כל אות לזיהוי ברשת ניורונים
  - הכנסת הנתונים שזוהו לאובייקט
  - השואות נתונים בין האובייקטים שהתקבלו מהתמונות שנשלחו למודל
  - החזרת אישור ללקוח(במקרה של שגיאה- מחזיר את הקבצים הלא חוקיים)
- חלק רביעי- תהליך הפרויקט ופעולותיו בצד הלקוח

- כניסת המשתמש לאתר ובחירת הפעולה הרצויה, העלאת תמונות של ת.ז. , כרטיס סטודנט, אישור לימודים, רב-קו מהמחשב (או פתיחת מצלמה כדי לצלם את המסמכים הנדרשים)
- שליחת התמונות לשרת
- קבלת אישור מהשרת (במקרה של שגיאה- הצגה אילו מסמכים אינם חוקיים ועצירה בשלב הזה)
- המשכת תהליך שינוי פרופיל

### אפיון פונקציונלי וביצועים עיקריים

- חלק ראשון – הכנת הנתונים ללמידה
  - blackAndWhite - פונקציה המשנה את התמונה לשחור לבן
  - cropImageRow – פונקציה שחותכת את התמונה לשורות (לפי השורות של האותיות)
  - cropLetters – פונקציה שחותכת את השורות לאותיות
  - resizeWithWhiteBackground – פונקציה המשנה את גודל התמונה (אות חתוכה) ל 32\*32 פיקסלים
  - EnLargelImage – פונקציה שמעבה את מאגר התמונות פי 9
  - saveTrainTestValidate – פונקציה שמחלקת את מאגר התמונות ל train , validate ו test כאשר כל אחת מכילה 37 תיקיות של הקטגוריות
- חלק שני – בניית הרשת
  - load\_dataset- פונקציה שטוענת את התמונות המחולקות ל train test בשביל המודל
  - prep\_pixels – פונקציה שמנרמלת את train וה-test
  - define\_model – פונקציה שמרכיבה את השכבות של המודל
  - run\_test\_harness- פונקציה שמזמנת את כל הפונקציות הנ"ל ומאמנת את המודל ומביאה את אחוז הדיוק
- חלק שלישי – תהליך הפרויקט בצד שרת
  - uploadImage – פונקציה המקבלת את הקבצים מצד לקוח
  - strRun – פונקציה שמזמנת את כל הפונקציות של העיבוד תמונה כולל החיתוך ושליחת התמונות למודל רשת הניורונים. ולאחר מכן החזרת מערך של מערכים כך שכל מערך פנימי מכיל את כל המילים שזוהו מהתמונה.
  - searchAndAnswer – פונקציה המקבלת את המערך שהתקבל מהתמונות ומשווה את הנתונים

- חלק רביעי – תהליך הפרויקט בצד לקוח
  - imageUploader – פונקציה שמעלה את התמונות שהמשתמש בחר ושליחתם לשרת
  - trueResponse – פונקציה שממשיכה בבחירת התהליך

#### אילוצים

- זיהוי מדויק של הנתונים מתוך הקבצים בזמן תגובה מהיר ככל הניתן
- זיהוי תווים מספרים ואותיות בעברית בלבד

## תיאור הארכיטקטורה

### תיאור שלבי העבודה

תכנון הפרויקט נעשה מן הכלל אל הפרט, בתחילה תכנון הפרויקט נעשה באופן כללי ובכל שלב ירדתי פנימה, יותר לפרטים ולעומק עד לביצוע מלא ומקיף של הפרויקט.

### תיאור רכיבים



### ארכיטקטורת רשת

אין- לא קיים

### תיאור פרוטוקולי רשת

http

### שרת – לקוח

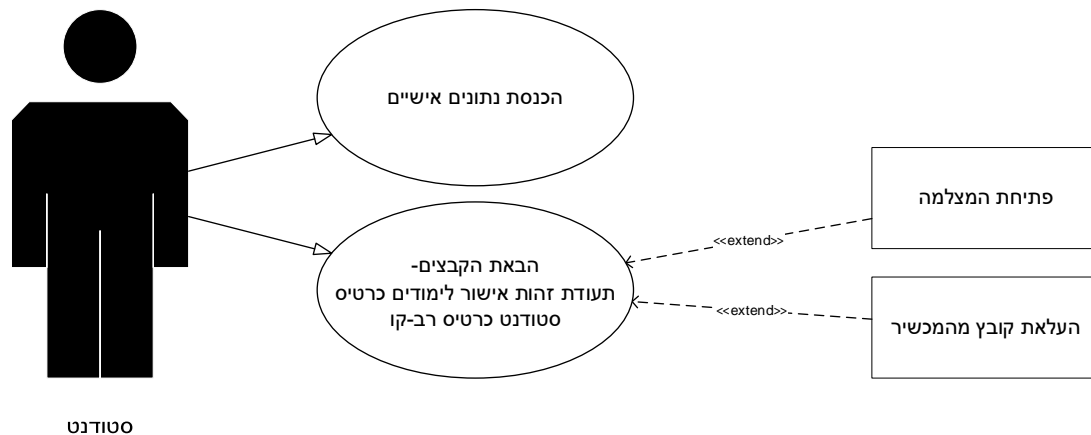
- בצד שרת – server
  - פותח בשפת Python עם תוכנת PyCharm גרסה 3.10 עם טכנולוגיית Flask -
  - ספריה ב Python המשמשת בסיס עבודה לפיתוח Web וממשק API בשפת Python
- בצד לקוח – client
  - פותח בשפות Html , css , js בטכנולוגיית React , עם תוכנת VisualStudioCode

### תיאור הצפנות

אין- לא קיים

## ניתוח ותרשים UML/Use cases של המערכת המוצעת

### תרשים UML



### רשימת useCase

- Uc1 - הכנסת נתונים אישיים- שם פרטי ומשפחה, מס' זהות
- Uc2 - העלאת קבצים- תעודת זהות אישור לימודים כרטיס סטודנט כרטיס רב-קו

### תיאור ה useCase

#### Use Case 1 ○

**Name** - הכנסת נתונים אישיים

**Identifier** - UC1

**Description** - הכנסה לתוך input שם פרטי ומשפחה, מספר זהות לצורך שליחתם

לשרת לבדיקת תקינות הקבצים

**Actors** - Client- student

**Frequency** - בעת הצגת הדמיה

**Pre-conditions** - מילוי פרטים אישיים

**Extended use cases** - None

**Included use cases** - None

**Basic course of action** - None

**Alternate course of action** - None

#### Use Case 2 ○

**Name** - העלאת קבצים

**Identifier** - UC2

**Description** - העלאת קבצים: כרטיס סטודנט אישור לימודים כרטיס ת"ז כרטיס רב-קו, השרת יעבד את התמונה וימצא את מיקומי המילים בתמונה, את מיקומי המילים הוא יחתוך לאותיות וישלח לרשת שמזהה את האות בתמונה ומחזירה אותה, את האותיות נרכיב חזרה למילה כמו שהיא מופיעה בתמונה וכניס למערך, לאחר שנקבל את המערך מכל קובץ שהועלה נשלח לפונקציה שבדקת בכל אחד מהמערכים אם הנתונים שהמשתמש הכניס זהים לנתונים שזהו מהקבצים, במידה וכל הנתונים זהים יחזור למשתמש אישור והוא יוכל להמשיך לתשלום וטעינת הרב-קו, במידה ולא יחזור למשתמש הודעה עם כל המקומות שלא היו טובים כמו שהשם לא תקין בקובץ רב-קו והוא יצטרך להעלות שוב את הקבצים

**Actors** - Client- student

**Frequency** - בעת הצגת הדמיה

**Pre-conditions** - בחירת קבצים להעלאה

**Extended use cases**

- פתיחת המצלמה לצילום קבצים
- פתיחת חלון הקבצים לבחירת קובץ מהמחשב

**Included use cases**

- העלאת הקבצים והנתונים האישיים לשרת
- צפיה בתוצאות

**Basic course of action** - None

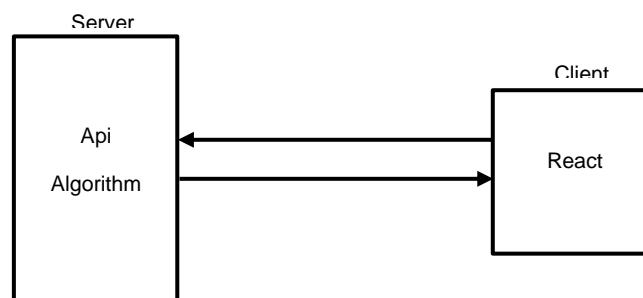
**Alternate course of action** - None

### מבנה הנתונים בהם השתמשתי בפרויקט

מטריצות- הפונקציות השונות של עיבוד תמונה עובדות על התמונה כמטריצת פיקסלים רשימות- לצורך החזקת מידע

### תרשים מודולים ותרשים מחלקות

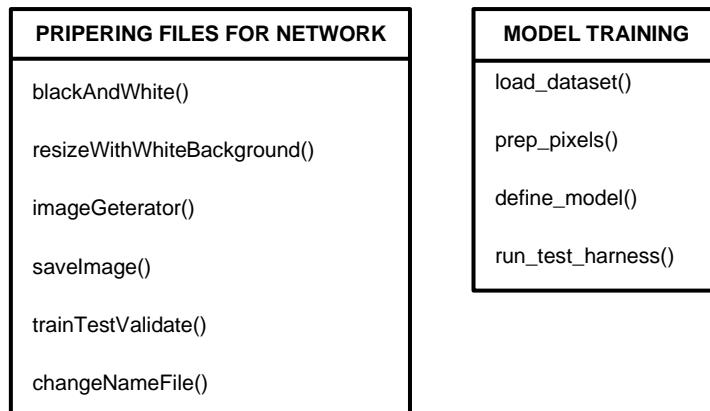
תרשים מודולים



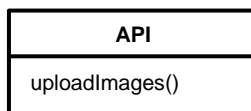


## תרשים מחלקות

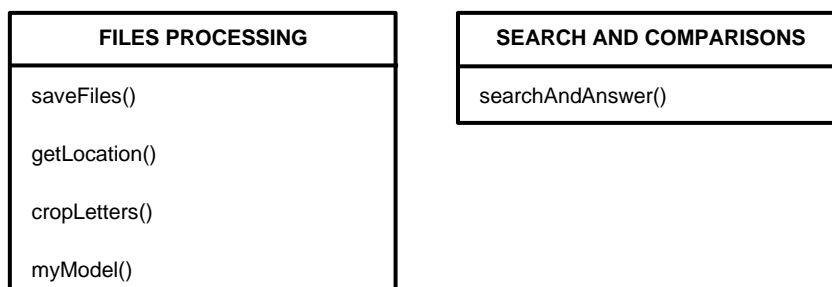
- הכנת הנתונים ובניית המודל



- API



- לוגיקה



## תיאור המחלקות המוצעות

### הכנת הנתונים ובניית המודל

#### מחלקת PRIPERING FILES FOR NETWORK

#### תפקידי המחלקה

מחלקה זו אחראית על עיבוד התמונות של המסד נתונים והכנתם לשליחה אל הרשת נוירונים בשביל האימון

## פונקציות המחלקה

blackAndWhite() - מקבלת תמונה והופכת אותה לשחור לבן

resizeWithWhiteBackground() - משנה את הרזולוציה של התמונות על ידי שמוסיפה רקע לבן מאחורי כל תמונה בגודל 28 על מנת לשמור על אחידות הגודל של התמונות

imageGeterator() - מקבלת תמונה ומעבה אותה – מכפילה את כמות התמונות ב 400

savelmage() - שומרת את כל התמונות המעובות..

trainTestValidate() - חלוקה ל train test validate באופן רנדומאלי כך ש כ70% ב train  
20% ב test ו10% ב validate

changeNameFile() - משנה את השמות של התמונות המחולקות לתיקיות בשביל המוחות

## מחלקת MODEL TRAINING

### תפקידי המחלקה

מחלקה זו היא המחלקה העיקרית בפרויקט והיא אחראית על האימון בפועל של המודל- רשת הנוירונים עד לקבלת רשת חזקה ומדויקת

### פונקציות המחלקה

load\_dataset() - טעינת המסד הנתונים בשביל האימון

prep\_pixels() - נרמול הנתונים של המסד נתונים

define\_model() - יצירת שכבות המודל

run\_test\_harness() - אימון רשת הנוירונים, אימון המודל

## API

### מחלקת API

### תפקידי המחלקה

מחלקה זו אחראית על התקשורת בין ה server ל client , מעבירה מידע מה client ומחזירה תגובה מה server

### פונקציות המחלקה

uploadImages() - אחראית על קבלת הקבצים מה client ושליחתם לעיבוד, בדיקות והחזרת תגובה ל client

## לוגיקה

### FILES PROCESSING מחלקת

#### תפקידי המחלקה

מחלקה זו אחראית על שמירת הקבצים כתמונות, עיבוד התמונה ושליחתה למודל לצורך הזיהוי של הטקסט

#### פונקציות המחלקה

saveFiles() - שמירת הקבצים כתמונות png

blackAndWhite() - הפיכת התמונות לשחור לבן

getLocation() - מציאת מיקומי המילים בתמונה

cropLetters() - חיתוך המילים לאותיות

myModel() - שליחה לפונקציה שתנרמל את התמונה ותשנה את גודלה ל 28\*28 ושליחה למודל לצורך הזיהוי

### SEARCH AND COMPARISONS מחלקת

#### תפקידי המחלקה

מחלקה זו אחראית על החיפוש של הנתונים הנדרשים- מספר זהות שם תאריכים וכד' מכל תמונה ותמונה שהתקבלה מהמשתמש והשוואת הנתונים

#### פונקציות המחלקה

searchAndAnswer() - פונקציה שמחפשת בטקסט שזוהה מכל הקבצים את השם ומשווה עם השם שהסטודנט הזין בתחילת הבקשה לשינוי הפרופיל, וכן היא מחפשת את המספר זהות בתעודת זהות בכרטיס סטודנט ובאישור לימודים ומשווה עם המספר שהסטודנט הזין בתחילת הבקשה לשינוי הפרופיל, וכן מחפשת את התאריך העיברי באישור לימודים ובודקת אם הוא תואם לתאריך העיברי של השנה הנוכחית.

## תיאור התוכנה

### סביבת פיתוח

חומרה- מעבד i7 , RAM 16GB  
 עמדת פיתוח- מחשב intel  
 מערכת הפעלה- windows 11  
 הפרויקט פותח בגרסת Python 3.10

### שפות תוכנה

צד שרת- פותח בשפת Python עם טכנולוגיית Flask עם תוכנת PyCharm גרסת 3.10  
 צד לקוח- פותח בטכנולוגיית React עם תוכנת VisualStudioCode  
 חיבור אינטרנט- לא נדרש  
 דפדפן- Chrome

## אלגוריתמים מרכזיים

האלגוריתם המרכזי אחראי על כל התפעול של המערכת מאחורה כלומר, אחראי על הניהול, תחילה הכנתי את הנתונים- התמונות בשביל אימון המודל- בשביל הלימוד שלו.. אחר כך בניתי את שכבות המודל ושלחתי לו את הנתונים המעובדים שהכנתי בשביל זה והרצתי את האימון, המודל התאמן ולמד את ההקשרים בין הנתונים כך שאחרי האימון קיבלתי מודל שיועד לקבל נתון מסוים- תמונה של אות והוא ידע להחזיר לי איזה אות זו.

לאחר שהמודל שלי מוכן יכולתי להתקדם לשלב הבא של הפרויקט בצד שרת שהוא להכין פונקציה שתדע לקבל תמונה גדולה שיש בה הרבה אותיות והיא תדע לחתוך את התמונה למילים ואחר כך לאותיות שאת האותיות נשלח למודל שיזהה ויחזיר לבסוף את כל המלל שהיה כתוב בתמונה בצורה של מחרוזת בשביל שאני יוכל לעבוד על המלל הזה.

את המחרוזת שהתקבלה מהתמונה אני שולחת לפונקציה שהולכת לחפש נתונים מסוימים אם הם קיימים ואם כן להשוות אותם עם נתונים מהמחרוזות האחרות - התמונות האחרות - הקבצים האחרים, במידה ששכל הבדיקות יצליחו והנתונים יהיו תואמים בכל הקבצים המערכת תחזיר תשובה חיובית במידה ולא היא תחזיר תשובה שלילית..

האלגוריתם המרכזי בנוי מכמה שלבים עיקריים

- הכנת הנתונים- DataBase לצורך בניית הרשת
- בניית מודל רשת נוירונים
- בדיקת נכונות המודל- Predict
- הפונקציה שמטפלת בכל עיבוד התמונה – מציאת מיקומי המילים חיתוכם חיתוך לאותיות ושליחה לזיהוי במודל
- חיבור ל client באמצעות ספריית flask
- שליחת הנתונים שזוהו מהתמונות לפונקציה של חיפוש הנתונים הרלוונטיים והשוואתם- בדיקת נכונותם

## פירוט

1. הכנת הנתונים- DataBase לצורך בניית הרשת

על מנת לאמן את הרשת בצורה הטובה והמדויקת ביותר יש להפעיל כמה פונקציות על התמונות שנאספו

ראשית יש להפוך אותן לצבעי שחור-לבן בשביל שיקל על הזיהוי של האות

```
def blackAndWhite(path,newPath):
    # running on the photos
    for dirLetters in os.listdir(fr'../{path}'):
        for image in
os.listdir(fr'../{path}/{dirLetters}')
```

```

        imageOriginul =
Image.open(fr'../{path}/{dirLetters}/{image}')
        # image height and width
        w, h = imageOriginul.size
        for i in range(h):
            for j in range(w):
                # pixel color
                r, g, b = imageOriginul.getpixel((i,
j))

                # if larger than 200 paint in white
                if (r+g+b) / 3 >= 200:
                    imageOriginul.putpixel((i,j),
(255, 255, 255))

                # if not paint in black
                else:
                    imageOriginul.putpixel((i, j), (0,
0, 0))

imageOriginul.save(fr'../{newPath}/{dirLetters}/{image}')
```

לאחר מכן יש לשנות את הגודל ל  $28 \times 28$  פיקסלים על מנת שיהיו חדות מספיק, ולא גדולות מדי

```

def resizeWithWhiteBackground(path_ori, path_dest,size):
    img = Image.open(path_ori)

    # resize and keep the aspect ratio
    img.thumbnail((size, size), Image.ANTIALIAS)

    # add the white background
    img_w, img_h = img.size
```

```
background = Image.new('RGB', (size, size), (255,
255, 255))

bg_w, bg_h = background.size
offset = ((bg_w - img_w) // 2, (bg_h - img_h) // 2)
background.paste(img, offset)
background.save(path_dest)
```

כמו כן על מנת שהרשת תתאמן בצורה מדויקת ככל היותר היא צריכה מספר גדול של תמונות.

אחת הדרכים להגדלת מאגר התמונות ללמידת מכונה היא באמצעות גישה של augmentation

ספריית Keras מציעה את הפונקציה ImageDataGenerator שיכולה להפיק תמונות מגוונות ושונות על בסיס התמונה המקורית שאספק לה הספריות הנדרשות הן

```
import numpy as np
from keras_preprocessing.image import ImageDataGenerator
from PIL import Image
from skimage import io
from pathlib import Path
import os
```

אופן הפעולה- הפונקציה מקבלת תמונה ועליה היא מגדילה, מקטינה מטה מסובבת לפי הכמות הנדרשת, לבסוף מחזירה רשימה של תמונות

```
def imageGenerator(path,num):
    # defining the changes for condensation
    imageGen = ImageDataGenerator(
        zoom_range=[0.9, 1.1],
        fill_mode='reflect',
        shear_range=0.15,
        width_shift_range=0.1,
        height_shift_range=0.1
    )
    imagePath = fr'../{path}'
    myFile = Path(imagePath)
```

```
if myFile.is_file():
    print('ok')
else:
    print('no image')
image = np.expand_dims(io.imread(imagePath),0)
augIter = imageGen.flow(image)
augImage = [next(augIter)[0].astype(np.uint8) for i
in range(num)]
return augImage
```

הפונקציה ששומרת את התמונות שהתקבלו לאחר העיבוי

```
def saveImage(images, numOfImages, path):
    index = 1
    for n in range(numOfImages):
        data = Image.fromarray(images[n])
        nameImage = fr'../{path}{index}.png'

        index += 1
```

לאחר עיבויי מאגר התמונות יש לחלק אותם לשלוש תיקיות שונות Test, Train,

Validate

החלוקה מתבצעת באופן הבא

20% מכלל התמונות במאגר יוכנסו באופן רנדומאלי (זאת בעזרת ספריית Scikit-

Learn) לתיקיית Test

והשאר (כ 80%) יוכנסו לתיקיית Train

לאחר מכן, מתוך התמונות שבתיקיית ה- Train, ניקח 10% לתיקיית ה Validate

תפקידה של כל תיקייה

Train- על תמונות אלו תתאמן הרשת ותלמד איך נראית כל אות

Test- עם התמונות האלו נבדוק את נכונותה של הרשת (עדיין בשלב של האימון

והלמידה)

Validate- תמונות אלו יוצגו לרשת בסיום האימון, כלומר אלו תמונות שהרשת עדיין

לא ראתה כלל במהלך האימון ואיתם בחנתי את הרשת אם היא אכן מזהה נכון

הקוד



```
def saveTrainTestValidate(pathImage, newPathImageTest,
newPathImageTrain, newPathImageValidate):
    X = []
    baseDir = fr'../{pathImage}'
    for i in sorted(os.listdir(baseDir)):
        for j in sorted(os.listdir(fr'{baseDir}/{i}')):
            X.append(j)

    # division for train and test
    dirTrainValidate, dirTest = train_test_split(X,
test_size=0.2, random_state=42)

    c = 1
    for item in dirTest:
        ori = fr'{baseDir}/{i}/{item}'
        dest = fr'../{newPathImageTest}/{i}/{c}.png'
        shutil.copy(ori, dest)
        c += 1

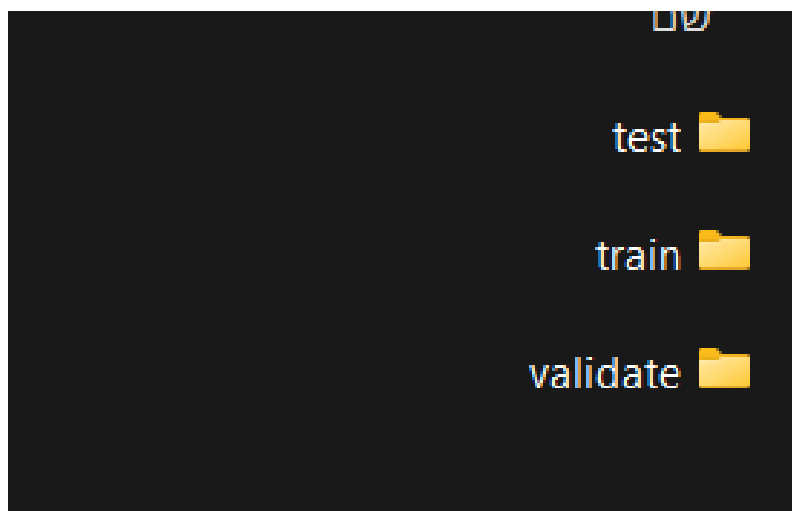
    # division for train and validate
    dirTrain, dirValidate =
train_test_split(dirTrainValidate, test_size=0.1,
random_state=42)

    c = 0
    for item in dirTrain:
        ori = fr'{baseDir}/{i}/{item}'
        dest = fr'../{newPathImageTrain}/{i}/{c}.png'
        shutil.copy(ori, dest)
```

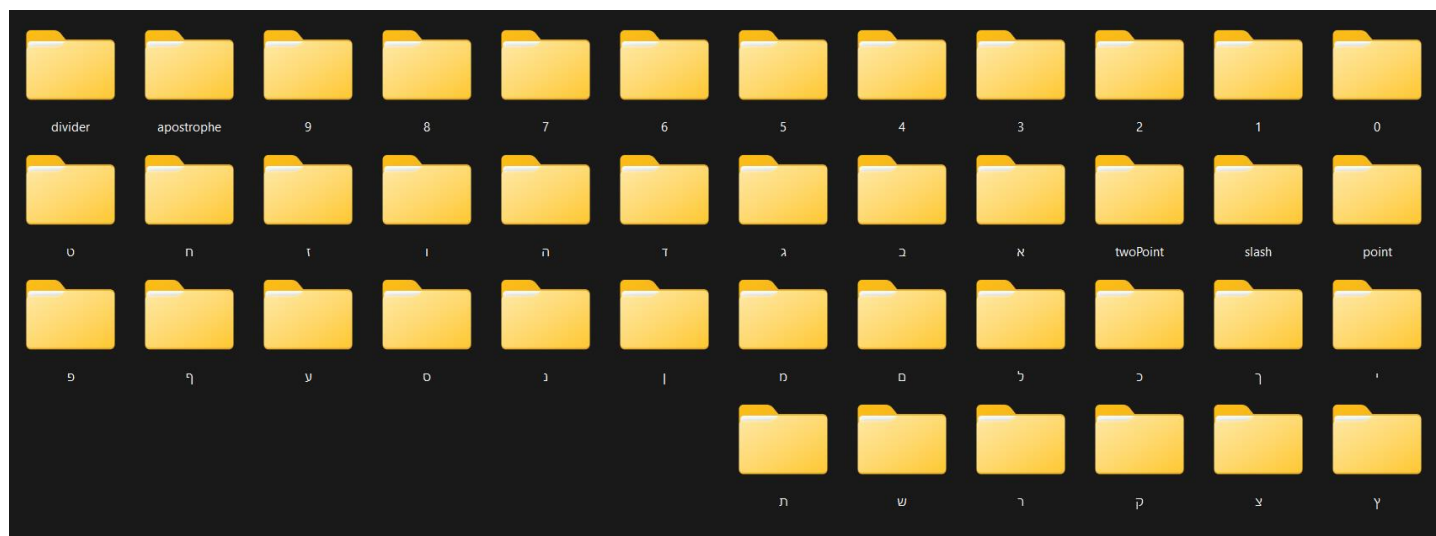
```
c += 1

c = 0
for item in dirValidate:
    ori = fr'{baseDir}/{i}/{item}'
    dest =
fr'../{newPathImageValidate}/{i}/{c}.png'
    shutil.copy(ori, dest)
    c += 1
X = []
```

תוצאה



כל אחת מהתיקיות מכילה 37 תיקיות של אותיות ומספרים



על מנת לשמור על הסדר שיניתי לכל התמונות את השם

```
def changeNamefile(path):
    for dirLetters in os.listdir(fr'../{path}'):
        i=1
        for image in
os.listdir(fr'../{path}/{dirLetters}'):
            if image.endswith('.png') or
image.endswith('.PNG') or image.endswith('jpg'):
                imageEnd = '.png'
                imageName = str(i)
                newName = fr'{imageName}{imageEnd}'

os.renames(fr'../{path}/{dirLetters}/{image}',

fr'../{path}/{dirLetters}/{newName}')

        i+=1
```

2. פיתוח המודל- בניית רשת הניורונים

ראשית ייבאתי את הספריות הבאות שעזרות לי בבניית רשת הניורונים

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.optimizers import SGD
from model.save_folders import trainTestValidat
from keras.layers import Dense, Flatten, Conv2D
from keras.callbacks import ReduceLROnPlateau,
EarlyStopping
```

לאחר מכן ישנה פונקציה שטוענת את התמונות בשביל האימון ברשת

```
# load train and test dataset
def load_dataset():
    trainX, trainY, testX, testY =
trainTestValidat(r'images/dataset/AlphaBetaHebrew28/ttv/t
est',
r'images/dataset/AlphaBetaHebrew28/ttv/train')

    # reshape dataset to have a single channel
    trainX = trainX.reshape((trainX.shape[0], 28, 28, 1))
    testX = testX.reshape((testX.shape[0], 28, 28, 1))

    return trainX, trainY, testX, testY
```

אחרי שהתמונות נטענות יש לנרמל את התמונות- כל פיקסל מיוצג לפי ממוצע אחוזי הצבע שלו ב-RGB כך שיוצא שהמספרים נעים בין 0-255 כך שכלול שהמספר קרוב ל0 זה שחור וכלול שהוא קרוב ל255 זה לבן, לכן על מנת שיהיה לי יותר קל להתעסק אם התמונה שיניתי את התווך של המספרים ל0-1 כך בהתאמה, וכן כל תא במטריצה

(פיקסל של תונה) אמור להיות מסוג float, וכל זה בשביל להגיעה לצורה שבה המודל יתאמן הכי טוב

```
def prep_pixels(train, test):
    # convert from integers to floats
    train_norm = train.astype('float32')
    test_norm = test.astype('float32')
    # normalize to range 0-1
    train_norm = train_norm / 255.0
    test_norm = test_norm / 255.0
    # return normalized images
    return train_norm, test_norm
```

בניית שכבות המודל

```
# define cnn model
def define_model():
    model = Sequential()
    # convolution layer-32 filters 3*3
    model.add(Conv2D(32, (3, 3), activation='relu',
kernel_initializer='he_uniform', input_shape=(28, 28,
1)))
    # [add this layer to decrease the loss]
    model.add(MaxPooling2D((2, 2)))
    # 2 layers of convolution layer-64 filters 3*3
    model.add(Conv2D(64, (3, 3), activation='relu',
kernel_initializer='he_uniform'))
    model.add(Conv2D(64, (3, 3), activation='relu',
kernel_initializer='he_uniform'))
    # [add this layer to decrease the loss]
    model.add(MaxPooling2D((2, 2)))
    # flattens the input to one long vector
```

```
model.add(Flatten())
# another neurons layers
model.add(Dense(100, activation='relu',
kernel_initializer='he_uniform'))
# input...
model.add(Dense(42, activation='softmax'))
# compile model
opt = SGD(learning_rate=0.01, momentum=0.9)
model.compile(optimizer=opt,
loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
reduce_lr = ReduceLRonPlateau(monitor='val_loss',
factor=0.2, patience=1, min_lr=0.0001)
early_stop = EarlyStopping(monitor='val_loss',
min_delta=0, patience=2, verbose=0, mode='auto')

return model, reduce_lr, early_stop
```

הסבר מורחב על תהליך בניית המודל

יצירת משתנה מודל מסוג שכבות

השכבה הראשונה היא שכבת הקונבולוציה

model.add(Conv2D(32, (3, 3), activation='relu',  
kernel\_initializer='he\_uniform', input\_shape=(28, 28, 1)))

שכבה זו מריצה 32 פילטרים בגודל 3\*3 פיקסלים על כל תמונה, הפילטרים הם הנוירונים של השכבה הכוללים פונקציית אקטיבציה. כל פילטר עובר על כל התמונה. לבסוף התוצאה הינה סידרה של ייצוגי שונים של התמונה. כל פילטר יצר ייצוג שונה ולכן במקרה הזה נוצרו 32 ייצוגים של התמונה.

השכבה הבאה היא שכבת דגימה – pooling (2, 2)))

שכבה זו מצמצמת את המטריצה (ריבוע פיקסלים) שנמסרה מהשכבה הקודמת. כל 4 פיקסלים יצומצמו לריבוע אחד בודד, תהליך זה נעשה ע"י בחירת הערך הגבוה

מבין ארבעת הפיקסלים. בסופו של דבר כך נקבל רבע מן המטריצה ואותה מעבירים לשכבה הבאה.

חזרתי שוב על התהליך של שכבת הקונבולוציה רק שעכשיו הוספתי עוד שתי שכבות של קונבולוציה עם 64 פילטרים בגודל  $3 \times 3$  פיקסלים.

שוב הוספתי עוד שכבה של pooling בשביל "לאזן" את המודל – שלא יצא יותר מידי מאומן..

הסיווג בפועל של התמונות נעשה על ידי רשת מסוג Dense .

כדי להעביר את המידע מבלוק הקונבולוציה (שכולל את חמשת השכבות הנ"ל) לרשת מסוג Dense, משתמשים בפונקציית flatten שתפקידה לשטח את המידע לווקטור, משום ששכבת ה Dense חייבת לעבוד על ווקטור ולא על מטריצה.

```
model.add(Flatten())
```

```
model.add(Dense(100, activation='relu', dense שכבת ה
kernel_initializer='he_uniform'))
```

מכילה 100 פיקסלים בלבד, ובה כל נירון מחובר לכל אחד מהנירונים של הקלט.

שכבה נוספת של dense היא כגודל המחלקות שיש לנו לאימון

```
model.add(Dense(42, activation='softmax'))
```

קמפול המודל-

```
opt = SGD(learning_rate=0.01, momentum=0.9)
```

```
model.compile(optimizer=opt, loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
```

```
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2,
patience=1, min_lr=0.0001)
```

```
early_stop = EarlyStopping(monitor='val_loss', min_delta=0,
patience=2, verbose=0, mode='auto')
```

בקומפילציה של המודל יש להשתמש במייעלים בשביל לשלוט על השיפוע-קצב הלמידה, אני השתמשתי במייעל SGD- ירידת שיפוע סטוכסטית, הוא אלגוריתם האופטימיזציה הקלאסי. ב-SGD אנו מחשבים את השיפוע של פונקציית אובדן הרשת ביחס לכל משקל בודד ברשת. כל מעבר קדימה ברשת מביא לפונקציית אובדן מסוימת עם פרמטר, ואז משתמשים בכל אחד מהשיפועים שיצרנו עבור כל אחד מהמשקולים, כפול קצב למידה מסוים, כדי להזיז את המשקולים לכל כיוון שהשיפוע שלו מצביע. SGD הוא האלגוריתם הפשוט ביותר הן מבחינה רעיונית והן מבחינת ההתנהגות שלו. המשקלים החדשים שנוצרו בכל איטרציה יהיו תמיד טובים יותר מהישנים מהאיטרציה הקודמת.

## אימון המודל

```
# fit model

history = model.fit(trainX, trainY, epochs=20,
callbacks=[reduce_lr, early_stop],
validation_data=(testX, testY))

print("The validation accuracy is :",
history.history['val_accuracy'])

print("The training accuracy is :",
history.history['accuracy'])

print("The validation loss is :",
history.history['val_loss'])

print("The training loss is :", history.history['loss'])

# save model

model.save(r'model_alphaBeta.h5')
```

לאחר האימון אחוזי הדיוק בזיהוי היו 99.98%

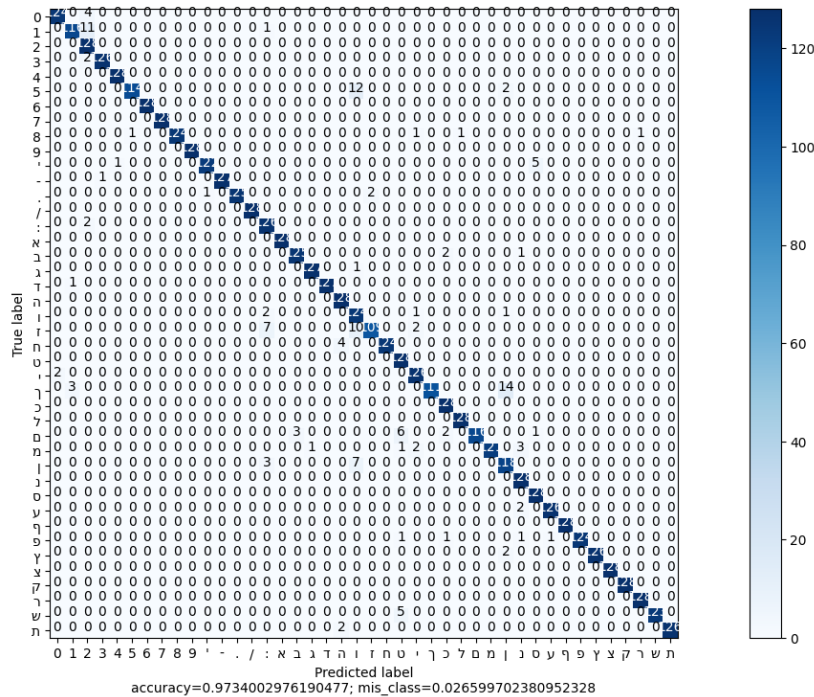
```
Epoch 1/40
1332/1332 [=====] - 17s 13ms/step - loss: 0.3709 - accuracy: 0.8885 - val_loss: 0.0072 - val_accuracy: 0.9980 - lr: 0.0100
Epoch 2/40
1332/1332 [=====] - 17s 13ms/step - loss: 0.0128 - accuracy: 0.9964 - val_loss: 0.0032 - val_accuracy: 0.9991 - lr: 0.0100
Epoch 3/40
1332/1332 [=====] - 18s 13ms/step - loss: 0.0040 - accuracy: 0.9989 - val_loss: 0.0013 - val_accuracy: 0.9997 - lr: 0.0100
Epoch 4/40
1332/1332 [=====] - 18s 13ms/step - loss: 0.0016 - accuracy: 0.9996 - val_loss: 6.6587e-04 - val_accuracy: 0.9997 - lr: 0.0100
Epoch 5/40
1332/1332 [=====] - 17s 13ms/step - loss: 9.3081e-04 - accuracy: 0.9998 - val_loss: 0.0014 - val_accuracy: 0.9997 - lr: 0.0100
Epoch 6/40
1332/1332 [=====] - 18s 13ms/step - loss: 3.9233e-04 - accuracy: 0.9999 - val_loss: 7.3930e-04 - val_accuracy: 0.9998 - lr: 0.0020
The validation accuracy is : [0.9979729652404785, 0.999070942401886, 0.9996621608734131, 0.9997466206550598, 0.9997466206550598, 0.9998310804367065]
The training accuracy is : [0.8885369896888733, 0.9964104890823364, 0.9988738894462585, 0.9996011853218079, 0.9998123049736023, 0.9999061822891235]
The validation loss is : [0.0071673342026770115, 0.00321318325586617, 0.0012793175410479307, 0.0006658665370196104, 0.0013817654689773917, 0.0007392959087155759]
The training loss is : [0.37085995078086853, 0.012785570695996284, 0.004009529482573271, 0.0016102351946756244, 0.0009308119188062847, 0.00039233200368471444]
```



### 3. בדיקת נכונות המודל Predict-

כדי לבדוק את נכונות המודל לקחתי את תקית ה Validate של התמונות ובניתי מטריצת בלבול כדי לראות מה רמת הדיוק של המודל, ואיפה הוא מזהה נכון ואיפה

הוא טועה



כפי שרואים המודל מזהה את הרוב כנכון פה ושם יש לא בלבולים קטנים אך אין כל כך מה לעשות איתם גם כי הם אפסיים וגם כי זה בלבול בדברים יחסית דומים.

4. הפונקציה שמטפלת בכל עיבוד התמונה והשליחה לרשת נעזרים

חיתוך התמונה לאותיות על מנת שנוכל לשלוח כל תמונה לרשת נירונים שתפענח,

שלב זה מתחלק לכמה שלבים

- מציאת המילים בתמונה
- חיתוך כל מילה לאותיות

את השלב הזה עשיתי בצורה מקבילית מפני שכל העיבוד תמונה, חיתוך המילים והאותיות, והשליחה לרשת נירונים לוקחת זמן כי הם פונקציות שמחשבות הרבה חישובים, ובזמן ריצה סיבוכיות הזמן הייתה מאוד גבוה (כ 10 דקות ) וכדי ליעל את הזמן עשיתי שחלק זה יעבוד במקביל על ידי threads וכך זמן התגובה שיחזור למשתמש תהיה יותר קצרה

## הפונקציה הראשית שמחלקת ל threads

```
def startRun(files):
    size = [4, 2, 4, 6]
    black = [68, 52, 55, 85]
    import multiprocessing
    with multiprocessing.Manager() as manager:
        listText = manager.list()
        listText.append([])
        listText.append([])
        listText.append([])
        listText.append([])
        p_arr = []
        for index, i_file in enumerate(files):
            p = multiprocessing.Process(target=my_func,
args=(i_file, size[index], black[index], index,
listText))
            p_arr.append(p)
        for p in p_arr:
            p.start()
        for p in p_arr:
            p.join()
        return list(listText)
```

```
def my_func(i_file, place, black, index, arrResult):
    if index == 1:
        removeLines(f'{i_file}')
        image = cv2.imread(i_file, cv2.IMREAD_GRAYSCALE)
        image = cv2.resize(image, (image.shape[1] // place,
image.shape[0] // place))
        (thresh, image) = cv2.threshold(image, black, 255,
cv2.THRESH_BINARY)
        loc = getLocationText(image)
        arrResult[index] = cropLetters(image, loc, index)
```

כעת נפרט על כל שלב

בשלב הבא יש למצוא את מיקומי המילים

לצורך זה השתמשתי בספריה בפיתון pytesseract שמוצאת את מיקומי המילים

בתמונה

להלן הקוד המבצע

```
def getLocationText(img):
    text_pos_list = []
    custom_config = r'-l heb --psm 6'
    d = pytesseract.image_to_data(img,
output_type=Output.DICT, config=custom_config)
```

```
n_boxes = len(d['level'])
for i in range(n_boxes):
    text = d['text'][i].strip()
    if len(text) == 0:
        continue
    (x, y, w, h) = (d['left'][i], d['top'][i],
d['width'][i], d['height'][i])

    text_pos_list.append((x, y, w, h))
return text_pos_list
```

כעת נותר לחלק כל מילה לאותיות, אם נמצאה עמודה שהיא כולה לבנה זה אומר שכאן נגמרה אות ובמיקום הזה חתכת

```
def cropLetters(image, loc, index):
    ind = 0
    for location in loc:
        (x, y, w, h) = location
        imgC = image[y:y + h, x:x + w]
cv2.imwrite(fr'myProject/cropLocation/imgCrop/imgC{index}
/i{ind}.png', imgC)

        ind += 1
        indexWord = 0
        wordStr = ''
        arrayWords = []
        for img in
sorted(os.listdir(fr'myProject/cropLocation/imgCrop/imgC{
index}')):
            with
Image.open(fr'myProject/cropLocation/imgCrop/imgC{index}/
{img}') as imgC:
                (width, height) = imgC.size
                if width > 200 or height > 200 or width < 5
or height < 5:
                    continue
                countWhite = [0] * width
                for row in range(height):
                    for col in range(width):
                        color = imgC.getpixel((col, row))
                        if color == 255:
                            countWhite[col] += 1
                start, end = 0, 0
                for col in range(0, width - 1, 1):
```

```

        if countWhite[col] != countWhite[col + 1]
or (
            (col == 0 or col == width - 2)
and countWhite[col] != height):
            if countWhite[col] == height or col
== 0:
                start = col + 1
            if countWhite[col + 1] == height or
col == width - 2:
                end = col + 1
            if end and start:
                imageCrop = imgC.crop((int(start
- 1), 0, int(end), int(height)))
                path =
fr'myProject/cropLocation/imgCrop/imgL{index}/letter{inde
xWord}.png'
                imageCrop.save(f'{path}',
format="png")
                indexWord += 1
                reSize(f'{path}', f'{path}', 28)
                char = myModel(path)
                if char.isdigit():
                    wordStr += char
                else:
                    wordStr = char + wordStr
                start, end = 0, 0
            arrayWords.append(wordStr)
            wordStr = ''
    return arrayWords

```

בשלב הבא יש לשלוח את התמונות של האותיות לרשת נירונים על מנת שתפענח את האות

```

def myModel(path):
    # load the image
    img = load_img(path, color_mode="grayscale",
target_size=(28, 28))
    # convert to array
    img = img_to_array(img)
    # reshape into a single sample with 1 channel
    img = img.reshape(1, 28, 28, 1)
    # prepare pixel data
    img = img.astype('float32')
    img = img / 255.0

```

```
# load model
model = load_model(fr'myProject/model/alpha-beta-
tubs-model.h5')
# predict the class
predict_value = model.predict(img)
digit = argmax(predict_value)
return convertDigit(digit)
```

מכיוון שהרשת מחזירה מספר קטגוריה שהיא בין 0 ל-42 ולא את האות עצמה יש צורך בפונקציה שתמיר את זה לתו

```
def convertDigit(digit):
    arrClass = ['0', '1', '2', '3', '4', '5', '6', '7',
'8', '9', '"', '-', '.', '/', ':', 'א', 'ב', 'ג', 'ד',
'ה', 'ו',
'ז', 'ח', 'ט', 'י', 'כ', 'ל', 'מ',
'נ', 'ס', 'ע', 'פ', 'ק', 'ר', 'ש', 'ת']
    return arrClass[int(digit)]
```

5. חיבור ל client באמצעות ספריית flask

שלב זה נותן את התקשורת בין צד לקוח לצד שרת – העברת נתונים אחד לשני..  
הספריות הנדרשות

```
from flask import Flask, request
from flask_cors import CORS
import sys
import asyncio
app = Flask(__name__)
CORS(app)
if sys.platform == "win32" and sys.version_info >= (3, 8,
0):
    asyncio.set_event_loop_policy(asyncio.WindowsSelectorEven
tLoopPolicy())
```

## פעולות

Upload- קלט: קבלת הקבצים מהמשתמש ושליחתם לעיבוד, פלט: החזרת התשובה למשתמש

```
i = 0
@app.route('/upload-images', methods=['POST'])
def uploadImages():
    global i

    idCard = request.files['idCard']
    studentPermit = request.files['studentPermit']
    studentCard = request.files['studentCard']
    ravKav = request.files['ravKav']
    firstName = request.form['firstName']
    lastName = request.form['lastName']
    idNumber = request.form['idNumber']
    files1 = [idCard, studentPermit, studentCard, ravKav]
    i += 1
    # saves the received files as an image
    files = saveFiles(files1, i)
    # sends to processing functions for model
    strFiles = startRun(files)
    # sends for testing and comparison functions
    answer = searchAndAnswer(strFiles, firstName,
lastName, idNumber)
    for di in
os.listdir(f'myProject/cropLocation/imgCrop'):
        for deli in
os.listdir(f'myProject/cropLocation/imgCrop/{di}'):
            if
os.path.isfile(f'myProject/cropLocation/imgCrop/{di}/{deli}'):
os.remove(f'myProject/cropLocation/imgCrop/{di}/{deli}'):
        for df in os.listdir(f'myProject/files/file'):
            if os.path.isfile(f'myProject/files/file/{df}'):
                os.remove(f'myProject/files/file/{df}')
    return json.dumps(answer)
```

## קוד התוכנית – על פי סטנדרטים בליווי תיעוד

### Server

#### קלט

הפונקציות ב server מקבלות את התמונות- הקבצים שהמשתמש שלח, ועלים הם בודקות את הנתונים

#### פלט

מחזיר את תוצאת ההשוואות

#### פונקציות עיקריות

ישנם כמה פונקציות נוספות שלא נמצאות באלגוריתם המרכזי אבל הן חשובות...  
 saveFile() - פונקציה שמקבלת את הקבצים שהתקבלו מה client וממירה אותם לקובץ מסוג png תמונה

```
def saveFiles(files, i):
    newFiles = []
    poppler_path = r'files/poppler-0.68.0_x86'
    for file in files:
        if file.filename.endswith('pdf'):
            file.save(f'{i}{file.filename}')
            pages =
convert_from_path(pdf_path=f'{i}{file.filename}',
dpi=500, poppler_path=poppler_path)
            for page in pages:
                path = f'{i}{file.name}.png'
                page.save(f'files/{path}', 'PNG')
                newFiles.append(f'files/{path}')
                os.remove(f'{i}{file.filename}')
            elif file.filename.lower().endswith('jpg'):
                path = f'{i}{file.name}.png'
                file.save(f'files/{path}', 'PNG')
                newFiles.append(f'files/{path}')
            else:
                path = f'{i}{file.name}.png'
```

```
file.save(f'files/{path}')
newFiles.append(f'files/{path}')

return newFiles
```

searchAndAnswer() - פונקציה שמחפשת בטקסט שזוהה את המספר זהות שם ותאריך ובודקת את נכונותם- משווה אם המספר זהות תואם בכל הקבצים, שם תואם בכל הקבצים ותאריך של השנה הנוכחית, במקרה של שגיאה הפונקציה מחזירה למשתמש איפה הייתה השגיאה

```
def searchAndAnswer(strFiles, firstName, lastName,
idNumber):
    res = []
    status = '1'
    yearToday = ['ב'תשפ', 'תשפב']

    if not any(element in idNumber for element in
strFiles[0]):
        res.append('The ID number in the ID card is
incorrect')
        status = '0'
    if not any(element in idNumber for element in
strFiles[1]):
        res.append('Identity number in incorrect study
certificate')
        status = '0'
    if not any(element in idNumber for element in
strFiles[2]):
        res.append('The ID number on the student card is
incorrect')
        status = '0'
    if not any(element in firstName for element in
strFiles[0]) or not any(element in lastName for element
in strFiles[0]):
        res.append('The name in the ID card is
incorrect')
        status = '0'
    if not any(element in firstName for element in
strFiles[1]) or not any(element in lastName for element
in strFiles[1]):
        res.append('The name in the certificate is
```



```
incorrect')
    status = '0'
    if not any(element in firstName for element in
strFiles[2]) or not any(element in lastName for element
in strFiles[2]):
        res.append('Invalid student card name')
        status = '0'
    if not any(element in firstName for element in
strFiles[3]) or not any(element in lastName for element
in strFiles[3]):
        res.append('The name in rav-kav is incorrect')
        status = '0'
    if not any(element in yearToday[0] for element in
strFiles[1]) and not any(element in yearToday[1] for
element in strFiles[1]):
        res.append('The date in the certificate is
incorrect')
        status = '0'
    return [status, res]
```

## Client

ב client יש את הקומפוננטה הראשית App שהיא אחראית על הניווט בין התצוגות השונות- תצוגה של הזנת נתונים אישיים, העלאת תמונה, תצוגה של השגיונות וכד'

## קלט

הפונקציות ב client מקבלות את התמונות- הקבצים שהמשתמש מעלה, וכן הפרטיים האישיים ושולחות אותם לשרת

## פלט

## פונקציות עיקריות

ישנם כמה פונקציות נוספות שקשורות למסך של העלאת הקבצים, שלא נמצאות באלגוריתם המרכזי אבל הן חשובות..

Form- אחראית על הצגת מסך של הזנת נתונים אישיים והעברתם לפונקציה של העלאת קבצים לצורך שליחתם לשרת

```
export default function Form(){

    const navigate=useNavigate();
    const [firstName, setFirstName] = useState();
    const [lastName, setLastName] = useState();
    const [idNumber, setidNumber] = useState();
    const [noFile, sentNofile] = useState(false)
```

```
// A function that sends the entered data to the file
upload component
const continue_uploader_image={()=>{
  if(firstName && lastName && idNumber){
    const form = [firstName, lastName, idNumber];
    navigate('/upload', { state: { data: form } })
  }
  else {
    sentNoFile(true)
  }
}

// A function that closes the modal
const cancelModel = () => {
  sentNoFile(false)
}

return(
  <div>
    {/* Displays the personal data entry buttons */}
    <Box component="form" sx={{ '& > :not(style)': {
m: 1, width: '25ch' }, }} noValidate autoComplete="off">
      <TextField id="standard-basic"
label="FirstName"
variant="standard" onChange={(event)=>setFirstName(event.targ
et.value)}>
        <TextField id="standard-basic"
label="LastName" variant="standard" onChange={(event) =>
setLastName(event.target.value)}>
        <TextField id="standard-basic"
label="IdNumber" variant="standard" onChange={(event) =>
setIdNumber(event.target.value)}>
      </Box>
      <button onClick={continue_uploader_image}
className="continue-btn" type="button" >continue</button>

      {/* error modal – when not enough files were
uploaded */}
      <MyModal show={noFile} title={'It is mandatory to
fill in all the details'} cancelModel={cancelModel} icon={<i
className="fa fa-exclamation-triangle warning-triangle col"
aria-hidden="true"></i>}></MyModal>

    </div>
  )
}
```

ImageUploader - אחראית על הצגת המסך של העלאת הקבצים ושליחתם לשרת

בתוכה נזמן את הקומפוננטה MyModal ImageDropZone

```
export default function ImageUploader() {
  // Variable settings
  const [noFile, sentNoFile] = useState(false)
  const idCardRef = useRef();
  const approvalOfStudiesRef = useRef();
  const studentCardRef = useRef();
  const ravKavRef = useRef();
  const navigate = useNavigate();
  const [invalidateFiles, setInvalidateFile] = useState(false);
  // A function that sends the server to the files
  const send = () => {
    // When all files are present
    if (idCardRef.current.returnFile() &&
    studentPermitRef.current.returnFile() &&
    studentCardRef.current.returnFile() &&
    ravKavRef.current.returnFile()) {
      const idCard = idCardRef.current.returnFile()
      const studentPermit =
    studentPermitRef.current.returnFile()
      const studentCard =
    studentCardRef.current.returnFile()
      const ravKav = ravKavRef.current.returnFile()
      const firstName = location.state.data[0]
      const lastName = location.state.data[1]
      const idNumber = location.state.data[2]
      // Create from all files one FormData object that is
    sent to the server
      const formData = new FormData();
      formData.append('firstName', firstName);
      formData.append('lastName', lastName);
      formData.append('idNumber', idNumber);
      formData.append('idCard', idCard);
      formData.append('studentPermit', studentPermit);
      formData.append('studentCard', studentCard);
      formData.append('ravKav', ravKav);

      // console.log(location.state.data[0])
      // Send to server
      axios.post('http://127.0.0.1:5000/upload-images',
    formData, {})
      // Getting the answer from the server
      .then(response => {
```

```

        if (response.data[0] === '1') { rout() }
        if (response.data[0] === '0') {
responseFalse(response.data[1]) }
        })
        .catch((error) => {
            if (error.response) {
                console.log(error.response)
                console.log(error.response.status)
                console.log(error.response.headers)
            }
        })
    }
    // In case all files are not uploaded, an appropriate
message is displayed
    else {
        sentNoFile(true)
    }
}
// When a positive response is returned from the server,
send for further operation
const rout={()=>{
    navigate('continue')
}
// A function that closes the modal
const cancelModel = () => {
    if(noFile){sentNoFile(false)}
    if(invalideFiles){setInvalideFile(false)}
}
return (
    <div >
        <h3>Please upload the appropriate photos</h3>
        <div style={{ display: 'flex' }}>
            /* The component of uploading files */
            <ImageDropZone ref={idCardRef} nameH3={"Id card"}
/>
            <ImageDropZone ref={approvalOfStudiesRef}
nameH3={"Approval of studies"} />
        </div>
        <div style={{ display: 'flex' }}>

```

```

        <ImageDropZone ref={studentCardRef}
nameH3={"Student card"} />
        <ImageDropZone ref={ravKavRef} nameH3={"Rav kav
Card"} />
    </div>
    { /* Button of sending the files */}
    <button onClick={send} className="file-upload-btn"
type="button" >I finished<i className="fa fa-paper-plane send"
aria-hidden="true"></i></button>
    { /* error modal – when not enough files were uploaded
*/}
    <MyModal show={noFile} title={'All appropriate forms
must be uploaded'} cancelModel={cancelModel} icon={<i
className="fa fa-exclamation-triangle warning-triangle col"
aria-hidden="true"></i>}></MyModal>
    { /* error modal – when a negative response is
returned from the server */}
    <MyModal show={invalidFiles} title={'Invalid files'}
cancelModel={cancelModel} icon={<i className="fa fa-
exclamation-triangle warning-triangle col" aria-
hidden="true"></i>}></MyModal>
    </div>
)
}

```

ImageDropZone – תפקידה להוסיף על הכפתור של העלאת תמונה אפשרות לגרירה ושחרור מעל שטח הגרירה וכן גם את האפשרות של הפתיחת מצלמה במכשיר (אם קיים)

```

// An object that contains the camera settings
const videoConstraints = {
    height: 200,
    facingMode: "user"
};
const ImageDropZone = forwardRef((props, ref) => {

```

```
// Variable settings
const [imageFile, setImageFile] = useState();
const [openCamera, setOpenCamera] = useState(false);
const webcamRef = useRef(null);
const [image, setImage] = useState('');
// A function responsible for deleting a selected file
const cancel = () => {
  setImageFile(undefined)
}
// A function responsible for opening the camera
const openCamera1 = () => {
  setOpenCamera(true)
}
// A function responsible for closing the camera
const cancelModel = () => {
  if (openCamera) { setOpenCamera(false) }
}
// A function responsible for capturing the image
const capture = useCallback(
  () => {
    const imageSrc =
webcamRef.current.getScreenshot();
    setImage(imageSrc)
  },
  [webcamRef]
);
// A function responsible for saving the image
const save = () => {
  const imageBlob = new File([image], 'File Camera', {
type: 'image/png' })
  setImageFile(imageBlob)
  setOpenCamera(false)
}
```

```
// A function responsible for saving the uploaded image by
dragging or dropping
const onDrop = useCallback(
  acceptedFiles => {
    const file = acceptedFiles[0]
    if (file.type === 'application/pdf' || file.type
=== 'image/png' || file.type === 'image/jpeg') {
      setImageFile(file)
    }
  },
  [setImageFile],
)
// Sending the image object uploaded to the imageUploader
component
useImperativeHandle(ref, () => ({
  returnFile() {
    return imageFile
  }
})));
// Set the drag and drop variable
const { getRootProps, getInputProps } = useDropzone({
onDrop })
return (
  <div>
    <div className='image-upload-wrap' >
      {/* Button for uploading a file from the
computer / dragging the file and dialog ... */}
      <div {...getRootProps()}>
        <input className='file-upload-input'
type='file' accept="image/*" {...getInputProps()} />
        <div className='drag-text'>
          <h3>{props.nameH3}</h3>

```

```

        {imageFile ? <h6>{imageFile.name}</h6>
: <h6>Click or drag here to upload</h6>}
      </div>
    </div>
    { /* Cancel the selected file */}
    <div >
      {imageFile ? <i onClick={cancel}
className="fa fa-times cancel" aria-hidden="true" /> : <></>}
    </div>
    { /* Button to open the camera to take the file
    */}

    <div >
      <i onClick={openCamera1} className="fa fa-
camera camera0" aria-hidden="true" />
    </div>
  </div>
  { /* modal for opening the camera */}
  <MyModal
    show={openCamera}
    title={props.nameH3}
    cancelModel={cancelModel}
    icon={<i className="fa fa-camera camera col"
aria-hidden="true" />}>
    <div className="webcam-container">
      { /* Displays the camera screen / View the
      captured image */}
      <div className="webcam-img">
        {image === '' ? <Webcam
          audio={false}
          ref={webcamRef}
          screenshotFormat="image/png"
          videoConstraints={videoConstraints
}

```



```

        /> : <img src={image} alt='' />}
      </div>
      { /* Displays the save button / save and
replay buttons */}
      {image === '' ?
        <button className='btn-camera'
onClick={e => { e.preventDefault(); capture();
}}>Capture</button> :
        <><button className='btn-camera'
onClick={e => { e.preventDefault(); save(); }}>Save</button>
        <button className='btn-camera
repeat' onClick={e => { e.preventDefault(); setImage('')
}}>Repeat</button></>}
      </div>
    </MyModal>
  </div>
)
})
ImageDropZone.displayName = "ImageDropZone";
export default ImageDropZone;

```

MyModal - תפקידה לקפוץ למשתמש כאשר לא העלו את כל הקבצים הנדרשים וכאשר חוזר מהשרת תשובה שלילית- כלומר הקבצים לא חוקיים, וכן גם כאשר רוצים לצלם את הקובץ על ידי המצלמה של המחשב

```

export default function MyModal(props) {
  return (
    <Modal dir="rtl" show={props.show} >
      { /* Displays the appropriate diameter of the modal
*/}
      <Modal.Header>
        <Modal.Title className='row
warning'>{props.icon}<div className='warning-text
col'>{props.title}</div></Modal.Title>
      </Modal.Header>

```

```

        {/* When it is necessary to also add a body to the
modal (in the camera trail) */}
        {props.children?
          <Modal.Body>
            {props.children}
          </Modal.Body>:<></>}
        {/* Modal close button */}
        <Modal.Footer dir="rtl">
          <Button onClick={props.cancelModel}
variant="secondary" className='btm-model'>
            Close
          </Button>
        </Modal.Footer>
      </Modal>
    )
  }

```

## תיאור המסד נתונים

מסד נתונים מורכב מתמונות רבות של אותיות אותם חתכתי מתוך תמונות שיצרתי בעצמי. בנוסף, על מנת לעבות את מאגר התמונות בניתי קוד עם יכולת פונקציונלית של עיבוד תמונה שיוצרת מכל תמונה – מספר תמונות שונות. עד שהגעתי לכמות שכוללת של כמעל 50,000 תמונות! בנוסף, נתינת גודל אחיד לכלל התמונות במאגר וכן שינוי הצבעים לשחור-לבן עבור אימון תקין ונכון של הרשתות השונות. לאחר מכן, חלוקת מאגר הנתונים ל - 3 תתי מאגרים train, test, validate בהן נמצאות כל התמונות עליהן המודל התאמן.

- Train - החלק הגדול ביותר ממדגם הנתונים המשמש לאימון המודל. אוסף התמונות עליהם המודל מתאמן באופן מעשי, לומד על פי הנתונים שהתמונות בתיקיה זו נתנו לו ועל פיהם מכון את המשקולות ובונה את הרשת.
- Test – החלק הבינוני ממאגר התמונות, הם היעד של המודל. האימון נעשה בצורה כזו שמזינים מראש לאיזו תוצאה צריך להגיע ולפי כוון זה "סוללים את הדרך". חלק זה ממאגר התמונות, הוא זה שנתן למודל את תמונת היעד, עד לאיזה כוון להגיע, האם להוריד או להעלות את רמת המשקולות וכד'.
- Validate - אחרי שהמודל קיבל תמונות ללמוד מהם נתונים – train ותמונות מהם למד איך ומה בדיוק לעשות עם הנתונים – test ישנו מדגם נוסף הקטן ביותר מתוך השלושה, שעליו בודקים האם באמת מודל הרשת שנבנה אכן נבנה בצורה טובה ונותן תוצאות טובות ומדויקות. מזינים לרשת את כל התמונות מחלק זה ורואים באחוזים בכמה תמונות המודל זיהה נכון. בתמונה הבאה ניתן לראות את החלוקה המפורטת לתיקיות, כל תיקיה מכילה 42 קטגוריות המכילות את כל הא-ב והספרות מ 0-9 וכן כמה תווים נוספים

```

train
  0
  1
  2
  3
  4
  5
  6
  7
  8
  9
  apostrophe
  divider
  point
  slash
  twoPoint
  א
  ב
  ג
  ד
  ה
  ו
  ז
  ח
  ט
  י
  יו
  כ
  ל
  מ
  נ
  ס
  ע
  פ
  פו
  צ
  ק
  קו
  ר
  ש
  ת

```

```

test
  0
  1
  2
  3
  4
  5
  6
  7
  8
  9
  apostrophe
  divider
  point
  slash
  twoPoint
  א
  ב
  ג
  ד
  ה
  ו
  ז
  ח
  ט
  י
  יו
  כ
  ל
  מ
  נ
  ס
  ע
  פ
  פו
  צ
  ק
  קו
  ר
  ש
  ת

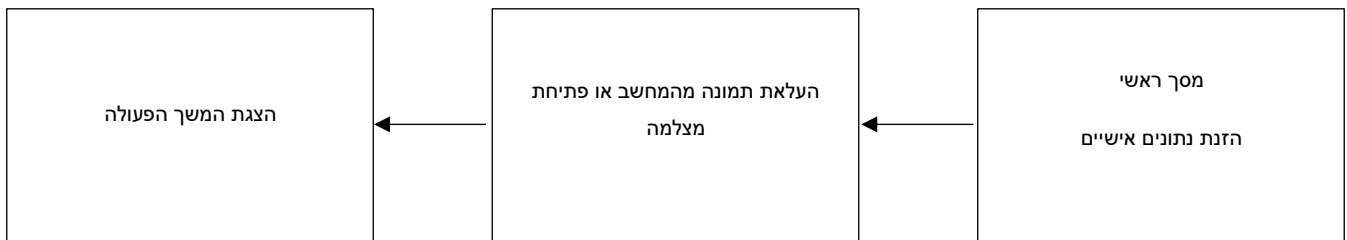
```

```

validate
  0
  1
  2
  3
  4
  5
  6
  7
  8
  9
  apostrophe
  divider
  point
  slash
  twoPoint
  א
  ב
  ג
  ד
  ה
  ו
  ז
  ח
  ט
  י
  יו
  כ
  ל
  מ
  נ
  ס
  ע
  פ
  פו
  צ
  ק
  קו
  ר
  ש
  ת

```

## תיאור מסכים



## תרשים Screen Flow Diagram

### תפקידי המסכים

1. מסך פתיחה בו המשתמש מזין את הפרטים האישיים שלו

PROFILE

First Name Last Name Id Number

CONTINUE

2. מסך שמוצג אחרי שהמשתמש מזין את הנתונים שלו בשביל שיעלה תמונות של הרב-קן תעודת זהות כרטיס סטודנט ואישור לימודים שלו

PROFILE

Please upload the appropriate photos

ID CARD Click or drag here to upload

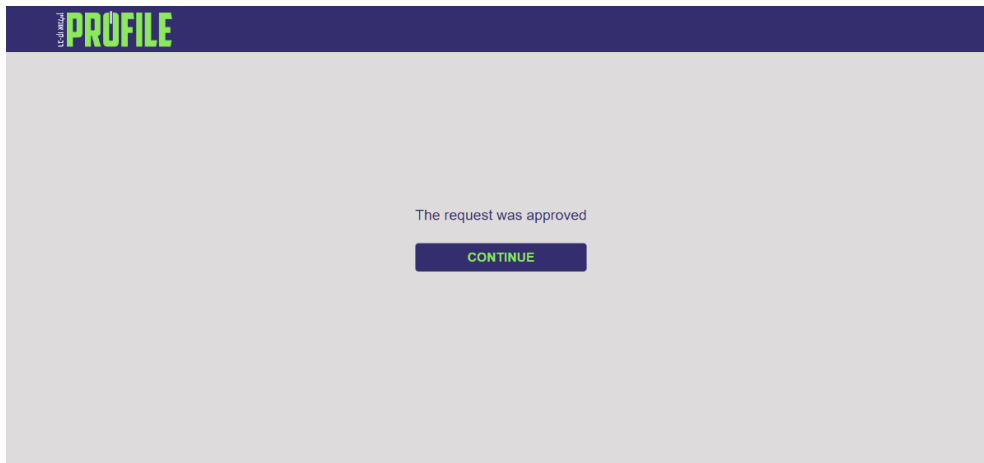
APPROVAL OF STUDIES Click or drag here to upload

STUDENT CARD Click or drag here to upload

RAV KAV CARD Click or drag here to upload

I FINISHED

3. מסך שמוצג רק כאשר מתקבל אישור מהמערכת להמשכת הפעולה



### תיאור מסך פתיחה

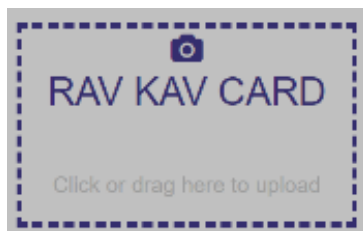
במסך הפתיחה המשתמש צריך להזין את הפרטיים האישיים שלו וכן הוא צריך להעלות קבצים של תעודת זהות כרטיס רב-קו כרטיס סטודנט ואישור לימודים. הוא יכול לעשות את העלאה בשני דרכים האחת פתיחת המצלמה(רק אם קיימת מצלמה במכשיר...) , והשנייה העלאת קובץ קיים במכשיר לאחר בחירת קובץ כל שהוא יוצג על המסך השם של הקובץ שהועלה והמשתמש יכול להתחרט ולבחור מחדש כאשר נבחרו כל הקבצים המשתמש צריך ללחוץ על הכפתור שליחה בה הוא שולח את הקבצים לשרת אם הקבצים חוקיים המסך יתחלף למסך השני בו יוצג כפתור להמשכת הפעולה

### כל מסכי המערכת בליווי הסברים (כולל הסברים על הכפתורים)

במסך הפתיחה

כפתור מסוג input שבו מזינים את הטקסט של הפרטים האישיים(עבור כל נתון אישי צריך ישנו כפתור מתאים)

כפתור להעברת הנתונים האישיים למסך של העלאת קבצים לצורך שליחתם בהמשך לשרת



הלחיצה על השטח הזה או בגרירה אליו הקובץ יעלה (כנ"ל לכל הארבעת המשטחים-  
כפתורים)

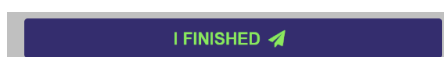


בלחיצה על הסמל מצלמה, המצלמה תפתח ועל המשתמש לשים את הקובץ המתאים מול  
המצלמה כך שהתמונה תצא ברורה

עבור כל קובץ יש להעלות את הקבצים מאחת מבצורות הנ"ל



אם המשתמש רוצה לבטל את אחד מהקבצים שכבר עלו עליו ללחוץ על הכפתור X שנמצע  
בפינה למעלה עבור כל קובץ...



כאשר כל הקבצים עלו ורוצים לשלוח לבדיקה יש ללחוץ על הכפתור "סיימתי"

במסך השני



המשתמש יגיע לפה רק כאשר התקבלה תשובה חיובית מהשרת ואז עליו ללחוץ על הכפתור  
"המשך" להמשך טעינת הפרופיל(טעינת הרב-קו)

בפתיחת המודל ישנם גם כן כמה כפתורים



כפתור סגירת המודל (כל סוגי המודל)

CAPTURE

כפתור לכידה של התמונה

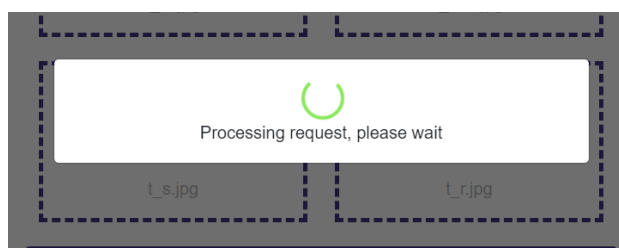
SAVE

כפתור שמירת התמונה

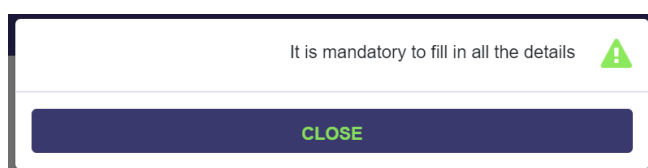
REPEAT

כפתור צילום חדש

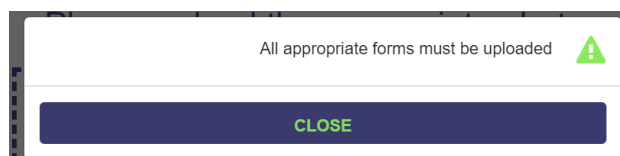
## הודעות למשתמש



כאשר הסטודנט לחץ על כפתור שליחה לשרת לבדיקת הנתונים, יוצג לו מסך זה עד לחזרת התשובה מהשרת

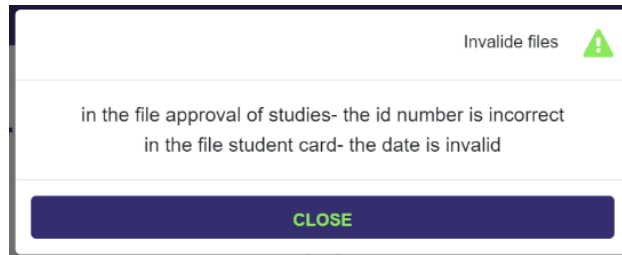


כאשר המשתמש לוחץ על הכפתור "המשך" לפני שהזין את כל הנתונים האישיים, קופצת לו האזהרה על כך שכל השדות חובה והוא חוזר חזרה למסך הפתיחה על ידי לחיצה על "סגירה"

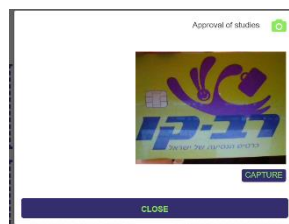


כאשר המשתמש לוחץ על הכפתור "סיימתי" לפני שכל הקבצים עלו קופצת לו אזהרה על כך שלא כל הקבצים עלו והוא חוזר חזרה למסך הפתיחה על ידי לחיצה על "סגירה"





כאשר אחד או יותר מהטפסים שהמשתמש העלה לא היו חוקיים וחזרה הודעת שגיאה מהמודל תקפוצ למשתמש אזהרה על כך שהקבצים לא חוקיים(אם אני יספיק גם מה לא היה חוקי..) ואז עליו ללחוץ על "אישור" והוא יחזור חזרה למסך הפתיחה



כאשר המצלמה נפתחת כאשר רוצים לצלם את המסמך

## מדריך למשתמש

בעת כניסת המשתמש למערכת יש לפניו שלוש כפתורים של הזנת נתונים וכפתור של המשך בפעולה.

לאחר שהוא מזין את הנתונים מוצג לו על המסך ארבע כפתורים לקבצים וכפתור להעלאה.

עליו להעלות כל קובץ באחת משתי הדרכים הבאות

האחת, העלאת קובץ מהמחשב, אם הוא בחר באפשרות הזאת עליו להעלות קובץ של תעודת הזהות שלו או כרטיס סטודנט או כרטיס רב-קו או אישור לימודים מתוך המחשב (בהתאמה לכפתור).

השנייה היא פתיחת המצלמה, כאן עליו לצלם את תעודת הזהות שלו או כרטיס סטודנט או כרטיס רב-קו או אישור לימודים (בהתאמה לכפתור)

(אפשר לעשות שכל קובץ יעלה בדרך אחרת)

לאחר מכן עליו הוא יכול ללחוץ על כפתור "ביטול"-X ואז יהיה עליו להעלות או לצלם את הקובץ מחדש, או על כפתור "אישור" וכך הוא שולח את הקבצים לעיבוד.

לאחר מכן מוצג לו על המסך כפתור להמשך הפעולה במקרה שחזר אישור מהמערכת או הודעת שגיאה- כאשר הקבצים לא אושרו על ידי המערכת אז הוא יצטרך לשלוח מחדש עם הטפסים התקינים.

## בדיקות והערכה

על מנת לבדוק האם רשת עובדת כראוי ומזהה את האותיות באופן מדויק היה צריך לעשות בדיקות תקינות רבות.

בהתחלה הרצתי את הפונקציות שמאמנות את המודל והם הביאו תוצאה שלא הייתה מדויקת מאד אז הוספתי עוד שכבות במודל והרצתי שוב ועדיין התוצאה לא הייתה מספקת אז הגדלתי את מספר ה-epoch, שזה מספר פעמים שהרשת מתאמנת על התמונות ולאחר כמה ניסיונות אחוז ההצלחה של המודל היה גבוה 98%

לאחר מכן, כדי לבדוק האם הרשת באמת מדויקת הרצתי את המודל על אלפי תמונות שהוא לא הכיר קודם ולא התאמן עליהם. ואכן הוא דייק וידע לסווג את התמונות במדויק. בנוסף הרצתי כמה פעמים על נתונים שונים לראות אם הוא מצליח לזהות את הנתונים ולהשוות אותם כאשר הם נכונים ולא נכונים...

## ניתוח יעילות

סיבוכיות אלגוריתם של פיתוח המודל הינו סיבוכיות גבוהה ולכן הרצתי מספר מוגדר של epochs, עד לקבלת התוצאה הרצויה.

אם כן הסיבוכיות הינה כמספר ה epochs - מספר סבבי האימון שהמודל מריץ על כל תמונה הנמצאת בתיקיית ה train, עבור הרשת הנבחרת ביצעתי 40 epochs. המודל ייעצר מלהתאמן עקב אחת מ-2 הסיבות

- גמר את סבבי האימון- הגיע למספר ה epoch המרבי שהוגדרו
  - ייעצר עקב אימון יתר, כך שאם ימשיך לבצע סבבים- ייתן תוצאות פחותות מהסבב הקודם (overfitting)
- המודל שלי נעצר כעבור 6 epochs. לאחר מכן כאשר הרשת מוכנה המערכת רצה כמספר שכבות הרשת.

## אבטחת מידע

השתמשתי בפרוטוקול תקשורת http ולא https כי השתמשתי בשרת מקומי ואז התקשורת בין React ל Python נחסמת ולא עובדת

## מסקנות

הפרויקט שהצגתי נכתב במשך השנה האחרונה בעבודה מאומצת ובהשקעה מרובה. בתחילת דרכי כאשר היה עלי לתכנן את הפרויקט הבנתי שהמשימה שעומדת לנגדי הולכת להיות לא פשוטה ואך יותר מכך...

ראיתי כי המשימה מורכבת ודורשת הן בתכנון הפרויקט בצורה נכונה, הן בלימוד החומר קודם תחילת בנייתו והן בבניה עצמה.

ראשית, השקעתי מחשבה רבה בתכנון הפרויקט. חשבתי כיצד לבצע את בניית האתר בצורה היעילה ביותר וכן תכננתי את מראה האתר.

מבנה הפרויקט המורכב מצד שרת וצד לקוח הינו מסובך ביותר. כל צד הוא בעל אלגוריתם הדורש חשיבה והשקעה מרובה.

צד הלקוח תוכנת בטכנולוגיית React ובצד השרת שימוש בספריות מיוחדות בשפת Python הייתי צריכה להשקיע שעות רבות ללמוד את החומר ולהבין אותו לעומק, כל זה עוד לפני תחילת הכתיבה, וכשהגעתי לשלב הכתיבה שהוא השלב העיקרי, נדרש ממני הרבה ידע והתנסות על מנת להגיע לתוצאות טובות.

כתיבת הקוד הייתה מורכבת אך גם מאד מאתגרת, למדתי המון נושאים חדשים שלולא הפרויקט לא הייתי יודעת אותם בתור סטודנטית, בעיקר בנושא של למידת מכונה ואימון רשת נירונים, וכן כל הנושא של עיבוד תמונה, ובכלל זכיתי לדעת פיתון ברמה גבוהה ולהכיר ספריות חשובות ומהותיות שקיימות בה לעומק.

עכשיו, בזמן סיום הפרויקט ובמבט לאחור על הדרך אותה עברתי ניתן לומר כי היא הביאה אותי להבנה מעמיקה יותר ביסודות התכנות והבינה מלאכותית.

וניתן לומר שהפרויקט קידם אותי המון בתור מהנדסת, בפעם הראשונה בניתי תוכנה שימושית שכל סטודנט יכול להנות ממנה, כמובן שזה גורם לי לסיפוק אישי גדול והלוואי שיועיל וישמש אנשים אחרים.

מבחינה אישית – למדתי איך לנהל את הזמנים שלי נכון ואיך לעבוד לבד בלי אדם שידאג שהדברים באמת יקרו. למדתי לעבוד על פרויקטים גדולים ובצורה טובה ומסודרת.

פרויקט זה אכן היווה אבן דרך משמעותית בהתמקצעות בתחום הנדסת התכנה.

## פיתוחים עתידיים

התוכנה יפה ומועילה אך תמיד ניתן לשפר ולהוסיף, כמו לדוגמא: להרחיב את התוכנה גם עבור שינוי כול סוגי הפרופילים ולא רק עבור סטודנטים, וכן להוסיף מודל שיזהה את הפנים בכול התמונות כאותו אדם

## ביבליוגרפיה

- <https://github.com/> GitHub
- [/https://stackoverflow.com](https://stackoverflow.com) Stack Overflow
- <https://reshetech.co.il/python-tutorials/all-the-tutorials> לימוד Python
- <https://www.programiz.com/python-programming/methods/built-in/ord>
- <https://reshetech.co.il/machine-learning-tutorials/all-the-tutorials> לימוד למידת מכונה
- <https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-from-scratch-for-mnist-handwritten-digit-classification>
- [/https://opencv.org](https://opencv.org) ספריית OpenCV
- ויקיפדיה
- <https://he.reactjs.org/tutorial/tutorial.html> מדריך ב React
- ספריית NumPy
- <https://numpy.org/doc/stable/reference/generated/numpy.ndarray.tolist.html>
- [/https://keras.io](https://keras.io) ספריית Keras
- [/https://www.tensorflow.org](https://www.tensorflow.org) ספריית TensorFlow
- <https://pyimagesearch.com/2020/08/24/ocr-with-opencv-keras-and-tensorflow/> עיבוד תמונה
- <https://reshetech.co.il/python-tutorials/pillow-imaging-library>
- רשת-טק
- [/https://programmingwithmosh.com](https://programmingwithmosh.com) חיבור Python ל React
- [/https://www.geeksforgeeks.org](https://www.geeksforgeeks.org) Geeks For Geeks
- [/https://nanonets.com/blog/ocr-with-tesseract](https://nanonets.com/blog/ocr-with-tesseract) ספריית pytesseract
- [/https://pillow.readthedocs.io/en/stable](https://pillow.readthedocs.io/en/stable) ספריית Pillow